

GDCM  
3.0.24

Generated by Doxygen 1.9.8



---

<b>1 GDCM Documentation</b>	<b>1</b>
<b>2 Todo List</b>	<b>3</b>
<b>3 Deprecated List</b>	<b>5</b>
<b>4 Bug List</b>	<b>7</b>
<b>5 Namespace Index</b>	<b>9</b>
5.1 Namespace List . . . . .	9
<b>6 Hierarchical Index</b>	<b>11</b>
6.1 Class Hierarchy . . . . .	11
<b>7 Class Index</b>	<b>21</b>
7.1 Class List . . . . .	21
<b>8 File Index</b>	<b>35</b>
8.1 File List . . . . .	35
<b>9 Namespace Documentation</b>	<b>43</b>
9.1 gdcmm Namespace Reference . . . . .	43
9.1.1 Detailed Description . . . . .	58
9.1.2 Typedef Documentation . . . . .	58
9.1.2.1 AECComp . . . . .	58
9.1.2.2 ASComp . . . . .	58
9.1.2.3 BOOL_FUNCTION_PFILE_PFILE_POINTER . . . . .	58
9.1.2.4 CSComp . . . . .	58
9.1.2.5 DAComp . . . . .	58
9.1.2.6 DTComp . . . . .	59
9.1.2.7 FileList . . . . .	59
9.1.2.8 IconImage . . . . .	59
9.1.2.9 LOComp . . . . .	59
9.1.2.10 LTComp . . . . .	59
9.1.2.11 MacroEntry . . . . .	59
9.1.2.12 NestedMacroEntries . . . . .	59
9.1.2.13 PNComp . . . . .	59
9.1.2.14 SHComp . . . . .	59
9.1.2.15 STComp . . . . .	59
9.1.2.16 TMComp . . . . .	60
9.1.2.17 UCComp . . . . .	60
9.1.2.18 UIComp . . . . .	60

9.1.2.19 URComp	60
9.1.2.20 UComp	60
9.1.3 Enumeration Type Documentation	60
9.1.3.1 CompOperators	60
9.1.3.2 ECharSet	60
9.1.3.3 ENQueryType	62
9.1.3.4 EQueryLevel	62
9.1.3.5 EQueryType	62
9.1.3.6 ERootType	63
9.1.3.7 LodModeType	63
9.1.4 Function Documentation	63
9.1.4.1 add1()	63
9.1.4.2 backslash()	63
9.1.4.3 Clamp()	64
9.1.4.4 clean()	64
9.1.4.5 doround()	64
9.1.4.6 GetVRFromTag()	64
9.1.4.7 operator"!="() [1/2]	64
9.1.4.8 operator"!="() [2/2]	64
9.1.4.9 operator<<() [1/59]	65
9.1.4.10 operator<<() [2/59]	65
9.1.4.11 operator<<() [3/59]	65
9.1.4.12 operator<<() [4/59]	65
9.1.4.13 operator<<() [5/59]	65
9.1.4.14 operator<<() [6/59]	65
9.1.4.15 operator<<() [7/59]	65
9.1.4.16 operator<<() [8/59]	66
9.1.4.17 operator<<() [9/59]	66
9.1.4.18 operator<<() [10/59]	66
9.1.4.19 operator<<() [11/59]	66
9.1.4.20 operator<<() [12/59]	66
9.1.4.21 operator<<() [13/59]	66
9.1.4.22 operator<<() [14/59]	66
9.1.4.23 operator<<() [15/59]	67
9.1.4.24 operator<<() [16/59]	67
9.1.4.25 operator<<() [17/59]	67
9.1.4.26 operator<<() [18/59]	67
9.1.4.27 operator<<() [19/59]	67
9.1.4.28 operator<<() [20/59]	67



---

9.1.4.29 operator<<()	[21/59]	67
9.1.4.30 operator<<()	[22/59]	68
9.1.4.31 operator<<()	[23/59]	68
9.1.4.32 operator<<()	[24/59]	68
9.1.4.33 operator<<()	[25/59]	68
9.1.4.34 operator<<()	[26/59]	68
9.1.4.35 operator<<()	[27/59]	68
9.1.4.36 operator<<()	[28/59]	68
9.1.4.37 operator<<()	[29/59]	69
9.1.4.38 operator<<()	[30/59]	69
9.1.4.39 operator<<()	[31/59]	69
9.1.4.40 operator<<()	[32/59]	69
9.1.4.41 operator<<()	[33/59]	69
9.1.4.42 operator<<()	[34/59]	69
9.1.4.43 operator<<()	[35/59]	70
9.1.4.44 operator<<()	[36/59]	70
9.1.4.45 operator<<()	[37/59]	70
9.1.4.46 operator<<()	[38/59]	70
9.1.4.47 operator<<()	[39/59]	70
9.1.4.48 operator<<()	[40/59]	70
9.1.4.49 operator<<()	[41/59]	70
9.1.4.50 operator<<()	[42/59]	71
9.1.4.51 operator<<()	[43/59]	71
9.1.4.52 operator<<()	[44/59]	71
9.1.4.53 operator<<()	[45/59]	71
9.1.4.54 operator<<()	[46/59]	71
9.1.4.55 operator<<()	[47/59]	71
9.1.4.56 operator<<()	[48/59]	71
9.1.4.57 operator<<()	[49/59]	72
9.1.4.58 operator<<()	[50/59]	72
9.1.4.59 operator<<()	[51/59]	72
9.1.4.60 operator<<()	[52/59]	72
9.1.4.61 operator<<()	[53/59]	72
9.1.4.62 operator<<()	[54/59]	72
9.1.4.63 operator<<()	[55/59]	72
9.1.4.64 operator<<()	[56/59]	73
9.1.4.65 operator<<()	[57/59]	73
9.1.4.66 operator<<()	[58/59]	73
9.1.4.67 operator<<()	[59/59]	73

9.1.4.68 operator==( )	73
9.1.4.69 operator>>( ) [1/3]	73
9.1.4.70 operator>>( ) [2/3]	73
9.1.4.71 operator>>( ) [3/3]	74
9.1.4.72 Round( )	74
9.1.4.73 roundat( )	74
9.1.4.74 x16printf( )	74
9.1.5 Variable Documentation	74
9.1.5.1 GlobalInstance	74
9.2 gdcm::network Namespace Reference	75
9.2.1 Enumeration Type Documentation	79
9.2.1.1 EEventID	79
9.2.1.2 EStateID	79
9.2.2 Function Documentation	80
9.2.2.1 GetStateIndex( )	80
9.2.3 Variable Documentation	80
9.2.3.1 cMaxEventID	80
9.2.3.2 cMaxStateID	80
9.3 gdcm::SegmentHelper Namespace Reference	81
9.4 gdcm::terminal Namespace Reference	81
9.4.1 Detailed Description	81
9.4.2 Enumeration Type Documentation	82
9.4.2.1 Attribute	82
9.4.2.2 Color	82
9.4.2.3 Mode	82
9.4.3 Function Documentation	83
9.4.3.1 setattribute( )	83
9.4.3.2 setbgcolor( )	83
9.4.3.3 setfgcolor( )	83
9.4.3.4 setmode( )	83
<b>10 Class Documentation</b>	<b>85</b>
10.1 gdcm::network::AAabortPDU Class Reference	85
10.1.1 Detailed Description	86
10.1.2 Constructor & Destructor Documentation	86
10.1.2.1 AAabortPDU( )	86
10.1.3 Member Function Documentation	86
10.1.3.1 IsLastFragment( )	86
10.1.3.2 Print( )	86

10.1.3.3 Read()	87
10.1.3.4 SetReason()	87
10.1.3.5 SetSource()	87
10.1.3.6 Size()	87
10.1.3.7 Write()	87
10.2 gdcmm::network::AAssociateACPDU Class Reference	88
10.2.1 Detailed Description	89
10.2.2 Member Typedef Documentation	89
10.2.2.1 SizeType	89
10.2.3 Constructor & Destructor Documentation	89
10.2.3.1 AAssociateACPDU()	89
10.2.4 Member Function Documentation	90
10.2.4.1 AddPresentationContextAC()	90
10.2.4.2 GetNumberOfPresentationContextAC()	90
10.2.4.3 GetPresentationContextAC()	90
10.2.4.4 GetUserInfo()	90
10.2.4.5 InitFromRQ()	90
10.2.4.6 IsLastFragment()	90
10.2.4.7 Print()	90
10.2.4.8 Read()	91
10.2.4.9 SetCalledAETitle()	91
10.2.4.10 SetCallingAETitle()	91
10.2.4.11 Size()	91
10.2.4.12 Write()	91
10.2.5 Friends And Related Symbol Documentation	91
10.2.5.1 AAssociateRQPDU	91
10.3 gdcmm::network::AAssociateRJPDU Class Reference	92
10.3.1 Detailed Description	93
10.3.2 Constructor & Destructor Documentation	93
10.3.2.1 AAssociateRJPDU()	93
10.3.3 Member Function Documentation	93
10.3.3.1 IsLastFragment()	93
10.3.3.2 Print()	93
10.3.3.3 Read()	93
10.3.3.4 Size()	93
10.3.3.5 Write()	94
10.4 gdcmm::network::AAssociateRQPDU Class Reference	94
10.4.1 Detailed Description	96
10.4.2 Member Typedef Documentation	96

10.4.2.1 PresentationContextArrayType . . . . .	96
10.4.2.2 SizeType . . . . .	96
10.4.3 Constructor & Destructor Documentation . . . . .	96
10.4.3.1 AAssociateRQPDU() [1/2] . . . . .	96
10.4.3.2 AAssociateRQPDU() [2/2] . . . . .	96
10.4.4 Member Function Documentation . . . . .	96
10.4.4.1 AddPresentationContext() . . . . .	96
10.4.4.2 GetCalledAETitle() . . . . .	96
10.4.4.3 GetCallingAETitle() . . . . .	97
10.4.4.4 GetNumberOfPresentationContext() . . . . .	97
10.4.4.5 GetPresentationContext() . . . . .	97
10.4.4.6 GetPresentationContextByAbstractSyntax() . . . . .	97
10.4.4.7 GetPresentationContextByID() . . . . .	97
10.4.4.8 GetPresentationContexts() . . . . .	97
10.4.4.9 GetReserved43_74() . . . . .	97
10.4.4.10 GetUserInfo() . . . . .	97
10.4.4.11 IsAETitleValid() . . . . .	97
10.4.4.12 IsLastFragment() . . . . .	98
10.4.4.13 Print() . . . . .	98
10.4.4.14 Read() . . . . .	98
10.4.4.15 SetCalledAETitle() . . . . .	98
10.4.4.16 SetCallingAETitle() . . . . .	98
10.4.4.17 SetUserInfo() . . . . .	98
10.4.4.18 Size() . . . . .	99
10.4.4.19 Write() . . . . .	99
10.4.5 Friends And Related Symbol Documentation . . . . .	99
10.4.5.1 AAssociateACPDU . . . . .	99
10.5 gdcmm::AbortEvent Class Reference . . . . .	99
10.6 gdcmm::network::AbstractSyntax Class Reference . . . . .	100
10.6.1 Detailed Description . . . . .	101
10.6.2 Constructor & Destructor Documentation . . . . .	101
10.6.2.1 AbstractSyntax() . . . . .	101
10.6.3 Member Function Documentation . . . . .	101
10.6.3.1 GetAsDataElement() . . . . .	101
10.6.3.2 GetName() . . . . .	101
10.6.3.3 operator==( ) . . . . .	101
10.6.3.4 Print() . . . . .	102
10.6.3.5 Read() . . . . .	102
10.6.3.6 SetName() . . . . .	102

10.6.3.7 SetNameFromUID()	102
10.6.3.8 Size()	102
10.6.3.9 Write()	102
10.7 gdcm::AnonymizeEvent Class Reference	103
10.7.1 Detailed Description	104
10.7.2 Member Typedef Documentation	104
10.7.2.1 Self	104
10.7.2.2 Superclass	104
10.7.3 Constructor & Destructor Documentation	105
10.7.3.1 AnonymizeEvent() [1/2]	105
10.7.3.2 ~AnonymizeEvent()	105
10.7.3.3 AnonymizeEvent() [2/2]	105
10.7.4 Member Function Documentation	105
10.7.4.1 CheckEvent()	105
10.7.4.2 GetEventName()	105
10.7.4.3 GetTag()	105
10.7.4.4 MakeObject()	106
10.7.4.5 operator=()	106
10.7.4.6 SetTag()	106
10.8 gdcm::Anonymizer Class Reference	106
10.8.1 Detailed Description	109
10.8.2 Constructor & Destructor Documentation	110
10.8.2.1 Anonymizer()	110
10.8.2.2 ~Anonymizer()	110
10.8.3 Member Function Documentation	110
10.8.3.1 BALCPPProtect()	110
10.8.3.2 BasicApplicationLevelConfidentialityProfile()	110
10.8.3.3 CanEmptyTag()	110
10.8.3.4 Clear() [1/2]	110
10.8.3.5 Clear() [2/2]	111
10.8.3.6 ClearInternalUIDs()	111
10.8.3.7 Empty() [1/2]	111
10.8.3.8 Empty() [2/2]	111
10.8.3.9 GetBasicApplicationLevelConfidentialityProfileAttributes()	111
10.8.3.10 GetCryptographicMessageSyntax()	112
10.8.3.11 GetFile()	112
10.8.3.12 New()	112
10.8.3.13 RecurseDataSet()	112
10.8.3.14 Remove() [1/2]	112

10.8.3.15 Remove() [2/2]	112
10.8.3.16 RemoveGroupLength()	113
10.8.3.17 RemovePrivateTags()	113
10.8.3.18 RemoveRetired()	113
10.8.3.19 Replace() [1/4]	113
10.8.3.20 Replace() [2/4]	113
10.8.3.21 Replace() [3/4]	114
10.8.3.22 Replace() [4/4]	114
10.8.3.23 SetCryptographicMessageSyntax()	114
10.8.3.24 SetFile()	114
10.9 gdcmm::AnyEvent Class Reference	115
10.10 gdcmm::network::ApplicationContext Class Reference	116
10.10.1 Detailed Description	117
10.10.2 Constructor & Destructor Documentation	117
10.10.2.1 ApplicationContext()	117
10.10.3 Member Function Documentation	117
10.10.3.1 GetName()	117
10.10.3.2 Print()	117
10.10.3.3 Read()	117
10.10.3.4 SetName()	117
10.10.3.5 Size()	117
10.10.3.6 Write()	118
10.11 gdcmm::ApplicationEntity Class Reference	118
10.11.1 Detailed Description	119
10.11.2 Member Function Documentation	119
10.11.2.1 IsValid()	119
10.11.2.2 Print()	119
10.11.2.3 SetBlob()	119
10.11.2.4 Squeeze()	119
10.11.3 Member Data Documentation	120
10.11.3.1 Internal	120
10.11.3.2 MaxLength	120
10.11.3.3 MaxNumberOfComponents	120
10.11.3.4 Padding	120
10.11.3.5 Separator	120
10.12 gdcmm::network::AReleaseRPPDU Class Reference	120
10.12.1 Detailed Description	121
10.12.2 Constructor & Destructor Documentation	121
10.12.2.1 AReleaseRPPDU()	121

10.12.3 Member Function Documentation	122
10.12.3.1 IsLastFragment()	122
10.12.3.2 Print()	122
10.12.3.3 Read()	122
10.12.3.4 Size()	122
10.12.3.5 Write()	122
10.13 gdcmm::network::AReleaseRQPDU Class Reference	123
10.13.1 Detailed Description	124
10.13.2 Constructor & Destructor Documentation	124
10.13.2.1 AReleaseRQPDU()	124
10.13.3 Member Function Documentation	124
10.13.3.1 IsLastFragment()	124
10.13.3.2 Print()	124
10.13.3.3 Read()	124
10.13.3.4 Size()	124
10.13.3.5 Write()	125
10.14 gdcmm::network::ARTIMTimer Class Reference	125
10.14.1 Detailed Description	125
10.14.2 Constructor & Destructor Documentation	125
10.14.2.1 ARTIMTimer()	125
10.14.3 Member Function Documentation	126
10.14.3.1 GetElapsedTime()	126
10.14.3.2 GetHasExpired()	126
10.14.3.3 GetTimeout()	126
10.14.3.4 SetTimeout()	126
10.14.3.5 Start()	126
10.14.3.6 Stop()	126
10.15 gdcmm::ASN1 Class Reference	126
10.15.1 Detailed Description	127
10.15.2 Constructor & Destructor Documentation	127
10.15.2.1 ASN1() [1/2]	127
10.15.2.2 ~ASN1()	127
10.15.2.3 ASN1() [2/2]	127
10.15.3 Member Function Documentation	127
10.15.3.1 operator=()	127
10.15.3.2 ParseDump()	128
10.15.3.3 ParseDumpFile()	128
10.15.3.4 TestPBKDF2()	128
10.16 gdcmm::network::AsynchronousOperationsWindowSub Class Reference	128

10.16.1 Detailed Description	128
10.16.2 Constructor & Destructor Documentation	129
10.16.2.1 AsynchronousOperationsWindowSub()	129
10.16.3 Member Function Documentation	129
10.16.3.1 Print()	129
10.16.3.2 Read()	129
10.16.3.3 Size()	129
10.16.3.4 Write()	129
10.17 gdcmm::Attribute< Group, Element, TVR, TVM > Class Template Reference	130
10.17.1 Detailed Description	131
10.17.2 Member Typedef Documentation	132
10.17.2.1 ArrayType	132
10.17.3 Member Enumeration Documentation	132
10.17.3.1 anonymous enum	132
10.17.4 Member Function Documentation	132
10.17.4.1 GDCM_STATIC_ASSERT() [1/3]	132
10.17.4.2 GDCM_STATIC_ASSERT() [2/3]	132
10.17.4.3 GDCM_STATIC_ASSERT() [3/3]	132
10.17.4.4 GetAsDataElement()	133
10.17.4.5 GetDictVM()	133
10.17.4.6 GetDictVR()	133
10.17.4.7 GetNumberOfValues()	133
10.17.4.8 GetTag()	133
10.17.4.9 GetValue() [1/2]	134
10.17.4.10 GetValue() [2/2]	134
10.17.4.11 GetValues()	134
10.17.4.12 GetVM()	134
10.17.4.13 GetVR()	134
10.17.4.14 operator!=(())	135
10.17.4.15 operator<()	135
10.17.4.16 operator==(())	135
10.17.4.17 operator[]() [1/2]	135
10.17.4.18 operator[]() [2/2]	135
10.17.4.19 Print()	135
10.17.4.20 Set()	136
10.17.4.21 SetByteValue()	136
10.17.4.22 SetByteValueNoSwap()	136
10.17.4.23 SetFromDataElement()	136
10.17.4.24 SetFromDataSet()	137



10.17.4.25 SetValue()	137
10.17.4.26 SetValues()	137
10.17.5 Member Data Documentation	137
10.17.5.1 Internal	137
10.18 gdcmm::Attribute< Group, Element, TVR, VM::VM1 > Class Template Reference	138
10.18.1 Member Typedef Documentation	139
10.18.1.1 ArrayType	139
10.18.2 Member Enumeration Documentation	139
10.18.2.1 anonymous enum	139
10.18.3 Member Function Documentation	139
10.18.3.1 GDCM_STATIC_ASSERT() [1/4]	139
10.18.3.2 GDCM_STATIC_ASSERT() [2/4]	140
10.18.3.3 GDCM_STATIC_ASSERT() [3/4]	140
10.18.3.4 GDCM_STATIC_ASSERT() [4/4]	140
10.18.3.5 GetAsDataElement()	140
10.18.3.6 GetDictVM()	140
10.18.3.7 GetDictVR()	140
10.18.3.8 GetNumberOfValues()	140
10.18.3.9 GetTag()	141
10.18.3.10 GetValue() [1/2]	141
10.18.3.11 GetValue() [2/2]	141
10.18.3.12 GetValues()	141
10.18.3.13 GetVM()	141
10.18.3.14 GetVR()	141
10.18.3.15 operator!=(())	141
10.18.3.16 operator<()	142
10.18.3.17 operator==(())	142
10.18.3.18 Print()	142
10.18.3.19 Set()	142
10.18.3.20 SetByteValue()	142
10.18.3.21 SetByteValueNoSwap()	142
10.18.3.22 SetFromDataElement()	143
10.18.3.23 SetFromDataSet()	143
10.18.3.24 SetValue()	143
10.18.4 Member Data Documentation	143
10.18.4.1 Internal	143
10.19 gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 > Class Template Reference	144
10.19.1 Member Function Documentation	146
10.19.1.1 GetVM()	146

10.20 gdcM::Attribute< Group, Element, TVR, VM::VM1_8 > Class Template Reference	146
10.20.1 Member Function Documentation	148
10.20.1.1 GetVM()	148
10.21 gdcM::Attribute< Group, Element, TVR, VM::VM1_n > Class Template Reference	148
10.21.1 Member Typedef Documentation	149
10.21.1.1 ArrayType	149
10.21.2 Constructor & Destructor Documentation	149
10.21.2.1 Attribute()	149
10.21.2.2 ~Attribute()	149
10.21.3 Member Function Documentation	150
10.21.3.1 GDCM_STATIC_ASSERT() [1/3]	150
10.21.3.2 GDCM_STATIC_ASSERT() [2/3]	150
10.21.3.3 GDCM_STATIC_ASSERT() [3/3]	150
10.21.3.4 GetAsDataElement()	150
10.21.3.5 GetDictVM()	150
10.21.3.6 GetDictVR()	150
10.21.3.7 GetNumberOfValues()	150
10.21.3.8 GetTag()	151
10.21.3.9 GetValue() [1/2]	151
10.21.3.10 GetValue() [2/2]	151
10.21.3.11 GetValues()	151
10.21.3.12 GetVM()	151
10.21.3.13 GetVR()	151
10.21.3.14 operator[]() [1/2]	151
10.21.3.15 operator[]() [2/2]	151
10.21.3.16 Print()	152
10.21.3.17 Set()	152
10.21.3.18 SetByteValue()	152
10.21.3.19 SetFromDataElement()	152
10.21.3.20 SetFromDataSet()	152
10.21.3.21 SetNumberOfValues()	152
10.21.3.22 SetValue() [1/2]	153
10.21.3.23 SetValue() [2/2]	153
10.21.3.24 SetValues()	153
10.22 gdcM::Attribute< Group, Element, TVR, VM::VM2_2n > Class Template Reference	153
10.22.1 Member Function Documentation	156
10.22.1.1 GetVM()	156
10.23 gdcM::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference	156
10.23.1 Member Function Documentation	158

10.23.1.1 GetVM()	158
10.24 gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n > Class Template Reference	158
10.24.1 Member Function Documentation	161
10.24.1.1 GetVM()	161
10.25 gdcmm::Attribute< Group, Element, TVR, VM::VM3_n > Class Template Reference	161
10.25.1 Member Function Documentation	163
10.25.1.1 GetVM()	163
10.26 gdcmm::AudioCodec Class Reference	163
10.26.1 Detailed Description	165
10.26.2 Constructor & Destructor Documentation	165
10.26.2.1 AudioCodec()	165
10.26.2.2 ~AudioCodec()	165
10.26.3 Member Function Documentation	165
10.26.3.1 CanCode()	165
10.26.3.2 CanDecode()	166
10.26.3.3 Decode()	166
10.27 gdcmm::Base64 Class Reference	166
10.27.1 Detailed Description	167
10.27.2 Constructor & Destructor Documentation	167
10.27.2.1 Base64()	167
10.27.3 Member Function Documentation	167
10.27.3.1 Decode()	167
10.27.3.2 Encode()	167
10.27.3.3 GetDecodeLength()	168
10.27.3.4 GetEncodeLength()	168
10.27.3.5 operator=()	168
10.28 gdcmm::network::BaseCompositeMessage Class Reference	169
10.28.1 Detailed Description	169
10.28.2 Constructor & Destructor Documentation	170
10.28.2.1 ~BaseCompositeMessage()	170
10.28.3 Member Function Documentation	170
10.28.3.1 ConstructPDV()	170
10.29 gdcmm::network::BaseNormalizedMessage Class Reference	170
10.29.1 Detailed Description	171
10.29.2 Constructor & Destructor Documentation	172
10.29.2.1 ~BaseNormalizedMessage()	172
10.29.3 Member Function Documentation	172
10.29.3.1 ConstructPDV()	172
10.30 gdcmm::network::BasePDU Class Reference	173

10.30.1 Detailed Description	173
10.30.2 Constructor & Destructor Documentation	174
10.30.2.1 ~BasePDU()	174
10.30.3 Member Function Documentation	174
10.30.3.1 IsLastFragment()	174
10.30.3.2 Print()	174
10.30.3.3 Read()	174
10.30.3.4 Size()	175
10.30.3.5 Write()	175
10.31 gdcmm::BaseQuery Class Reference	175
10.31.1 Detailed Description	177
10.31.2 Constructor & Destructor Documentation	177
10.31.2.1 BaseQuery()	177
10.31.2.2 ~BaseQuery()	177
10.31.3 Member Function Documentation	177
10.31.3.1 AddQueryDataSet()	177
10.31.3.2 GetAbstractSyntaxUID()	177
10.31.3.3 GetQueryDataSet() [1/2]	178
10.31.3.4 GetQueryDataSet() [2/2]	178
10.31.3.5 GetSOPInstanceUID()	178
10.31.3.6 Print()	178
10.31.3.7 SetSearchParameter() [1/3]	178
10.31.3.8 SetSearchParameter() [2/3]	178
10.31.3.9 SetSearchParameter() [3/3]	178
10.31.3.10 SetSOPInstanceUID()	179
10.31.3.11 ValidateQuery()	179
10.31.3.12 ValidDataSet()	179
10.31.3.13 WriteHelpFile()	179
10.31.3.14 WriteQuery()	179
10.31.4 Friends And Related Symbol Documentation	179
10.31.4.1 QueryFactory	179
10.31.5 Member Data Documentation	179
10.31.5.1 mDataSet	179
10.31.5.2 mSopInstanceUID	180
10.32 gdcmm::BaseRootQuery Class Reference	180
10.32.1 Detailed Description	182
10.32.2 Constructor & Destructor Documentation	182
10.32.2.1 BaseRootQuery()	182
10.32.2.2 ~BaseRootQuery()	183

10.32.3 Member Function Documentation	183
10.32.3.1 Construct()	183
10.32.3.2 GetQueryLevelFromQueryRoot()	183
10.32.3.3 GetQueryLevelFromString()	183
10.32.3.4 GetQueryLevelString()	183
10.32.3.5 GetTagListByLevel()	183
10.32.3.6 InitializeDataSet()	184
10.32.3.7 ValidateQuery()	184
10.32.4 Friends And Related Symbol Documentation	184
10.32.4.1 QueryFactory	184
10.32.5 Member Data Documentation	184
10.32.5.1 mHelpDescription	184
10.32.5.2 mImage	184
10.32.5.3 mPatient	185
10.32.5.4 mRootType	185
10.32.5.5 mSeries	185
10.32.5.6 mStudy	185
10.33 gdcM::SegmentHelper::BasicCodedEntry Struct Reference	185
10.33.1 Detailed Description	186
10.33.2 Constructor & Destructor Documentation	186
10.33.2.1 BasicCodedEntry() [1/3]	186
10.33.2.2 BasicCodedEntry() [2/3]	186
10.33.2.3 BasicCodedEntry() [3/3]	187
10.33.3 Member Function Documentation	187
10.33.3.1 IsEmpty()	187
10.33.4 Member Data Documentation	187
10.33.4.1 CM	187
10.33.4.2 CSD	187
10.33.4.3 CSV	187
10.33.4.4 CV	188
10.34 gdcM::BasicOffsetTable Class Reference	188
10.34.1 Detailed Description	191
10.34.2 Constructor & Destructor Documentation	191
10.34.2.1 BasicOffsetTable()	191
10.34.3 Member Function Documentation	191
10.34.3.1 Read()	191
10.34.4 Friends And Related Symbol Documentation	192
10.34.4.1 operator<<	192
10.35 gdcM::Bitmap Class Reference	192

10.35.1 Detailed Description	195
10.35.2 Member Typedef Documentation	195
10.35.2.1 LUTPtr	195
10.35.3 Constructor & Destructor Documentation	195
10.35.3.1 Bitmap()	195
10.35.3.2 ~Bitmap()	196
10.35.4 Member Function Documentation	196
10.35.4.1 AreOverlaysInPixelData()	196
10.35.4.2 Clear()	196
10.35.4.3 ComputeLossyFlag()	196
10.35.4.4 GetBuffer()	196
10.35.4.5 GetBuffer2()	196
10.35.4.6 GetBufferLength()	197
10.35.4.7 GetColumns()	197
10.35.4.8 GetDataElement() [1/2]	197
10.35.4.9 GetDataElement() [2/2]	197
10.35.4.10 GetDimension()	197
10.35.4.11 GetDimensions()	198
10.35.4.12 GetLUT() [1/2]	198
10.35.4.13 GetLUT() [2/2]	198
10.35.4.14 GetNeedByteSwap()	198
10.35.4.15 GetNumberOfDimensions()	198
10.35.4.16 GetPhotometricInterpretation()	199
10.35.4.17 GetPixelFormat() [1/2]	199
10.35.4.18 GetPixelFormat() [2/2]	199
10.35.4.19 GetPlanarConfiguration()	199
10.35.4.20 GetRows()	199
10.35.4.21 GetTransferSyntax()	199
10.35.4.22 IsEmpty()	200
10.35.4.23 IsLossy()	200
10.35.4.24 IsTransferSyntaxCompatible()	200
10.35.4.25 Print()	200
10.35.4.26 SetColumns()	200
10.35.4.27 SetDataElement()	200
10.35.4.28 SetDimension()	201
10.35.4.29 SetDimensions()	201
10.35.4.30 SetLossyFlag()	201
10.35.4.31 SetLUT()	201
10.35.4.32 SetNeedByteSwap()	201

10.35.4.33 SetNumberOfDimensions()	202
10.35.4.34 SetPhotometricInterpretation()	202
10.35.4.35 SetPixelFormat()	202
10.35.4.36 SetPlanarConfiguration()	202
10.35.4.37 SetRows()	203
10.35.4.38 SetTransferSyntax()	203
10.35.4.39 TryJPEG2000Codec()	203
10.35.4.40 TryJPEG2000Codec2()	203
10.35.4.41 TryJPEGCodec()	203
10.35.4.42 TryJPEGCodec2()	203
10.35.4.43 TryJPEGLSCodec()	203
10.35.4.44 TryKAKADUCodec()	204
10.35.4.45 TryPVRGCodec()	204
10.35.4.46 TryRAWCodec()	204
10.35.4.47 TryRLECodec()	204
10.35.4.48 UnusedBitsPresentInPixelData()	204
10.35.5 Friends And Related Symbol Documentation	204
10.35.5.1 ImageChangeTransferSyntax	204
10.35.5.2 PixmapReader	204
10.35.6 Member Data Documentation	205
10.35.6.1 Dimensions	205
10.35.6.2 LossyFlag	205
10.35.6.3 LUT	205
10.35.6.4 NeedByteSwap	205
10.35.6.5 NumberOfDimensions	205
10.35.6.6 PF	205
10.35.6.7 PI	205
10.35.6.8 PixelData	205
10.35.6.9 PlanarConfiguration	205
10.35.6.10 TS	206
10.36 gdcmm::BitmapToBitmapFilter Class Reference	206
10.36.1 Detailed Description	207
10.36.2 Constructor & Destructor Documentation	207
10.36.2.1 BitmapToBitmapFilter()	207
10.36.2.2 ~BitmapToBitmapFilter()	207
10.36.3 Member Function Documentation	207
10.36.3.1 GetOutput()	207
10.36.3.2 GetOutputAsBitmap()	207
10.36.3.3 SetInput()	208

10.36.4 Member Data Documentation	208
10.36.4.1 Input	208
10.36.4.2 Output	208
10.37 gdcm::BoxRegion Class Reference	208
10.37.1 Detailed Description	210
10.37.2 Constructor & Destructor Documentation	210
10.37.2.1 BoxRegion() [1/2]	210
10.37.2.2 ~BoxRegion()	210
10.37.2.3 BoxRegion() [2/2]	210
10.37.3 Member Function Documentation	210
10.37.3.1 Area()	210
10.37.3.2 BoundingBox()	211
10.37.3.3 Clone()	211
10.37.3.4 ComputeBoundingBox()	211
10.37.3.5 Empty()	211
10.37.3.6 GetXMax()	211
10.37.3.7 GetXMin()	211
10.37.3.8 GetYMax()	211
10.37.3.9 GetYMin()	212
10.37.3.10 GetZMax()	212
10.37.3.11 GetZMin()	212
10.37.3.12 IsValid()	212
10.37.3.13 operator=()	212
10.37.3.14 Print()	212
10.37.3.15 SetDomain()	213
10.38 gdcm::ByteBuffer Class Reference	213
10.38.1 Detailed Description	213
10.38.2 Constructor & Destructor Documentation	214
10.38.2.1 ByteBuffer()	214
10.38.3 Member Function Documentation	214
10.38.3.1 Get()	214
10.38.3.2 GetStart()	214
10.38.3.3 ShiftEnd()	214
10.38.3.4 UpdatePosition()	214
10.39 gdcm::ByteSwap< T > Class Template Reference	214
10.39.1 Detailed Description	215
10.39.2 Member Function Documentation	215
10.39.2.1 Swap()	215
10.39.2.2 SwapFromSwapCodeIntoSystem()	215



10.39.2.3 SwapRange()	215
10.39.2.4 SwapRangeFromSwapCodeIntoSystem()	216
10.39.2.5 SystemIsBigEndian()	216
10.39.2.6 SystemIsLittleEndian()	216
10.40 gdcm::ByteSwapFilter Class Reference	216
10.40.1 Detailed Description	217
10.40.2 Constructor & Destructor Documentation	217
10.40.2.1 ByteSwapFilter() [1/2]	217
10.40.2.2 ~ByteSwapFilter()	217
10.40.2.3 ByteSwapFilter() [2/2]	217
10.40.3 Member Function Documentation	217
10.40.3.1 ByteSwap()	217
10.40.3.2 operator=()	217
10.40.3.3 SetByteSwapTag()	217
10.41 gdcm::ByteValue Class Reference	218
10.41.1 Detailed Description	220
10.41.2 Constructor & Destructor Documentation	220
10.41.2.1 ByteValue() [1/2]	220
10.41.2.2 ByteValue() [2/2]	220
10.41.2.3 ~ByteValue()	220
10.41.3 Member Function Documentation	221
10.41.3.1 Append()	221
10.41.3.2 Clear()	221
10.41.3.3 ComputeLength()	221
10.41.3.4 Fill()	221
10.41.3.5 GetBuffer()	221
10.41.3.6 GetLength()	222
10.41.3.7 GetPointer()	222
10.41.3.8 GetVoidPointer() [1/2]	222
10.41.3.9 GetVoidPointer() [2/2]	222
10.41.3.10 IsEmpty()	223
10.41.3.11 IsPrintable()	223
10.41.3.12 operator const std::vector< char > &()	223
10.41.3.13 operator=()	223
10.41.3.14 operator==(1/2)	223
10.41.3.15 operator==(2/2)	223
10.41.3.16 Print()	223
10.41.3.17 PrintASCII()	224
10.41.3.18 PrintASCIIXML()	224

10.41.3.19 PrintGroupLength()	224
10.41.3.20 PrintHex()	224
10.41.3.21 PrintHexXML()	224
10.41.3.22 PrintPNXML()	224
10.41.3.23 Read() [1/2]	224
10.41.3.24 Read() [2/2]	225
10.41.3.25 SetLength()	225
10.41.3.26 SetLengthOnly()	225
10.41.3.27 Write() [1/2]	225
10.41.3.28 Write() [2/2]	225
10.41.3.29 WriteBuffer()	225
10.42 gdcM::CAPICryptoFactory Class Reference	226
10.42.1 Constructor & Destructor Documentation	227
10.42.1.1 CAPICryptoFactory()	227
10.42.2 Member Function Documentation	227
10.42.2.1 CreateCMSProvider()	227
10.43 gdcM::CAPICryptographicMessageSyntax Class Reference	227
10.43.1 Constructor & Destructor Documentation	229
10.43.1.1 CAPICryptographicMessageSyntax()	229
10.43.1.2 ~CAPICryptographicMessageSyntax()	229
10.43.2 Member Function Documentation	229
10.43.2.1 Decrypt()	229
10.43.2.2 Encrypt()	229
10.43.2.3 GetCipherType()	229
10.43.2.4 GetInitialized()	229
10.43.2.5 ParseCertificateFile()	230
10.43.2.6 ParseKeyFile()	230
10.43.2.7 SetCipherType()	230
10.43.2.8 SetPassword()	230
10.44 gdcM::network::CEchoRQ Class Reference	231
10.44.1 Detailed Description	232
10.44.2 Member Function Documentation	232
10.44.2.1 ConstructPDV()	232
10.44.3 Member Data Documentation	232
10.44.3.1 AffectedSOPClassUID	232
10.44.3.2 MessageID	232
10.45 gdcM::network::CEchoRSP Class Reference	233
10.45.1 Detailed Description	234
10.45.2 Member Function Documentation	234

10.45.2.1 ConstructPDVByDataSet()	234
10.46 gdcmm::network::CFind Class Reference	234
10.46.1 Detailed Description	234
10.47 gdcmm::network::CFindCancelRQ Class Reference	234
10.47.1 Detailed Description	235
10.47.2 Member Function Documentation	235
10.47.2.1 ConstructPDVByDataSet()	235
10.48 gdcmm::network::CFindRQ Class Reference	236
10.48.1 Detailed Description	237
10.48.2 Member Function Documentation	237
10.48.2.1 ConstructPDV()	237
10.49 gdcmm::network::CFindRSP Class Reference	237
10.49.1 Detailed Description	238
10.49.2 Member Function Documentation	238
10.49.2.1 ConstructPDVByDataSet()	238
10.50 gdcmm::Cleaner Class Reference	239
10.50.1 Detailed Description	241
10.50.2 Constructor & Destructor Documentation	241
10.50.2.1 Cleaner()	241
10.50.2.2 ~Cleaner()	242
10.50.3 Member Function Documentation	242
10.50.3.1 Clean()	242
10.50.3.2 Empty() [1/4]	242
10.50.3.3 Empty() [2/4]	242
10.50.3.4 Empty() [3/4]	242
10.50.3.5 Empty() [4/4]	242
10.50.3.6 EmptyWhenScrubFails()	243
10.50.3.7 GetFile()	243
10.50.3.8 New()	243
10.50.3.9 Preserve()	243
10.50.3.10 Remove() [1/4]	243
10.50.3.11 Remove() [2/4]	243
10.50.3.12 Remove() [3/4]	244
10.50.3.13 Remove() [4/4]	244
10.50.3.14 RemoveAllGroupLength()	244
10.50.3.15 RemoveAllIllegal()	244
10.50.3.16 RemoveAllMissingPrivateCreator()	244
10.50.3.17 RemoveMissingPrivateCreator()	244
10.50.3.18 Scrub() [1/4]	245

10.50.3.19 Scrub() [2/4]	245
10.50.3.20 Scrub() [3/4]	245
10.50.3.21 Scrub() [4/4]	245
10.50.3.22 SetFile()	245
10.51 gdcm::network::CMoveCancelRq Class Reference	246
10.51.1 Member Function Documentation	247
10.51.1.1 ConstructPDVByDataSet()	247
10.52 gdcm::network::CMoveRQ Class Reference	247
10.52.1 Detailed Description	248
10.52.2 Member Function Documentation	248
10.52.2.1 ConstructPDV()	248
10.53 gdcm::network::CMoveRSP Class Reference	248
10.53.1 Detailed Description	249
10.53.2 Member Function Documentation	249
10.53.2.1 ConstructPDVByDataSet()	249
10.54 gdcm::Codec Class Reference	250
10.54.1 Detailed Description	251
10.55 gdcm::Coder Class Reference	251
10.55.1 Detailed Description	252
10.55.2 Constructor & Destructor Documentation	252
10.55.2.1 ~Coder()	252
10.55.3 Member Function Documentation	252
10.55.3.1 CanCode()	252
10.55.3.2 Code()	252
10.55.3.3 InternalCode()	253
10.56 gdcm::CodeString Class Reference	253
10.56.1 Detailed Description	254
10.56.2 Member Typedef Documentation	254
10.56.2.1 const_iterator	254
10.56.2.2 const_reference	254
10.56.2.3 const_reverse_iterator	254
10.56.2.4 difference_type	255
10.56.2.5 iterator	255
10.56.2.6 pointer	255
10.56.2.7 reference	255
10.56.2.8 reverse_iterator	255
10.56.2.9 size_type	255
10.56.2.10 value_type	255
10.56.3 Constructor & Destructor Documentation	255

10.56.3.1 CodeString() [1/4]	255
10.56.3.2 CodeString() [2/4]	255
10.56.3.3 CodeString() [3/4]	256
10.56.3.4 CodeString() [4/4]	256
10.56.4 Member Function Documentation	256
10.56.4.1 GetAsString()	256
10.56.4.2 IsValid()	256
10.56.4.3 Size()	256
10.56.4.4 TrimInternal()	256
10.56.5 Friends And Related Symbol Documentation	256
10.56.5.1 operator"!="	256
10.56.5.2 operator<<	257
10.56.5.3 operator==	257
10.57 gdcmm::Command Class Reference	257
10.57.1 Detailed Description	259
10.57.2 Constructor & Destructor Documentation	259
10.57.2.1 Command() [1/2]	259
10.57.2.2 Command() [2/2]	259
10.57.2.3 ~Command()	259
10.57.3 Member Function Documentation	259
10.57.3.1 Execute() [1/2]	259
10.57.3.2 Execute() [2/2]	260
10.57.3.3 operator=()	260
10.58 gdcmm::CommandDataSet Class Reference	260
10.58.1 Detailed Description	263
10.58.2 Constructor & Destructor Documentation	263
10.58.2.1 CommandDataSet()	263
10.58.2.2 ~CommandDataSet()	263
10.58.3 Member Function Documentation	263
10.58.3.1 Insert()	263
10.58.3.2 Read()	264
10.58.3.3 Replace()	264
10.58.3.4 Write()	264
10.58.4 Friends And Related Symbol Documentation	264
10.58.4.1 operator<<	264
10.59 gdcmm::network::CompositeMessageFactory Class Reference	264
10.59.1 Detailed Description	265
10.59.2 Member Function Documentation	265
10.59.2.1 ConstructCEchoRQ()	265

10.59.2.2 ConstructCFindRQ()	265
10.59.2.3 ConstructCMoveRQ()	265
10.59.2.4 ConstructCStoreRQ()	266
10.59.2.5 ConstructCStoreRSP()	266
10.60 gdcm::CompositeNetworkFunctions Class Reference	266
10.60.1 Detailed Description	267
10.60.2 Member Typedef Documentation	267
10.60.2.1 KeyValuePairArrayType	267
10.60.2.2 KeyValuePairType	267
10.60.3 Member Function Documentation	267
10.60.3.1 CEcho()	267
10.60.3.2 CFind()	268
10.60.3.3 CMove()	269
10.60.3.4 ConstructQuery() [1/2]	269
10.60.3.5 ConstructQuery() [2/2]	269
10.60.3.6 CStore()	270
10.61 gdcm::ConstCharWrapper Class Reference	271
10.61.1 Detailed Description	271
10.61.2 Constructor & Destructor Documentation	271
10.61.2.1 ConstCharWrapper()	271
10.61.3 Member Function Documentation	271
10.61.3.1 operator const char *()	271
10.62 gdcm::CP246ExplicitDataElement Class Reference	272
10.62.1 Detailed Description	274
10.62.2 Member Function Documentation	274
10.62.2.1 GetLength()	274
10.62.2.2 Read()	275
10.62.2.3 ReadPreValue()	275
10.62.2.4 ReadValue()	275
10.62.2.5 ReadWithLength()	275
10.63 gdcm::CryptoFactory Class Reference	275
10.63.1 Detailed Description	276
10.63.2 Member Enumeration Documentation	276
10.63.2.1 CryptoLib	276
10.63.3 Constructor & Destructor Documentation	277
10.63.3.1 CryptoFactory() [1/2]	277
10.63.3.2 CryptoFactory() [2/2]	277
10.63.3.3 ~CryptoFactory()	277
10.63.4 Member Function Documentation	277

10.63.4.1 CreateCMSProvider()	277
10.63.4.2 GetFactoryInstance()	277
10.64 gdcmm::CryptographicMessageSyntax Class Reference	278
10.64.1 Detailed Description	278
10.64.2 Member Enumeration Documentation	279
10.64.2.1 CipherTypes	279
10.64.3 Constructor & Destructor Documentation	279
10.64.3.1 CryptographicMessageSyntax() [1/2]	279
10.64.3.2 ~CryptographicMessageSyntax()	279
10.64.3.3 CryptographicMessageSyntax() [2/2]	279
10.64.4 Member Function Documentation	279
10.64.4.1 Decrypt()	279
10.64.4.2 Encrypt()	280
10.64.4.3 GetCipherType()	280
10.64.4.4 operator=()	280
10.64.4.5 ParseCertificateFile()	280
10.64.4.6 ParseKeyFile()	280
10.64.4.7 SetCipherType()	281
10.64.4.8 SetPassword()	281
10.65 gdcmm::CSAElement Class Reference	281
10.65.1 Detailed Description	283
10.65.2 Member Typedef Documentation	283
10.65.2.1 DataPtr	283
10.65.3 Constructor & Destructor Documentation	283
10.65.3.1 CSAElement() [1/2]	283
10.65.3.2 CSAElement() [2/2]	283
10.65.4 Member Function Documentation	283
10.65.4.1 GetByteValue()	283
10.65.4.2 GetKey()	284
10.65.4.3 GetName()	284
10.65.4.4 GetNumberOfItems()	284
10.65.4.5 GetSyngoDT()	284
10.65.4.6 GetValue() [1/2]	284
10.65.4.7 GetValue() [2/2]	284
10.65.4.8 GetVM()	285
10.65.4.9 GetVR()	285
10.65.4.10 IsEmpty()	285
10.65.4.11 operator<()	285
10.65.4.12 operator=()	285

10.65.4.13 operator==( ) . . . . .	285
10.65.4.14 SetByteValue() . . . . .	286
10.65.4.15 SetKey() . . . . .	286
10.65.4.16 SetName() . . . . .	286
10.65.4.17 SetNoOfItems() . . . . .	286
10.65.4.18 SetSyngoDT() . . . . .	286
10.65.4.19 SetValue() . . . . .	286
10.65.4.20 SetVM() . . . . .	286
10.65.4.21 SetVR() . . . . .	286
10.65.5 Friends And Related Symbol Documentation . . . . .	287
10.65.5.1 operator<< . . . . .	287
10.65.6 Member Data Documentation . . . . .	287
10.65.6.1 DataField . . . . .	287
10.65.6.2 KeyField . . . . .	287
10.65.6.3 NameField . . . . .	287
10.65.6.4 NoOfItemsField . . . . .	287
10.65.6.5 SyngoDTField . . . . .	287
10.65.6.6 ValueMultiplicityField . . . . .	287
10.65.6.7 VRField . . . . .	288
10.66 gdcmm::CSAHeader Class Reference . . . . .	288
10.66.1 Detailed Description . . . . .	289
10.66.2 Member Enumeration Documentation . . . . .	289
10.66.2.1 CSAHeaderType . . . . .	289
10.66.3 Constructor & Destructor Documentation . . . . .	290
10.66.3.1 CSAHeader() . . . . .	290
10.66.3.2 ~CSAHeader() . . . . .	290
10.66.4 Member Function Documentation . . . . .	290
10.66.4.1 FindCSAElementByName() . . . . .	290
10.66.4.2 GetCSADataInfo() . . . . .	290
10.66.4.3 GetCSAEEnd() . . . . .	291
10.66.4.4 GetCSAElementByName() . . . . .	291
10.66.4.5 GetCSAImageHeaderInfoTag() . . . . .	291
10.66.4.6 GetCSASeriesHeaderInfoTag() . . . . .	291
10.66.4.7 GetDataSet() . . . . .	291
10.66.4.8 GetFormat() . . . . .	292
10.66.4.9 GetInterfile() . . . . .	292
10.66.4.10 GetMrProtocol() . . . . .	292
10.66.4.11 LoadFromDataElement() . . . . .	292
10.66.4.12 Print() . . . . .	292



10.66.5 Friends And Related Symbol Documentation	293
10.66.5.1 operator<<	293
10.67 gdcM::CSAHeaderDict Class Reference	293
10.67.1 Detailed Description	294
10.67.2 Member Typedef Documentation	294
10.67.2.1 ConstIterator	294
10.67.2.2 Iterator	294
10.67.2.3 MapCSAHeaderDictEntry	294
10.67.3 Constructor & Destructor Documentation	294
10.67.3.1 CSAHeaderDict() [1/2]	294
10.67.3.2 CSAHeaderDict() [2/2]	294
10.67.4 Member Function Documentation	294
10.67.4.1 AddCSAHeaderDictEntry()	294
10.67.4.2 Begin()	295
10.67.4.3 End()	295
10.67.4.4 GetCSAHeaderDictEntry()	295
10.67.4.5 IsEmpty()	295
10.67.4.6 LoadDefault()	295
10.67.4.7 operator=()	295
10.67.5 Friends And Related Symbol Documentation	295
10.67.5.1 Dicts	295
10.67.5.2 operator<<	296
10.68 gdcM::CSAHeaderDictEntry Class Reference	296
10.68.1 Detailed Description	297
10.68.2 Constructor & Destructor Documentation	297
10.68.2.1 CSAHeaderDictEntry()	297
10.68.3 Member Function Documentation	297
10.68.3.1 GetDescription()	297
10.68.3.2 GetName()	297
10.68.3.3 GetVM()	298
10.68.3.4 GetVR()	298
10.68.3.5 operator<()	298
10.68.3.6 SetDescription()	298
10.68.3.7 SetName()	298
10.68.3.8 SetVM()	298
10.68.3.9 SetVR()	298
10.68.4 Friends And Related Symbol Documentation	299
10.68.4.1 operator<<	299
10.69 gdcM::CSAHeaderDictException Class Reference	299

10.70 gdcm::network::CStoreRQ Class Reference	300
10.70.1 Detailed Description	301
10.70.2 Member Function Documentation	301
10.70.2.1 ConstructPDV()	301
10.71 gdcm::network::CStoreRSP Class Reference	301
10.71.1 Detailed Description	302
10.71.2 Member Function Documentation	302
10.71.2.1 ConstructPDV()	302
10.72 gdcm::Curve Class Reference	303
10.72.1 Detailed Description	304
10.72.2 Constructor & Destructor Documentation	305
10.72.2.1 Curve() [1/2]	305
10.72.2.2 ~Curve()	305
10.72.2.3 Curve() [2/2]	305
10.72.3 Member Function Documentation	305
10.72.3.1 Decode()	305
10.72.3.2 GetAsPoints()	305
10.72.3.3 GetCurveDataDescriptor()	305
10.72.3.4 GetDataValueRepresentation()	305
10.72.3.5 GetDimensions()	305
10.72.3.6 GetGroup()	306
10.72.3.7 GetNumberOfCurves()	306
10.72.3.8 GetNumberOfPoints()	306
10.72.3.9 GetTypeOfData()	306
10.72.3.10 GetTypeOfDataDescription()	306
10.72.3.11 IsEmpty()	306
10.72.3.12 Print()	306
10.72.3.13 SetCoordinateStartValue()	306
10.72.3.14 SetCoordinateStepValue()	306
10.72.3.15 SetCurve()	307
10.72.3.16 SetCurveDataDescriptor()	307
10.72.3.17 SetCurveDescription()	307
10.72.3.18 SetDataValueRepresentation()	307
10.72.3.19 SetDimensions()	307
10.72.3.20 SetGroup()	307
10.72.3.21 SetNumberOfPoints()	307
10.72.3.22 SetTypeOfData()	307
10.72.3.23 Update()	308
10.73 gdcm::DataElement Class Reference	308

10.73.1 Detailed Description	311
10.73.2 Member Typedef Documentation	311
10.73.2.1 ValuePtr	311
10.73.3 Constructor & Destructor Documentation	311
10.73.3.1 DataElement() [1/2]	311
10.73.3.2 DataElement() [2/2]	312
10.73.4 Member Function Documentation	312
10.73.4.1 Clear()	312
10.73.4.2 Empty()	312
10.73.4.3 GetByteValue()	312
10.73.4.4 GetLength()	312
10.73.4.5 GetSequenceOfFragments() [1/2]	313
10.73.4.6 GetSequenceOfFragments() [2/2]	313
10.73.4.7 GetTag() [1/2]	313
10.73.4.8 GetTag() [2/2]	313
10.73.4.9 GetValue() [1/2]	313
10.73.4.10 GetValue() [2/2]	314
10.73.4.11 GetValueAsSQ()	314
10.73.4.12 GetVL() [1/2]	314
10.73.4.13 GetVL() [2/2]	314
10.73.4.14 GetVR()	315
10.73.4.15 IsEmpty()	315
10.73.4.16 IsUndefinedLength()	315
10.73.4.17 operator<()	315
10.73.4.18 operator=()	316
10.73.4.19 operator==(())	316
10.73.4.20 Read()	316
10.73.4.21 ReadOrSkip()	316
10.73.4.22 ReadPreValue()	316
10.73.4.23 ReadValue()	316
10.73.4.24 ReadValueWithLength()	317
10.73.4.25 ReadWithLength()	317
10.73.4.26 SetByteValue()	317
10.73.4.27 SetTag()	318
10.73.4.28 SetValue()	318
10.73.4.29 SetValueFieldLength()	318
10.73.4.30 SetVL()	318
10.73.4.31 SetVLToUndefined()	319
10.73.4.32 SetVR()	319

10.73.4.33 Write()	319
10.73.5 Friends And Related Symbol Documentation	319
10.73.5.1 operator<<	319
10.73.6 Member Data Documentation	320
10.73.6.1 TagField	320
10.73.6.2 ValueField	320
10.73.6.3 ValueLengthField	320
10.73.6.4 VRField	320
10.74 gdcM::DataElementException Class Reference	321
10.75 gdcM::DataEvent Class Reference	321
10.75.1 Detailed Description	323
10.75.2 Member Typedef Documentation	323
10.75.2.1 Self	323
10.75.2.2 Superclass	323
10.75.3 Constructor & Destructor Documentation	323
10.75.3.1 DataEvent() [1/2]	323
10.75.3.2 ~DataEvent()	324
10.75.3.3 DataEvent() [2/2]	324
10.75.4 Member Function Documentation	324
10.75.4.1 CheckEvent()	324
10.75.4.2 GetData()	324
10.75.4.3 GetDataLength()	324
10.75.4.4 GetEventName()	324
10.75.4.5 MakeObject()	324
10.75.4.6 operator=()	325
10.75.4.7 SetData()	325
10.76 gdcM::DataSet Class Reference	325
10.76.1 Detailed Description	327
10.76.2 Member Typedef Documentation	328
10.76.2.1 ConstIterator	328
10.76.2.2 DataElementSet	328
10.76.2.3 Iterator	328
10.76.2.4 SizeType	328
10.76.3 Member Function Documentation	328
10.76.3.1 Begin() [1/2]	328
10.76.3.2 Begin() [2/2]	329
10.76.3.3 Clear()	329
10.76.3.4 ComputeDataElement()	329
10.76.3.5 ComputeGroupLength()	329

10.76.3.6 End() [1/2]	329
10.76.3.7 End() [2/2]	329
10.76.3.8 FindDataElement() [1/2]	330
10.76.3.9 FindDataElement() [2/2]	330
10.76.3.10 FindNextDataElement()	330
10.76.3.11 GetDataElement() [1/2]	330
10.76.3.12 GetDataElement() [2/2]	331
10.76.3.13 GetDEEnd()	331
10.76.3.14 GetDES() [1/2]	331
10.76.3.15 GetDES() [2/2]	331
10.76.3.16 GetLength()	331
10.76.3.17 GetMediaStorage()	332
10.76.3.18 GetPrivateCreator()	332
10.76.3.19 GetPrivateTag()	332
10.76.3.20 Insert()	332
10.76.3.21 InsertDataElement()	333
10.76.3.22 IsEmpty()	333
10.76.3.23 operator()()	333
10.76.3.24 operator=()	333
10.76.3.25 operator[]()	333
10.76.3.26 Print()	333
10.76.3.27 Read()	333
10.76.3.28 ReadNested()	334
10.76.3.29 ReadSelectedPrivateTags()	334
10.76.3.30 ReadSelectedPrivateTagsWithLength()	334
10.76.3.31 ReadSelectedTags()	334
10.76.3.32 ReadSelectedTagsWithLength()	334
10.76.3.33 ReadUpToTag()	334
10.76.3.34 ReadUpToTagWithLength()	335
10.76.3.35 ReadWithLength()	335
10.76.3.36 Remove()	335
10.76.3.37 Replace()	335
10.76.3.38 ReplaceEmpty()	336
10.76.3.39 Size()	336
10.76.3.40 Write()	336
10.76.4 Friends And Related Symbol Documentation	336
10.76.4.1 CSAHeader	336
10.76.4.2 operator<<	336
10.77 gdcm::DataSetEvent Class Reference	337

10.77.1 Detailed Description	338
10.77.2 Member Typedef Documentation	338
10.77.2.1 Self	338
10.77.2.2 Superclass	338
10.77.3 Constructor & Destructor Documentation	339
10.77.3.1 DataSetEvent() [1/2]	339
10.77.3.2 ~DataSetEvent()	339
10.77.3.3 DataSetEvent() [2/2]	339
10.77.4 Member Function Documentation	339
10.77.4.1 CheckEvent()	339
10.77.4.2 GetDataSet()	339
10.77.4.3 GetEventName()	339
10.77.4.4 MakeObject()	339
10.77.4.5 operator=()	340
10.77.5 Member Data Documentation	340
10.77.5.1 m_DataSet	340
10.78 gdcm::DataSetHelper Class Reference	340
10.78.1 Detailed Description	340
10.78.2 Member Function Documentation	341
10.78.2.1 ComputeVR()	341
10.79 gdcm::Decoder Class Reference	341
10.79.1 Detailed Description	342
10.79.2 Constructor & Destructor Documentation	342
10.79.2.1 ~Decoder()	342
10.79.3 Member Function Documentation	342
10.79.3.1 CanDecode()	342
10.79.3.2 Decode()	342
10.79.3.3 DecodeByStreams()	343
10.80 gdcm::DefinedTerms Class Reference	343
10.80.1 Detailed Description	343
10.80.2 Constructor & Destructor Documentation	343
10.80.2.1 DefinedTerms()	343
10.81 gdcm::Defs Class Reference	344
10.81.1 Detailed Description	344
10.81.2 Constructor & Destructor Documentation	345
10.81.2.1 Defs() [1/2]	345
10.81.2.2 ~Defs()	345
10.81.2.3 Defs() [2/2]	345
10.81.3 Member Function Documentation	345

10.81.3.1 GetIODFromFile()	345
10.81.3.2 GetIODNameFromMediaStorage()	345
10.81.3.3 GetIODs() [1/2]	345
10.81.3.4 GetIODs() [2/2]	345
10.81.3.5 GetMacros() [1/2]	346
10.81.3.6 GetMacros() [2/2]	346
10.81.3.7 GetModules() [1/2]	346
10.81.3.8 GetModules() [2/2]	346
10.81.3.9 GetTypeFromTag()	346
10.81.3.10 IsEmpty()	346
10.81.3.11 LoadDefaults()	346
10.81.3.12 LoadFromFile()	347
10.81.3.13 operator=()	347
10.81.3.14 Verify() [1/2]	347
10.81.3.15 Verify() [2/2]	347
10.81.4 Friends And Related Symbol Documentation	347
10.81.4.1 Global	347
10.82 gdcmm::DeltaEncodingCodec Class Reference	348
10.82.1 Detailed Description	350
10.82.2 Constructor & Destructor Documentation	351
10.82.2.1 DeltaEncodingCodec()	351
10.82.2.2 ~DeltaEncodingCodec()	351
10.82.3 Member Function Documentation	351
10.82.3.1 CanDecode()	351
10.82.3.2 Decode() [1/2]	351
10.82.3.3 Decode() [2/2]	351
10.83 gdcmm::DICOMDIR Class Reference	351
10.83.1 Detailed Description	352
10.83.2 Constructor & Destructor Documentation	352
10.83.2.1 DICOMDIR() [1/2]	352
10.83.2.2 DICOMDIR() [2/2]	352
10.84 gdcmm::DICOMDIRGenerator Class Reference	352
10.84.1 Detailed Description	353
10.84.2 Member Typedef Documentation	354
10.84.2.1 FilenamesType	354
10.84.2.2 FilenameType	354
10.84.3 Constructor & Destructor Documentation	354
10.84.3.1 DICOMDIRGenerator()	354
10.84.3.2 ~DICOMDIRGenerator()	354

10.84.4 Member Function Documentation	354
10.84.4.1 AddImageDirectoryRecord()	354
10.84.4.2 AddPatientDirectoryRecord()	354
10.84.4.3 AddSeriesDirectoryRecord()	354
10.84.4.4 AddStudyDirectoryRecord()	354
10.84.4.5 Generate()	355
10.84.4.6 GetFile()	355
10.84.4.7 GetScanner()	355
10.84.4.8 SetDescriptor()	355
10.84.4.9 SetFile()	355
10.84.4.10 SetFilenames()	356
10.84.4.11 SetRootDirectory()	356
10.85 gdcmm::Dict Class Reference	356
10.85.1 Detailed Description	357
10.85.2 Member Typedef Documentation	357
10.85.2.1 ConstIterator	357
10.85.2.2 Iterator	357
10.85.2.3 MapDictEntry	357
10.85.3 Constructor & Destructor Documentation	357
10.85.3.1 Dict() [1/2]	357
10.85.3.2 Dict() [2/2]	358
10.85.4 Member Function Documentation	358
10.85.4.1 AddDictEntry()	358
10.85.4.2 Begin()	358
10.85.4.3 End()	358
10.85.4.4 GetDictEntry()	358
10.85.4.5 GetDictEntryByKeyword()	359
10.85.4.6 GetDictEntryByName()	359
10.85.4.7 GetKeywordFromTag()	359
10.85.4.8 IsEmpty()	359
10.85.4.9 LoadDefault()	359
10.85.4.10 operator=()	359
10.85.5 Friends And Related Symbol Documentation	360
10.85.5.1 Dicts	360
10.85.5.2 operator<<	360
10.86 gdcmm::DictConverter Class Reference	360
10.86.1 Detailed Description	361
10.86.2 Member Enumeration Documentation	361
10.86.2.1 OutputTypes	361



10.86.3 Constructor & Destructor Documentation	361
10.86.3.1 DictConverter()	361
10.86.3.2 ~DictConverter()	362
10.86.4 Member Function Documentation	362
10.86.4.1 AddGroupLength()	362
10.86.4.2 Convert()	362
10.86.4.3 ConvertToCXX()	362
10.86.4.4 ConvertToXML()	362
10.86.4.5 GetDictName()	362
10.86.4.6 GetInputFilename()	362
10.86.4.7 GetOutputFilename()	362
10.86.4.8 GetOutputType()	363
10.86.4.9 Readuint16()	363
10.86.4.10 ReadVM()	363
10.86.4.11 ReadVR()	363
10.86.4.12 SetDictName()	363
10.86.4.13 SetInputFileName()	363
10.86.4.14 SetOutputFileName()	363
10.86.4.15 SetOutputType()	363
10.86.4.16 WriteFooter()	364
10.86.4.17 WriteHeader()	364
10.87 gdcmm::DictEntry Class Reference	364
10.87.1 Detailed Description	365
10.87.2 Constructor & Destructor Documentation	365
10.87.2.1 DictEntry()	365
10.87.3 Member Function Documentation	365
10.87.3.1 GetKeyword()	365
10.87.3.2 GetName()	366
10.87.3.3 GetRetired()	366
10.87.3.4 GetVM()	366
10.87.3.5 GetVR()	366
10.87.3.6 IsUnique()	366
10.87.3.7 SetElementXX()	367
10.87.3.8 SetGroupXX()	367
10.87.3.9 SetKeyword()	367
10.87.3.10 SetName()	367
10.87.3.11 SetRetired()	367
10.87.3.12 SetVM()	367
10.87.3.13 SetVR()	367

10.87.4 Friends And Related Symbol Documentation	368
10.87.4.1 Dict	368
10.87.4.2 operator<<	368
10.88 gdcm::DictPrinter Class Reference	368
10.88.1 Detailed Description	370
10.88.2 Constructor & Destructor Documentation	370
10.88.2.1 DictPrinter()	370
10.88.2.2 ~DictPrinter()	370
10.88.3 Member Function Documentation	371
10.88.3.1 Print()	371
10.88.3.2 PrintDataElement2()	371
10.88.3.3 PrintDataSet2()	371
10.89 gdcm::Dicts Class Reference	371
10.89.1 Detailed Description	372
10.89.2 Member Enumeration Documentation	372
10.89.2.1 ConstructorType	372
10.89.3 Constructor & Destructor Documentation	373
10.89.3.1 Dicts() [1/2]	373
10.89.3.2 ~Dicts()	373
10.89.3.3 Dicts() [2/2]	373
10.89.4 Member Function Documentation	373
10.89.4.1 GetConstructorString()	373
10.89.4.2 GetCSAHeaderDict()	373
10.89.4.3 GetDictEntry() [1/2]	373
10.89.4.4 GetDictEntry() [2/2]	374
10.89.4.5 GetPrivateDict() [1/2]	374
10.89.4.6 GetPrivateDict() [2/2]	374
10.89.4.7 GetPublicDict()	374
10.89.4.8 IsEmpty()	374
10.89.4.9 LoadDefaults()	374
10.89.4.10 operator=()	375
10.89.5 Friends And Related Symbol Documentation	375
10.89.5.1 Global	375
10.89.5.2 operator<<	375
10.90 gdcm::network::DIMSE Class Reference	375
10.90.1 Detailed Description	376
10.90.2 Member Enumeration Documentation	376
10.90.2.1 CommandTypes	376
10.91 gdcm::DirectionCosines Class Reference	376

10.91.1 Detailed Description	377
10.91.2 Constructor & Destructor Documentation	377
10.91.2.1 DirectionCosines() [1/2]	377
10.91.2.2 DirectionCosines() [2/2]	378
10.91.2.3 ~DirectionCosines()	378
10.91.3 Member Function Documentation	378
10.91.3.1 ComputeDistAlongNormal()	378
10.91.3.2 Cross()	378
10.91.3.3 CrossDot()	378
10.91.3.4 Dot() [1/2]	378
10.91.3.5 Dot() [2/2]	379
10.91.3.6 IsValid()	379
10.91.3.7 Norm()	379
10.91.3.8 Normalize() [1/2]	379
10.91.3.9 Normalize() [2/2]	379
10.91.3.10 operator const double *()	379
10.91.3.11 Print()	379
10.91.3.12 SetFromString()	380
10.92 gdcmm::Directory Class Reference	380
10.92.1 Detailed Description	381
10.92.2 Member Typedef Documentation	381
10.92.2.1 FilenamesType	381
10.92.2.2 FilenameType	381
10.92.3 Constructor & Destructor Documentation	381
10.92.3.1 Directory()	381
10.92.3.2 ~Directory()	381
10.92.4 Member Function Documentation	382
10.92.4.1 Explore()	382
10.92.4.2 GetDirectories()	382
10.92.4.3 GetFilenames()	382
10.92.4.4 GetToplevel()	382
10.92.4.5 Load()	382
10.92.4.6 Print()	383
10.92.5 Friends And Related Symbol Documentation	383
10.92.5.1 operator<<	383
10.93 gdcmm::DirectoryHelper Class Reference	383
10.93.1 Detailed Description	384
10.93.2 Member Function Documentation	384
10.93.2.1 GetCTImageSeriesUIDs()	384

10.93.2.2 GetFileNamesFromSeriesUIDs()	384
10.93.2.3 GetFrameOfReference()	384
10.93.2.4 GetMRImageSeriesUIDs()	384
10.93.2.5 GetRTStructSeriesUIDs()	384
10.93.2.6 GetSeriesUIDsBySOPClassUID()	384
10.93.2.7 GetSOPClassUID()	385
10.93.2.8 GetStringValueFromTag()	385
10.93.2.9 LoadImageFromFiles()	385
10.93.2.10 RetrieveSOPInstanceUIDFromIndex()	385
10.93.2.11 RetrieveSOPInstanceUIDFromZPosition()	385
10.94 gdcmm::DPath Class Reference	385
10.94.1 Detailed Description	386
10.94.2 Constructor & Destructor Documentation	386
10.94.2.1 DPath()	386
10.94.2.2 ~DPath()	386
10.94.3 Member Function Documentation	387
10.94.3.1 ConstructFromString()	387
10.94.3.2 IsValid()	387
10.94.3.3 Match()	387
10.94.3.4 operator<()	387
10.94.3.5 Print()	387
10.94.4 Friends And Related Symbol Documentation	387
10.94.4.1 operator<<	387
10.95 gdcmm::DummyValueGenerator Class Reference	388
10.95.1 Detailed Description	388
10.95.2 Member Function Documentation	388
10.95.2.1 Generate()	388
10.96 gdcmm::Dumper Class Reference	389
10.96.1 Detailed Description	390
10.96.2 Constructor & Destructor Documentation	391
10.96.2.1 Dumper()	391
10.96.2.2 ~Dumper()	391
10.97 gdcmm::Element< TVR, TVM > Class Template Reference	391
10.97.1 Detailed Description	393
10.97.2 Member Typedef Documentation	393
10.97.2.1 Type	393
10.97.3 Member Function Documentation	393
10.97.3.1 GetAsDataElement()	393
10.97.3.2 GetLength()	394

10.97.3.3 GetValue() [1/2]	394
10.97.3.4 GetValue() [2/2]	394
10.97.3.5 GetValues()	394
10.97.3.6 GetVM()	394
10.97.3.7 GetVR()	394
10.97.3.8 operator[]()	395
10.97.3.9 Print()	395
10.97.3.10 Read()	395
10.97.3.11 Set()	395
10.97.3.12 SetFromDataElement()	395
10.97.3.13 SetNoSwap()	396
10.97.3.14 SetValue()	396
10.97.3.15 Write()	396
10.97.4 Member Data Documentation	396
10.97.4.1 Internal	396
10.98 gdcmm::Element< TVR, VM::VM1_2 > Class Template Reference	397
10.98.1 Member Typedef Documentation	398
10.98.1.1 Parent	398
10.98.2 Member Function Documentation	399
10.98.2.1 SetLength()	399
10.99 gdcmm::Element< TVR, VM::VM1_n > Class Template Reference	399
10.99.1 Member Typedef Documentation	400
10.99.1.1 Type	400
10.99.2 Constructor & Destructor Documentation	400
10.99.2.1 Element() [1/2]	400
10.99.2.2 ~Element()	400
10.99.2.3 Element() [2/2]	401
10.99.3 Member Function Documentation	401
10.99.3.1 GetAsDataElement()	401
10.99.3.2 GetLength()	401
10.99.3.3 GetValue() [1/2]	401
10.99.3.4 GetValue() [2/2]	401
10.99.3.5 GetVM()	401
10.99.3.6 GetVR()	401
10.99.3.7 operator=()	402
10.99.3.8 operator[]()	402
10.99.3.9 Print()	402
10.99.3.10 Read()	402
10.99.3.11 Set()	402

10.99.3.12 SetArray()	402
10.99.3.13 SetFromDataElement()	403
10.99.3.14 SetLength()	403
10.99.3.15 SetNoSwap()	403
10.99.3.16 SetValue()	403
10.99.3.17 Write()	403
10.99.3.18 WriteASCII()	403
10.100 gdcmm::Element< TVR, VM::VM2_2n > Class Template Reference	404
10.100.1 Member Typedef Documentation	406
10.100.1.1 Parent	406
10.100.2 Member Function Documentation	406
10.100.2.1 SetLength()	406
10.101 gdcmm::Element< TVR, VM::VM2_n > Class Template Reference	406
10.101.1 Member Typedef Documentation	408
10.101.1.1 Parent	408
10.101.2 Member Function Documentation	408
10.101.2.1 SetLength()	408
10.102 gdcmm::Element< TVR, VM::VM3_3n > Class Template Reference	408
10.102.1 Member Typedef Documentation	411
10.102.1.1 Parent	411
10.102.2 Member Function Documentation	411
10.102.2.1 SetLength()	411
10.103 gdcmm::Element< TVR, VM::VM3_4 > Class Template Reference	411
10.103.1 Member Typedef Documentation	413
10.103.1.1 Parent	413
10.103.2 Member Function Documentation	413
10.103.2.1 SetLength()	413
10.104 gdcmm::Element< TVR, VM::VM3_n > Class Template Reference	413
10.104.1 Member Typedef Documentation	415
10.104.1.1 Parent	415
10.104.2 Member Function Documentation	416
10.104.2.1 SetLength()	416
10.105 gdcmm::Element< VR::AS, VM::VM5 > Class Reference	416
10.105.1 Member Function Documentation	416
10.105.1.1 GetLength()	416
10.105.1.2 Print()	416
10.105.2 Member Data Documentation	417
10.105.2.1 Internal	417
10.106 gdcmm::Element< VR::OB, VM::VM1 > Class Reference	417

10.107 gdcM::Element< VR::OW, VM::VM1 > Class Reference . . . . .	419
10.108 gdcM::ElementDisableCombinations< TVR, TVM > Class Template Reference . . . . .	421
10.108.1 Detailed Description . . . . .	421
10.109 gdcM::ElementDisableCombinations< VR::OB, VM::VM1_n > Class Reference . . . . .	422
10.110 gdcM::ElementDisableCombinations< VR::OW, VM::VM1_n > Class Reference . . . . .	422
10.111 gdcM::EmptyMaskGenerator Class Reference . . . . .	422
10.111.1 Detailed Description . . . . .	423
10.111.2 Member Enumeration Documentation . . . . .	423
10.111.2.1 SOPClassUIDMode . . . . .	423
10.111.3 Constructor & Destructor Documentation . . . . .	424
10.111.3.1 EmptyMaskGenerator() . . . . .	424
10.111.3.2 ~EmptyMaskGenerator() . . . . .	424
10.111.4 Member Function Documentation . . . . .	424
10.111.4.1 Execute() . . . . .	424
10.111.4.2 SetInputDirectory() . . . . .	424
10.111.4.3 SetOutputDirectory() . . . . .	424
10.111.4.4 SetSOPClassUIDMode() . . . . .	425
10.112 gdcM::EncapsulatedDocument Class Reference . . . . .	425
10.112.1 Detailed Description . . . . .	425
10.112.2 Constructor & Destructor Documentation . . . . .	425
10.112.2.1 EncapsulatedDocument() . . . . .	425
10.113 gdcM::EncodingImplementation< T > Class Template Reference . . . . .	425
10.113.1 Detailed Description . . . . .	426
10.114 gdcM::EncodingImplementation< VR::VRASCII > Class Reference . . . . .	426
10.114.1 Member Function Documentation . . . . .	426
10.114.1.1 Read() . . . . .	426
10.114.1.2 ReadComputeLength() . . . . .	427
10.114.1.3 ReadNoSwap() . . . . .	427
10.114.1.4 Write() [1/2] . . . . .	427
10.114.1.5 Write() [2/2] . . . . .	427
10.115 gdcM::EncodingImplementation< VR::VRBINARY > Class Reference . . . . .	427
10.115.1 Member Function Documentation . . . . .	428
10.115.1.1 Read() . . . . .	428
10.115.1.2 ReadComputeLength() . . . . .	428
10.115.1.3 ReadNoSwap() . . . . .	428
10.115.1.4 Write() . . . . .	428
10.116 gdcM::EndEvent Class Reference . . . . .	429
10.117 gdcM::EnumeratedValues Class Reference . . . . .	430
10.117.1 Detailed Description . . . . .	430

10.117.2 Constructor & Destructor Documentation	430
10.117.2.1 EnumeratedValues()	430
10.118 gdcmm::EquipmentManufacturer Class Reference	431
10.118.1 Detailed Description	431
10.118.2 Member Enumeration Documentation	431
10.118.2.1 Type	431
10.118.3 Member Function Documentation	432
10.118.3.1 Compute()	432
10.118.3.2 ToString()	432
10.119 gdcmm::Event Class Reference	432
10.119.1 Detailed Description	434
10.119.2 Constructor & Destructor Documentation	434
10.119.2.1 Event() [1/2]	434
10.119.2.2 ~Event()	434
10.119.2.3 Event() [2/2]	434
10.119.3 Member Function Documentation	434
10.119.3.1 CheckEvent()	434
10.119.3.2 GetEventName()	434
10.119.3.3 MakeObject()	435
10.119.3.4 operator=()	435
10.119.3.5 Print()	435
10.120 gdcmm::Exception Class Reference	435
10.120.1 Detailed Description	436
10.120.2 Constructor & Destructor Documentation	436
10.120.2.1 Exception()	436
10.120.2.2 ~Exception()	437
10.120.3 Member Function Documentation	437
10.120.3.1 GetDescription()	437
10.120.3.2 what()	437
10.121 gdcmm::ExitEvent Class Reference	438
10.122 gdcmm::ExplicitDataElement Class Reference	439
10.122.1 Detailed Description	442
10.122.2 Member Function Documentation	442
10.122.2.1 GetLength()	442
10.122.2.2 Read()	442
10.122.2.3 ReadPreValue()	442
10.122.2.4 ReadValue()	442
10.122.2.5 ReadWithLength()	443
10.122.2.6 Write()	443



10.123 gdcmm::ExplicitImplicitDataElement Class Reference . . . . .	443
10.123.1 Detailed Description . . . . .	446
10.123.2 Member Function Documentation . . . . .	446
10.123.2.1 GetLength() . . . . .	446
10.123.2.2 Read() . . . . .	446
10.123.2.3 ReadPreValue() . . . . .	446
10.123.2.4 ReadValue() . . . . .	446
10.123.2.5 ReadWithLength() . . . . .	447
10.124 gdcmm::Fiducials Class Reference . . . . .	447
10.124.1 Detailed Description . . . . .	447
10.124.2 Constructor & Destructor Documentation . . . . .	447
10.124.2.1 Fiducials() . . . . .	447
10.125 gdcmm::File Class Reference . . . . .	448
10.125.1 Detailed Description . . . . .	449
10.125.2 Constructor & Destructor Documentation . . . . .	450
10.125.2.1 File() . . . . .	450
10.125.2.2 ~File() . . . . .	450
10.125.3 Member Function Documentation . . . . .	450
10.125.3.1 GetDataSet() [1/2] . . . . .	450
10.125.3.2 GetDataSet() [2/2] . . . . .	451
10.125.3.3 GetHeader() [1/2] . . . . .	451
10.125.3.4 GetHeader() [2/2] . . . . .	451
10.125.3.5 Read() . . . . .	451
10.125.3.6 SetDataSet() . . . . .	452
10.125.3.7 SetHeader() . . . . .	452
10.125.3.8 Write() . . . . .	452
10.125.4 Friends And Related Symbol Documentation . . . . .	452
10.125.4.1 operator<< . . . . .	452
10.126 gdcmm::FileAnonymizer Class Reference . . . . .	453
10.126.1 Detailed Description . . . . .	455
10.126.2 Constructor & Destructor Documentation . . . . .	455
10.126.2.1 FileAnonymizer() . . . . .	455
10.126.2.2 ~FileAnonymizer() . . . . .	455
10.126.3 Member Function Documentation . . . . .	455
10.126.3.1 Empty() . . . . .	455
10.126.3.2 Remove() . . . . .	456
10.126.3.3 Replace() [1/2] . . . . .	456
10.126.3.4 Replace() [2/2] . . . . .	456
10.126.3.5 SetInputFileName() . . . . .	456

10.126.3.6 SetOutputFileName()	457
10.126.3.7 Write()	457
10.127 gdcm::FileChangeTransferSyntax Class Reference	457
10.127.1 Detailed Description	459
10.127.2 Constructor & Destructor Documentation	459
10.127.2.1 FileChangeTransferSyntax()	459
10.127.2.2 ~FileChangeTransferSyntax()	460
10.127.3 Member Function Documentation	460
10.127.3.1 Change()	460
10.127.3.2 GetCodec()	460
10.127.3.3 New()	460
10.127.3.4 SetInputFileName()	460
10.127.3.5 SetOutputFileName()	461
10.127.3.6 SetTransferSyntax()	461
10.128 gdcm::FileDecompressLookupTable Class Reference	461
10.128.1 Detailed Description	463
10.128.2 Constructor & Destructor Documentation	463
10.128.2.1 FileDecompressLookupTable()	463
10.128.2.2 ~FileDecompressLookupTable()	463
10.128.3 Member Function Documentation	463
10.128.3.1 Change()	463
10.128.3.2 GetFile()	464
10.128.3.3 GetPixmap() [1/2]	464
10.128.3.4 GetPixmap() [2/2]	464
10.128.3.5 SetFile()	464
10.128.3.6 SetPixmap()	464
10.129 gdcm::FileDerivation Class Reference	464
10.129.1 Detailed Description	465
10.129.2 Constructor & Destructor Documentation	465
10.129.2.1 FileDerivation()	465
10.129.2.2 ~FileDerivation()	466
10.129.3 Member Function Documentation	466
10.129.3.1 AddDerivationDescription()	466
10.129.3.2 AddPurposeOfReferenceCodeSequence()	466
10.129.3.3 AddReference()	466
10.129.3.4 AddSourceImageSequence()	466
10.129.3.5 Derive()	466
10.129.3.6 GetFile() [1/2]	467
10.129.3.7 GetFile() [2/2]	467

10.129.3.8 SetAppendDerivationHistory()	467
10.129.3.9 SetDerivationCodeSequenceCodeValue()	467
10.129.3.10 SetDerivationDescription()	467
10.129.3.11 SetFile()	468
10.129.3.12 SetPurposeOfReferenceCodeSequenceCodeValue()	468
10.130 gdcm::FileExplicitFilter Class Reference	468
10.130.1 Detailed Description	469
10.130.2 Constructor & Destructor Documentation	469
10.130.2.1 FileExplicitFilter()	469
10.130.2.2 ~FileExplicitFilter()	469
10.130.3 Member Function Documentation	469
10.130.3.1 Change()	469
10.130.3.2 ChangeFMI()	470
10.130.3.3 GetFile()	470
10.130.3.4 ProcessDataSet()	470
10.130.3.5 SetChangePrivateTags()	470
10.130.3.6 SetFile()	470
10.130.3.7 SetRecomputeItemLength()	470
10.130.3.8 SetRecomputeSequenceLength()	470
10.130.3.9 SetUseVRUN()	471
10.131 gdcm::FileMetaInformation Class Reference	471
10.131.1 Detailed Description	475
10.131.2 Constructor & Destructor Documentation	475
10.131.2.1 FileMetaInformation() [1/2]	475
10.131.2.2 ~FileMetaInformation()	475
10.131.2.3 FileMetaInformation() [2/2]	475
10.131.3 Member Function Documentation	475
10.131.3.1 AppendImplementationClassUID()	475
10.131.3.2 ComputeDataSetMediaStorageSOPClass()	476
10.131.3.3 ComputeDataSetTransferSyntax()	476
10.131.3.4 Default()	476
10.131.3.5 FillFromDataSet()	476
10.131.3.6 GetDataSetTransferSyntax()	476
10.131.3.7 GetFileMetaInformationVersion()	476
10.131.3.8 GetFullLength()	476
10.131.3.9 GetGDCMImplementationClassUID()	476
10.131.3.10 GetGDCMImplementationVersionName()	477
10.131.3.11 GetGDCMSourceApplicationEntityTitle()	477
10.131.3.12 GetImplementationClassUID()	477

10.131.3.13 GetImplementationVersionName()	477
10.131.3.14 GetMediaStorage()	477
10.131.3.15 GetMediaStorageAsString()	477
10.131.3.16 GetMetaInformationTS()	477
10.131.3.17 GetPreamble() [1/2]	477
10.131.3.18 GetPreamble() [2/2]	477
10.131.3.19 GetSourceApplicationEntityTitle()	478
10.131.3.20 Insert()	478
10.131.3.21 IsValid()	478
10.131.3.22 operator=()	478
10.131.3.23 Read()	478
10.131.3.24 ReadCompat()	478
10.131.3.25 ReadCompatInternal()	478
10.131.3.26 Replace()	479
10.131.3.27 SetDataSetTransferSyntax()	479
10.131.3.28 SetImplementationClassUID()	479
10.131.3.29 SetImplementationVersionName()	479
10.131.3.30 SetPreamble()	479
10.131.3.31 SetSourceApplicationEntityTitle()	480
10.131.3.32 Write()	480
10.131.4 Friends And Related Symbol Documentation	480
10.131.4.1 operator<<	480
10.131.5 Member Data Documentation	480
10.131.5.1 DataSetMS	480
10.131.5.2 DataSetTS	480
10.131.5.3 MetaInformationTS	480
10.132 gdcm::Filename Class Reference	481
10.132.1 Detailed Description	481
10.132.2 Constructor & Destructor Documentation	482
10.132.2.1 Filename()	482
10.132.3 Member Function Documentation	482
10.132.3.1 EndWith()	482
10.132.3.2 GetExtension()	482
10.132.3.3 GetFileName()	482
10.132.3.4 GetName()	482
10.132.3.5 GetPath()	482
10.132.3.6 IsEmpty()	483
10.132.3.7 IsIdentical()	483
10.132.3.8 Join()	483

10.132.3.9 operator const char *()	483
10.132.3.10 ToUnixSlashes()	483
10.132.3.11 ToWindowsSlashes()	483
10.133 gdcM::FileNameEvent Class Reference	484
10.133.1 Detailed Description	485
10.133.2 Member Typedef Documentation	485
10.133.2.1 Self	485
10.133.2.2 Superclass	485
10.133.3 Constructor & Destructor Documentation	486
10.133.3.1 FileNameEvent() [1/2]	486
10.133.3.2 ~FileNameEvent()	486
10.133.3.3 FileNameEvent() [2/2]	486
10.133.4 Member Function Documentation	486
10.133.4.1 CheckEvent()	486
10.133.4.2 GetEventName()	486
10.133.4.3 GetFileName()	486
10.133.4.4 MakeObject()	487
10.133.4.5 operator=()	487
10.133.4.6 SetFileName()	487
10.134 gdcM::FilenameGenerator Class Reference	487
10.134.1 Detailed Description	488
10.134.2 Member Typedef Documentation	488
10.134.2.1 FilenamesType	488
10.134.2.2 FilenameType	488
10.134.2.3 SizeType	489
10.134.3 Constructor & Destructor Documentation	489
10.134.3.1 FilenameGenerator()	489
10.134.3.2 ~FilenameGenerator()	489
10.134.4 Member Function Documentation	489
10.134.4.1 Generate()	489
10.134.4.2 GetFilename()	489
10.134.4.3 GetFilenames()	489
10.134.4.4 GetNumberOfFilenames()	490
10.134.4.5 GetPattern()	490
10.134.4.6 GetPrefix()	490
10.134.4.7 SetNumberOfFilenames()	490
10.134.4.8 SetPattern()	490
10.134.4.9 SetPrefix()	491
10.135 gdcM::FileSet Class Reference	491

10.135.1 Detailed Description	491
10.135.2 Member Typedef Documentation	491
10.135.2.1 FileType	491
10.135.2.2 FileType	492
10.135.3 Constructor & Destructor Documentation	492
10.135.3.1 FileSet()	492
10.135.4 Member Function Documentation	492
10.135.4.1 AddFile() [1/2]	492
10.135.4.2 AddFile() [2/2]	492
10.135.4.3 GetFiles()	492
10.135.4.4 SetFiles()	492
10.135.5 Friends And Related Symbol Documentation	493
10.135.5.1 operator<<	493
10.136 gdcm::FileStreamer Class Reference	493
10.136.1 Detailed Description	495
10.136.2 Constructor & Destructor Documentation	496
10.136.2.1 FileStreamer()	496
10.136.2.2 ~FileStreamer()	496
10.136.3 Member Function Documentation	496
10.136.3.1 AppendToDataElement()	496
10.136.3.2 AppendToGroupDataElement()	496
10.136.3.3 CheckDataElement()	496
10.136.3.4 CheckTemplateFileName()	497
10.136.3.5 New()	497
10.136.3.6 ReserveDataElement()	497
10.136.3.7 ReserveGroupDataElement()	497
10.136.3.8 SetOutputFileName()	497
10.136.3.9 SetTemplateFileName()	498
10.136.3.10 StartDataElement()	498
10.136.3.11 StartGroupDataElement()	498
10.136.3.12 StopDataElement()	498
10.136.3.13 StopGroupDataElement()	499
10.137 gdcm::FileWithName Class Reference	499
10.137.1 Detailed Description	501
10.137.2 Constructor & Destructor Documentation	501
10.137.2.1 FileWithName()	501
10.137.3 Member Data Documentation	501
10.137.3.1 filename	501
10.138 gdcm::FindPatientRootQuery Class Reference	502

10.138.1 Detailed Description . . . . .	504
10.138.2 Constructor & Destructor Documentation . . . . .	504
10.138.2.1 FindPatientRootQuery() . . . . .	504
10.138.3 Member Function Documentation . . . . .	504
10.138.3.1 GetAbstractSyntaxUID() . . . . .	504
10.138.3.2 GetTagListByLevel() . . . . .	505
10.138.3.3 InitializeDataSet() . . . . .	505
10.138.3.4 ValidateQuery() . . . . .	505
10.138.4 Friends And Related Symbol Documentation . . . . .	505
10.138.4.1 QueryFactory . . . . .	505
10.139 gdcmm::FindStudyRootQuery Class Reference . . . . .	506
10.139.1 Detailed Description . . . . .	508
10.139.2 Constructor & Destructor Documentation . . . . .	508
10.139.2.1 FindStudyRootQuery() . . . . .	508
10.139.3 Member Function Documentation . . . . .	508
10.139.3.1 GetAbstractSyntaxUID() . . . . .	508
10.139.3.2 GetTagListByLevel() . . . . .	509
10.139.3.3 InitializeDataSet() . . . . .	509
10.139.3.4 ValidateQuery() . . . . .	509
10.139.4 Friends And Related Symbol Documentation . . . . .	509
10.139.4.1 QueryFactory . . . . .	509
10.140 gdcmm::Fragment Class Reference . . . . .	510
10.140.1 Detailed Description . . . . .	512
10.140.2 Constructor & Destructor Documentation . . . . .	513
10.140.2.1 Fragment() . . . . .	513
10.140.3 Member Function Documentation . . . . .	513
10.140.3.1 ComputeLength() . . . . .	513
10.140.3.2 GetLength() . . . . .	513
10.140.3.3 Read() . . . . .	513
10.140.3.4 ReadBacktrack() . . . . .	513
10.140.3.5 ReadPreValue() . . . . .	513
10.140.3.6 ReadValue() . . . . .	514
10.140.3.7 Write() . . . . .	514
10.140.4 Friends And Related Symbol Documentation . . . . .	514
10.140.4.1 operator<< . . . . .	514
10.141 gdcmm::Global Class Reference . . . . .	514
10.141.1 Detailed Description . . . . .	515
10.141.2 Constructor & Destructor Documentation . . . . .	515
10.141.2.1 Global() [1/2] . . . . .	515

10.141.2.2 ~Global()	515
10.141.2.3 Global() [2/2]	515
10.141.3 Member Function Documentation	516
10.141.3.1 Append()	516
10.141.3.2 GetDefs()	516
10.141.3.3 GetDicts() [1/2]	516
10.141.3.4 GetDicts() [2/2]	516
10.141.3.5 GetInstance()	516
10.141.3.6 LoadResourcesFiles()	517
10.141.3.7 Locate()	517
10.141.3.8 operator=()	517
10.141.3.9 Prepend()	517
10.141.4 Friends And Related Symbol Documentation	517
10.141.4.1 operator<<	517
10.142 gdcm::GroupDict Class Reference	518
10.142.1 Detailed Description	518
10.142.2 Member Typedef Documentation	518
10.142.2.1 GroupStringVector	518
10.142.3 Constructor & Destructor Documentation	519
10.142.3.1 GroupDict()	519
10.142.3.2 ~GroupDict()	519
10.142.4 Member Function Documentation	519
10.142.4.1 Add()	519
10.142.4.2 GetAbbreviation()	519
10.142.4.3 GetName()	519
10.142.4.4 Insert()	519
10.142.4.5 Size()	519
10.142.5 Friends And Related Symbol Documentation	520
10.142.5.1 operator<<	520
10.143 gdcm::IconImageFilter Class Reference	520
10.143.1 Detailed Description	521
10.143.2 Constructor & Destructor Documentation	521
10.143.2.1 IconImageFilter()	521
10.143.2.2 ~IconImageFilter()	521
10.143.3 Member Function Documentation	522
10.143.3.1 Extract()	522
10.143.3.2 ExtractIconImages()	522
10.143.3.3 ExtractVeprolIconImages()	522
10.143.3.4 GetFile() [1/2]	522



10.143.3.5	GetFile() [2/2]	522
10.143.3.6	GetIconImage()	522
10.143.3.7	GetNumberOfIconImages()	523
10.143.3.8	SetFile()	523
10.144	gdcm::IconImageGenerator Class Reference	523
10.144.1	Detailed Description	524
10.144.2	Constructor & Destructor Documentation	524
10.144.2.1	IconImageGenerator()	524
10.144.2.2	~IconImageGenerator()	524
10.144.3	Member Function Documentation	524
10.144.3.1	AutoPixelMinMax()	524
10.144.3.2	ConvertRGBToPaletteColor()	525
10.144.3.3	Generate()	525
10.144.3.4	GetIconImage()	525
10.144.3.5	GetPixmap() [1/2]	525
10.144.3.6	GetPixmap() [2/2]	525
10.144.3.7	SetOutputDimensions()	525
10.144.3.8	SetOutsideValuePixel()	526
10.144.3.9	SetPixelMinMax()	526
10.144.3.10	SetPixmap()	526
10.145	gdcm::ignore_char Struct Reference	526
10.145.1	Constructor & Destructor Documentation	527
10.145.1.1	ignore_char()	527
10.145.2	Member Data Documentation	527
10.145.2.1	m_char	527
10.146	gdcm::Image Class Reference	527
10.146.1	Detailed Description	532
10.146.2	Constructor & Destructor Documentation	532
10.146.2.1	Image()	532
10.146.2.2	~Image()	532
10.146.3	Member Function Documentation	533
10.146.3.1	GetDirectionCosines() [1/2]	533
10.146.3.2	GetDirectionCosines() [2/2]	533
10.146.3.3	GetIntercept()	533
10.146.3.4	GetOrigin() [1/2]	533
10.146.3.5	GetOrigin() [2/2]	533
10.146.3.6	GetSlope()	533
10.146.3.7	GetSpacing() [1/2]	533
10.146.3.8	GetSpacing() [2/2]	534

10.146.3.9 Print()	534
10.146.3.10 SetDirectionCosines() [1/3]	534
10.146.3.11 SetDirectionCosines() [2/3]	534
10.146.3.12 SetDirectionCosines() [3/3]	534
10.146.3.13 SetIntercept()	534
10.146.3.14 SetOrigin() [1/3]	535
10.146.3.15 SetOrigin() [2/3]	535
10.146.3.16 SetOrigin() [3/3]	535
10.146.3.17 SetSlope()	535
10.146.3.18 SetSpacing() [1/2]	535
10.146.3.19 SetSpacing() [2/2]	535
10.147 gdcm::ImageApplyLookupTable Class Reference	536
10.147.1 Detailed Description	538
10.147.2 Constructor & Destructor Documentation	538
10.147.2.1 ImageApplyLookupTable()	538
10.147.2.2 ~ImageApplyLookupTable()	539
10.147.3 Member Function Documentation	539
10.147.3.1 Apply()	539
10.147.3.2 SetRGB8()	539
10.148 gdcm::ImageChangePhotometricInterpretation Class Reference	539
10.148.1 Detailed Description	542
10.148.2 Constructor & Destructor Documentation	542
10.148.2.1 ImageChangePhotometricInterpretation()	542
10.148.2.2 ~ImageChangePhotometricInterpretation()	542
10.148.3 Member Function Documentation	542
10.148.3.1 Change()	542
10.148.3.2 ChangeMonochrome()	542
10.148.3.3 ChangeRGB2YBR()	542
10.148.3.4 ChangeYBR2RGB()	542
10.148.3.5 GetPhotometricInterpretation()	542
10.148.3.6 RGB2YBR()	543
10.148.3.7 SetPhotometricInterpretation()	543
10.148.3.8 YBR2RGB()	543
10.149 gdcm::ImageChangePlanarConfiguration Class Reference	543
10.149.1 Detailed Description	546
10.149.2 Constructor & Destructor Documentation	547
10.149.2.1 ImageChangePlanarConfiguration()	547
10.149.2.2 ~ImageChangePlanarConfiguration()	547
10.149.3 Member Function Documentation	547

10.149.3.1 Change()	547
10.149.3.2 GetPlanarConfiguration()	547
10.149.3.3 RGBPixelsToRGBPlanes()	547
10.149.3.4 RGBPlanesToRGBPixels()	548
10.149.3.5 SetPlanarConfiguration()	548
10.150 gdcmm::ImageChangeTransferSyntax Class Reference	548
10.150.1 Detailed Description	551
10.150.2 Constructor & Destructor Documentation	551
10.150.2.1 ImageChangeTransferSyntax()	551
10.150.2.2 ~ImageChangeTransferSyntax()	551
10.150.3 Member Function Documentation	551
10.150.3.1 Change()	551
10.150.3.2 GetTransferSyntax()	552
10.150.3.3 SetCompressIconImage()	552
10.150.3.4 SetForce()	552
10.150.3.5 SetTransferSyntax()	552
10.150.3.6 SetUserCodec()	553
10.150.3.7 TryJPEG2000Codec()	553
10.150.3.8 TryJPEGCodec()	553
10.150.3.9 TryJPEGLSCodec()	553
10.150.3.10 TryRAWCodec()	553
10.150.3.11 TryRLECodec()	554
10.151 gdcmm::ImageCodec Class Reference	554
10.151.1 Detailed Description	557
10.151.2 Member Typedef Documentation	557
10.151.2.1 LUTPtr	557
10.151.3 Constructor & Destructor Documentation	557
10.151.3.1 ImageCodec()	557
10.151.3.2 ~ImageCodec()	557
10.151.4 Member Function Documentation	557
10.151.4.1 AppendFrameEncode()	557
10.151.4.2 AppendRowEncode()	558
10.151.4.3 CanCode()	558
10.151.4.4 CanDecode()	558
10.151.4.5 CleanupUnusedBits()	558
10.151.4.6 Clone()	558
10.151.4.7 Decode()	559
10.151.4.8 DecodeByStreams()	559
10.151.4.9 DoByteSwap()	559

10.151.4.10 DoInvertMonochrome()	559
10.151.4.11 DoOverlayCleanup()	559
10.151.4.12 DoPaddedCompositePixelCode()	559
10.151.4.13 DoPlanarConfiguration()	560
10.151.4.14 DoSimpleCopy()	560
10.151.4.15 DoYBR()	560
10.151.4.16 DoYBRFull422()	560
10.151.4.17 GetDimensions()	560
10.151.4.18 GetHeaderInfo()	560
10.151.4.19 GetLossyFlag()	560
10.151.4.20 GetLUT()	561
10.151.4.21 GetNeedByteSwap()	561
10.151.4.22 GetNumberOfDimensions()	561
10.151.4.23 GetPhotometricInterpretation()	561
10.151.4.24 GetPixelFormat() [1/2]	561
10.151.4.25 GetPixelFormat() [2/2]	561
10.151.4.26 GetPlanarConfiguration()	561
10.151.4.27 IsFrameEncoder()	561
10.151.4.28 IsLossy()	562
10.151.4.29 IsRowEncoder()	562
10.151.4.30 IsValid()	562
10.151.4.31 SetDimensions() [1/2]	562
10.151.4.32 SetDimensions() [2/2]	562
10.151.4.33 SetLossyFlag()	562
10.151.4.34 SetLUT()	562
10.151.4.35 SetNeedByteSwap()	563
10.151.4.36 SetNeedOverlayCleanup()	563
10.151.4.37 SetNumberOfDimensions()	563
10.151.4.38 SetPhotometricInterpretation()	563
10.151.4.39 SetPixelFormat()	563
10.151.4.40 SetPlanarConfiguration()	563
10.151.4.41 StartEncode()	564
10.151.4.42 StopEncode()	564
10.151.5 Friends And Related Symbol Documentation	564
10.151.5.1 FileChangeTransferSyntax	564
10.151.5.2 ImageChangePhotometricInterpretation	564
10.151.6 Member Data Documentation	564
10.151.6.1 Dimensions	564
10.151.6.2 LossyFlag	564

10.151.6.3 LUT	565
10.151.6.4 NeedByteSwap	565
10.151.6.5 NeedOverlayCleanup	565
10.151.6.6 NumberOfDimensions	565
10.151.6.7 PF	565
10.151.6.8 PI	565
10.151.6.9 PlanarConfiguration	565
10.151.6.10 RequestPaddedCompositePixelCode	565
10.151.6.11 RequestPlanarConfiguration	565
10.152 gdcm::ImageConverter Class Reference	566
10.152.1 Detailed Description	566
10.152.2 Constructor & Destructor Documentation	566
10.152.2.1 ImageConverter()	566
10.152.2.2 ~ImageConverter()	566
10.152.3 Member Function Documentation	566
10.152.3.1 Convert()	566
10.152.3.2 GetOutput()	567
10.152.3.3 SetInput()	567
10.153 gdcm::ImageFragmentSplitter Class Reference	567
10.153.1 Detailed Description	569
10.153.2 Constructor & Destructor Documentation	569
10.153.2.1 ImageFragmentSplitter()	569
10.153.2.2 ~ImageFragmentSplitter()	570
10.153.3 Member Function Documentation	570
10.153.3.1 GetFragmentSizeMax()	570
10.153.3.2 SetForce()	570
10.153.3.3 SetFragmentSizeMax()	570
10.153.3.4 Split()	570
10.154 gdcm::ImageHelper Class Reference	570
10.154.1 Detailed Description	572
10.154.2 Member Function Documentation	572
10.154.2.1 ComputeMediaStorageFromModality()	572
10.154.2.2 ComputeSpacingFromImagePositionPatient()	572
10.154.2.3 GetDimensionsValue()	572
10.154.2.4 GetDirectionCosinesFromDataSet()	573
10.154.2.5 GetDirectionCosinesValue()	573
10.154.2.6 GetForcePixelSpacing()	573
10.154.2.7 GetForceRescaleInterceptSlope()	573
10.154.2.8 GetLUT()	573

10.154.2.9 GetOriginValue()	573
10.154.2.10 GetPhotometricInterpretationValue()	573
10.154.2.11 GetPixelFormatValue()	574
10.154.2.12 GetPlanarConfigurationValue()	574
10.154.2.13 GetPMSRescaleInterceptSlope()	574
10.154.2.14 GetPointerFromElement()	574
10.154.2.15 GetRealWorldValueMappingContent()	574
10.154.2.16 GetRescaleInterceptSlopeValue()	574
10.154.2.17 GetSecondaryCaptureImagePlaneModule()	575
10.154.2.18 GetSpacingTagFromMediaStorage()	575
10.154.2.19 GetSpacingValue()	575
10.154.2.20 GetZSpacingTagFromMediaStorage()	575
10.154.2.21 SetDimensionsValue()	575
10.154.2.22 SetDirectionCosinesValue()	575
10.154.2.23 SetForcePixelSpacing()	575
10.154.2.24 SetForceRescaleInterceptSlope()	576
10.154.2.25 SetOriginValue()	576
10.154.2.26 SetPMSRescaleInterceptSlope()	576
10.154.2.27 SetRescaleInterceptSlopeValue()	576
10.154.2.28 SetSecondaryCaptureImagePlaneModule()	576
10.154.2.29 SetSpacingValue()	577
10.155 gdcm::ImageReader Class Reference	577
10.155.1 Detailed Description	580
10.155.2 Constructor & Destructor Documentation	580
10.155.2.1 ImageReader()	580
10.155.2.2 ~ImageReader()	580
10.155.3 Member Function Documentation	580
10.155.3.1 GetImage() [1/2]	580
10.155.3.2 GetImage() [2/2]	581
10.155.3.3 Read()	581
10.155.3.4 ReadACRNEMAImage()	581
10.155.3.5 ReadImage()	581
10.156 gdcm::ImageRegionReader Class Reference	582
10.156.1 Detailed Description	585
10.156.2 Constructor & Destructor Documentation	585
10.156.2.1 ImageRegionReader()	585
10.156.2.2 ~ImageRegionReader()	586
10.156.3 Member Function Documentation	586
10.156.3.1 ComputeBufferLength()	586

10.156.3.2 GetRegion()	586
10.156.3.3 Read()	586
10.156.3.4 ReadInformation()	586
10.156.3.5 ReadIntoBuffer()	587
10.156.3.6 SetRegion()	587
10.157 gdcm::ImageToImageFilter Class Reference	587
10.157.1 Detailed Description	589
10.157.2 Constructor & Destructor Documentation	589
10.157.2.1 ImageToImageFilter()	589
10.157.2.2 ~ImageToImageFilter()	589
10.157.3 Member Function Documentation	589
10.157.3.1 GetInput()	589
10.157.3.2 GetOutput()	590
10.158 gdcm::ImageWriter Class Reference	590
10.158.1 Detailed Description	593
10.158.2 Constructor & Destructor Documentation	593
10.158.2.1 ImageWriter()	593
10.158.2.2 ~ImageWriter()	593
10.158.3 Member Function Documentation	593
10.158.3.1 ComputeTargetMediaStorage()	593
10.158.3.2 GetImage() [1/2]	594
10.158.3.3 GetImage() [2/2]	594
10.158.3.4 Write()	594
10.159 gdcm::network::ImplementationClassUIDSub Class Reference	594
10.159.1 Detailed Description	595
10.159.2 Constructor & Destructor Documentation	595
10.159.2.1 ImplementationClassUIDSub()	595
10.159.3 Member Function Documentation	595
10.159.3.1 Print()	595
10.159.3.2 Read()	595
10.159.3.3 Size()	595
10.159.3.4 Write()	595
10.160 gdcm::network::ImplementationUIDSub Class Reference	596
10.160.1 Detailed Description	596
10.160.2 Constructor & Destructor Documentation	596
10.160.2.1 ImplementationUIDSub()	596
10.160.3 Member Function Documentation	596
10.160.3.1 Write()	596
10.161 gdcm::network::ImplementationVersionNameSub Class Reference	596

10.161.1 Detailed Description . . . . .	597
10.161.2 Constructor & Destructor Documentation . . . . .	597
10.161.2.1 ImplementationVersionNameSub() . . . . .	597
10.161.3 Member Function Documentation . . . . .	597
10.161.3.1 Print() . . . . .	597
10.161.3.2 Read() . . . . .	597
10.161.3.3 Size() . . . . .	597
10.161.3.4 Write() . . . . .	597
10.162 gdcm::ImplicitDataElement Class Reference . . . . .	598
10.162.1 Detailed Description . . . . .	600
10.162.2 Member Function Documentation . . . . .	601
10.162.2.1 GetLength() . . . . .	601
10.162.2.2 Read() . . . . .	601
10.162.2.3 ReadPreValue() . . . . .	601
10.162.2.4 ReadValue() . . . . .	601
10.162.2.5 ReadValueWithLength() . . . . .	601
10.162.2.6 ReadWithLength() . . . . .	601
10.162.2.7 Write() . . . . .	602
10.163 gdcm::InitializeEvent Class Reference . . . . .	602
10.164 gdcm::IOD Class Reference . . . . .	603
10.164.1 Detailed Description . . . . .	604
10.164.2 Member Typedef Documentation . . . . .	604
10.164.2.1 MapIODEntry . . . . .	604
10.164.2.2 SizeType . . . . .	604
10.164.3 Constructor & Destructor Documentation . . . . .	604
10.164.3.1 IOD() . . . . .	604
10.164.4 Member Function Documentation . . . . .	605
10.164.4.1 AddIODEntry() . . . . .	605
10.164.4.2 Clear() . . . . .	605
10.164.4.3 GetIODEntry() . . . . .	605
10.164.4.4 GetNumberOfIODs() . . . . .	605
10.164.4.5 GetTypeFromTag() . . . . .	605
10.164.5 Friends And Related Symbol Documentation . . . . .	605
10.164.5.1 operator<< . . . . .	605
10.165 gdcm::IODEntry Class Reference . . . . .	606
10.165.1 Detailed Description . . . . .	606
10.165.2 Constructor & Destructor Documentation . . . . .	607
10.165.2.1 IODEntry() . . . . .	607
10.165.3 Member Function Documentation . . . . .	607



10.165.3.1 GetIE()	607
10.165.3.2 GetName()	607
10.165.3.3 GetRef()	607
10.165.3.4 GetUsage()	607
10.165.3.5 GetUsageType()	607
10.165.3.6 SetIE()	607
10.165.3.7 SetName()	608
10.165.3.8 SetRef()	608
10.165.3.9 SetUsage()	608
10.165.4 Friends And Related Symbol Documentation	608
10.165.4.1 operator<<	608
10.166 gdcmm::IODs Class Reference	608
10.166.1 Detailed Description	609
10.166.2 Member Typedef Documentation	609
10.166.2.1 IODMapType	609
10.166.2.2 IODMapTypeConstIterator	609
10.166.2.3 IODName	609
10.166.3 Constructor & Destructor Documentation	610
10.166.3.1 IODs()	610
10.166.4 Member Function Documentation	610
10.166.4.1 AddIOD()	610
10.166.4.2 Begin()	610
10.166.4.3 Clear()	610
10.166.4.4 End()	610
10.166.4.5 GetIOD()	610
10.166.5 Friends And Related Symbol Documentation	611
10.166.5.1 operator<<	611
10.167 gdcmm::IPPSorter Class Reference	611
10.167.1 Detailed Description	613
10.167.2 Constructor & Destructor Documentation	613
10.167.2.1 IPPSorter()	613
10.167.3 Member Function Documentation	613
10.167.3.1 GetDirectionCosinesTolerance()	613
10.167.3.2 GetZSpacing()	614
10.167.3.3 GetZSpacingTolerance()	614
10.167.3.4 SetComputeZSpacing()	614
10.167.3.5 SetDirectionCosinesTolerance()	614
10.167.3.6 SetDropDuplicatePositions()	615
10.167.3.7 SetZSpacingTolerance()	615

10.167.3.8 Sort()	615
10.167.4 Member Data Documentation	615
10.167.4.1 ComputeZSpacing	615
10.167.4.2 DirCosTolerance	615
10.167.4.3 DropDuplicatePositions	616
10.167.4.4 ZSpacing	616
10.167.4.5 ZTolerance	616
10.168 gdcm::Item Class Reference	616
10.168.1 Detailed Description	619
10.168.2 Constructor & Destructor Documentation	619
10.168.2.1 Item() [1/2]	619
10.168.2.2 Item() [2/2]	620
10.168.3 Member Function Documentation	620
10.168.3.1 Clear()	620
10.168.3.2 FindDataElement()	620
10.168.3.3 GetDataElement()	620
10.168.3.4 GetLength()	620
10.168.3.5 GetNestedDataSet() [1/2]	620
10.168.3.6 GetNestedDataSet() [2/2]	620
10.168.3.7 InsertDataElement()	621
10.168.3.8 Read()	621
10.168.3.9 SetNestedDataSet()	621
10.168.3.10 Write()	621
10.168.4 Friends And Related Symbol Documentation	621
10.168.4.1 operator<<	621
10.169 gdcm::IterationEvent Class Reference	622
10.170 gdcm::JPEG12Codec Class Reference	623
10.170.1 Detailed Description	626
10.170.2 Constructor & Destructor Documentation	627
10.170.2.1 JPEG12Codec()	627
10.170.2.2 ~JPEG12Codec()	627
10.170.3 Member Function Documentation	627
10.170.3.1 DecodeByStreams()	627
10.170.3.2 EncodeBuffer()	627
10.170.3.3 GetHeaderInfo()	627
10.170.3.4 InternalCode()	627
10.170.3.5 IsStateSuspension()	628
10.171 gdcm::JPEG16Codec Class Reference	628
10.171.1 Detailed Description	631

10.171.2 Constructor & Destructor Documentation	632
10.171.2.1 JPEG16Codec()	632
10.171.2.2 ~JPEG16Codec()	632
10.171.3 Member Function Documentation	632
10.171.3.1 DecodeByStreams()	632
10.171.3.2 EncodeBuffer()	632
10.171.3.3 GetHeaderInfo()	632
10.171.3.4 InternalCode()	632
10.171.3.5 IsStateSuspension()	633
10.172 gdcm::JPEG2000Codec Class Reference	633
10.172.1 Detailed Description	636
10.172.2 Constructor & Destructor Documentation	636
10.172.2.1 JPEG2000Codec()	636
10.172.2.2 ~JPEG2000Codec()	636
10.172.3 Member Function Documentation	636
10.172.3.1 AppendFrameEncode()	636
10.172.3.2 AppendRowEncode()	637
10.172.3.3 CanCode()	637
10.172.3.4 CanDecode()	637
10.172.3.5 Clone()	637
10.172.3.6 Code()	637
10.172.3.7 Decode()	638
10.172.3.8 DecodeByStreams()	638
10.172.3.9 DecodeExtent()	638
10.172.3.10 GetHeaderInfo()	638
10.172.3.11 GetQuality()	638
10.172.3.12 GetRate()	638
10.172.3.13 IsFrameEncoder()	639
10.172.3.14 IsRowEncoder()	639
10.172.3.15 SetMCT()	639
10.172.3.16 SetNumberOfResolutions()	639
10.172.3.17 SetNumberOfThreadsForDecompression()	639
10.172.3.18 SetQuality()	639
10.172.3.19 SetRate()	640
10.172.3.20 SetReversible()	640
10.172.3.21 SetTileSize()	640
10.172.3.22 StartEncode()	640
10.172.3.23 StopEncode()	640
10.172.4 Friends And Related Symbol Documentation	640

10.172.4.1 Bitmap	640
10.172.4.2 ImageRegionReader	640
10.173 gdcm::JPEG8Codec Class Reference	641
10.173.1 Detailed Description	644
10.173.2 Constructor & Destructor Documentation	644
10.173.2.1 JPEG8Codec()	644
10.173.2.2 ~JPEG8Codec()	644
10.173.3 Member Function Documentation	644
10.173.3.1 DecodeByStreams()	644
10.173.3.2 EncodeBuffer()	645
10.173.3.3 GetHeaderInfo()	645
10.173.3.4 InternalCode()	645
10.173.3.5 IsStateSuspension()	645
10.174 gdcm::JPEGCodec Class Reference	646
10.174.1 Detailed Description	649
10.174.2 Constructor & Destructor Documentation	649
10.174.2.1 JPEGCodec()	649
10.174.2.2 ~JPEGCodec()	649
10.174.3 Member Function Documentation	650
10.174.3.1 AppendFrameEncode()	650
10.174.3.2 AppendRowEncode()	650
10.174.3.3 CanCode()	650
10.174.3.4 CanDecode()	650
10.174.3.5 Clone()	650
10.174.3.6 Code()	651
10.174.3.7 ComputeOffsetTable()	651
10.174.3.8 Decode()	651
10.174.3.9 DecodeByStreams()	651
10.174.3.10 DecodeExtent()	651
10.174.3.11 EncodeBuffer()	652
10.174.3.12 GetHeaderInfo()	652
10.174.3.13 GetLossless()	652
10.174.3.14 GetQuality()	652
10.174.3.15 IsFrameEncoder()	652
10.174.3.16 IsRowEncoder()	652
10.174.3.17 IsStateSuspension()	653
10.174.3.18 IsValid()	653
10.174.3.19 SetBitSample()	653
10.174.3.20 SetLossless()	653

10.174.3.21 SetPixelFormat()	653
10.174.3.22 SetQuality()	653
10.174.3.23 StartEncode()	654
10.174.3.24 StopEncode()	654
10.174.4 Friends And Related Symbol Documentation	654
10.174.4.1 ImageRegionReader	654
10.174.5 Member Data Documentation	654
10.174.5.1 BitSample	654
10.174.5.2 Quality	654
10.175 gdcm::JPEGLSCodec Class Reference	655
10.175.1 Detailed Description	658
10.175.2 Constructor & Destructor Documentation	658
10.175.2.1 JPEGLSCodec()	658
10.175.2.2 ~JPEGLSCodec()	658
10.175.3 Member Function Documentation	658
10.175.3.1 AppendFrameEncode()	658
10.175.3.2 AppendRowEncode()	659
10.175.3.3 CanCode()	659
10.175.3.4 CanDecode()	659
10.175.3.5 Clone()	659
10.175.3.6 Code()	659
10.175.3.7 Decode() [1/2]	660
10.175.3.8 Decode() [2/2]	660
10.175.3.9 DecodeExtent()	660
10.175.3.10 GetBufferLength()	660
10.175.3.11 GetHeaderInfo()	660
10.175.3.12 GetLossless()	661
10.175.3.13 IsFrameEncoder()	661
10.175.3.14 IsRowEncoder()	661
10.175.3.15 SetBufferLength()	661
10.175.3.16 SetLossless()	661
10.175.3.17 SetLossyError()	661
10.175.3.18 StartEncode()	661
10.175.3.19 StopEncode()	662
10.175.4 Friends And Related Symbol Documentation	662
10.175.4.1 ImageRegionReader	662
10.176 gdcm::JSON Class Reference	662
10.176.1 Detailed Description	662
10.176.2 Constructor & Destructor Documentation	662

10.176.2.1 JSON()	662
10.176.2.2 ~JSON()	663
10.176.3 Member Function Documentation	663
10.176.3.1 Code()	663
10.176.3.2 Decode()	663
10.176.3.3 GetPrettyPrint()	663
10.176.3.4 PrettyPrintOff()	663
10.176.3.5 PrettyPrintOn()	663
10.176.3.6 SetPrettyPrint()	664
10.177 gdcm::KAKADUCodec Class Reference	664
10.177.1 Detailed Description	666
10.177.2 Constructor & Destructor Documentation	667
10.177.2.1 KAKADUCodec()	667
10.177.2.2 ~KAKADUCodec()	667
10.177.3 Member Function Documentation	667
10.177.3.1 CanCode()	667
10.177.3.2 CanDecode()	667
10.177.3.3 Clone()	667
10.177.3.4 Code()	667
10.177.3.5 Decode()	668
10.178 gdcm::LO Class Reference	668
10.178.1 Detailed Description	669
10.178.2 Member Typedef Documentation	669
10.178.2.1 const_iterator	669
10.178.2.2 const_reference	669
10.178.2.3 const_reverse_iterator	669
10.178.2.4 difference_type	670
10.178.2.5 iterator	670
10.178.2.6 pointer	670
10.178.2.7 reference	670
10.178.2.8 reverse_iterator	670
10.178.2.9 size_type	670
10.178.2.10 Superclass	670
10.178.2.11 value_type	670
10.178.3 Constructor & Destructor Documentation	670
10.178.3.1 LO() [1/4]	670
10.178.3.2 LO() [2/4]	671
10.178.3.3 LO() [3/4]	671
10.178.3.4 LO() [4/4]	671

10.178.4 Member Function Documentation . . . . .	671
10.178.4.1 IsValid() . . . . .	671
10.179 gdcmm::LookupTable Class Reference . . . . .	671
10.179.1 Detailed Description . . . . .	674
10.179.2 Member Enumeration Documentation . . . . .	674
10.179.2.1 LookupTableType . . . . .	674
10.179.3 Constructor & Destructor Documentation . . . . .	674
10.179.3.1 LookupTable() [1/2] . . . . .	674
10.179.3.2 ~LookupTable() . . . . .	675
10.179.3.3 LookupTable() [2/2] . . . . .	675
10.179.4 Member Function Documentation . . . . .	675
10.179.4.1 Allocate() . . . . .	675
10.179.4.2 Clear() . . . . .	675
10.179.4.3 Decode() [1/2] . . . . .	675
10.179.4.4 Decode() [2/2] . . . . .	675
10.179.4.5 Decode8() . . . . .	676
10.179.4.6 GetBitSample() . . . . .	676
10.179.4.7 GetBufferAsRGBA() . . . . .	676
10.179.4.8 GetLUT() . . . . .	676
10.179.4.9 GetLUTDescriptor() . . . . .	676
10.179.4.10 GetLUTLength() . . . . .	676
10.179.4.11 GetPointer() . . . . .	677
10.179.4.12 InitializeBlueLUT() . . . . .	677
10.179.4.13 Initialized() . . . . .	677
10.179.4.14 InitializeGreenLUT() . . . . .	677
10.179.4.15 InitializeLUT() . . . . .	677
10.179.4.16 InitializeRedLUT() . . . . .	677
10.179.4.17 IsRGB8() . . . . .	678
10.179.4.18 Print() . . . . .	678
10.179.4.19 SetBlueLUT() . . . . .	678
10.179.4.20 SetGreenLUT() . . . . .	678
10.179.4.21 SetLUT() . . . . .	678
10.179.4.22 SetRedLUT() . . . . .	678
10.179.4.23 WriteBufferAsRGBA() . . . . .	679
10.179.5 Member Data Documentation . . . . .	679
10.179.5.1 BitSample . . . . .	679
10.179.5.2 IncompleteLUT . . . . .	679
10.179.5.3 Internal . . . . .	679
10.180 gdcmm::Scanner2::Itstr Struct Reference . . . . .	679

10.180.1 Member Function Documentation . . . . .	679
10.180.1.1 operator>() . . . . .	679
10.181 gdcmm::Scanner::ltstr Struct Reference . . . . .	680
10.181.1 Member Function Documentation . . . . .	680
10.181.1.1 operator>() . . . . .	680
10.182 gdcmm::StrictScanner2::ltstr Struct Reference . . . . .	680
10.182.1 Member Function Documentation . . . . .	680
10.182.1.1 operator>() . . . . .	680
10.183 gdcmm::StrictScanner::ltstr Struct Reference . . . . .	681
10.183.1 Member Function Documentation . . . . .	681
10.183.1.1 operator>() . . . . .	681
10.184 gdcmm::Macro Class Reference . . . . .	681
10.184.1 Detailed Description . . . . .	682
10.184.2 Member Typedef Documentation . . . . .	682
10.184.2.1 ArrayIncludeMacrosType . . . . .	682
10.184.2.2 MapModuleEntry . . . . .	682
10.184.3 Constructor & Destructor Documentation . . . . .	682
10.184.3.1 Macro() . . . . .	682
10.184.4 Member Function Documentation . . . . .	682
10.184.4.1 AddMacroEntry() . . . . .	682
10.184.4.2 Clear() . . . . .	683
10.184.4.3 FindMacroEntry() . . . . .	683
10.184.4.4 GetMacroEntry() . . . . .	683
10.184.4.5 GetName() . . . . .	683
10.184.4.6 SetName() . . . . .	683
10.184.4.7 Verify() . . . . .	683
10.184.5 Friends And Related Symbol Documentation . . . . .	683
10.184.5.1 operator<< . . . . .	683
10.185 gdcmm::Macros Class Reference . . . . .	684
10.185.1 Detailed Description . . . . .	684
10.185.2 Member Typedef Documentation . . . . .	684
10.185.2.1 ModuleMapType . . . . .	684
10.185.3 Constructor & Destructor Documentation . . . . .	685
10.185.3.1 Macros() . . . . .	685
10.185.4 Member Function Documentation . . . . .	685
10.185.4.1 AddMacro() . . . . .	685
10.185.4.2 Clear() . . . . .	685
10.185.4.3 GetMacro() . . . . .	685
10.185.4.4 IsEmpty() . . . . .	685



10.185.5 Friends And Related Symbol Documentation . . . . .	685
10.185.5.1 operator<< . . . . .	685
10.186 gdcm::network::MaximumLengthSub Class Reference . . . . .	686
10.186.1 Detailed Description . . . . .	686
10.186.2 Constructor & Destructor Documentation . . . . .	686
10.186.2.1 MaximumLengthSub() . . . . .	686
10.186.3 Member Function Documentation . . . . .	686
10.186.3.1 GetMaximumLength() . . . . .	686
10.186.3.2 Print() . . . . .	686
10.186.3.3 Read() . . . . .	687
10.186.3.4 SetMaximumLength() . . . . .	687
10.186.3.5 Size() . . . . .	687
10.186.3.6 Write() . . . . .	687
10.187 gdcm::MD5 Class Reference . . . . .	687
10.187.1 Detailed Description . . . . .	687
10.187.2 Member Function Documentation . . . . .	688
10.187.2.1 Compute() . . . . .	688
10.187.2.2 ComputeFile() . . . . .	688
10.188 gdcm::MEC_MR3 Class Reference . . . . .	688
10.188.1 Detailed Description . . . . .	688
10.188.2 Member Function Documentation . . . . .	688
10.188.2.1 GetCanonMECMR3Tag() . . . . .	688
10.188.2.2 GetPMTFInformationDataTag() . . . . .	689
10.188.2.3 GetToshibaMECMR3Tag() . . . . .	689
10.188.2.4 Print() . . . . .	689
10.189 gdcm::MediaStorage Class Reference . . . . .	689
10.189.1 Detailed Description . . . . .	692
10.189.2 Member Enumeration Documentation . . . . .	692
10.189.2.1 MStype . . . . .	692
10.189.2.2 ObjectType . . . . .	695
10.189.3 Constructor & Destructor Documentation . . . . .	695
10.189.3.1 MediaStorage() . . . . .	695
10.189.4 Member Function Documentation . . . . .	696
10.189.4.1 GetModality() . . . . .	696
10.189.4.2 GetModalityDimension() . . . . .	696
10.189.4.3 GetMSString() . . . . .	696
10.189.4.4 GetMStype() . . . . .	696
10.189.4.5 GetNumberOfModality() . . . . .	696
10.189.4.6 GetNumberOfMSString() . . . . .	696

10.189.4.7	GetNumberOfMSType()	696
10.189.4.8	GetString()	697
10.189.4.9	GuessFromModality()	697
10.189.4.10	IsImage()	697
10.189.4.11	IsUndefined()	697
10.189.4.12	operator MSType()	698
10.189.4.13	SetFromDataSet()	698
10.189.4.14	SetFromFile()	698
10.189.4.15	SetFromHeader()	698
10.189.4.16	SetFromModality()	698
10.189.4.17	SetFromSourceImageSequence()	698
10.189.5	Friends And Related Symbol Documentation	699
10.189.5.1	operator<<	699
10.190	gdcM::MemberCommand< T > Class Template Reference	699
10.190.1	Detailed Description	702
10.190.2	Member Typedef Documentation	702
10.190.2.1	Self	702
10.190.2.2	TConstMemberFunctionPointer	702
10.190.2.3	TMemberFunctionPointer	702
10.190.3	Constructor & Destructor Documentation	702
10.190.3.1	MemberCommand() [1/2]	702
10.190.3.2	MemberCommand() [2/2]	702
10.190.3.3	~MemberCommand()	703
10.190.4	Member Function Documentation	703
10.190.4.1	Execute() [1/2]	703
10.190.4.2	Execute() [2/2]	703
10.190.4.3	New()	703
10.190.4.4	operator=()	703
10.190.4.5	SetCallbackFunction() [1/2]	704
10.190.4.6	SetCallbackFunction() [2/2]	704
10.190.5	Member Data Documentation	704
10.190.5.1	m_ConstMemberFunction	704
10.190.5.2	m_MemberFunction	704
10.190.5.3	m_This	704
10.191	gdcM::MeshPrimitive Class Reference	705
10.191.1	Detailed Description	707
10.191.2	Member Typedef Documentation	707
10.191.2.1	PrimitivesData	707
10.191.3	Member Enumeration Documentation	707

10.191.3.1 MPTType	707
10.191.4 Constructor & Destructor Documentation	708
10.191.4.1 MeshPrimitive()	708
10.191.4.2 ~MeshPrimitive()	708
10.191.5 Member Function Documentation	708
10.191.5.1 AddPrimitiveData()	708
10.191.5.2 GetMPTType()	708
10.191.5.3 GetMPTTypeString()	708
10.191.5.4 GetNumberOfPrimitivesData()	708
10.191.5.5 GetPrimitiveData() [1/4]	708
10.191.5.6 GetPrimitiveData() [2/4]	708
10.191.5.7 GetPrimitiveData() [3/4]	709
10.191.5.8 GetPrimitiveData() [4/4]	709
10.191.5.9 GetPrimitivesData() [1/2]	709
10.191.5.10 GetPrimitivesData() [2/2]	709
10.191.5.11 GetPrimitiveType()	709
10.191.5.12 SetPrimitiveData() [1/2]	709
10.191.5.13 SetPrimitiveData() [2/2]	709
10.191.5.14 SetPrimitivesData()	709
10.191.5.15 SetPrimitiveType()	710
10.191.6 Member Data Documentation	710
10.191.6.1 PrimitiveData	710
10.191.6.2 PrimitiveType	710
10.192 gdcmm::ModalityPerformedProcedureStepCreateQuery Class Reference	710
10.192.1 Detailed Description	712
10.192.2 Constructor & Destructor Documentation	712
10.192.2.1 ModalityPerformedProcedureStepCreateQuery()	712
10.192.3 Member Function Documentation	713
10.192.3.1 GetAbstractSyntaxUID()	713
10.192.3.2 GetRequiredDataSet()	713
10.192.3.3 ValidateQuery()	713
10.192.4 Friends And Related Symbol Documentation	713
10.192.4.1 QueryFactory	713
10.193 gdcmm::ModalityPerformedProcedureStepSetQuery Class Reference	713
10.193.1 Detailed Description	716
10.193.2 Constructor & Destructor Documentation	716
10.193.2.1 ModalityPerformedProcedureStepSetQuery()	716
10.193.3 Member Function Documentation	716
10.193.3.1 GetAbstractSyntaxUID()	716

10.193.3.2 GetRequiredDataSet()	716
10.193.3.3 ValidateQuery()	716
10.193.4 Friends And Related Symbol Documentation	717
10.193.4.1 QueryFactory	717
10.194 gdcM::ModifiedEvent Class Reference	717
10.195 gdcM::Module Class Reference	718
10.195.1 Detailed Description	719
10.195.2 Member Typedef Documentation	719
10.195.2.1 ArrayIncludeMacroType	719
10.195.2.2 MapModuleEntry	719
10.195.3 Constructor & Destructor Documentation	720
10.195.3.1 Module()	720
10.195.4 Member Function Documentation	720
10.195.4.1 AddMacro()	720
10.195.4.2 AddModuleEntry()	720
10.195.4.3 Clear()	720
10.195.4.4 FindModuleEntryInMacros()	720
10.195.4.5 GetModuleEntryInMacros()	720
10.195.4.6 GetName()	721
10.195.4.7 SetName()	721
10.195.4.8 Verify()	721
10.195.5 Friends And Related Symbol Documentation	721
10.195.5.1 operator<<	721
10.196 gdcM::ModuleEntry Class Reference	721
10.196.1 Detailed Description	723
10.196.2 Member Typedef Documentation	723
10.196.2.1 Description	723
10.196.3 Constructor & Destructor Documentation	723
10.196.3.1 ModuleEntry()	723
10.196.3.2 ~ModuleEntry()	723
10.196.4 Member Function Documentation	723
10.196.4.1 GetDescription()	723
10.196.4.2 GetName()	724
10.196.4.3 GetType()	724
10.196.4.4 SetDescription()	724
10.196.4.5 SetName()	724
10.196.4.6 SetType()	724
10.196.5 Friends And Related Symbol Documentation	724
10.196.5.1 operator<<	724

10.196.6 Member Data Documentation . . . . .	724
10.196.6.1 DataElementType . . . . .	724
10.196.6.2 DescriptionField . . . . .	725
10.196.6.3 Name . . . . .	725
10.197 gdcM::Modules Class Reference . . . . .	725
10.197.1 Detailed Description . . . . .	725
10.197.2 Member Typedef Documentation . . . . .	726
10.197.2.1 ModuleMapType . . . . .	726
10.197.3 Constructor & Destructor Documentation . . . . .	726
10.197.3.1 Modules() . . . . .	726
10.197.4 Member Function Documentation . . . . .	726
10.197.4.1 AddModule() . . . . .	726
10.197.4.2 Clear() . . . . .	726
10.197.4.3 GetModule() . . . . .	726
10.197.4.4 IsEmpty() . . . . .	726
10.197.5 Friends And Related Symbol Documentation . . . . .	726
10.197.5.1 operator<< . . . . .	726
10.198 gdcM::MovePatientRootQuery Class Reference . . . . .	727
10.198.1 Detailed Description . . . . .	729
10.198.2 Constructor & Destructor Documentation . . . . .	729
10.198.2.1 MovePatientRootQuery() . . . . .	729
10.198.3 Member Function Documentation . . . . .	729
10.198.3.1 GetAbstractSyntaxUID() . . . . .	729
10.198.3.2 GetTagListByLevel() . . . . .	730
10.198.3.3 InitializeDataSet() . . . . .	730
10.198.3.4 ValidateQuery() . . . . .	730
10.198.4 Friends And Related Symbol Documentation . . . . .	730
10.198.4.1 QueryFactory . . . . .	730
10.199 gdcM::MoveStudyRootQuery Class Reference . . . . .	731
10.199.1 Detailed Description . . . . .	733
10.199.2 Constructor & Destructor Documentation . . . . .	733
10.199.2.1 MoveStudyRootQuery() . . . . .	733
10.199.3 Member Function Documentation . . . . .	733
10.199.3.1 GetAbstractSyntaxUID() . . . . .	733
10.199.3.2 GetTagListByLevel() . . . . .	734
10.199.3.3 InitializeDataSet() . . . . .	734
10.199.3.4 ValidateQuery() . . . . .	734
10.199.4 Friends And Related Symbol Documentation . . . . .	734
10.199.4.1 QueryFactory . . . . .	734

10.200 gdcmm::MrProtocol Class Reference . . . . .	735
10.200.1 Detailed Description . . . . .	735
10.200.2 Constructor & Destructor Documentation . . . . .	735
10.200.2.1 MrProtocol() . . . . .	735
10.200.2.2 ~MrProtocol() . . . . .	735
10.200.3 Member Function Documentation . . . . .	736
10.200.3.1 FindMrProtocolByName() . . . . .	736
10.200.3.2 GetMrProtocolByName() . . . . .	736
10.200.3.3 GetSliceArray() . . . . .	736
10.200.3.4 GetVersion() . . . . .	736
10.200.3.5 Load() . . . . .	736
10.200.3.6 Print() . . . . .	736
10.200.4 Friends And Related Symbol Documentation . . . . .	736
10.200.4.1 operator<< . . . . .	736
10.201 gdcmm::network::NActionRQ Class Reference . . . . .	737
10.201.1 Detailed Description . . . . .	738
10.201.2 Member Function Documentation . . . . .	738
10.201.2.1 ConstructPDV() . . . . .	738
10.202 gdcmm::network::NActionRSP Class Reference . . . . .	738
10.202.1 Detailed Description . . . . .	739
10.202.2 Member Function Documentation . . . . .	739
10.202.2.1 ConstructPDVByDataSet() . . . . .	739
10.203 gdcmm::network::NCreateRQ Class Reference . . . . .	740
10.203.1 Detailed Description . . . . .	741
10.203.2 Member Function Documentation . . . . .	741
10.203.2.1 ConstructPDV() . . . . .	741
10.204 gdcmm::network::NCreateRSP Class Reference . . . . .	741
10.204.1 Detailed Description . . . . .	742
10.204.2 Member Function Documentation . . . . .	742
10.204.2.1 ConstructPDVByDataSet() . . . . .	742
10.205 gdcmm::network::NDeleteRQ Class Reference . . . . .	743
10.205.1 Detailed Description . . . . .	744
10.205.2 Member Function Documentation . . . . .	744
10.205.2.1 ConstructPDV() . . . . .	744
10.206 gdcmm::network::NDeleteRSP Class Reference . . . . .	744
10.206.1 Detailed Description . . . . .	745
10.206.2 Member Function Documentation . . . . .	745
10.206.2.1 ConstructPDVByDataSet() . . . . .	745
10.207 gdcmm::NestedModuleEntries Class Reference . . . . .	746

---

10.207.1 Detailed Description . . . . .	748
10.207.2 Member Typedef Documentation . . . . .	748
10.207.2.1 SizeType . . . . .	748
10.207.3 Constructor & Destructor Documentation . . . . .	748
10.207.3.1 NestedModuleEntries() . . . . .	748
10.207.4 Member Function Documentation . . . . .	748
10.207.4.1 AddModuleEntry() . . . . .	748
10.207.4.2 GetModuleEntry() [1/2] . . . . .	748
10.207.4.3 GetModuleEntry() [2/2] . . . . .	748
10.207.4.4 GetNumberOfModuleEntries() . . . . .	749
10.207.5 Friends And Related Symbol Documentation . . . . .	749
10.207.5.1 operator<< . . . . .	749
10.208 gdcn::network::NEventReportRQ Class Reference . . . . .	749
10.208.1 Detailed Description . . . . .	750
10.208.2 Member Function Documentation . . . . .	750
10.208.2.1 ConstructPDV() . . . . .	750
10.209 gdcn::network::NEventReportRSP Class Reference . . . . .	751
10.209.1 Detailed Description . . . . .	752
10.209.2 Member Function Documentation . . . . .	752
10.209.2.1 ConstructPDVByDataSet() . . . . .	752
10.210 gdcn::network::NGetRQ Class Reference . . . . .	752
10.210.1 Detailed Description . . . . .	753
10.210.2 Member Function Documentation . . . . .	753
10.210.2.1 ConstructPDV() . . . . .	753
10.211 gdcn::network::NGetRSP Class Reference . . . . .	754
10.211.1 Detailed Description . . . . .	755
10.211.2 Member Function Documentation . . . . .	755
10.211.2.1 ConstructPDVByDataSet() . . . . .	755
10.212 gdcn::NoEvent Class Reference . . . . .	755
10.212.1 Detailed Description . . . . .	756
10.213 gdcn::network::NormalizedMessageFactory Class Reference . . . . .	756
10.213.1 Member Function Documentation . . . . .	756
10.213.1.1 ConstructNAction() . . . . .	756
10.213.1.2 ConstructNCreate() . . . . .	757
10.213.1.3 ConstructNDelete() . . . . .	757
10.213.1.4 ConstructNEventReport() . . . . .	757
10.213.1.5 ConstructNGet() . . . . .	757
10.213.1.6 ConstructNSet() . . . . .	757
10.214 gdcn::NormalizedNetworkFunctions Class Reference . . . . .	757

10.214.1 Detailed Description . . . . .	758
10.214.2 Member Function Documentation . . . . .	758
10.214.2.1 ConstructQuery() . . . . .	758
10.214.2.2 NAction() . . . . .	759
10.214.2.3 NCreate() . . . . .	759
10.214.2.4 NDelete() . . . . .	759
10.214.2.5 NEventReport() . . . . .	759
10.214.2.6 NGet() . . . . .	759
10.214.2.7 NSet() . . . . .	760
10.215 gdcmm::network::NSetRQ Class Reference . . . . .	760
10.215.1 Detailed Description . . . . .	761
10.215.2 Member Function Documentation . . . . .	761
10.215.2.1 ConstructPDV() . . . . .	761
10.216 gdcmm::network::NSetRSP Class Reference . . . . .	761
10.216.1 Detailed Description . . . . .	762
10.216.2 Member Function Documentation . . . . .	762
10.216.2.1 ConstructPDVByDataSet() . . . . .	762
10.217 gdcmm::Object Class Reference . . . . .	763
10.217.1 Detailed Description . . . . .	764
10.217.2 Constructor & Destructor Documentation . . . . .	764
10.217.2.1 Object() [1/2] . . . . .	764
10.217.2.2 ~Object() . . . . .	764
10.217.2.3 Object() [2/2] . . . . .	764
10.217.3 Member Function Documentation . . . . .	765
10.217.3.1 operator=() . . . . .	765
10.217.3.2 Print() . . . . .	765
10.217.3.3 Register() . . . . .	765
10.217.3.4 UnRegister() . . . . .	765
10.217.4 Friends And Related Symbol Documentation . . . . .	765
10.217.4.1 operator<< . . . . .	765
10.217.4.2 SmartPointer . . . . .	765
10.218 gdcmm::OpenSSLCryptoFactory Class Reference . . . . .	766
10.218.1 Constructor & Destructor Documentation . . . . .	767
10.218.1.1 OpenSSLCryptoFactory() . . . . .	767
10.218.2 Member Function Documentation . . . . .	767
10.218.2.1 CreateCMSProvider() . . . . .	767
10.218.2.2 InitOpenSSL() . . . . .	767
10.219 gdcmm::OpenSSLCryptographicMessageSyntax Class Reference . . . . .	768
10.219.1 Constructor & Destructor Documentation . . . . .	769



10.219.1.1 OpenSSLCryptographicMessageSyntax()	. 769
10.219.1.2 ~OpenSSLCryptographicMessageSyntax()	. 769
10.219.2 Member Function Documentation	. 769
10.219.2.1 Decrypt()	. 769
10.219.2.2 Encrypt()	. 770
10.219.2.3 GetCipherType()	. 770
10.219.2.4 ParseCertificateFile()	. 770
10.219.2.5 ParseKeyFile()	. 770
10.219.2.6 SetCipherType()	. 770
10.219.2.7 SetPassword()	. 771
10.220 gdcmm::OpenSSLP7CryptoFactory Class Reference	. 771
10.220.1 Constructor & Destructor Documentation	. 772
10.220.1.1 OpenSSLP7CryptoFactory()	. 772
10.220.2 Member Function Documentation	. 772
10.220.2.1 CreateCMSProvider()	. 772
10.221 gdcmm::OpenSSLP7CryptographicMessageSyntax Class Reference	. 773
10.221.1 Detailed Description	. 774
10.221.2 Constructor & Destructor Documentation	. 774
10.221.2.1 OpenSSLP7CryptographicMessageSyntax()	. 774
10.221.2.2 ~OpenSSLP7CryptographicMessageSyntax()	. 774
10.221.3 Member Function Documentation	. 774
10.221.3.1 Decrypt()	. 774
10.221.3.2 Encrypt()	. 775
10.221.3.3 GetCipherType()	. 775
10.221.3.4 ParseCertificateFile()	. 775
10.221.3.5 ParseKeyFile()	. 775
10.221.3.6 SetCipherType()	. 775
10.221.3.7 SetPassword()	. 776
10.222 gdcmm::Orientation Class Reference	. 776
10.222.1 Detailed Description	. 777
10.222.2 Member Enumeration Documentation	. 777
10.222.2.1 OrientationType	. 777
10.222.3 Constructor & Destructor Documentation	. 777
10.222.3.1 Orientation()	. 777
10.222.3.2 ~Orientation()	. 777
10.222.4 Member Function Documentation	. 778
10.222.4.1 GetLabel()	. 778
10.222.4.2 GetMajorAxisFromPatientRelativeDirectionCosine()	. 778
10.222.4.3 GetObliquityThresholdCosineValue()	. 778

10.222.4.4 GetType()	778
10.222.4.5 Print()	778
10.222.4.6 SetObliquityThresholdCosineValue()	779
10.222.5 Friends And Related Symbol Documentation	779
10.222.5.1 operator<<	779
10.223 gdcmm::Overlay Class Reference	779
10.223.1 Detailed Description	782
10.223.2 Member Enumeration Documentation	782
10.223.2.1 OverlayType	782
10.223.3 Constructor & Destructor Documentation	782
10.223.3.1 Overlay() [1/2]	782
10.223.3.2 ~Overlay()	783
10.223.3.3 Overlay() [2/2]	783
10.223.4 Member Function Documentation	783
10.223.4.1 Decompress()	783
10.223.4.2 GetBitPosition()	783
10.223.4.3 GetBitsAllocated()	783
10.223.4.4 GetColumns()	783
10.223.4.5 GetDescription()	783
10.223.4.6 GetGroup()	784
10.223.4.7 GetOrigin()	784
10.223.4.8 GetOverlayData()	784
10.223.4.9 GetOverlayTypeAsString()	784
10.223.4.10 GetOverlayTypeFromString()	784
10.223.4.11 GetRows()	784
10.223.4.12 GetType()	784
10.223.4.13 GetTypeAsEnum()	785
10.223.4.14 GetUnpackBuffer()	785
10.223.4.15 GetUnpackBufferLength()	785
10.223.4.16 GrabOverlayFromPixelData()	785
10.223.4.17 IsEmpty()	785
10.223.4.18 IsInPixelData() [1/2]	785
10.223.4.19 IsInPixelData() [2/2]	785
10.223.4.20 IsZero()	786
10.223.4.21 operator=()	786
10.223.4.22 Print()	786
10.223.4.23 SetBitPosition()	786
10.223.4.24 SetBitsAllocated()	786
10.223.4.25 SetColumns()	786

10.223.4.26 SetDescription()	787
10.223.4.27 SetFrameOrigin()	787
10.223.4.28 SetGroup()	787
10.223.4.29 SetNumberOfFrames()	787
10.223.4.30 SetOrigin()	787
10.223.4.31 SetOverlay()	787
10.223.4.32 SetRows()	788
10.223.4.33 SetType()	788
10.223.4.34 Update()	788
10.224 gdcmm::ParseException Class Reference	788
10.224.1 Detailed Description	789
10.224.2 Constructor & Destructor Documentation	789
10.224.2.1 ParseException() [1/2]	789
10.224.2.2 ~ParseException()	790
10.224.2.3 ParseException() [2/2]	790
10.224.3 Member Function Documentation	790
10.224.3.1 GetLastElement()	790
10.224.3.2 operator=()	790
10.224.3.3 SetLastElement()	790
10.225 gdcmm::Parser Class Reference	790
10.225.1 Detailed Description	791
10.225.2 Member Typedef Documentation	792
10.225.2.1 EndElementHandler	792
10.225.2.2 StartElementHandler	792
10.225.3 Member Enumeration Documentation	792
10.225.3.1 ErrorType	792
10.225.4 Constructor & Destructor Documentation	792
10.225.4.1 Parser()	792
10.225.4.2 ~Parser()	792
10.225.5 Member Function Documentation	793
10.225.5.1 GetBuffer()	793
10.225.5.2 GetCurrentByteIndex()	793
10.225.5.3 GetErrorCode()	793
10.225.5.4 GetErrorString()	793
10.225.5.5 GetUserData()	793
10.225.5.6 Parse()	793
10.225.5.7 ParseBuffer()	793
10.225.5.8 Process()	793
10.225.5.9 SetElementHandler()	794

10.225.5.10 SetUserData()	794
10.226 gdcm::Patient Class Reference	794
10.226.1 Detailed Description	794
10.226.2 Constructor & Destructor Documentation	794
10.226.2.1 Patient()	794
10.227 gdcm::network::PDataTFPDU Class Reference	795
10.227.1 Detailed Description	796
10.227.2 Member Typedef Documentation	796
10.227.2.1 SizeType	796
10.227.3 Constructor & Destructor Documentation	796
10.227.3.1 PDataTFPDU()	796
10.227.4 Member Function Documentation	796
10.227.4.1 AddPresentationDataValue()	796
10.227.4.2 GetNumberOfPresentationDataValues()	796
10.227.4.3 GetPresentationDataValue()	796
10.227.4.4 IsLastFragment()	797
10.227.4.5 Print()	797
10.227.4.6 Read()	797
10.227.4.7 ReadInto()	797
10.227.4.8 Size()	797
10.227.4.9 Write()	797
10.228 gdcm::PDBelement Class Reference	798
10.228.1 Detailed Description	799
10.228.2 Constructor & Destructor Documentation	799
10.228.2.1 PDBelement()	799
10.228.3 Member Function Documentation	799
10.228.3.1 GetName()	799
10.228.3.2 GetValue()	799
10.228.3.3 operator==()	799
10.228.3.4 SetName()	799
10.228.3.5 SetValue()	800
10.228.4 Friends And Related Symbol Documentation	800
10.228.4.1 operator<<	800
10.228.5 Member Data Documentation	800
10.228.5.1 NameField	800
10.228.5.2 ValueField	800
10.229 gdcm::PDBHeader Class Reference	800
10.229.1 Detailed Description	801
10.229.2 Constructor & Destructor Documentation	802

---

10.229.2.1 PDBHeader()	802
10.229.2.2 ~PDBHeader()	802
10.229.3 Member Function Documentation	802
10.229.3.1 FindPDBELEMENTByName()	802
10.229.3.2 GetPDBEEnd()	802
10.229.3.3 GetPDBELEMENTByName()	802
10.229.3.4 GetPDBInfoTag()	802
10.229.3.5 LoadFromDataElement()	803
10.229.3.6 Print()	803
10.229.4 Friends And Related Symbol Documentation	803
10.229.4.1 operator<<	803
10.230 gdcm::PDFCodec Class Reference	803
10.230.1 Detailed Description	805
10.230.2 Constructor & Destructor Documentation	805
10.230.2.1 PDFCodec()	805
10.230.2.2 ~PDFCodec()	805
10.230.3 Member Function Documentation	805
10.230.3.1 CanCode()	805
10.230.3.2 CanDecode()	805
10.230.3.3 Decode()	806
10.231 gdcm::network::PDUFactory Class Reference	806
10.231.1 Detailed Description	807
10.231.2 Member Function Documentation	807
10.231.2.1 ConstructAbortPDU()	807
10.231.2.2 ConstructPDU()	807
10.231.2.3 ConstructReleasePDU()	807
10.231.2.4 CreateCEchoPDU()	807
10.231.2.5 CreateCFindPDU()	807
10.231.2.6 CreateCMovePDU()	807
10.231.2.7 CreateCStoreRQPDU()	808
10.231.2.8 CreateCStoreRSPPDU()	808
10.231.2.9 CreateNActionPDU()	808
10.231.2.10 CreateNCreatePDU()	808
10.231.2.11 CreateNDeletePDU()	808
10.231.2.12 CreateNEventReportPDU()	808
10.231.2.13 CreateNGetPDU()	808
10.231.2.14 CreateNSetPDU()	809
10.231.2.15 DetermineEventByPDU()	809
10.231.2.16 GetPDVs()	809

10.232 gdc::PersonName Class Reference . . . . .	809
10.232.1 Detailed Description . . . . .	810
10.232.2 Member Function Documentation . . . . .	810
10.232.2.1 GetMaxLength() . . . . .	810
10.232.2.2 GetNumberOfComponents() . . . . .	810
10.232.2.3 Print() . . . . .	810
10.232.2.4 SetBlob() . . . . .	810
10.232.2.5 SetComponents() [1/2] . . . . .	810
10.232.2.6 SetComponents() [2/2] . . . . .	811
10.232.3 Member Data Documentation . . . . .	811
10.232.3.1 Component . . . . .	811
10.232.3.2 MaxLength . . . . .	811
10.232.3.3 MaxNumberOfComponents . . . . .	811
10.232.3.4 Padding . . . . .	811
10.232.3.5 Separator . . . . .	811
10.233 gdc::PGXCodec Class Reference . . . . .	812
10.233.1 Detailed Description . . . . .	814
10.233.2 Constructor & Destructor Documentation . . . . .	815
10.233.2.1 PGXCodec() . . . . .	815
10.233.2.2 ~PGXCodec() . . . . .	815
10.233.3 Member Function Documentation . . . . .	815
10.233.3.1 CanCode() . . . . .	815
10.233.3.2 CanDecode() . . . . .	815
10.233.3.3 Clone() . . . . .	815
10.233.3.4 GetHeaderInfo() . . . . .	815
10.233.3.5 Read() . . . . .	816
10.233.3.6 Write() . . . . .	816
10.234 gdc::PhotometricInterpretation Class Reference . . . . .	816
10.234.1 Detailed Description . . . . .	817
10.234.2 Member Enumeration Documentation . . . . .	817
10.234.2.1 PType . . . . .	817
10.234.3 Constructor & Destructor Documentation . . . . .	818
10.234.3.1 PhotometricInterpretation() . . . . .	818
10.234.4 Member Function Documentation . . . . .	818
10.234.4.1 GetPIString() . . . . .	818
10.234.4.2 GetPType() . . . . .	818
10.234.4.3 GetSamplesPerPixel() . . . . .	818
10.234.4.4 GetString() . . . . .	819
10.234.4.5 GetType() . . . . .	819

10.234.4.6 IsLossless()	819
10.234.4.7 IsLossy()	819
10.234.4.8 IsRetired()	819
10.234.4.9 IsSameColorSpace()	819
10.234.4.10 operator PType()	819
10.234.5 Friends And Related Symbol Documentation	819
10.234.5.1 operator<<	819
10.235 gdcmm::PixelFormat Class Reference	820
10.235.1 Detailed Description	821
10.235.2 Member Enumeration Documentation	822
10.235.2.1 ScalarType	822
10.235.3 Constructor & Destructor Documentation	822
10.235.3.1 PixelFormat() [1/3]	822
10.235.3.2 PixelFormat() [2/3]	822
10.235.3.3 PixelFormat() [3/3]	822
10.235.4 Member Function Documentation	823
10.235.4.1 GetBitsAllocated()	823
10.235.4.2 GetBitsStored()	823
10.235.4.3 GetHighBit()	823
10.235.4.4 GetMax()	823
10.235.4.5 GetMin()	823
10.235.4.6 GetPixelRepresentation()	824
10.235.4.7 GetPixelSize()	824
10.235.4.8 GetSamplesPerPixel()	824
10.235.4.9 GetScalarType()	824
10.235.4.10 GetScalarTypeAsString()	825
10.235.4.11 IsCompatible()	825
10.235.4.12 IsValid()	825
10.235.4.13 operator ScalarType()	825
10.235.4.14 operator!=(()) [1/2]	825
10.235.4.15 operator!=(()) [2/2]	825
10.235.4.16 operator==(()) [1/2]	825
10.235.4.17 operator==(()) [2/2]	825
10.235.4.18 Print()	826
10.235.4.19 SetBitsAllocated()	826
10.235.4.20 SetBitsStored()	826
10.235.4.21 SetHighBit()	826
10.235.4.22 SetPixelRepresentation()	826
10.235.4.23 SetSamplesPerPixel()	826

10.235.4.24 SetScalarType()	827
10.235.4.25 Validate()	827
10.235.5 Friends And Related Symbol Documentation	827
10.235.5.1 Bitmap	827
10.235.5.2 operator<<	827
10.236 gdcm::Pixmap Class Reference	828
10.236.1 Detailed Description	831
10.236.2 Constructor & Destructor Documentation	832
10.236.2.1 Pixmap()	832
10.236.2.2 ~Pixmap()	832
10.236.3 Member Function Documentation	832
10.236.3.1 AreOverlaysInPixelData()	832
10.236.3.2 GetCurve() [1/2]	832
10.236.3.3 GetCurve() [2/2]	832
10.236.3.4 GetIconImage() [1/2]	832
10.236.3.5 GetIconImage() [2/2]	832
10.236.3.6 GetNumberOfCurves()	833
10.236.3.7 GetNumberOfOverlays()	833
10.236.3.8 GetOverlay() [1/2]	833
10.236.3.9 GetOverlay() [2/2]	833
10.236.3.10 Print()	833
10.236.3.11 RemoveOverlay()	833
10.236.3.12 SetIconImage()	833
10.236.3.13 SetNumberOfCurves()	833
10.236.3.14 SetNumberOfOverlays()	834
10.236.3.15 UnusedBitsPresentInPixelData()	834
10.236.4 Member Data Documentation	834
10.236.4.1 Curves	834
10.236.4.2 Icon	834
10.236.4.3 Overlays	834
10.237 gdcm::PixmapReader Class Reference	835
10.237.1 Detailed Description	837
10.237.2 Constructor & Destructor Documentation	838
10.237.2.1 PixmapReader()	838
10.237.2.2 ~PixmapReader()	838
10.237.3 Member Function Documentation	838
10.237.3.1 GetPixmap() [1/2]	838
10.237.3.2 GetPixmap() [2/2]	838
10.237.3.3 Read()	838



10.237.3.4 ReadACRNEMAIImage()	839
10.237.3.5 ReadImage()	839
10.237.3.6 ReadImageInternal()	839
10.237.4 Member Data Documentation	839
10.237.4.1 PixelData	839
10.238 gdcm::PixmapToPixmapFilter Class Reference	839
10.238.1 Detailed Description	841
10.238.2 Constructor & Destructor Documentation	841
10.238.2.1 PixmapToPixmapFilter()	841
10.238.2.2 ~PixmapToPixmapFilter()	841
10.238.3 Member Function Documentation	841
10.238.3.1 GetInput()	841
10.238.3.2 GetOutput()	841
10.238.3.3 GetOutputAsPixmap()	842
10.239 gdcm::PixmapWriter Class Reference	842
10.239.1 Detailed Description	844
10.239.2 Constructor & Destructor Documentation	845
10.239.2.1 PixmapWriter()	845
10.239.2.2 ~PixmapWriter()	845
10.239.3 Member Function Documentation	845
10.239.3.1 DoIcnImage()	845
10.239.3.2 GetImage() [1/2]	845
10.239.3.3 GetImage() [2/2]	845
10.239.3.4 GetPixmap() [1/2]	845
10.239.3.5 GetPixmap() [2/2]	845
10.239.3.6 PrepareWrite()	846
10.239.3.7 SetImage()	846
10.239.3.8 SetPixmap()	846
10.239.3.9 Write()	846
10.239.4 Member Data Documentation	846
10.239.4.1 PixelData	846
10.240 gdcm::PNMCodec Class Reference	847
10.240.1 Detailed Description	850
10.240.2 Constructor & Destructor Documentation	850
10.240.2.1 PNMCodec()	850
10.240.2.2 ~PNMCodec()	850
10.240.3 Member Function Documentation	850
10.240.3.1 CanCode()	850
10.240.3.2 CanDecode()	850

10.240.3.3 Clone()	851
10.240.3.4 GetBufferLength()	851
10.240.3.5 GetHeaderInfo()	851
10.240.3.6 Read()	851
10.240.3.7 SetBufferLength()	851
10.240.3.8 Write()	851
10.241 gdcmm::Preamble Class Reference	852
10.241.1 Detailed Description	852
10.241.2 Constructor & Destructor Documentation	853
10.241.2.1 Preamble() [1/2]	853
10.241.2.2 ~Preamble()	853
10.241.2.3 Preamble() [2/2]	853
10.241.3 Member Function Documentation	853
10.241.3.1 Clear()	853
10.241.3.2 Create()	853
10.241.3.3 GetInternal()	853
10.241.3.4 GetLength()	853
10.241.3.5 IsEmpty()	854
10.241.3.6 IsValid()	854
10.241.3.7 operator=()	854
10.241.3.8 Print()	854
10.241.3.9 Read()	854
10.241.3.10 Remove()	854
10.241.3.11 Valid()	854
10.241.3.12 Write()	855
10.241.4 Friends And Related Symbol Documentation	855
10.241.4.1 operator<<	855
10.242 gdcmm::PresentationContext Class Reference	855
10.242.1 Detailed Description	856
10.242.2 Member Typedef Documentation	856
10.242.2.1 SizeType	856
10.242.2.2 TransferSyntaxArrayType	856
10.242.3 Constructor & Destructor Documentation	857
10.242.3.1 PresentationContext() [1/2]	857
10.242.3.2 PresentationContext() [2/2]	857
10.242.4 Member Function Documentation	857
10.242.4.1 AddTransferSyntax()	857
10.242.4.2 GetAbstractSyntax()	857
10.242.4.3 GetNumberOfTransferSyntaxes()	857

10.242.4.4 GetPresentationContextID()	857
10.242.4.5 GetTransferSyntax()	857
10.242.4.6 operator==()	858
10.242.4.7 Print()	858
10.242.4.8 SetAbstractSyntax()	858
10.242.4.9 SetPresentationContextID()	858
10.242.5 Member Data Documentation	858
10.242.5.1 AbstractSyntax	858
10.242.5.2 ID	858
10.242.5.3 TransferSyntaxes	858
10.243 gdcmm::network::PresentationContextAC Class Reference	859
10.243.1 Detailed Description	859
10.243.2 Constructor & Destructor Documentation	859
10.243.2.1 PresentationContextAC()	859
10.243.3 Member Function Documentation	859
10.243.3.1 GetPresentationContextID()	859
10.243.3.2 GetReason()	860
10.243.3.3 GetTransferSyntax()	860
10.243.3.4 Print()	860
10.243.3.5 Read()	860
10.243.3.6 SetPresentationContextID()	860
10.243.3.7 SetReason()	860
10.243.3.8 SetTransferSyntax()	860
10.243.3.9 Size()	860
10.243.3.10 Write()	861
10.244 gdcmm::PresentationContextGenerator Class Reference	861
10.244.1 Detailed Description	862
10.244.2 Member Typedef Documentation	862
10.244.2.1 PresentationContextArrayType	862
10.244.2.2 SizeType	862
10.244.3 Constructor & Destructor Documentation	862
10.244.3.1 PresentationContextGenerator()	862
10.244.4 Member Function Documentation	863
10.244.4.1 AddFromFile()	863
10.244.4.2 AddPresentationContext()	863
10.244.4.3 GenerateFromFilenames()	863
10.244.4.4 GenerateFromUID()	863
10.244.4.5 GetDefaultTransferSyntax()	863
10.244.4.6 GetPresentationContexts()	863

10.244.4.7 SetDefaultTransferSyntax()	864
10.244.4.8 SetMergeModeToAbstractSyntax()	864
10.244.4.9 SetMergeModeToTransferSyntax()	864
10.245 gdcmm::network::PresentationContextRQ Class Reference	864
10.245.1 Detailed Description	865
10.245.2 Member Typedef Documentation	865
10.245.2.1 SizeType	865
10.245.3 Constructor & Destructor Documentation	865
10.245.3.1 PresentationContextRQ() [1/3]	865
10.245.3.2 PresentationContextRQ() [2/3]	865
10.245.3.3 PresentationContextRQ() [3/3]	865
10.245.4 Member Function Documentation	865
10.245.4.1 AddTransferSyntax()	865
10.245.4.2 GetAbstractSyntax() [1/2]	866
10.245.4.3 GetAbstractSyntax() [2/2]	866
10.245.4.4 GetNumberOfTransferSyntaxes()	866
10.245.4.5 GetPresentationContextID()	866
10.245.4.6 GetTransferSyntax() [1/2]	866
10.245.4.7 GetTransferSyntax() [2/2]	866
10.245.4.8 GetTransferSyntaxes()	866
10.245.4.9 operator==( )	866
10.245.4.10 Print()	866
10.245.4.11 Read()	867
10.245.4.12 SetAbstractSyntax()	867
10.245.4.13 SetPresentationContextID()	867
10.245.4.14 Size()	867
10.245.4.15 Write()	867
10.246 gdcmm::network::PresentationDataValue Class Reference	867
10.246.1 Detailed Description	868
10.246.2 Constructor & Destructor Documentation	868
10.246.2.1 PresentationDataValue()	868
10.246.3 Member Function Documentation	868
10.246.3.1 ConcatenatePDVBlobs()	868
10.246.3.2 ConcatenatePDVBlobsAsExplicit()	869
10.246.3.3 GetBlob()	869
10.246.3.4 GetIsCommand()	869
10.246.3.5 GetIsLastFragment()	869
10.246.3.6 GetMessageHeader()	869
10.246.3.7 GetPresentationContextID()	869

10.246.3.8 Print()	869
10.246.3.9 Read()	869
10.246.3.10 ReadInto()	869
10.246.3.11 SetBlob()	870
10.246.3.12 SetCommand()	870
10.246.3.13 SetDataSet()	870
10.246.3.14 SetLastFragment()	870
10.246.3.15 SetMessageHeader()	870
10.246.3.16 SetPresentationContextID()	870
10.246.3.17 Size()	870
10.246.3.18 Write()	871
10.247 gdcmm::Printer Class Reference	871
10.247.1 Detailed Description	872
10.247.2 Member Enumeration Documentation	872
10.247.2.1 PrintStyles	872
10.247.3 Constructor & Destructor Documentation	873
10.247.3.1 Printer()	873
10.247.3.2 ~Printer()	873
10.247.4 Member Function Documentation	873
10.247.4.1 GetPrintStyle()	873
10.247.4.2 Print()	873
10.247.4.3 PrintDataElement()	873
10.247.4.4 PrintDataSet()	874
10.247.4.5 PrintSQ()	874
10.247.4.6 SetColor()	874
10.247.4.7 SetFile()	874
10.247.4.8 SetStyle()	874
10.247.5 Member Data Documentation	874
10.247.5.1 F	874
10.247.5.2 MaxPrintLength	875
10.247.5.3 PrintStyle	875
10.248 gdcmm::PrivateDict Class Reference	875
10.248.1 Detailed Description	875
10.248.2 Constructor & Destructor Documentation	876
10.248.2.1 PrivateDict()	876
10.248.2.2 ~PrivateDict()	876
10.248.3 Member Function Documentation	876
10.248.3.1 AddDictEntry()	876
10.248.3.2 FindDictEntry()	876

10.248.3.3 GetDictEntry()	876
10.248.3.4 IsEmpty()	876
10.248.3.5 LoadDefault()	876
10.248.3.6 PrintXML()	877
10.248.3.7 RemoveDictEntry()	877
10.248.4 Friends And Related Symbol Documentation	877
10.248.4.1 Dicts	877
10.248.4.2 operator<<	877
10.249 gdcmm::PrivateTag Class Reference	877
10.249.1 Detailed Description	880
10.249.2 Constructor & Destructor Documentation	880
10.249.2.1 PrivateTag() [1/2]	880
10.249.2.2 PrivateTag() [2/2]	880
10.249.3 Member Function Documentation	881
10.249.3.1 GetAsDataElement()	881
10.249.3.2 GetOwner()	881
10.249.3.3 operator!=( ) [1/2]	881
10.249.3.4 operator!=( ) [2/2]	881
10.249.3.5 operator<()	881
10.249.3.6 operator=()	881
10.249.3.7 operator==( ) [1/2]	882
10.249.3.8 operator==( ) [2/2]	882
10.249.3.9 ReadFromCommaSeparatedString()	882
10.249.3.10 SetOwner()	882
10.249.4 Friends And Related Symbol Documentation	882
10.249.4.1 operator<<	882
10.250 gdcmm::ProgressEvent Class Reference	883
10.250.1 Detailed Description	884
10.250.2 Member Typedef Documentation	884
10.250.2.1 Self	884
10.250.2.2 Superclass	884
10.250.3 Constructor & Destructor Documentation	885
10.250.3.1 ProgressEvent() [1/2]	885
10.250.3.2 ~ProgressEvent()	885
10.250.3.3 ProgressEvent() [2/2]	885
10.250.4 Member Function Documentation	885
10.250.4.1 CheckEvent()	885
10.250.4.2 GetEventName()	885
10.250.4.3 GetProgress()	885

10.250.4.4 MakeObject()	886
10.250.4.5 operator=()	886
10.250.4.6 SetProgress()	886
10.251 gdcm::PVRGCodec Class Reference	886
10.251.1 Detailed Description	889
10.251.2 Constructor & Destructor Documentation	889
10.251.2.1 PVRGCodec()	889
10.251.2.2 ~PVRGCodec()	889
10.251.3 Member Function Documentation	889
10.251.3.1 CanCode()	889
10.251.3.2 CanDecode()	889
10.251.3.3 Clone()	890
10.251.3.4 Code()	890
10.251.3.5 Decode()	890
10.251.3.6 SetLossyFlag()	890
10.252 gdcm::PythonFilter Class Reference	890
10.252.1 Detailed Description	891
10.252.2 Constructor & Destructor Documentation	891
10.252.2.1 PythonFilter()	891
10.252.2.2 ~PythonFilter()	891
10.252.3 Member Function Documentation	891
10.252.3.1 GetFile() [1/2]	891
10.252.3.2 GetFile() [2/2]	891
10.252.3.3 SetDicts()	891
10.252.3.4 SetFile()	892
10.252.3.5 ToPyObject()	892
10.252.3.6 UseDictAlways()	892
10.253 gdcm::QueryBase Class Reference	892
10.253.1 Detailed Description	893
10.253.2 Constructor & Destructor Documentation	893
10.253.2.1 ~QueryBase()	893
10.253.3 Member Function Documentation	893
10.253.3.1 GetAllRequiredTags()	893
10.253.3.2 GetAllTags()	893
10.253.3.3 GetHierarchicalSearchTags()	894
10.253.3.4 GetName()	894
10.253.3.5 GetOptionalTags()	894
10.253.3.6 GetQueryLevel()	894
10.253.3.7 GetRequiredTags()	894

10.253.3.8 GetUniqueTags()	894
10.254 gdcM::QueryFactory Class Reference	895
10.254.1 Detailed Description	895
10.254.2 Member Function Documentation	895
10.254.2.1 GetCharacterFromCurrentLocale()	895
10.254.2.2 ListCharSets()	895
10.254.2.3 ProduceCharacterSetDataElement()	896
10.254.2.4 ProduceQuery() [1/2]	896
10.254.2.5 ProduceQuery() [2/2]	896
10.255 gdcM::QueryImage Class Reference	896
10.255.1 Detailed Description	897
10.255.2 Member Function Documentation	897
10.255.2.1 GetHierarchicalSearchTags()	897
10.255.2.2 GetName()	898
10.255.2.3 GetOptionalTags()	898
10.255.2.4 GetQueryLevel()	898
10.255.2.5 GetRequiredTags()	898
10.255.2.6 GetUniqueTags()	898
10.256 gdcM::QueryPatient Class Reference	899
10.256.1 Detailed Description	900
10.256.2 Member Function Documentation	900
10.256.2.1 GetHierarchicalSearchTags()	900
10.256.2.2 GetName()	900
10.256.2.3 GetOptionalTags()	900
10.256.2.4 GetQueryLevel()	900
10.256.2.5 GetRequiredTags()	901
10.256.2.6 GetUniqueTags()	901
10.257 gdcM::QuerySeries Class Reference	901
10.257.1 Detailed Description	902
10.257.2 Member Function Documentation	902
10.257.2.1 GetHierarchicalSearchTags()	902
10.257.2.2 GetName()	903
10.257.2.3 GetOptionalTags()	903
10.257.2.4 GetQueryLevel()	903
10.257.2.5 GetRequiredTags()	903
10.257.2.6 GetUniqueTags()	903
10.258 gdcM::QueryStudy Class Reference	904
10.258.1 Detailed Description	905
10.258.2 Member Function Documentation	905



10.258.2.1 GetHierarchicalSearchTags()	905
10.258.2.2 GetName()	905
10.258.2.3 GetOptionalTags()	905
10.258.2.4 GetQueryLevel()	905
10.258.2.5 GetRequiredTags()	906
10.258.2.6 GetUniqueTags()	906
10.259 gdcmm::RAWCodec Class Reference	906
10.259.1 Detailed Description	909
10.259.2 Constructor & Destructor Documentation	909
10.259.2.1 RAWCodec()	909
10.259.2.2 ~RAWCodec()	909
10.259.3 Member Function Documentation	909
10.259.3.1 CanCode()	909
10.259.3.2 CanDecode()	909
10.259.3.3 Clone()	910
10.259.3.4 Code()	910
10.259.3.5 Decode()	910
10.259.3.6 DecodeByStreams()	910
10.259.3.7 DecodeBytes()	910
10.259.3.8 GetHeaderInfo()	911
10.260 gdcmm::Reader Class Reference	911
10.260.1 Detailed Description	913
10.260.2 Constructor & Destructor Documentation	914
10.260.2.1 Reader()	914
10.260.2.2 ~Reader()	914
10.260.3 Member Function Documentation	914
10.260.3.1 CanRead()	914
10.260.3.2 GetFile() [1/2]	914
10.260.3.3 GetFile() [2/2]	914
10.260.3.4 GetStreamCurrentPosition()	915
10.260.3.5 GetStreamPtr()	915
10.260.3.6 Read()	915
10.260.3.7 ReadDataSet()	915
10.260.3.8 ReadMetaInformation()	915
10.260.3.9 ReadPreamble()	916
10.260.3.10 ReadSelectedPrivateTags()	916
10.260.3.11 ReadSelectedTags()	916
10.260.3.12 ReadUpToTag()	916
10.260.3.13 SetFile()	916

10.260.3.14 SetFileName()	917
10.260.3.15 SetStream()	917
10.260.4 Friends And Related Symbol Documentation	917
10.260.4.1 StreamImageReader	917
10.260.5 Member Data Documentation	917
10.260.5.1 F	917
10.261 gdcm::RealWorldValueMappingContent Struct Reference	918
10.261.1 Member Data Documentation	918
10.261.1.1 CodeMeaning	918
10.261.1.2 CodeValue	918
10.261.1.3 RealWorldValueIntercept	919
10.261.1.4 RealWorldValueSlope	919
10.262 gdcm::Region Class Reference	919
10.262.1 Detailed Description	920
10.262.2 Constructor & Destructor Documentation	920
10.262.2.1 Region()	920
10.262.2.2 ~Region()	920
10.262.3 Member Function Documentation	920
10.262.3.1 Area()	920
10.262.3.2 Clone()	920
10.262.3.3 ComputeBoundingBox()	920
10.262.3.4 Empty()	921
10.262.3.5 IsValid()	921
10.262.3.6 Print()	921
10.263 gdcm::Rescaler Class Reference	921
10.263.1 Detailed Description	922
10.263.2 Constructor & Destructor Documentation	923
10.263.2.1 Rescaler()	923
10.263.2.2 ~Rescaler()	923
10.263.3 Member Function Documentation	923
10.263.3.1 ComputeInterceptSlopePixelType()	923
10.263.3.2 ComputePixelTypeFromMinMax()	924
10.263.3.3 GetIntercept()	924
10.263.3.4 GetSlope()	924
10.263.3.5 InverseRescale()	924
10.263.3.6 InverseRescaleFunctionIntoBestFit()	924
10.263.3.7 Rescale()	924
10.263.3.8 RescaleFunctionIntoBestFit()	925
10.263.3.9 SetIntercept()	925

10.263.3.10 SetMinMaxForPixelType()	925
10.263.3.11 SetPixelFormat()	925
10.263.3.12 SetSlope()	925
10.263.3.13 SetTargetPixelType()	926
10.263.3.14 SetUseTargetPixelType()	926
10.264 gdcmm::RLECodec Class Reference	926
10.264.1 Detailed Description	929
10.264.2 Constructor & Destructor Documentation	929
10.264.2.1 RLECodec()	929
10.264.2.2 ~RLECodec()	929
10.264.3 Member Function Documentation	930
10.264.3.1 AppendFrameEncode()	930
10.264.3.2 AppendRowEncode()	930
10.264.3.3 CanCode()	930
10.264.3.4 CanDecode()	930
10.264.3.5 Clone()	930
10.264.3.6 Code()	931
10.264.3.7 Decode()	931
10.264.3.8 DecodeByStreams()	931
10.264.3.9 DecodeExtent()	931
10.264.3.10 GetBufferLength()	931
10.264.3.11 GetHeaderInfo()	932
10.264.3.12 IsFrameEncoder()	932
10.264.3.13 IsRowEncoder()	932
10.264.3.14 SetBufferLength()	932
10.264.3.15 SetLength()	932
10.264.3.16 StartEncode()	932
10.264.3.17 StopEncode()	932
10.264.4 Friends And Related Symbol Documentation	933
10.264.4.1 ImageRegionReader	933
10.265 gdcmm::network::RoleSelectionSub Class Reference	933
10.265.1 Detailed Description	933
10.265.2 Constructor & Destructor Documentation	933
10.265.2.1 RoleSelectionSub()	933
10.265.3 Member Function Documentation	933
10.265.3.1 Print()	933
10.265.3.2 Read()	934
10.265.3.3 SetTuple()	934
10.265.3.4 Size()	934

10.265.3.5 Write()	934
10.266 gdcmm::Scanner Class Reference	934
10.266.1 Detailed Description	937
10.266.2 Member Typedef Documentation	938
10.266.2.1 ConstIterator	938
10.266.2.2 MappingType	938
10.266.2.3 TagToValue	938
10.266.2.4 TagToValueValueType	938
10.266.2.5 ValuesType	938
10.266.3 Constructor & Destructor Documentation	938
10.266.3.1 Scanner()	938
10.266.3.2 ~Scanner()	938
10.266.4 Member Function Documentation	938
10.266.4.1 AddPrivateTag()	938
10.266.4.2 AddSkipTag()	939
10.266.4.3 AddTag()	939
10.266.4.4 Begin()	939
10.266.4.5 ClearSkipTags()	939
10.266.4.6 ClearTags()	939
10.266.4.7 End()	939
10.266.4.8 GetAllFilenamesFromTagToValue()	939
10.266.4.9 GetFilenameFromTagToValue()	940
10.266.4.10 GetFilenames()	940
10.266.4.11 GetKeys()	940
10.266.4.12 GetMapping()	940
10.266.4.13 GetMappingFromTagToValue()	940
10.266.4.14 GetMappings()	941
10.266.4.15 GetOrderedValues()	941
10.266.4.16 GetValue()	941
10.266.4.17 GetValues() [1/2]	941
10.266.4.18 GetValues() [2/2]	941
10.266.4.19 IsKey()	942
10.266.4.20 New()	942
10.266.4.21 Print()	942
10.266.4.22 PrintTable()	942
10.266.4.23 ProcessPublicTag()	942
10.266.4.24 Scan()	942
10.266.5 Friends And Related Symbol Documentation	943
10.266.5.1 operator<<	943

---

10.267	gdcmm::Scanner2 Class Reference	943
10.267.1	Detailed Description	946
10.267.2	Member Typedef Documentation	947
10.267.2.1	PrivateConstIterator	947
10.267.2.2	PrivateMappingType	947
10.267.2.3	PrivateTagToValue	947
10.267.2.4	PrivateTagToValueValueType	947
10.267.2.5	PublicConstIterator	947
10.267.2.6	PublicMappingType	947
10.267.2.7	PublicTagToValue	947
10.267.2.8	PublicTagToValueValueType	947
10.267.2.9	ValuesType	948
10.267.3	Constructor & Destructor Documentation	948
10.267.3.1	Scanner2()	948
10.267.3.2	~Scanner2()	948
10.267.4	Member Function Documentation	948
10.267.4.1	AddPrivateTag()	948
10.267.4.2	AddPublicTag()	948
10.267.4.3	AddSkipTag()	948
10.267.4.4	Begin()	948
10.267.4.5	ClearPrivateTags()	949
10.267.4.6	ClearPublicTags()	949
10.267.4.7	ClearSkipTags()	949
10.267.4.8	End()	949
10.267.4.9	GetAllFilenamesFromPrivateTagToValue()	949
10.267.4.10	GetAllFilenamesFromPublicTagToValue()	949
10.267.4.11	GetFilenameFromPrivateTagToValue()	949
10.267.4.12	GetFilenameFromPublicTagToValue()	949
10.267.4.13	GetFilenames()	950
10.267.4.14	GetKeys()	950
10.267.4.15	GetMappingFromPrivateTagToValue()	950
10.267.4.16	GetMappingFromPublicTagToValue()	950
10.267.4.17	GetPrivateMapping()	950
10.267.4.18	GetPrivateMappings()	950
10.267.4.19	GetPrivateOrderedValues()	950
10.267.4.20	GetPrivateValue()	951
10.267.4.21	GetPrivateValues()	951
10.267.4.22	GetPublicMapping()	951
10.267.4.23	GetPublicMappings()	951

10.267.4.24	GetPublicOrderedValues()	951
10.267.4.25	GetPublicValue()	951
10.267.4.26	GetPublicValues()	952
10.267.4.27	GetValues()	952
10.267.4.28	IsKey()	952
10.267.4.29	New()	952
10.267.4.30	Print()	952
10.267.4.31	PrintTable()	952
10.267.4.32	PrivateBegin()	953
10.267.4.33	PrivateEnd()	953
10.267.4.34	ProcessPrivateTag()	953
10.267.4.35	ProcessPublicTag()	953
10.267.4.36	Scan()	953
10.267.5	Friends And Related Symbol Documentation	953
10.267.5.1	operator<<	953
10.268	gdcmm::Segment Class Reference	954
10.268.1	Detailed Description	956
10.268.2	Member Typedef Documentation	956
10.268.2.1	BasicCodedEntryVector	956
10.268.2.2	SurfaceVector	956
10.268.3	Member Enumeration Documentation	956
10.268.3.1	ALGOType	956
10.268.4	Constructor & Destructor Documentation	957
10.268.4.1	Segment()	957
10.268.4.2	~Segment()	957
10.268.5	Member Function Documentation	957
10.268.5.1	AddSurface()	957
10.268.5.2	GetALGOType()	957
10.268.5.3	GetALGOTypeString()	957
10.268.5.4	GetAnatomicRegion() [1/2]	957
10.268.5.5	GetAnatomicRegion() [2/2]	958
10.268.5.6	GetAnatomicRegionModifiers() [1/2]	958
10.268.5.7	GetAnatomicRegionModifiers() [2/2]	958
10.268.5.8	GetPropertyCategory() [1/2]	958
10.268.5.9	GetPropertyCategory() [2/2]	958
10.268.5.10	GetPropertyType() [1/2]	958
10.268.5.11	GetPropertyType() [2/2]	958
10.268.5.12	GetPropertyTypeModifiers() [1/2]	958
10.268.5.13	GetPropertyTypeModifiers() [2/2]	958

10.268.5.14	GetSegmentAlgorithmName()	958
10.268.5.15	GetSegmentAlgorithmType()	959
10.268.5.16	GetSegmentDescription()	959
10.268.5.17	GetSegmentLabel()	959
10.268.5.18	GetSegmentNumber()	959
10.268.5.19	GetSurface()	959
10.268.5.20	GetSurfaceCount()	959
10.268.5.21	GetSurfaces() [1/2]	959
10.268.5.22	GetSurfaces() [2/2]	959
10.268.5.23	SetAnatomicRegion()	959
10.268.5.24	SetAnatomicRegionModifiers()	960
10.268.5.25	SetPropertyCategory()	960
10.268.5.26	SetPropertyType()	960
10.268.5.27	SetPropertyTypeModifiers()	960
10.268.5.28	SetSegmentAlgorithmName()	960
10.268.5.29	SetSegmentAlgorithmType() [1/2]	960
10.268.5.30	SetSegmentAlgorithmType() [2/2]	960
10.268.5.31	SetSegmentDescription()	960
10.268.5.32	SetSegmentLabel()	961
10.268.5.33	SetSegmentNumber()	961
10.268.5.34	SetSurfaceCount()	961
10.268.6	Member Data Documentation	961
10.268.6.1	AnatomicRegion	961
10.268.6.2	AnatomicRegionModifiers	961
10.268.6.3	PropertyCategory	961
10.268.6.4	PropertyType	961
10.268.6.5	PropertyTypeModifiers	961
10.268.6.6	SegmentAlgorithmName	961
10.268.6.7	SegmentAlgorithmType	962
10.268.6.8	SegmentDescription	962
10.268.6.9	SegmentLabel	962
10.268.6.10	SegmentNumber	962
10.268.6.11	SurfaceCount	962
10.268.6.12	Surfaces	962
10.269	gdcm::SegmentedPaletteColorLookupTable Class Reference	963
10.269.1	Detailed Description	965
10.269.2	Constructor & Destructor Documentation	965
10.269.2.1	SegmentedPaletteColorLookupTable()	965
10.269.2.2	~SegmentedPaletteColorLookupTable()	965

10.269.3 Member Function Documentation	966
10.269.3.1 Print()	966
10.269.3.2 SetLUT()	966
10.270 gdcm::SegmentReader Class Reference	966
10.270.1 Detailed Description	968
10.270.2 Member Typedef Documentation	969
10.270.2.1 SegmentMap	969
10.270.2.2 SegmentVector	969
10.270.3 Constructor & Destructor Documentation	969
10.270.3.1 SegmentReader()	969
10.270.3.2 ~SegmentReader()	969
10.270.4 Member Function Documentation	969
10.270.4.1 GetSegments() [1/2]	969
10.270.4.2 GetSegments() [2/2]	969
10.270.4.3 Read()	969
10.270.4.4 ReadSegment()	970
10.270.4.5 ReadSegments()	970
10.270.5 Member Data Documentation	970
10.270.5.1 Segments	970
10.271 gdcm::SegmentWriter Class Reference	970
10.271.1 Detailed Description	972
10.271.2 Member Typedef Documentation	972
10.271.2.1 SegmentVector	972
10.271.3 Constructor & Destructor Documentation	972
10.271.3.1 SegmentWriter()	972
10.271.3.2 ~SegmentWriter()	972
10.271.4 Member Function Documentation	973
10.271.4.1 AddSegment()	973
10.271.4.2 GetNumberOfSegments()	973
10.271.4.3 GetSegment()	973
10.271.4.4 GetSegments() [1/2]	973
10.271.4.5 GetSegments() [2/2]	973
10.271.4.6 PrepareWrite()	973
10.271.4.7 SetNumberOfSegments()	973
10.271.4.8 SetSegments()	973
10.271.4.9 Write()	974
10.271.5 Member Data Documentation	974
10.271.5.1 Segments	974
10.272 gdcm::SequenceOfFragments Class Reference	974



10.272.1 Detailed Description	977
10.272.2 Member Typedef Documentation	977
10.272.2.1 ConstIterator	977
10.272.2.2 FragmentVector	977
10.272.2.3 Iterator	977
10.272.2.4 SizeType	977
10.272.3 Constructor & Destructor Documentation	977
10.272.3.1 SequenceOfFragments()	977
10.272.4 Member Function Documentation	978
10.272.4.1 AddFragment()	978
10.272.4.2 Begin() [1/2]	978
10.272.4.3 Begin() [2/2]	978
10.272.4.4 Clear()	978
10.272.4.5 ComputeByteLength()	978
10.272.4.6 ComputeLength()	978
10.272.4.7 End() [1/2]	978
10.272.4.8 End() [2/2]	978
10.272.4.9 GetBuffer()	979
10.272.4.10 GetFragBuffer()	979
10.272.4.11 GetFragment()	979
10.272.4.12 GetLength()	979
10.272.4.13 GetNumberOfFragments()	979
10.272.4.14 GetTable() [1/2]	979
10.272.4.15 GetTable() [2/2]	980
10.272.4.16 New()	980
10.272.4.17 operator==()	980
10.272.4.18 Print()	980
10.272.4.19 Read()	980
10.272.4.20 ReadPreValue()	980
10.272.4.21 ReadValue()	981
10.272.4.22 SetLength()	981
10.272.4.23 Write()	981
10.272.4.24 WriteBuffer()	981
10.273 gdcm::SequenceOfItems Class Reference	982
10.273.1 Detailed Description	984
10.273.2 Member Typedef Documentation	985
10.273.2.1 ConstIterator	985
10.273.2.2 ItemVector	985
10.273.2.3 Iterator	985

10.273.2.4 SizeType	985
10.273.3 Constructor & Destructor Documentation	985
10.273.3.1 SequenceOfItems()	985
10.273.4 Member Function Documentation	985
10.273.4.1 AddItem()	985
10.273.4.2 AddNewUndefinedLengthItem()	986
10.273.4.3 Begin() [1/2]	986
10.273.4.4 Begin() [2/2]	986
10.273.4.5 Clear()	986
10.273.4.6 ComputeLength()	986
10.273.4.7 End() [1/2]	986
10.273.4.8 End() [2/2]	986
10.273.4.9 FindDataElement()	986
10.273.4.10 GetItem() [1/2]	987
10.273.4.11 GetItem() [2/2]	987
10.273.4.12 GetLength()	987
10.273.4.13 GetNumberOfItems()	987
10.273.4.14 IsEmpty()	987
10.273.4.15 IsUndefinedLength()	987
10.273.4.16 New()	988
10.273.4.17 operator=()	988
10.273.4.18 operator==(())	988
10.273.4.19 Print()	988
10.273.4.20 Read()	988
10.273.4.21 RemoveItemByIndex()	989
10.273.4.22 SetLength()	989
10.273.4.23 SetLengthToUndefined()	989
10.273.4.24 SetNumberOfItems()	989
10.273.4.25 Write()	989
10.273.5 Member Data Documentation	989
10.273.5.1 Items	989
10.273.5.2 SequenceLengthField	990
10.274 gdcm::SerieHelper Class Reference	990
10.274.1 Detailed Description	991
10.274.2 Member Typedef Documentation	992
10.274.2.1 Rule	992
10.274.2.2 SerieRestrictions	992
10.274.2.3 SingleSerieUIDFileSetmap	992
10.274.3 Constructor & Destructor Documentation	992

10.274.3.1 SerieHelper()	992
10.274.3.2 ~SerieHelper()	992
10.274.4 Member Function Documentation	992
10.274.4.1 AddFile()	992
10.274.4.2 AddFileName()	992
10.274.4.3 AddRestriction() [1/3]	992
10.274.4.4 AddRestriction() [2/3]	993
10.274.4.5 AddRestriction() [3/3]	993
10.274.4.6 Clear()	993
10.274.4.7 CreateDefaultUniqueSeriesIdentifier()	993
10.274.4.8 CreateUniqueSeriesIdentifier()	993
10.274.4.9 FileNameOrdering()	993
10.274.4.10 GetFirstSingleSerieUIDFileSet()	993
10.274.4.11 GetNextSingleSerieUIDFileSet()	993
10.274.4.12 ImageNumberOrdering()	994
10.274.4.13 ImagePositionPatientOrdering()	994
10.274.4.14 OrderFileList()	994
10.274.4.15 SetDirectory()	994
10.274.4.16 SetLoadMode()	994
10.274.4.17 SetUseSeriesDetails()	994
10.274.4.18 UserOrdering()	994
10.274.5 Member Data Documentation	994
10.274.5.1 elem	994
10.274.5.2 ItFileSetHt	995
10.274.5.3 op	995
10.274.5.4 SingleSerieUIDFileSetHT	995
10.274.5.5 value	995
10.275 gdcm::Series Class Reference	995
10.275.1 Detailed Description	995
10.275.2 Constructor & Destructor Documentation	995
10.275.2.1 Series()	995
10.276 gdcm::network::ServiceClassApplicationInformation Class Reference	996
10.276.1 Detailed Description	996
10.276.2 Constructor & Destructor Documentation	996
10.276.2.1 ServiceClassApplicationInformation()	996
10.276.3 Member Function Documentation	996
10.276.3.1 Print()	996
10.276.3.2 Read()	996
10.276.3.3 SetTuple()	996

10.276.3.4 Size()	997
10.276.3.5 Write()	997
10.277 gdcmm::ServiceClassUser Class Reference	997
10.277.1 Detailed Description	1000
10.277.2 Constructor & Destructor Documentation	1000
10.277.2.1 ServiceClassUser() [1/2]	1000
10.277.2.2 ~ServiceClassUser()	1000
10.277.2.3 ServiceClassUser() [2/2]	1000
10.277.3 Member Function Documentation	1000
10.277.3.1 GetAETitle()	1000
10.277.3.2 GetCalledAETitle()	1001
10.277.3.3 GetTimeout()	1001
10.277.3.4 InitializeConnection()	1001
10.277.3.5 IsPresentationContextAccepted()	1001
10.277.3.6 New()	1001
10.277.3.7 operator=()	1001
10.277.3.8 SendEcho()	1001
10.277.3.9 SendFind()	1002
10.277.3.10 SendMove() [1/3]	1002
10.277.3.11 SendMove() [2/3]	1002
10.277.3.12 SendMove() [3/3]	1002
10.277.3.13 SendStore() [1/3]	1002
10.277.3.14 SendStore() [2/3]	1003
10.277.3.15 SendStore() [3/3]	1003
10.277.3.16 SetAETitle()	1003
10.277.3.17 SetCalledAETitle()	1003
10.277.3.18 SetHostname()	1003
10.277.3.19 SetPort()	1004
10.277.3.20 SetPortSCP()	1004
10.277.3.21 SetPresentationContexts()	1004
10.277.3.22 SetTimeout()	1004
10.277.3.23 StartAssociation()	1005
10.277.3.24 StopAssociation()	1005
10.278 gdcmm::SHA1 Class Reference	1005
10.278.1 Detailed Description	1006
10.278.2 Constructor & Destructor Documentation	1006
10.278.2.1 SHA1() [1/2]	1006
10.278.2.2 ~SHA1()	1006
10.278.2.3 SHA1() [2/2]	1006

10.278.3 Member Function Documentation . . . . .	1006
10.278.3.1 Compute() . . . . .	1006
10.278.3.2 ComputeFile() . . . . .	1006
10.278.3.3 operator=() . . . . .	1007
10.279 gdcmm::SimpleMemberCommand< T > Class Template Reference . . . . .	1007
10.279.1 Detailed Description . . . . .	1010
10.279.2 Member Typedef Documentation . . . . .	1010
10.279.2.1 Self . . . . .	1010
10.279.2.2 TMemberFunctionPointer . . . . .	1010
10.279.3 Constructor & Destructor Documentation . . . . .	1010
10.279.3.1 SimpleMemberCommand() [1/2] . . . . .	1010
10.279.3.2 SimpleMemberCommand() [2/2] . . . . .	1010
10.279.3.3 ~SimpleMemberCommand() . . . . .	1010
10.279.4 Member Function Documentation . . . . .	1011
10.279.4.1 Execute() [1/2] . . . . .	1011
10.279.4.2 Execute() [2/2] . . . . .	1011
10.279.4.3 New() . . . . .	1011
10.279.4.4 operator=() . . . . .	1011
10.279.4.5 SetCallbackFunction() . . . . .	1012
10.279.5 Member Data Documentation . . . . .	1012
10.279.5.1 m_MemberFunction . . . . .	1012
10.279.5.2 m_This . . . . .	1012
10.280 gdcmm::SimpleSubjectWatcher Class Reference . . . . .	1012
10.280.1 Detailed Description . . . . .	1013
10.280.2 Constructor & Destructor Documentation . . . . .	1013
10.280.2.1 SimpleSubjectWatcher() [1/2] . . . . .	1013
10.280.2.2 ~SimpleSubjectWatcher() . . . . .	1013
10.280.2.3 SimpleSubjectWatcher() [2/2] . . . . .	1013
10.280.3 Member Function Documentation . . . . .	1014
10.280.3.1 EndFilter() . . . . .	1014
10.280.3.2 operator=() . . . . .	1014
10.280.3.3 ShowAbort() . . . . .	1014
10.280.3.4 ShowAnonymization() . . . . .	1014
10.280.3.5 ShowData() . . . . .	1014
10.280.3.6 ShowDataSet() . . . . .	1014
10.280.3.7 ShowFileName() . . . . .	1014
10.280.3.8 ShowIteration() . . . . .	1015
10.280.3.9 ShowProgress() . . . . .	1015
10.280.3.10 StartFilter() . . . . .	1015

10.280.3.11 TestAbortOff()	1015
10.280.3.12 TestAbortOn()	1015
10.281 gdcm::MrProtocol::Slice Struct Reference	1015
10.281.1 Member Data Documentation	1016
10.281.1.1 Normal	1016
10.281.1.2 Position	1016
10.282 gdcm::MrProtocol::SliceArray Struct Reference	1016
10.282.1 Member Data Documentation	1017
10.282.1.1 Slices	1017
10.283 gdcm::SmartPointer< ObjectType > Class Template Reference	1017
10.283.1 Detailed Description	1018
10.283.2 Constructor & Destructor Documentation	1019
10.283.2.1 SmartPointer() [1/4]	1019
10.283.2.2 SmartPointer() [2/4]	1019
10.283.2.3 SmartPointer() [3/4]	1019
10.283.2.4 SmartPointer() [4/4]	1019
10.283.2.5 ~SmartPointer()	1019
10.283.3 Member Function Documentation	1019
10.283.3.1 GetPointer()	1019
10.283.3.2 operator ObjectType *()	1020
10.283.3.3 operator*()	1020
10.283.3.4 operator->()	1020
10.283.3.5 operator=() [1/3]	1020
10.283.3.6 operator=() [2/3]	1020
10.283.3.7 operator=() [3/3]	1020
10.284 gdcm::network::SOPClassExtendedNegociationSub Class Reference	1021
10.284.1 Detailed Description	1021
10.284.2 Constructor & Destructor Documentation	1021
10.284.2.1 SOPClassExtendedNegociationSub()	1021
10.284.3 Member Function Documentation	1021
10.284.3.1 Print()	1021
10.284.3.2 Read()	1021
10.284.3.3 SetTuple()	1022
10.284.3.4 Size()	1022
10.284.3.5 Write()	1022
10.285 gdcm::SOPClassUIDToIOD Class Reference	1022
10.285.1 Detailed Description	1023
10.285.2 Member Typedef Documentation	1023
10.285.2.1 const	1023

---

10.285.3 Member Function Documentation . . . . .	1023
10.285.3.1 GetIOD() . . . . .	1023
10.285.3.2 GetIODFromSOPClassUID() . . . . .	1023
10.285.3.3 GetNumberOfSOPClassToIOD() . . . . .	1023
10.285.3.4 GetSOPClassUIDFromIOD() . . . . .	1023
10.285.3.5 GetSOPClassUIDToIOD() . . . . .	1024
10.285.3.6 GetSOPClassUIDToIODs() . . . . .	1024
10.286 gdcm::Sorter Class Reference . . . . .	1024
10.286.1 Detailed Description . . . . .	1025
10.286.2 Member Typedef Documentation . . . . .	1026
10.286.2.1 SelectionMap . . . . .	1026
10.286.2.2 SortFunction . . . . .	1026
10.286.3 Constructor & Destructor Documentation . . . . .	1026
10.286.3.1 Sorter() . . . . .	1026
10.286.3.2 ~Sorter() . . . . .	1026
10.286.4 Member Function Documentation . . . . .	1026
10.286.4.1 AddSelect() . . . . .	1026
10.286.4.2 GetFileNames() . . . . .	1026
10.286.4.3 Print() . . . . .	1027
10.286.4.4 SetSortFunction() . . . . .	1027
10.286.4.5 SetTagsToRead() . . . . .	1027
10.286.4.6 Sort() . . . . .	1027
10.286.4.7 StableSort() . . . . .	1028
10.286.5 Friends And Related Symbol Documentation . . . . .	1028
10.286.5.1 operator<< . . . . .	1028
10.286.6 Member Data Documentation . . . . .	1028
10.286.6.1 FileNames . . . . .	1028
10.286.6.2 Selection . . . . .	1028
10.286.6.3 SortFunc . . . . .	1028
10.286.6.4 TagsToRead . . . . .	1028
10.287 gdcm::Spacing Class Reference . . . . .	1029
10.287.1 Detailed Description . . . . .	1029
10.287.2 Member Enumeration Documentation . . . . .	1030
10.287.2.1 SpacingType . . . . .	1030
10.287.3 Constructor & Destructor Documentation . . . . .	1030
10.287.3.1 Spacing() . . . . .	1030
10.287.3.2 ~Spacing() . . . . .	1031
10.287.4 Member Function Documentation . . . . .	1031
10.287.4.1 ComputePixelAspectRatioFromPixelSpacing() . . . . .	1031

10.288 gdcm::Spectroscopy Class Reference . . . . .	1031
10.288.1 Detailed Description . . . . .	1031
10.288.2 Constructor & Destructor Documentation . . . . .	1031
10.288.2.1 Spectroscopy() . . . . .	1031
10.289 gdcm::SplitMosaicFilter Class Reference . . . . .	1032
10.289.1 Detailed Description . . . . .	1032
10.289.2 Constructor & Destructor Documentation . . . . .	1033
10.289.2.1 SplitMosaicFilter() . . . . .	1033
10.289.2.2 ~SplitMosaicFilter() . . . . .	1033
10.289.3 Member Function Documentation . . . . .	1033
10.289.3.1 ComputeMOSAICDimensions() . . . . .	1033
10.289.3.2 ComputeMOSAICImagePositionPatient() . . . . .	1033
10.289.3.3 ComputeMOSAICSliceNormal() . . . . .	1033
10.289.3.4 ComputeMOSAICSlicePosition() . . . . .	1033
10.289.3.5 GetAcquisitionSize() . . . . .	1034
10.289.3.6 GetFile() [1/2] . . . . .	1034
10.289.3.7 GetFile() [2/2] . . . . .	1034
10.289.3.8 GetImage() [1/2] . . . . .	1034
10.289.3.9 GetImage() [2/2] . . . . .	1034
10.289.3.10 GetNumberOfImagesInMosaic() . . . . .	1034
10.289.3.11 SetFile() . . . . .	1034
10.289.3.12 SetImage() . . . . .	1034
10.289.3.13 Split() . . . . .	1035
10.290 gdcm::StartEvent Class Reference . . . . .	1035
10.291 gdcm::static_assert_test< x > Struct Template Reference . . . . .	1036
10.292 gdcm::STATIC_ASSERTION_FAILURE< x > Struct Template Reference . . . . .	1037
10.293 gdcm::STATIC_ASSERTION_FAILURE< true > Struct Reference . . . . .	1037
10.293.1 Member Enumeration Documentation . . . . .	1037
10.293.1.1 anonymous enum . . . . .	1037
10.294 gdcm::StreamImageReader Class Reference . . . . .	1037
10.294.1 Detailed Description . . . . .	1038
10.294.2 Constructor & Destructor Documentation . . . . .	1038
10.294.2.1 StreamImageReader() . . . . .	1038
10.294.2.2 ~StreamImageReader() . . . . .	1038
10.294.3 Member Function Documentation . . . . .	1039
10.294.3.1 CanReadImage() . . . . .	1039
10.294.3.2 DefinePixelExtent() . . . . .	1039
10.294.3.3 DefineProperBufferLength() . . . . .	1039
10.294.3.4 GetDimensionsValueForResolution() . . . . .	1040



10.294.3.5 GetFile()	1040
10.294.3.6 Read()	1040
10.294.3.7 ReadImageInformation()	1040
10.294.3.8 SetFileName()	1041
10.294.3.9 SetStream()	1041
10.295 gdcm::StreamImageWriter Class Reference	1041
10.295.1 Detailed Description	1043
10.295.2 Constructor & Destructor Documentation	1043
10.295.2.1 StreamImageWriter()	1043
10.295.2.2 ~StreamImageWriter()	1043
10.295.3 Member Function Documentation	1044
10.295.3.1 CanWriteFile()	1044
10.295.3.2 DefinePixelExtent()	1044
10.295.3.3 DefineProperBufferLength()	1044
10.295.3.4 SetFile()	1045
10.295.3.5 SetFileName()	1045
10.295.3.6 SetStream()	1045
10.295.3.7 Write()	1045
10.295.3.8 WriteImageInformation()	1046
10.295.3.9 WriteImageSubregionRAW()	1046
10.295.3.10 WriteRawHeader()	1046
10.295.4 Member Data Documentation	1046
10.295.4.1 mElementOffsets	1046
10.295.4.2 mElementOffsets1	1046
10.295.4.3 mspFile	1047
10.295.4.4 mWriter	1047
10.295.4.5 mXMax	1047
10.295.4.6 mXMin	1047
10.295.4.7 mYMax	1047
10.295.4.8 mYMin	1047
10.295.4.9 mZMax	1047
10.295.4.10 mZMin	1047
10.296 gdcm::StrictScanner Class Reference	1048
10.296.1 Detailed Description	1051
10.296.2 Member Typedef Documentation	1051
10.296.2.1 ConstIterator	1051
10.296.2.2 MappingType	1051
10.296.2.3 TagToValue	1051
10.296.2.4 TagToValueValueType	1052

10.296.2.5 ValueType	1052
10.296.3 Constructor & Destructor Documentation	1052
10.296.3.1 StrictScanner()	1052
10.296.3.2 ~StrictScanner()	1052
10.296.4 Member Function Documentation	1052
10.296.4.1 AddPrivateTag()	1052
10.296.4.2 AddSkipTag()	1052
10.296.4.3 AddTag()	1052
10.296.4.4 Begin()	1053
10.296.4.5 ClearSkipTags()	1053
10.296.4.6 ClearTags()	1053
10.296.4.7 End()	1053
10.296.4.8 GetAllFilenamesFromTagToValue()	1053
10.296.4.9 GetFilenameFromTagToValue()	1053
10.296.4.10 GetFilenames()	1053
10.296.4.11 GetKeys()	1053
10.296.4.12 GetMapping()	1054
10.296.4.13 GetMappingFromTagToValue()	1054
10.296.4.14 GetMappings()	1054
10.296.4.15 GetOrderedValues()	1054
10.296.4.16 GetValue()	1054
10.296.4.17 GetValues() [1/2]	1055
10.296.4.18 GetValues() [2/2]	1055
10.296.4.19 IsKey()	1055
10.296.4.20 New()	1055
10.296.4.21 Print()	1055
10.296.4.22 PrintTable()	1056
10.296.4.23 ProcessPublicTag()	1056
10.296.4.24 Scan()	1056
10.296.5 Friends And Related Symbol Documentation	1056
10.296.5.1 operator<<	1056
10.297 gdcmm::StrictScanner2 Class Reference	1057
10.297.1 Detailed Description	1060
10.297.2 Member Typedef Documentation	1060
10.297.2.1 PrivateConstIterator	1060
10.297.2.2 PrivateMappingType	1060
10.297.2.3 PrivateTagToValue	1060
10.297.2.4 PrivateTagToValueValueType	1061
10.297.2.5 PublicConstIterator	1061

10.297.2.6 PublicMappingType . . . . .	1061
10.297.2.7 PublicTagToValue . . . . .	1061
10.297.2.8 PublicTagToValueValueType . . . . .	1061
10.297.2.9 ValuesType . . . . .	1061
10.297.3 Constructor & Destructor Documentation . . . . .	1061
10.297.3.1 StrictScanner2() . . . . .	1061
10.297.3.2 ~StrictScanner2() . . . . .	1061
10.297.4 Member Function Documentation . . . . .	1062
10.297.4.1 AddPrivateTag() . . . . .	1062
10.297.4.2 AddPublicTag() . . . . .	1062
10.297.4.3 AddSkipTag() . . . . .	1062
10.297.4.4 Begin() . . . . .	1062
10.297.4.5 ClearPrivateTags() . . . . .	1062
10.297.4.6 ClearPublicTags() . . . . .	1062
10.297.4.7 ClearSkipTags() . . . . .	1062
10.297.4.8 End() . . . . .	1062
10.297.4.9 GetAllFilenamesFromPrivateTagToValue() . . . . .	1063
10.297.4.10 GetAllFilenamesFromPublicTagToValue() . . . . .	1063
10.297.4.11 GetFilenameFromPrivateTagToValue() . . . . .	1063
10.297.4.12 GetFilenameFromPublicTagToValue() . . . . .	1063
10.297.4.13 GetFilenames() . . . . .	1063
10.297.4.14 GetKeys() . . . . .	1063
10.297.4.15 GetMappingFromPrivateTagToValue() . . . . .	1063
10.297.4.16 GetMappingFromPublicTagToValue() . . . . .	1064
10.297.4.17 GetPrivateMapping() . . . . .	1064
10.297.4.18 GetPrivateMappings() . . . . .	1064
10.297.4.19 GetPrivateOrderedValues() . . . . .	1064
10.297.4.20 GetPrivateValue() . . . . .	1064
10.297.4.21 GetPrivateValues() . . . . .	1064
10.297.4.22 GetPublicMapping() . . . . .	1064
10.297.4.23 GetPublicMappings() . . . . .	1065
10.297.4.24 GetPublicOrderedValues() . . . . .	1065
10.297.4.25 GetPublicValue() . . . . .	1065
10.297.4.26 GetPublicValues() . . . . .	1065
10.297.4.27 GetValues() . . . . .	1065
10.297.4.28 IsKey() . . . . .	1065
10.297.4.29 New() . . . . .	1066
10.297.4.30 Print() . . . . .	1066
10.297.4.31 PrintTable() . . . . .	1066

10.297.4.32 PrivateBegin()	1066
10.297.4.33 PrivateEnd()	1066
10.297.4.34 ProcessPrivateTag()	1066
10.297.4.35 ProcessPublicTag()	1066
10.297.4.36 Scan()	1067
10.297.5 Friends And Related Symbol Documentation	1067
10.297.5.1 operator<<	1067
10.298 gdcmm::String< TDelimiter, TMaxLength, TPadChar > Class Template Reference	1067
10.298.1 Detailed Description	1069
10.298.2 Member Typedef Documentation	1069
10.298.2.1 const_iterator	1069
10.298.2.2 const_reference	1069
10.298.2.3 const_reverse_iterator	1069
10.298.2.4 difference_type	1069
10.298.2.5 iterator	1070
10.298.2.6 pointer	1070
10.298.2.7 reference	1070
10.298.2.8 reverse_iterator	1070
10.298.2.9 size_type	1070
10.298.2.10 value_type	1070
10.298.3 Constructor & Destructor Documentation	1070
10.298.3.1 String() [1/4]	1070
10.298.3.2 String() [2/4]	1071
10.298.3.3 String() [3/4]	1071
10.298.3.4 String() [4/4]	1071
10.298.4 Member Function Documentation	1071
10.298.4.1 IsValid()	1071
10.298.4.2 operator const char *()	1071
10.298.4.3 Trim() [1/2]	1072
10.298.4.4 Trim() [2/2]	1072
10.298.4.5 Truncate()	1072
10.299 gdcmm::StringFilter Class Reference	1072
10.299.1 Detailed Description	1073
10.299.2 Constructor & Destructor Documentation	1073
10.299.2.1 StringFilter()	1073
10.299.2.2 ~StringFilter()	1074
10.299.3 Member Function Documentation	1074
10.299.3.1 ExecuteQuery() [1/2]	1074
10.299.3.2 ExecuteQuery() [2/2]	1074

---

10.299.3.3 FromString()	1074
10.299.3.4 GetFile() [1/2]	1074
10.299.3.5 GetFile() [2/2]	1074
10.299.3.6 SetDicts()	1074
10.299.3.7 SetFile()	1075
10.299.3.8 ToString() [1/3]	1075
10.299.3.9 ToString() [2/3]	1075
10.299.3.10 ToString() [3/3]	1075
10.299.3.11 ToStringPair() [1/3]	1075
10.299.3.12 ToStringPair() [2/3]	1076
10.299.3.13 ToStringPair() [3/3]	1076
10.299.3.14 UseDictAlways()	1076
10.300 gdcmm::Study Class Reference	1076
10.300.1 Detailed Description	1076
10.300.2 Constructor & Destructor Documentation	1076
10.300.2.1 Study()	1076
10.301 gdcmm::Subject Class Reference	1077
10.301.1 Detailed Description	1078
10.301.2 Constructor & Destructor Documentation	1078
10.301.2.1 Subject()	1078
10.301.2.2 ~Subject()	1079
10.301.3 Member Function Documentation	1079
10.301.3.1 AddObserver() [1/2]	1079
10.301.3.2 AddObserver() [2/2]	1079
10.301.3.3 GetCommand()	1079
10.301.3.4 HasObserver()	1079
10.301.3.5 InvokeEvent() [1/2]	1080
10.301.3.6 InvokeEvent() [2/2]	1080
10.301.3.7 RemoveAllObservers()	1080
10.301.3.8 RemoveObserver()	1080
10.302 gdcmm::Surface Class Reference	1081
10.302.1 Detailed Description	1083
10.302.2 Member Enumeration Documentation	1084
10.302.2.1 STATES	1084
10.302.2.2 VIEWType	1084
10.302.3 Constructor & Destructor Documentation	1084
10.302.3.1 Surface()	1084
10.302.3.2 ~Surface()	1084
10.302.4 Member Function Documentation	1085

10.302.4.1 GetAlgorithmFamily() [1/2]	1085
10.302.4.2 GetAlgorithmFamily() [2/2]	1085
10.302.4.3 GetAlgorithmName()	1085
10.302.4.4 GetAlgorithmVersion()	1085
10.302.4.5 GetAxisOfRotation()	1085
10.302.4.6 GetCenterOfRotation()	1085
10.302.4.7 GetFiniteVolume()	1085
10.302.4.8 GetManifold()	1086
10.302.4.9 GetMaximumPointDistance()	1086
10.302.4.10 GetMeanPointDistance()	1086
10.302.4.11 GetMeshPrimitive() [1/2]	1086
10.302.4.12 GetMeshPrimitive() [2/2]	1086
10.302.4.13 GetNumberOfSurfacePoints()	1086
10.302.4.14 GetNumberOfVectors()	1086
10.302.4.15 GetPointCoordinatesData() [1/2]	1086
10.302.4.16 GetPointCoordinatesData() [2/2]	1086
10.302.4.17 GetPointPositionAccuracy()	1087
10.302.4.18 GetPointsBoundingBoxCoordinates()	1087
10.302.4.19 GetProcessingAlgorithm() [1/2]	1087
10.302.4.20 GetProcessingAlgorithm() [2/2]	1087
10.302.4.21 GetRecommendedDisplayCIELabValue() [1/2]	1087
10.302.4.22 GetRecommendedDisplayCIELabValue() [2/2]	1087
10.302.4.23 GetRecommendedDisplayGrayscaleValue()	1087
10.302.4.24 GetRecommendedPresentationOpacity()	1087
10.302.4.25 GetRecommendedPresentationType()	1088
10.302.4.26 GetSTATES()	1088
10.302.4.27 GetSTATESString()	1088
10.302.4.28 GetSurfaceComments()	1088
10.302.4.29 GetSurfaceNumber()	1088
10.302.4.30 GetSurfaceProcessing()	1088
10.302.4.31 GetSurfaceProcessingDescription()	1088
10.302.4.32 GetSurfaceProcessingRatio()	1088
10.302.4.33 GetVectorAccuracy()	1088
10.302.4.34 GetVectorCoordinateData() [1/2]	1089
10.302.4.35 GetVectorCoordinateData() [2/2]	1089
10.302.4.36 GetVectorDimensionality()	1089
10.302.4.37 GetVIEWType()	1089
10.302.4.38 GetVIEWTypeString()	1089
10.302.4.39 SetAlgorithmFamily()	1089

10.302.4.40 SetAlgorithmName()	1089
10.302.4.41 SetAlgorithmVersion()	1089
10.302.4.42 SetAxisOfRotation()	1089
10.302.4.43 SetCenterOfRotation()	1090
10.302.4.44 SetFiniteVolume()	1090
10.302.4.45 SetManifold()	1090
10.302.4.46 SetMaximumPointDistance()	1090
10.302.4.47 SetMeanPointDistance()	1090
10.302.4.48 SetMeshPrimitive()	1090
10.302.4.49 SetNumberOfSurfacePoints()	1090
10.302.4.50 SetNumberOfVectors()	1090
10.302.4.51 SetPointCoordinatesData()	1091
10.302.4.52 SetPointPositionAccuracy()	1091
10.302.4.53 SetPointsBoundingBoxCoordinates()	1091
10.302.4.54 SetProcessingAlgorithm()	1091
10.302.4.55 SetRecommendedDisplayCIELabValue() [1/3]	1091
10.302.4.56 SetRecommendedDisplayCIELabValue() [2/3]	1091
10.302.4.57 SetRecommendedDisplayCIELabValue() [3/3]	1091
10.302.4.58 SetRecommendedDisplayGrayscaleValue()	1091
10.302.4.59 SetRecommendedPresentationOpacity()	1092
10.302.4.60 SetRecommendedPresentationType()	1092
10.302.4.61 SetSurfaceComments()	1092
10.302.4.62 SetSurfaceNumber()	1092
10.302.4.63 SetSurfaceProcessing()	1092
10.302.4.64 SetSurfaceProcessingDescription()	1092
10.302.4.65 SetSurfaceProcessingRatio()	1092
10.302.4.66 SetVectorAccuracy()	1092
10.302.4.67 SetVectorCoordinateData()	1093
10.302.4.68 SetVectorDimensionality()	1093
10.303 gdcmm::SurfaceHelper Class Reference	1093
10.303.1 Detailed Description	1094
10.303.2 Member Typedef Documentation	1094
10.303.2.1 ColorArray	1094
10.303.3 Member Function Documentation	1094
10.303.3.1 RecommendedDisplayCIELabToRGB() [1/2]	1094
10.303.3.2 RecommendedDisplayCIELabToRGB() [2/2]	1094
10.303.3.3 RGBToRecommendedDisplayCIELab()	1095
10.303.3.4 RGBToRecommendedDisplayGrayscale()	1095
10.304 gdcmm::SurfaceReader Class Reference	1096

10.304.1 Detailed Description . . . . .	1099
10.304.2 Constructor & Destructor Documentation . . . . .	1099
10.304.2.1 SurfaceReader() . . . . .	1099
10.304.2.2 ~SurfaceReader() . . . . .	1099
10.304.3 Member Function Documentation . . . . .	1099
10.304.3.1 GetNumberOfSurfaces() . . . . .	1099
10.304.3.2 Read() . . . . .	1099
10.304.3.3 ReadPointMacro() . . . . .	1100
10.304.3.4 ReadSurface() . . . . .	1100
10.304.3.5 ReadSurfaces() . . . . .	1100
10.305 gdcm::SurfaceWriter Class Reference . . . . .	1100
10.305.1 Detailed Description . . . . .	1102
10.305.2 Constructor & Destructor Documentation . . . . .	1103
10.305.2.1 SurfaceWriter() . . . . .	1103
10.305.2.2 ~SurfaceWriter() . . . . .	1103
10.305.3 Member Function Documentation . . . . .	1103
10.305.3.1 ComputeNumberOfSurfaces() . . . . .	1103
10.305.3.2 GetNumberOfSurfaces() . . . . .	1103
10.305.3.3 PrepareWrite() . . . . .	1103
10.305.3.4 PrepareWritePointMacro() . . . . .	1103
10.305.3.5 SetNumberOfSurfaces() . . . . .	1103
10.305.3.6 Write() . . . . .	1103
10.305.4 Member Data Documentation . . . . .	1104
10.305.4.1 NumberOfSurfaces . . . . .	1104
10.306 gdcm::SwapCode Class Reference . . . . .	1104
10.306.1 Detailed Description . . . . .	1105
10.306.2 Member Enumeration Documentation . . . . .	1105
10.306.2.1 SwapCodeType . . . . .	1105
10.306.3 Constructor & Destructor Documentation . . . . .	1105
10.306.3.1 SwapCode() . . . . .	1105
10.306.4 Member Function Documentation . . . . .	1105
10.306.4.1 GetIndex() . . . . .	1105
10.306.4.2 GetSwapCodeString() . . . . .	1105
10.306.4.3 operator SwapCode::SwapCodeType() . . . . .	1106
10.306.5 Friends And Related Symbol Documentation . . . . .	1106
10.306.5.1 operator<< . . . . .	1106
10.307 gdcm::SwapperDoOp Class Reference . . . . .	1106
10.307.1 Member Function Documentation . . . . .	1106
10.307.1.1 Swap() . . . . .	1106



10.307.1.2 SwapArray()	1106
10.308 gdcmm::SwapperNoOp Class Reference	1107
10.308.1 Detailed Description	1107
10.308.2 Member Function Documentation	1107
10.308.2.1 Swap()	1107
10.308.2.2 SwapArray()	1107
10.309 gdcmm::System Class Reference	1107
10.309.1 Detailed Description	1109
10.309.2 Member Function Documentation	1109
10.309.2.1 ConvertToUNC()	1109
10.309.2.2 DeleteDirectory()	1109
10.309.2.3 EncodeBytes()	1109
10.309.2.4 FileExists()	1109
10.309.2.5 FileIsDirectory()	1110
10.309.2.6 FileIsSymlink()	1110
10.309.2.7 FileSize()	1110
10.309.2.8 FileTime()	1110
10.309.2.9 FormatDateTime()	1111
10.309.2.10 GetCurrentDateTime()	1111
10.309.2.11 GetCurrentModuleFileName()	1111
10.309.2.12 GetCurrentProcessFileName()	1111
10.309.2.13 GetCurrentResourcesDirectory()	1111
10.309.2.14 GetCWD()	1111
10.309.2.15 GetHostName()	1112
10.309.2.16 GetLastSystemError()	1112
10.309.2.17 GetLocaleCharset()	1112
10.309.2.18 GetPermissions()	1112
10.309.2.19 GetTimezoneOffsetFromUTC()	1112
10.309.2.20 MakeDirectory()	1112
10.309.2.21 ParseDateTime() [1/2]	1113
10.309.2.22 ParseDateTime() [2/2]	1113
10.309.2.23 RemoveFile()	1113
10.309.2.24 SetPermissions()	1113
10.309.2.25 StrCaseCmp()	1113
10.309.2.26 StrNCaseCmp()	1114
10.309.2.27 StrSep()	1114
10.309.2.28 StrTokR()	1114
10.310 gdcmm::Table Class Reference	1114
10.310.1 Detailed Description	1116

10.310.2 Member Typedef Documentation	1116
10.310.2.1 MapTableEntry	1116
10.310.3 Constructor & Destructor Documentation	1116
10.310.3.1 Table() [1/2]	1116
10.310.3.2 ~Table()	1116
10.310.3.3 Table() [2/2]	1116
10.310.4 Member Function Documentation	1116
10.310.4.1 GetTableEntry()	1116
10.310.4.2 InsertEntry()	1116
10.310.4.3 operator=()	1117
10.310.5 Friends And Related Symbol Documentation	1117
10.310.5.1 operator<<	1117
10.310.6 Member Data Documentation	1117
10.310.6.1 TableInternal	1117
10.311 gdcm::TableEntry Class Reference	1117
10.311.1 Detailed Description	1117
10.311.2 Constructor & Destructor Documentation	1118
10.311.2.1 TableEntry()	1118
10.311.2.2 ~TableEntry()	1118
10.312 gdcm::TableReader Class Reference	1118
10.312.1 Detailed Description	1119
10.312.2 Constructor & Destructor Documentation	1119
10.312.2.1 TableReader()	1119
10.312.2.2 ~TableReader()	1119
10.312.3 Member Function Documentation	1119
10.312.3.1 CharacterDataHandler()	1119
10.312.3.2 EndElement()	1120
10.312.3.3 GetDefs()	1120
10.312.3.4 GetFilename()	1120
10.312.3.5 HandleIOD()	1120
10.312.3.6 HandleIODEntry()	1120
10.312.3.7 HandleMacro()	1120
10.312.3.8 HandleMacroEntry()	1120
10.312.3.9 HandleMacroEntryDescription()	1120
10.312.3.10 HandleModule()	1121
10.312.3.11 HandleModuleEntry()	1121
10.312.3.12 HandleModuleEntryDescription()	1121
10.312.3.13 HandleModuleInclude()	1121
10.312.3.14 Read()	1121

10.312.3.15 SetFilename()	1121
10.312.3.16 StartElement()	1121
10.313 gdcmm::network::TableRow Class Reference	1122
10.313.1 Constructor & Destructor Documentation	1122
10.313.1.1 TableRow()	1122
10.313.1.2 ~TableRow()	1122
10.313.2 Member Data Documentation	1123
10.313.2.1 transitions	1123
10.314 gdcmm::Tag Class Reference	1123
10.314.1 Detailed Description	1125
10.314.2 Constructor & Destructor Documentation	1125
10.314.2.1 Tag() [1/3]	1125
10.314.2.2 Tag() [2/3]	1125
10.314.2.3 Tag() [3/3]	1126
10.314.3 Member Function Documentation	1126
10.314.3.1 GetElement()	1126
10.314.3.2 GetElementTag()	1126
10.314.3.3 GetGroup()	1126
10.314.3.4 GetLength()	1127
10.314.3.5 GetPrivateCreator()	1127
10.314.3.6 IsGroupLength()	1127
10.314.3.7 IsGroupXX()	1127
10.314.3.8 IsIllegal()	1127
10.314.3.9 IsPrivate()	1127
10.314.3.10 IsPrivateCreator()	1128
10.314.3.11 IsPublic()	1128
10.314.3.12 operator!=(())	1128
10.314.3.13 operator<()	1128
10.314.3.14 operator<=()	1128
10.314.3.15 operator=()	1128
10.314.3.16 operator==(())	1129
10.314.3.17 operator[]() [1/2]	1129
10.314.3.18 operator[]() [2/2]	1129
10.314.3.19 PrintAsContinuousString()	1129
10.314.3.20 PrintAsContinuousUpperCaseString()	1129
10.314.3.21 PrintAsPipeSeparatedString()	1129
10.314.3.22 Read()	1130
10.314.3.23 ReadFromCommaSeparatedString()	1130
10.314.3.24 ReadFromContinuousString()	1130

10.314.3.25 ReadFromPipeSeparatedString()	1130
10.314.3.26 SetElement()	1130
10.314.3.27 SetElementTag() [1/2]	1131
10.314.3.28 SetElementTag() [2/2]	1131
10.314.3.29 SetGroup()	1131
10.314.3.30 SetPrivateCreator()	1131
10.314.3.31 Write()	1131
10.314.4 Friends And Related Symbol Documentation	1132
10.314.4.1 operator<<	1132
10.314.4.2 operator>>	1132
10.314.5 Member Data Documentation	1132
10.314.5.1 bytes	1132
10.314.5.2 tag	1132
10.314.5.3 tags	1132
10.315 gdcmm::TagPath Class Reference	1132
10.315.1 Detailed Description	1133
10.315.2 Constructor & Destructor Documentation	1133
10.315.2.1 TagPath()	1133
10.315.2.2 ~TagPath()	1133
10.315.3 Member Function Documentation	1133
10.315.3.1 ConstructFromString()	1133
10.315.3.2 ConstructFromTagList()	1134
10.315.3.3 IsValid()	1134
10.315.3.4 Print()	1134
10.315.3.5 Push() [1/2]	1134
10.315.3.6 Push() [2/2]	1134
10.316 gdcmm::Testing Class Reference	1134
10.316.1 Detailed Description	1136
10.316.2 Member Typedef Documentation	1136
10.316.2.1 MD5DataImagesType	1136
10.316.2.2 MediaStorageDataFilesType	1136
10.316.3 Constructor & Destructor Documentation	1136
10.316.3.1 Testing()	1136
10.316.3.2 ~Testing()	1136
10.316.4 Member Function Documentation	1136
10.316.4.1 ComputeFileMD5()	1136
10.316.4.2 ComputeMD5()	1137
10.316.4.3 GetDataExtraRoot()	1137
10.316.4.4 GetDataRoot()	1137

10.316.4.5 GetFileName()	1137
10.316.4.6 GetFileNames()	1138
10.316.4.7 GetLossyFlagFromFile()	1138
10.316.4.8 GetMD5DataImage()	1138
10.316.4.9 GetMD5DataImages()	1138
10.316.4.10 GetMD5FromBrokenFile()	1138
10.316.4.11 GetMD5FromFile()	1138
10.316.4.12 GetMediaStorageDataFile()	1138
10.316.4.13 GetMediaStorageDataFiles()	1139
10.316.4.14 GetMediaStorageFromFile()	1139
10.316.4.15 GetNumberOfFileNames()	1139
10.316.4.16 GetNumberOfMD5DataImages()	1139
10.316.4.17 GetNumberOfMediaStorageDataFiles()	1139
10.316.4.18 GetPixelSpacingDataRoot()	1139
10.316.4.19 GetSelectedPrivateGroupOffsetFromFile()	1139
10.316.4.20 GetSelectedTagsOffsetFromFile()	1140
10.316.4.21 GetSourceDirectory()	1140
10.316.4.22 GetStreamOffsetFromFile()	1140
10.316.4.23 GetTempDirectory()	1140
10.316.4.24 GetTempDirectoryW()	1140
10.316.4.25 GetTempFilename()	1141
10.316.4.26 GetTempFilenameW()	1141
10.316.4.27 Print()	1141
10.317 gdcm::Trace Class Reference	1141
10.317.1 Detailed Description	1142
10.317.2 Constructor & Destructor Documentation	1143
10.317.2.1 Trace()	1143
10.317.2.2 ~Trace()	1143
10.317.3 Member Function Documentation	1143
10.317.3.1 DebugOff()	1143
10.317.3.2 DebugOn()	1143
10.317.3.3 ErrorOff()	1143
10.317.3.4 ErrorOn()	1143
10.317.3.5 GetDebugFlag()	1144
10.317.3.6 GetDebugStream()	1144
10.317.3.7 GetErrorFlag()	1144
10.317.3.8 GetErrorStream()	1144
10.317.3.9 GetStream()	1144
10.317.3.10 GetWarningFlag()	1144

10.317.3.11	<a href="#">GetWarningStream()</a>	1144
10.317.3.12	<a href="#">SetDebug()</a>	1144
10.317.3.13	<a href="#">SetDebugStream()</a>	1145
10.317.3.14	<a href="#">SetError()</a>	1145
10.317.3.15	<a href="#">SetErrorStream()</a>	1145
10.317.3.16	<a href="#">SetStream()</a>	1145
10.317.3.17	<a href="#">SetStreamToFile()</a>	1145
10.317.3.18	<a href="#">SetWarning()</a>	1146
10.317.3.19	<a href="#">SetWarningStream()</a>	1146
10.317.3.20	<a href="#">WarningOff()</a>	1146
10.317.3.21	<a href="#">WarningOn()</a>	1146
10.318	<a href="#">gdcm::TransferSyntax Class Reference</a>	1146
10.318.1	<a href="#">Detailed Description</a>	1148
10.318.2	<a href="#">Member Enumeration Documentation</a>	1148
10.318.2.1	<a href="#">NegociatedType</a>	1148
10.318.2.2	<a href="#">TSType</a>	1149
10.318.3	<a href="#">Constructor &amp; Destructor Documentation</a>	1149
10.318.3.1	<a href="#">TransferSyntax()</a>	1149
10.318.4	<a href="#">Member Function Documentation</a>	1150
10.318.4.1	<a href="#">CanStoreLossy()</a>	1150
10.318.4.2	<a href="#">GetNegociatedType()</a>	1150
10.318.4.3	<a href="#">GetString()</a>	1150
10.318.4.4	<a href="#">GetSwapCode()</a>	1150
10.318.4.5	<a href="#">GetTSSString()</a>	1150
10.318.4.6	<a href="#">GetTSType()</a>	1150
10.318.4.7	<a href="#">IsEncapsulated()</a>	1151
10.318.4.8	<a href="#">IsEncoded()</a>	1151
10.318.4.9	<a href="#">IsExplicit()</a>	1151
10.318.4.10	<a href="#">IsImplicit()</a>	1151
10.318.4.11	<a href="#">IsLossless()</a>	1151
10.318.4.12	<a href="#">IsLossy()</a>	1151
10.318.4.13	<a href="#">IsValid()</a>	1151
10.318.4.14	<a href="#">operator TSType()</a>	1151
10.318.5	<a href="#">Friends And Related Symbol Documentation</a>	1152
10.318.5.1	<a href="#">operator&lt;&lt;</a>	1152
10.319	<a href="#">gdcm::network::TransferSyntaxSub Class Reference</a>	1152
10.319.1	<a href="#">Detailed Description</a>	1152
10.319.2	<a href="#">Constructor &amp; Destructor Documentation</a>	1152
10.319.2.1	<a href="#">TransferSyntaxSub()</a>	1152

10.319.3 Member Function Documentation . . . . .	1153
10.319.3.1 GetName() . . . . .	1153
10.319.3.2 operator==( ) . . . . .	1153
10.319.3.3 Print() . . . . .	1153
10.319.3.4 Read() . . . . .	1153
10.319.3.5 SetName() . . . . .	1153
10.319.3.6 SetNameFromUID() . . . . .	1153
10.319.3.7 Size() . . . . .	1153
10.319.3.8 Write() . . . . .	1153
10.320 gdcmm::network::Transition Struct Reference . . . . .	1154
10.320.1 Constructor & Destructor Documentation . . . . .	1154
10.320.1.1 Transition() [1/2] . . . . .	1154
10.320.1.2 ~Transition() . . . . .	1155
10.320.1.3 Transition() [2/2] . . . . .	1155
10.320.2 Member Function Documentation . . . . .	1155
10.320.2.1 MakeNew() . . . . .	1155
10.320.3 Member Data Documentation . . . . .	1155
10.320.3.1 mAction . . . . .	1155
10.320.3.2 mEnd . . . . .	1155
10.321 gdcmm::Type Class Reference . . . . .	1156
10.321.1 Detailed Description . . . . .	1156
10.321.2 Member Enumeration Documentation . . . . .	1156
10.321.2.1 TypeType . . . . .	1156
10.321.3 Constructor & Destructor Documentation . . . . .	1157
10.321.3.1 Type() . . . . .	1157
10.321.4 Member Function Documentation . . . . .	1157
10.321.4.1 GetTypeString() . . . . .	1157
10.321.4.2 GetTypeType() . . . . .	1157
10.321.4.3 operator TypeType() . . . . .	1157
10.321.5 Friends And Related Symbol Documentation . . . . .	1157
10.321.5.1 operator<< . . . . .	1157
10.322 gdcmm::UI Struct Reference . . . . .	1158
10.322.1 Friends And Related Symbol Documentation . . . . .	1158
10.322.1.1 operator<< . . . . .	1158
10.322.2 Member Data Documentation . . . . .	1158
10.322.2.1 Internal . . . . .	1158
10.323 gdcmm::UIDGenerator Class Reference . . . . .	1158
10.323.1 Detailed Description . . . . .	1159
10.323.2 Constructor & Destructor Documentation . . . . .	1159

10.323.2.1 UIDGenerator()	1159
10.323.3 Member Function Documentation	1159
10.323.3.1 Generate()	1159
10.323.3.2 GenerateUUID()	1160
10.323.3.3 GetGDCMUID()	1160
10.323.3.4 GetRoot()	1160
10.323.3.5 IsValid()	1160
10.323.3.6 SetRoot()	1160
10.324 gdcmm::UIDs Class Reference	1161
10.324.1 Detailed Description	1176
10.324.2 Member Typedef Documentation	1177
10.324.2.1 TransferSyntaxStringsType	1177
10.324.3 Member Enumeration Documentation	1177
10.324.3.1 TSName	1177
10.324.3.2 TSType	1186
10.324.4 Member Function Documentation	1196
10.324.4.1 GetName()	1196
10.324.4.2 GetNumberOfTransferSyntaxStrings()	1196
10.324.4.3 GetString()	1196
10.324.4.4 GetTransferSyntaxString()	1196
10.324.4.5 GetTransferSyntaxStrings()	1196
10.324.4.6 GetUIDName()	1196
10.324.4.7 GetUIDString()	1197
10.324.4.8 operator TSType()	1197
10.324.4.9 SetFromUID()	1197
10.325 gdcmm::network::ULAction Class Reference	1197
10.325.1 Detailed Description	1199
10.325.2 Constructor & Destructor Documentation	1199
10.325.2.1 ULAction() [1/2]	1199
10.325.2.2 ~ULAction()	1199
10.325.2.3 ULAction() [2/2]	1200
10.325.3 Member Function Documentation	1200
10.325.3.1 operator=()	1200
10.325.3.2 PerformAction()	1200
10.326 gdcmm::network::ULActionAA1 Class Reference	1200
10.326.1 Member Function Documentation	1201
10.326.1.1 PerformAction()	1201
10.327 gdcmm::network::ULActionAA2 Class Reference	1202
10.327.1 Member Function Documentation	1203



---

10.327.1.1 PerformAction()	1203
10.328 gdcn::network::ULActionAA3 Class Reference	1203
10.328.1 Member Function Documentation	1204
10.328.1.1 PerformAction()	1204
10.329 gdcn::network::ULActionAA4 Class Reference	1204
10.329.1 Member Function Documentation	1205
10.329.1.1 PerformAction()	1205
10.330 gdcn::network::ULActionAA5 Class Reference	1206
10.330.1 Member Function Documentation	1207
10.330.1.1 PerformAction()	1207
10.331 gdcn::network::ULActionAA6 Class Reference	1207
10.331.1 Member Function Documentation	1208
10.331.1.1 PerformAction()	1208
10.332 gdcn::network::ULActionAA7 Class Reference	1208
10.332.1 Member Function Documentation	1209
10.332.1.1 PerformAction()	1209
10.333 gdcn::network::ULActionAA8 Class Reference	1210
10.333.1 Member Function Documentation	1211
10.333.1.1 PerformAction()	1211
10.334 gdcn::network::ULActionAE1 Class Reference	1211
10.334.1 Member Function Documentation	1212
10.334.1.1 PerformAction()	1212
10.335 gdcn::network::ULActionAE2 Class Reference	1212
10.335.1 Member Function Documentation	1213
10.335.1.1 PerformAction()	1213
10.336 gdcn::network::ULActionAE3 Class Reference	1214
10.336.1 Member Function Documentation	1215
10.336.1.1 PerformAction()	1215
10.337 gdcn::network::ULActionAE4 Class Reference	1215
10.337.1 Member Function Documentation	1216
10.337.1.1 PerformAction()	1216
10.338 gdcn::network::ULActionAE5 Class Reference	1216
10.338.1 Member Function Documentation	1217
10.338.1.1 PerformAction()	1217
10.339 gdcn::network::ULActionAE6 Class Reference	1218
10.339.1 Member Function Documentation	1219
10.339.1.1 PerformAction()	1219
10.340 gdcn::network::ULActionAE7 Class Reference	1219
10.340.1 Member Function Documentation	1220

10.340.1.1 PerformAction()	1220
10.341 gdcmm::network::ULActionAE8 Class Reference	1220
10.341.1 Member Function Documentation	1221
10.341.1.1 PerformAction()	1221
10.342 gdcmm::network::ULActionAR1 Class Reference	1222
10.342.1 Member Function Documentation	1223
10.342.1.1 PerformAction()	1223
10.343 gdcmm::network::ULActionAR10 Class Reference	1223
10.343.1 Member Function Documentation	1224
10.343.1.1 PerformAction()	1224
10.344 gdcmm::network::ULActionAR2 Class Reference	1224
10.344.1 Member Function Documentation	1225
10.344.1.1 PerformAction()	1225
10.345 gdcmm::network::ULActionAR3 Class Reference	1226
10.345.1 Member Function Documentation	1227
10.345.1.1 PerformAction()	1227
10.346 gdcmm::network::ULActionAR4 Class Reference	1227
10.346.1 Member Function Documentation	1228
10.346.1.1 PerformAction()	1228
10.347 gdcmm::network::ULActionAR5 Class Reference	1228
10.347.1 Member Function Documentation	1229
10.347.1.1 PerformAction()	1229
10.348 gdcmm::network::ULActionAR6 Class Reference	1230
10.348.1 Member Function Documentation	1231
10.348.1.1 PerformAction()	1231
10.349 gdcmm::network::ULActionAR7 Class Reference	1231
10.349.1 Member Function Documentation	1232
10.349.1.1 PerformAction()	1232
10.350 gdcmm::network::ULActionAR8 Class Reference	1232
10.350.1 Member Function Documentation	1233
10.350.1.1 PerformAction()	1233
10.351 gdcmm::network::ULActionAR9 Class Reference	1234
10.351.1 Member Function Documentation	1235
10.351.1.1 PerformAction()	1235
10.352 gdcmm::network::ULActionDT1 Class Reference	1235
10.352.1 Member Function Documentation	1236
10.352.1.1 PerformAction()	1236
10.353 gdcmm::network::ULActionDT2 Class Reference	1236
10.353.1 Member Function Documentation	1237

10.353.1.1 PerformAction()	1237
10.354 gdcn::network::ULBasicCallback Class Reference	1238
10.354.1 Detailed Description	1239
10.354.2 Constructor & Destructor Documentation	1239
10.354.2.1 ULBasicCallback()	1239
10.354.2.2 ~ULBasicCallback()	1239
10.354.3 Member Function Documentation	1239
10.354.3.1 GetDataSets()	1239
10.354.3.2 GetResponses()	1240
10.354.3.3 HandleDataSet()	1240
10.354.3.4 HandleResponse()	1240
10.355 gdcn::network::ULConnection Class Reference	1240
10.355.1 Detailed Description	1241
10.355.2 Constructor & Destructor Documentation	1241
10.355.2.1 ULConnection() [1/2]	1241
10.355.2.2 ~ULConnection()	1242
10.355.2.3 ULConnection() [2/2]	1242
10.355.3 Member Function Documentation	1242
10.355.3.1 AddAcceptedPresentationContext()	1242
10.355.3.2 FindContext()	1242
10.355.3.3 GetAcceptedPresentationContexts() [1/2]	1242
10.355.3.4 GetAcceptedPresentationContexts() [2/2]	1242
10.355.3.5 GetConnectionInfo()	1242
10.355.3.6 GetMaxPDUSize()	1242
10.355.3.7 GetPresentationContextACByID()	1243
10.355.3.8 GetPresentationContextIDFromPresentationContext()	1243
10.355.3.9 GetPresentationContextRQByID()	1243
10.355.3.10 GetPresentationContexts()	1243
10.355.3.11 GetProtocol()	1243
10.355.3.12 GetState()	1243
10.355.3.13 GetTimer()	1243
10.355.3.14 InitializeConnection()	1243
10.355.3.15 InitializeIncomingConnection()	1244
10.355.3.16 operator=()	1244
10.355.3.17 SetMaxPDUSize()	1244
10.355.3.18 SetPresentationContexts() [1/2]	1244
10.355.3.19 SetPresentationContexts() [2/2]	1244
10.355.3.20 SetState()	1244
10.355.3.21 StopProtocol()	1244

10.355.4 Friends And Related Symbol Documentation . . . . .	1244
10.355.4.1 ULActionAE6 . . . . .	1244
10.355.4.2 ULConnectionManager . . . . .	1245
10.356 gdcmm::network::ULConnectionCallback Class Reference . . . . .	1245
10.356.1 Detailed Description . . . . .	1246
10.356.2 Constructor & Destructor Documentation . . . . .	1246
10.356.2.1 ULConnectionCallback() . . . . .	1246
10.356.2.2 ~ULConnectionCallback() . . . . .	1246
10.356.3 Member Function Documentation . . . . .	1246
10.356.3.1 DataSetHandled() . . . . .	1246
10.356.3.2 DataSetHandles() . . . . .	1246
10.356.3.3 HandleDataSet() . . . . .	1246
10.356.3.4 HandleResponse() . . . . .	1246
10.356.3.5 ResetHandledDataSet() . . . . .	1247
10.356.3.6 SetImplicitFlag() . . . . .	1247
10.356.4 Member Data Documentation . . . . .	1247
10.356.4.1 mImplicit . . . . .	1247
10.357 gdcmm::network::ULConnectionInfo Class Reference . . . . .	1247
10.357.1 Detailed Description . . . . .	1247
10.357.2 Constructor & Destructor Documentation . . . . .	1248
10.357.2.1 ULConnectionInfo() . . . . .	1248
10.357.3 Member Function Documentation . . . . .	1248
10.357.3.1 GetCalledAETitle() . . . . .	1248
10.357.3.2 GetCalledComputerName() . . . . .	1248
10.357.3.3 GetCalledIPAddress() . . . . .	1248
10.357.3.4 GetCalledIPPort() . . . . .	1248
10.357.3.5 GetCallingAETitle() . . . . .	1248
10.357.3.6 GetMaxPDULength() . . . . .	1248
10.357.3.7 Initialize() . . . . .	1248
10.357.3.8 SetMaxPDULength() . . . . .	1249
10.358 gdcmm::network::ULConnectionManager Class Reference . . . . .	1249
10.358.1 Detailed Description . . . . .	1251
10.358.2 Constructor & Destructor Documentation . . . . .	1251
10.358.2.1 ULConnectionManager() [1/2] . . . . .	1251
10.358.2.2 ULConnectionManager() [2/2] . . . . .	1252
10.358.2.3 ~ULConnectionManager() . . . . .	1252
10.358.3 Member Function Documentation . . . . .	1252
10.358.3.1 BreakConnection() . . . . .	1252
10.358.3.2 BreakConnectionNow() . . . . .	1252

10.358.3.3 EstablishConnection()	1252
10.358.3.4 EstablishConnectionMove()	1252
10.358.3.5 RunEventLoop()	1253
10.358.3.6 RunMoveEventLoop()	1253
10.358.3.7 SendEcho()	1253
10.358.3.8 SendFind() [1/2]	1253
10.358.3.9 SendFind() [2/2]	1253
10.358.3.10 SendMove() [1/2]	1253
10.358.3.11 SendMove() [2/2]	1253
10.358.3.12 SendNAction() [1/2]	1254
10.358.3.13 SendNAction() [2/2]	1254
10.358.3.14 SendNCreate() [1/2]	1254
10.358.3.15 SendNCreate() [2/2]	1254
10.358.3.16 SendNDelete() [1/2]	1254
10.358.3.17 SendNDelete() [2/2]	1254
10.358.3.18 SendNEventReport() [1/2]	1254
10.358.3.19 SendNEventReport() [2/2]	1254
10.358.3.20 SendNGet() [1/2]	1255
10.358.3.21 SendNGet() [2/2]	1255
10.358.3.22 SendNSet() [1/2]	1255
10.358.3.23 SendNSet() [2/2]	1255
10.358.3.24 SendStore() [1/2]	1255
10.358.3.25 SendStore() [2/2]	1255
10.358.4 Member Data Documentation	1255
10.358.4.1 mConnection	1255
10.358.4.2 mSecondaryConnection	1256
10.358.4.3 mTransitions	1256
10.359 gdcn::network::ULError Class Reference	1256
10.359.1 Detailed Description	1256
10.359.2 Constructor & Destructor Documentation	1257
10.359.2.1 ULError() [1/2]	1257
10.359.2.2 ULError() [2/2]	1257
10.359.2.3 ~ULError()	1257
10.359.3 Member Function Documentation	1257
10.359.3.1 GetDataSetPos()	1257
10.359.3.2 GetEvent()	1257
10.359.3.3 GetIStream()	1257
10.359.3.4 GetPDUs()	1257
10.359.3.5 SetEvent()	1258

10.359.3.6 SetPDU()	1258
10.360 gdcm::network::ULTransitionTable Class Reference	1258
10.360.1 Detailed Description	1258
10.360.2 Constructor & Destructor Documentation	1258
10.360.2.1 ULTransitionTable()	1258
10.360.3 Member Function Documentation	1259
10.360.3.1 HandleEvent()	1259
10.360.3.2 PrintTable()	1259
10.361 gdcm::network::ULWritingCallback Class Reference	1259
10.361.1 Constructor & Destructor Documentation	1261
10.361.1.1 ULWritingCallback()	1261
10.361.1.2 ~ULWritingCallback()	1261
10.361.2 Member Function Documentation	1261
10.361.2.1 HandleDataSet()	1261
10.361.2.2 HandleResponse()	1261
10.361.2.3 SetDirectory()	1261
10.362 gdcm::UNExplicitDataElement Class Reference	1262
10.362.1 Detailed Description	1264
10.362.2 Member Function Documentation	1264
10.362.2.1 GetLength()	1264
10.362.2.2 Read()	1265
10.362.2.3 ReadPreValue()	1265
10.362.2.4 ReadValue()	1265
10.362.2.5 ReadWithLength()	1265
10.363 gdcm::UNExplicitImplicitDataElement Class Reference	1265
10.363.1 Detailed Description	1268
10.363.2 Member Function Documentation	1268
10.363.2.1 GetLength()	1268
10.363.2.2 Read()	1268
10.363.2.3 ReadPreValue()	1269
10.363.2.4 ReadValue()	1269
10.364 gdcm::Unpacker12Bits Class Reference	1269
10.364.1 Detailed Description	1269
10.364.2 Member Function Documentation	1270
10.364.2.1 Pack()	1270
10.364.2.2 Unpack()	1270
10.365 gdcm::Usage Class Reference	1270
10.365.1 Detailed Description	1271
10.365.2 Member Enumeration Documentation	1271

10.365.2.1 UsageType . . . . .	1271
10.365.3 Constructor & Destructor Documentation . . . . .	1272
10.365.3.1 Usage() . . . . .	1272
10.365.4 Member Function Documentation . . . . .	1272
10.365.4.1 GetUsageString() . . . . .	1272
10.365.4.2 GetUsageType() . . . . .	1272
10.365.4.3 operator UsageType() . . . . .	1272
10.365.5 Friends And Related Symbol Documentation . . . . .	1272
10.365.5.1 operator<< . . . . .	1272
10.366 gdcm::UserEvent Class Reference . . . . .	1273
10.367 gdcm::network::UserInformation Class Reference . . . . .	1274
10.367.1 Detailed Description . . . . .	1274
10.367.2 Constructor & Destructor Documentation . . . . .	1275
10.367.2.1 UserInformation() [1/2] . . . . .	1275
10.367.2.2 ~UserInformation() . . . . .	1275
10.367.2.3 UserInformation() [2/2] . . . . .	1275
10.367.3 Member Function Documentation . . . . .	1275
10.367.3.1 AddRoleSelectionSub() . . . . .	1275
10.367.3.2 AddSOPClassExtendedNegociationSub() . . . . .	1275
10.367.3.3 GetMaximumLengthSub() [1/2] . . . . .	1275
10.367.3.4 GetMaximumLengthSub() [2/2] . . . . .	1275
10.367.3.5 operator=() . . . . .	1275
10.367.3.6 Print() . . . . .	1276
10.367.3.7 Read() . . . . .	1276
10.367.3.8 Size() . . . . .	1276
10.367.3.9 Write() . . . . .	1276
10.368 gdcm::UUIDGenerator Class Reference . . . . .	1276
10.368.1 Detailed Description . . . . .	1277
10.368.2 Member Function Documentation . . . . .	1277
10.368.2.1 Generate() . . . . .	1277
10.368.2.2 IsValid() . . . . .	1277
10.369 gdcm::Validate Class Reference . . . . .	1277
10.369.1 Detailed Description . . . . .	1278
10.369.2 Constructor & Destructor Documentation . . . . .	1278
10.369.2.1 Validate() . . . . .	1278
10.369.2.2 ~Validate() . . . . .	1278
10.369.3 Member Function Documentation . . . . .	1278
10.369.3.1 GetValidatedFile() . . . . .	1278
10.369.3.2 SetFile() . . . . .	1278

10.369.3.3 Validation()	1278
10.369.4 Member Data Documentation	1279
10.369.4.1 F	1279
10.369.4.2 V	1279
10.370 gdcm::Value Class Reference	1279
10.370.1 Detailed Description	1280
10.370.2 Constructor & Destructor Documentation	1280
10.370.2.1 Value()	1280
10.370.2.2 ~Value()	1281
10.370.3 Member Function Documentation	1281
10.370.3.1 Clear()	1281
10.370.3.2 GetLength()	1281
10.370.3.3 operator==()	1281
10.370.3.4 SetLength()	1281
10.370.3.5 SetLengthOnly()	1281
10.370.4 Friends And Related Symbol Documentation	1282
10.370.4.1 DataElement	1282
10.371 gdcm::ValueIO< TDE, TSwap, TType > Class Template Reference	1282
10.371.1 Detailed Description	1282
10.371.2 Member Function Documentation	1282
10.371.2.1 Read()	1282
10.371.2.2 Write()	1283
10.372 gdcm::MrProtocol::Vector3 Struct Reference	1283
10.372.1 Member Data Documentation	1283
10.372.1.1 dCor	1283
10.372.1.2 dSag	1283
10.372.1.3 dTra	1283
10.373 gdcm::Version Class Reference	1284
10.373.1 Detailed Description	1284
10.373.2 Constructor & Destructor Documentation	1284
10.373.2.1 Version()	1284
10.373.2.2 ~Version()	1284
10.373.3 Member Function Documentation	1284
10.373.3.1 GetBuildVersion()	1284
10.373.3.2 GetMajorVersion()	1285
10.373.3.3 GetMinorVersion()	1285
10.373.3.4 GetVersion()	1285
10.373.3.5 Print()	1285
10.373.4 Friends And Related Symbol Documentation	1285



10.373.4.1 operator<<	1285
10.374 gdcm::VL Class Reference	1285
10.374.1 Detailed Description	1286
10.374.2 Member Typedef Documentation	1286
10.374.2.1 Type	1286
10.374.3 Constructor & Destructor Documentation	1287
10.374.3.1 VL()	1287
10.374.4 Member Function Documentation	1287
10.374.4.1 GetLength()	1287
10.374.4.2 GetVL16Max()	1287
10.374.4.3 GetVL32Max()	1287
10.374.4.4 IsOdd()	1287
10.374.4.5 IsUndefined()	1287
10.374.4.6 operator uint32_t()	1287
10.374.4.7 operator++() [1/2]	1288
10.374.4.8 operator++() [2/2]	1288
10.374.4.9 operator+=()	1288
10.374.4.10 Read()	1288
10.374.4.11 Read16()	1288
10.374.4.12 SetToUndefined()	1288
10.374.4.13 Write()	1288
10.374.4.14 Write16()	1289
10.374.5 Friends And Related Symbol Documentation	1289
10.374.5.1 operator<<	1289
10.375 gdcm::VM Class Reference	1289
10.375.1 Detailed Description	1291
10.375.2 Member Enumeration Documentation	1291
10.375.2.1 VMType	1291
10.375.3 Constructor & Destructor Documentation	1292
10.375.3.1 VM()	1292
10.375.4 Member Function Documentation	1292
10.375.4.1 Compatible()	1292
10.375.4.2 GetIndex()	1292
10.375.4.3 GetLength()	1292
10.375.4.4 GetNumberOfElementsFromArray()	1293
10.375.4.5 GetVMString()	1293
10.375.4.6 GetVMType()	1293
10.375.4.7 GetVMTypeFromLength()	1293
10.375.4.8 IsValid()	1293

10.375.4.9 operator VMType()	1293
10.375.5 Friends And Related Symbol Documentation	1293
10.375.5.1 operator<<	1293
10.376 gdcm::VMToLength< T > Struct Template Reference	1294
10.377 gdcm::VR Class Reference	1294
10.377.1 Detailed Description	1296
10.377.2 Member Enumeration Documentation	1296
10.377.2.1 VRType	1296
10.377.3 Constructor & Destructor Documentation	1297
10.377.3.1 VR()	1297
10.377.4 Member Function Documentation	1297
10.377.4.1 CanDisplay()	1297
10.377.4.2 Compatible()	1298
10.377.4.3 GetLength() [1/2]	1298
10.377.4.4 GetLength() [2/2]	1298
10.377.4.5 GetSize()	1298
10.377.4.6 GetSizeof()	1298
10.377.4.7 GetVRString()	1298
10.377.4.8 GetVRStringFromFile()	1298
10.377.4.9 GetVRType()	1299
10.377.4.10 GetVRTypeFromFile()	1299
10.377.4.11 IsASCII()	1299
10.377.4.12 IsASCII2()	1299
10.377.4.13 IsBinary()	1299
10.377.4.14 IsBinary2()	1299
10.377.4.15 IsDual()	1299
10.377.4.16 IsSwap()	1299
10.377.4.17 IsValid() [1/2]	1300
10.377.4.18 IsValid() [2/2]	1300
10.377.4.19 IsVRFile()	1300
10.377.4.20 operator VRType()	1300
10.377.4.21 Read()	1300
10.377.4.22 Write()	1300
10.377.5 Friends And Related Symbol Documentation	1300
10.377.5.1 operator<<	1300
10.378 gdcm::VR16ExplicitDataElement Class Reference	1301
10.378.1 Detailed Description	1303
10.378.2 Member Function Documentation	1303
10.378.2.1 GetLength()	1303

10.378.2.2 Read()	1304
10.378.2.3 ReadPreValue()	1304
10.378.2.4 ReadValue()	1304
10.378.2.5 ReadWithLength()	1304
10.379 gdcmm::VRToEncoding< T > Struct Template Reference	1304
10.380 gdcmm::VRToType< T > Struct Template Reference	1305
10.380.1 Detailed Description	1305
10.381 gdcmm::VRVLSIZE< T > Class Template Reference	1305
10.382 gdcmm::VRVLSIZE< 0 > Class Reference	1305
10.382.1 Member Function Documentation	1306
10.382.1.1 Read()	1306
10.382.1.2 Write()	1306
10.383 gdcmm::VRVLSIZE< 1 > Class Reference	1306
10.383.1 Member Function Documentation	1306
10.383.1.1 Read()	1306
10.383.1.2 Write()	1307
10.384 vtkGDCMImageReader Class Reference	1307
10.384.1 Detailed Description	1309
10.384.2 Constructor & Destructor Documentation	1310
10.384.2.1 vtkGDCMImageReader()	1310
10.384.2.2 ~vtkGDCMImageReader()	1310
10.384.3 Member Function Documentation	1310
10.384.3.1 CanReadFile()	1310
10.384.3.2 ExecuteData()	1310
10.384.3.3 ExecuteInformation()	1310
10.384.3.4 FillMedicalImageInformation()	1310
10.384.3.5 GetDescriptiveName()	1311
10.384.3.6 GetFileExtensions()	1311
10.384.3.7 GetIconImage()	1311
10.384.3.8 GetOverlay()	1311
10.384.3.9 LoadSingleFile()	1311
10.384.3.10 New()	1311
10.384.3.11 PrintSelf()	1311
10.384.3.12 RequestDataCompat()	1312
10.384.3.13 RequestInformationCompat()	1312
10.384.3.14 SetCurve()	1312
10.384.3.15 SetFileNames()	1312
10.384.3.16 SetFilePattern()	1312
10.384.3.17 SetFilePrefix()	1312

10.384.3.18 SetMedicalImageProperties()	1312
10.384.3.19 vtkBooleanMacro() [1/5]	1313
10.384.3.20 vtkBooleanMacro() [2/5]	1313
10.384.3.21 vtkBooleanMacro() [3/5]	1313
10.384.3.22 vtkBooleanMacro() [4/5]	1313
10.384.3.23 vtkBooleanMacro() [5/5]	1313
10.384.3.24 vtkGetMacro() [1/11]	1313
10.384.3.25 vtkGetMacro() [2/11]	1313
10.384.3.26 vtkGetMacro() [3/11]	1314
10.384.3.27 vtkGetMacro() [4/11]	1314
10.384.3.28 vtkGetMacro() [5/11]	1314
10.384.3.29 vtkGetMacro() [6/11]	1314
10.384.3.30 vtkGetMacro() [7/11]	1314
10.384.3.31 vtkGetMacro() [8/11]	1314
10.384.3.32 vtkGetMacro() [9/11]	1314
10.384.3.33 vtkGetMacro() [10/11]	1315
10.384.3.34 vtkGetMacro() [11/11]	1315
10.384.3.35 vtkGetObjectMacro() [1/4]	1315
10.384.3.36 vtkGetObjectMacro() [2/4]	1315
10.384.3.37 vtkGetObjectMacro() [3/4]	1315
10.384.3.38 vtkGetObjectMacro() [4/4]	1315
10.384.3.39 vtkGetStringMacro() [1/2]	1315
10.384.3.40 vtkGetStringMacro() [2/2]	1316
10.384.3.41 vtkGetVector3Macro()	1316
10.384.3.42 vtkGetVector6Macro()	1316
10.384.3.43 vtkSetMacro() [1/4]	1316
10.384.3.44 vtkSetMacro() [2/4]	1316
10.384.3.45 vtkSetMacro() [3/4]	1316
10.384.3.46 vtkSetMacro() [4/4]	1316
10.384.3.47 vtkSetVector6Macro()	1317
10.384.3.48 vtkTypeMacro()	1317
10.384.4 Member Data Documentation	1317
10.384.4.1 ApplyInverseVideo	1317
10.384.4.2 ApplyLookupTable	1317
10.384.4.3 ApplyPlanarConfiguration	1317
10.384.4.4 ApplyShiftScale	1317
10.384.4.5 ApplyYBRToRGB	1317
10.384.4.6 Curve	1317
10.384.4.7 DirectionCosines	1318

10.384.4.8 FileNames	1318
10.384.4.9 ForceRescale	1318
10.384.4.10 IconDataScalarType	1318
10.384.4.11 IconImageDataExtent	1318
10.384.4.12 IconNumberOfScalarComponents	1318
10.384.4.13 ImageFormat	1318
10.384.4.14 ImageOrientationPatient	1318
10.384.4.15 ImagePositionPatient	1318
10.384.4.16 LoadIconImage	1318
10.384.4.17 LoadOverlays	1319
10.384.4.18 LossyFlag	1319
10.384.4.19 MedicalImageProperties	1319
10.384.4.20 NumberOfIconImages	1319
10.384.4.21 NumberOfOverlays	1319
10.384.4.22 PlanarConfiguration	1319
10.384.4.23 Scale	1319
10.384.4.24 Shift	1319
10.385 vtkGDCMImageReader2 Class Reference	1320
10.385.1 Detailed Description	1322
10.385.2 Constructor & Destructor Documentation	1322
10.385.2.1 vtkGDCMImageReader2()	1322
10.385.2.2 ~vtkGDCMImageReader2()	1322
10.385.3 Member Function Documentation	1322
10.385.3.1 CanReadFile()	1322
10.385.3.2 FillMedicalImageInformation()	1323
10.385.3.3 GetDescriptiveName()	1323
10.385.3.4 GetFileExtensions()	1323
10.385.3.5 GetIconImage()	1323
10.385.3.6 GetIconImagePort()	1323
10.385.3.7 GetOverlay()	1323
10.385.3.8 GetOverlayPort()	1323
10.385.3.9 LoadSingleFile()	1323
10.385.3.10 New()	1324
10.385.3.11 PrintSelf()	1324
10.385.3.12 ProcessRequest()	1324
10.385.3.13 RequestData()	1324
10.385.3.14 RequestDataCompat()	1324
10.385.3.15 RequestInformation()	1324
10.385.3.16 RequestInformationCompat()	1324

10.385.3.17 SetCurve()	1325
10.385.3.18 SetFilePattern()	1325
10.385.3.19 SetFilePrefix()	1325
10.385.3.20 SetMedicalImageProperties()	1325
10.385.3.21 vtkBooleanMacro() [1/5]	1325
10.385.3.22 vtkBooleanMacro() [2/5]	1325
10.385.3.23 vtkBooleanMacro() [3/5]	1325
10.385.3.24 vtkBooleanMacro() [4/5]	1325
10.385.3.25 vtkBooleanMacro() [5/5]	1326
10.385.3.26 vtkGetMacro() [1/11]	1326
10.385.3.27 vtkGetMacro() [2/11]	1326
10.385.3.28 vtkGetMacro() [3/11]	1326
10.385.3.29 vtkGetMacro() [4/11]	1326
10.385.3.30 vtkGetMacro() [5/11]	1326
10.385.3.31 vtkGetMacro() [6/11]	1326
10.385.3.32 vtkGetMacro() [7/11]	1327
10.385.3.33 vtkGetMacro() [8/11]	1327
10.385.3.34 vtkGetMacro() [9/11]	1327
10.385.3.35 vtkGetMacro() [10/11]	1327
10.385.3.36 vtkGetMacro() [11/11]	1327
10.385.3.37 vtkGetObjectMacro() [1/2]	1327
10.385.3.38 vtkGetObjectMacro() [2/2]	1327
10.385.3.39 vtkGetStringMacro() [1/2]	1328
10.385.3.40 vtkGetStringMacro() [2/2]	1328
10.385.3.41 vtkGetVector3Macro()	1328
10.385.3.42 vtkGetVector6Macro()	1328
10.385.3.43 vtkSetMacro() [1/4]	1328
10.385.3.44 vtkSetMacro() [2/4]	1328
10.385.3.45 vtkSetMacro() [3/4]	1328
10.385.3.46 vtkSetMacro() [4/4]	1329
10.385.3.47 vtkSetVector6Macro()	1329
10.385.3.48 vtkTypeMacro()	1329
10.385.4 Member Data Documentation	1329
10.385.4.1 ApplyInverseVideo	1329
10.385.4.2 ApplyLookupTable	1329
10.385.4.3 ApplyPlanarConfiguration	1329
10.385.4.4 ApplyShiftScale	1329
10.385.4.5 ApplyYBRToRGB	1329
10.385.4.6 Curve	1330

10.385.4.7 DirectionCosines . . . . .	1330
10.385.4.8 ForceRescale . . . . .	1330
10.385.4.9 IconDataScalarType . . . . .	1330
10.385.4.10 IconImageDataExtent . . . . .	1330
10.385.4.11 IconNumberOfScalarComponents . . . . .	1330
10.385.4.12 ImageFormat . . . . .	1330
10.385.4.13 ImageOrientationPatient . . . . .	1330
10.385.4.14 ImagePositionPatient . . . . .	1330
10.385.4.15 LoadIconImage . . . . .	1330
10.385.4.16 LoadOverlays . . . . .	1331
10.385.4.17 LossyFlag . . . . .	1331
10.385.4.18 NumberOfIconImages . . . . .	1331
10.385.4.19 NumberOfOverlays . . . . .	1331
10.385.4.20 PlanarConfiguration . . . . .	1331
10.385.4.21 Scale . . . . .	1331
10.385.4.22 Shift . . . . .	1331
10.386 vtkGDCMImageWriter Class Reference . . . . .	1332
10.386.1 Detailed Description . . . . .	1334
10.386.2 Member Enumeration Documentation . . . . .	1334
10.386.2.1 CompressionTypes . . . . .	1334
10.386.3 Constructor & Destructor Documentation . . . . .	1334
10.386.3.1 vtkGDCMImageWriter() . . . . .	1334
10.386.3.2 ~vtkGDCMImageWriter() . . . . .	1334
10.386.4 Member Function Documentation . . . . .	1334
10.386.4.1 GetDescriptiveName() . . . . .	1334
10.386.4.2 GetFileExtensions() . . . . .	1334
10.386.4.3 GetFileName() . . . . .	1335
10.386.4.4 New() . . . . .	1335
10.386.4.5 PrintSelf() . . . . .	1335
10.386.4.6 SetDirectionCosines() . . . . .	1335
10.386.4.7 SetDirectionCosinesFromImageOrientationPatient() . . . . .	1335
10.386.4.8 SetFileNames() . . . . .	1335
10.386.4.9 SetMedicalImageProperties() . . . . .	1336
10.386.4.10 vtkBooleanMacro() [1/2] . . . . .	1336
10.386.4.11 vtkBooleanMacro() [2/2] . . . . .	1336
10.386.4.12 vtkGetMacro() [1/7] . . . . .	1336
10.386.4.13 vtkGetMacro() [2/7] . . . . .	1336
10.386.4.14 vtkGetMacro() [3/7] . . . . .	1336
10.386.4.15 vtkGetMacro() [4/7] . . . . .	1337

10.386.4.16	<a href="#">vtkGetMacro()</a> [5/7]	1337
10.386.4.17	<a href="#">vtkGetMacro()</a> [6/7]	1337
10.386.4.18	<a href="#">vtkGetMacro()</a> [7/7]	1337
10.386.4.19	<a href="#">vtkGetObjectMacro()</a> [1/3]	1337
10.386.4.20	<a href="#">vtkGetObjectMacro()</a> [2/3]	1337
10.386.4.21	<a href="#">vtkGetObjectMacro()</a> [3/3]	1337
10.386.4.22	<a href="#">vtkGetStringMacro()</a> [1/2]	1338
10.386.4.23	<a href="#">vtkGetStringMacro()</a> [2/2]	1338
10.386.4.24	<a href="#">vtkSetMacro()</a> [1/7]	1338
10.386.4.25	<a href="#">vtkSetMacro()</a> [2/7]	1338
10.386.4.26	<a href="#">vtkSetMacro()</a> [3/7]	1338
10.386.4.27	<a href="#">vtkSetMacro()</a> [4/7]	1338
10.386.4.28	<a href="#">vtkSetMacro()</a> [5/7]	1338
10.386.4.29	<a href="#">vtkSetMacro()</a> [6/7]	1339
10.386.4.30	<a href="#">vtkSetMacro()</a> [7/7]	1339
10.386.4.31	<a href="#">vtkSetStringMacro()</a> [1/2]	1339
10.386.4.32	<a href="#">vtkSetStringMacro()</a> [2/2]	1339
10.386.4.33	<a href="#">vtkTypeMacro()</a>	1339
10.386.4.34	<a href="#">Write()</a>	1339
10.386.4.35	<a href="#">WriteGDCMData()</a>	1339
10.386.4.36	<a href="#">WriteSlice()</a>	1340
10.387	<a href="#">vtkGDCMMedicalImageProperties Class Reference</a>	1340
10.387.1	<a href="#">Constructor &amp; Destructor Documentation</a>	1341
10.387.1.1	<a href="#">vtkGDCMMedicalImageProperties()</a>	1341
10.387.1.2	<a href="#">~vtkGDCMMedicalImageProperties()</a>	1341
10.387.2	<a href="#">Member Function Documentation</a>	1341
10.387.2.1	<a href="#">Clear()</a>	1341
10.387.2.2	<a href="#">GetFile()</a>	1341
10.387.2.3	<a href="#">New()</a>	1342
10.387.2.4	<a href="#">PrintSelf()</a>	1342
10.387.2.5	<a href="#">PushBackFile()</a>	1342
10.387.2.6	<a href="#">vtkTypeMacro()</a>	1342
10.387.3	<a href="#">Friends And Related Symbol Documentation</a>	1342
10.387.3.1	<a href="#">vtkGDCMImageReader</a>	1342
10.387.3.2	<a href="#">vtkGDCMImageReader2</a>	1342
10.387.3.3	<a href="#">vtkGDCMImageWriter</a>	1342
10.388	<a href="#">vtkGDCMPolyDataReader Class Reference</a>	1343
10.388.1	<a href="#">Detailed Description</a>	1344
10.388.2	<a href="#">Constructor &amp; Destructor Documentation</a>	1344



10.388.2.1	<a href="#">vtkGDCMPolyDataReader()</a>	1344
10.388.2.2	<a href="#">~vtkGDCMPolyDataReader()</a>	1344
10.388.3	<a href="#">Member Function Documentation</a>	1344
10.388.3.1	<a href="#">FillMedicalImageInformation()</a>	1344
10.388.3.2	<a href="#">New()</a>	1345
10.388.3.3	<a href="#">PrintSelf()</a>	1345
10.388.3.4	<a href="#">RequestData()</a>	1345
10.388.3.5	<a href="#">RequestData_HemodynamicWaveformStorage()</a>	1345
10.388.3.6	<a href="#">RequestData_RTStructureSetStorage()</a>	1345
10.388.3.7	<a href="#">RequestInformation()</a>	1345
10.388.3.8	<a href="#">RequestInformation_HemodynamicWaveformStorage()</a>	1345
10.388.3.9	<a href="#">RequestInformation_RTStructureSetStorage()</a>	1346
10.388.3.10	<a href="#">vtkGetObjectMacro()</a> [1/2]	1346
10.388.3.11	<a href="#">vtkGetObjectMacro()</a> [2/2]	1346
10.388.3.12	<a href="#">vtkGetStringMacro()</a>	1346
10.388.3.13	<a href="#">vtkSetStringMacro()</a>	1346
10.388.3.14	<a href="#">vtkTypeMacro()</a>	1346
10.388.4	<a href="#">Member Data Documentation</a>	1346
10.388.4.1	<a href="#">FileName</a>	1346
10.388.4.2	<a href="#">MedicalImageProperties</a>	1347
10.388.4.3	<a href="#">RTStructSetProperties</a>	1347
10.389	<a href="#">vtkGDCMPolyDataWriter Class Reference</a>	1347
10.389.1	<a href="#">Detailed Description</a>	1348
10.389.2	<a href="#">Constructor &amp; Destructor Documentation</a>	1348
10.389.2.1	<a href="#">vtkGDCMPolyDataWriter()</a>	1348
10.389.2.2	<a href="#">~vtkGDCMPolyDataWriter()</a>	1348
10.389.3	<a href="#">Member Function Documentation</a>	1349
10.389.3.1	<a href="#">InitializeRTStructSet()</a>	1349
10.389.3.2	<a href="#">New()</a>	1349
10.389.3.3	<a href="#">PrintSelf()</a>	1349
10.389.3.4	<a href="#">SetMedicalImageProperties()</a>	1349
10.389.3.5	<a href="#">SetNumberOfInputPorts()</a>	1349
10.389.3.6	<a href="#">SetRTStructSetProperties()</a>	1350
10.389.3.7	<a href="#">vtkTypeMacro()</a>	1350
10.389.3.8	<a href="#">WriteData()</a>	1350
10.389.3.9	<a href="#">WriteRTSTRUCTData()</a>	1350
10.389.3.10	<a href="#">WriteRTSTRUCTInfo()</a>	1350
10.389.4	<a href="#">Member Data Documentation</a>	1350
10.389.4.1	<a href="#">MedicalImageProperties</a>	1350

10.389.4.2 RTStructSetProperties	1350
10.390 vtkGDCMTesting Class Reference	1351
10.390.1 Detailed Description	1352
10.390.2 Member Typedef Documentation	1352
10.390.2.1 MD5MetainImagesType	1352
10.390.3 Constructor & Destructor Documentation	1352
10.390.3.1 vtkGDCMTesting()	1352
10.390.3.2 ~vtkGDCMTesting()	1352
10.390.4 Member Function Documentation	1352
10.390.4.1 GetGDCMDataRoot()	1352
10.390.4.2 GetMD5MetainImage()	1353
10.390.4.3 GetMHDMD5FromFile()	1353
10.390.4.4 GetNumberOfMD5MetainImages()	1353
10.390.4.5 GetRAWMD5FromFile()	1353
10.390.4.6 GetVTKDataRoot()	1353
10.390.4.7 New()	1353
10.390.4.8 PrintSelf()	1354
10.390.4.9 vtkTypeMacro()	1354
10.391 vtkGDCMThreadedImageReader Class Reference	1354
10.391.1 Constructor & Destructor Documentation	1357
10.391.1.1 vtkGDCMThreadedImageReader()	1357
10.391.1.2 ~vtkGDCMThreadedImageReader()	1357
10.391.2 Member Function Documentation	1357
10.391.2.1 ExecuteData()	1357
10.391.2.2 ExecuteInformation()	1358
10.391.2.3 New()	1358
10.391.2.4 PrintSelf()	1358
10.391.2.5 ReadFiles()	1358
10.391.2.6 RequestDataCompat()	1358
10.391.2.7 vtkBooleanMacro()	1358
10.391.2.8 vtkGetMacro()	1358
10.391.2.9 vtkSetMacro() [1/3]	1358
10.391.2.10 vtkSetMacro() [2/3]	1359
10.391.2.11 vtkSetMacro() [3/3]	1359
10.391.2.12 vtkTypeMacro()	1359
10.392 vtkGDCMThreadedImageReader2 Class Reference	1359
10.392.1 Constructor & Destructor Documentation	1361
10.392.1.1 vtkGDCMThreadedImageReader2()	1361
10.392.1.2 ~vtkGDCMThreadedImageReader2()	1361

---

10.392.2 Member Function Documentation	1361
10.392.2.1 GetFileName()	1361
10.392.2.2 New()	1361
10.392.2.3 PrintSelf()	1361
10.392.2.4 RequestInformation()	1362
10.392.2.5 SetFileName()	1362
10.392.2.6 SetFileNames()	1362
10.392.2.7 SplitExtent()	1362
10.392.2.8 ThreadedRequestData()	1362
10.392.2.9 vtkBooleanMacro() [1/3]	1362
10.392.2.10 vtkBooleanMacro() [2/3]	1363
10.392.2.11 vtkBooleanMacro() [3/3]	1363
10.392.2.12 vtkGetMacro() [1/8]	1363
10.392.2.13 vtkGetMacro() [2/8]	1363
10.392.2.14 vtkGetMacro() [3/8]	1363
10.392.2.15 vtkGetMacro() [4/8]	1363
10.392.2.16 vtkGetMacro() [5/8]	1363
10.392.2.17 vtkGetMacro() [6/8]	1364
10.392.2.18 vtkGetMacro() [7/8]	1364
10.392.2.19 vtkGetMacro() [8/8]	1364
10.392.2.20 vtkGetObjectMacro()	1364
10.392.2.21 vtkGetVector3Macro() [1/2]	1364
10.392.2.22 vtkGetVector3Macro() [2/2]	1364
10.392.2.23 vtkGetVector6Macro()	1364
10.392.2.24 vtkSetMacro() [1/7]	1365
10.392.2.25 vtkSetMacro() [2/7]	1365
10.392.2.26 vtkSetMacro() [3/7]	1365
10.392.2.27 vtkSetMacro() [4/7]	1365
10.392.2.28 vtkSetMacro() [5/7]	1365
10.392.2.29 vtkSetMacro() [6/7]	1365
10.392.2.30 vtkSetMacro() [7/7]	1365
10.392.2.31 vtkSetVector3Macro() [1/2]	1366
10.392.2.32 vtkSetVector3Macro() [2/2]	1366
10.392.2.33 vtkSetVector6Macro()	1366
10.392.2.34 vtkTypeMacro()	1366
10.393 vtkImageColorViewer Class Reference	1366
10.393.1 Detailed Description	1369
10.393.2 Member Enumeration Documentation	1369
10.393.2.1 anonymous enum	1369

10.393.3 Constructor & Destructor Documentation	1369
10.393.3.1 vtkImageColorViewer()	1369
10.393.3.2 ~vtkImageColorViewer()	1369
10.393.4 Member Function Documentation	1370
10.393.4.1 AddInput()	1370
10.393.4.2 AddInputConnection()	1370
10.393.4.3 GetColorLevel()	1370
10.393.4.4 GetColorWindow()	1370
10.393.4.5 GetInput()	1370
10.393.4.6 GetOffScreenRendering()	1370
10.393.4.7 GetOverlayVisibility()	1370
10.393.4.8 GetPosition()	1370
10.393.4.9 GetSize()	1370
10.393.4.10 GetSliceMax()	1371
10.393.4.11 GetSliceMin()	1371
10.393.4.12 GetSliceRange() [1/3]	1371
10.393.4.13 GetSliceRange() [2/3]	1371
10.393.4.14 GetSliceRange() [3/3]	1371
10.393.4.15 GetWindowName()	1371
10.393.4.16 InstallPipeline()	1371
10.393.4.17 New()	1371
10.393.4.18 PrintSelf()	1372
10.393.4.19 Render()	1372
10.393.4.20 SetColorLevel()	1372
10.393.4.21 SetColorWindow()	1372
10.393.4.22 SetDisplayId()	1372
10.393.4.23 SetInput()	1372
10.393.4.24 SetInputConnection()	1372
10.393.4.25 SetOffScreenRendering()	1373
10.393.4.26 SetOverlayVisibility()	1373
10.393.4.27 SetParentId()	1373
10.393.4.28 SetPosition() [1/2]	1373
10.393.4.29 SetPosition() [2/2]	1373
10.393.4.30 SetRenderer()	1373
10.393.4.31 SetRenderWindow()	1373
10.393.4.32 SetSize() [1/2]	1374
10.393.4.33 SetSize() [2/2]	1374
10.393.4.34 SetSlice()	1374
10.393.4.35 SetSliceOrientation()	1374

10.393.4.36 SetSliceOrientationToXY()	1374
10.393.4.37 SetSliceOrientationToXZ()	1374
10.393.4.38 SetSliceOrientationToYZ()	1375
10.393.4.39 SetupInteractor()	1375
10.393.4.40 SetWindowId()	1375
10.393.4.41 UnInstallPipeline()	1375
10.393.4.42 UpdateDisplayExtent()	1375
10.393.4.43 UpdateOrientation()	1375
10.393.4.44 vtkBooleanMacro()	1375
10.393.4.45 vtkGetMacro() [1/2]	1376
10.393.4.46 vtkGetMacro() [2/2]	1376
10.393.4.47 vtkGetObjectMacro() [1/5]	1376
10.393.4.48 vtkGetObjectMacro() [2/5]	1376
10.393.4.49 vtkGetObjectMacro() [3/5]	1376
10.393.4.50 vtkGetObjectMacro() [4/5]	1376
10.393.4.51 vtkGetObjectMacro() [5/5]	1376
10.393.4.52 vtkTypeMacro()	1377
10.393.5 Friends And Related Symbol Documentation	1377
10.393.5.1 vtkImageColorViewerCallback	1377
10.393.6 Member Data Documentation	1377
10.393.6.1 FirstRender	1377
10.393.6.2 ImageActor	1377
10.393.6.3 Interactor	1377
10.393.6.4 InteractorStyle	1377
10.393.6.5 OverlayImageActor	1377
10.393.6.6 Renderer	1377
10.393.6.7 RenderWindow	1378
10.393.6.8 Slice	1378
10.393.6.9 SliceOrientation	1378
10.393.6.10 WindowLevel	1378
10.394 vtkImageMapToColors16 Class Reference	1378
10.394.1 Constructor & Destructor Documentation	1380
10.394.1.1 vtkImageMapToColors16()	1380
10.394.1.2 ~vtkImageMapToColors16()	1380
10.394.2 Member Function Documentation	1380
10.394.2.1 GetMTime()	1380
10.394.2.2 New()	1380
10.394.2.3 PrintSelf()	1380
10.394.2.4 RequestData()	1380

10.394.2.5 RequestInformation()	1381
10.394.2.6 SetLookupTable()	1381
10.394.2.7 SetOutputFormatToLuminance()	1381
10.394.2.8 SetOutputFormatToLuminanceAlpha()	1381
10.394.2.9 SetOutputFormatToRGB()	1381
10.394.2.10 SetOutputFormatToRGBA()	1381
10.394.2.11 ThreadedRequestData()	1381
10.394.2.12 vtkBooleanMacro()	1382
10.394.2.13 vtkGetMacro() [1/3]	1382
10.394.2.14 vtkGetMacro() [2/3]	1382
10.394.2.15 vtkGetMacro() [3/3]	1382
10.394.2.16 vtkGetObjectMacro()	1382
10.394.2.17 vtkSetMacro() [1/3]	1382
10.394.2.18 vtkSetMacro() [2/3]	1382
10.394.2.19 vtkSetMacro() [3/3]	1383
10.394.2.20 vtkTypeMacro()	1383
10.394.3 Member Data Documentation	1383
10.394.3.1 ActiveComponent	1383
10.394.3.2 DataWasPassed	1383
10.394.3.3 LookupTable	1383
10.394.3.4 OutputFormat	1383
10.394.3.5 PassAlphaToOutput	1383
10.395 vtkImageMapToWindowLevelColors2 Class Reference	1384
10.395.1 Constructor & Destructor Documentation	1385
10.395.1.1 vtkImageMapToWindowLevelColors2()	1385
10.395.1.2 ~vtkImageMapToWindowLevelColors2()	1385
10.395.2 Member Function Documentation	1385
10.395.2.1 New()	1385
10.395.2.2 PrintSelf()	1385
10.395.2.3 RequestData()	1385
10.395.2.4 RequestInformation()	1386
10.395.2.5 ThreadedRequestData()	1386
10.395.2.6 vtkGetMacro() [1/2]	1386
10.395.2.7 vtkGetMacro() [2/2]	1386
10.395.2.8 vtkSetMacro() [1/2]	1386
10.395.2.9 vtkSetMacro() [2/2]	1386
10.395.2.10 vtkTypeMacro()	1387
10.395.3 Member Data Documentation	1387
10.395.3.1 Level	1387

10.395.3.2 Window	1387
10.396 vtkImagePlanarComponentsToComponents Class Reference	1387
10.396.1 Constructor & Destructor Documentation	1388
10.396.1.1 vtkImagePlanarComponentsToComponents()	1388
10.396.1.2 ~vtkImagePlanarComponentsToComponents()	1388
10.396.2 Member Function Documentation	1389
10.396.2.1 New()	1389
10.396.2.2 PrintSelf()	1389
10.396.2.3 RequestData()	1389
10.396.2.4 vtkTypeMacro()	1389
10.397 vtkImageRGBToYBR Class Reference	1389
10.397.1 Constructor & Destructor Documentation	1390
10.397.1.1 vtkImageRGBToYBR()	1390
10.397.1.2 ~vtkImageRGBToYBR()	1390
10.397.2 Member Function Documentation	1391
10.397.2.1 New()	1391
10.397.2.2 PrintSelf()	1391
10.397.2.3 ThreadedExecute()	1391
10.397.2.4 vtkTypeMacro()	1391
10.398 vtkImageYBRToRGB Class Reference	1391
10.398.1 Constructor & Destructor Documentation	1392
10.398.1.1 vtkImageYBRToRGB()	1392
10.398.1.2 ~vtkImageYBRToRGB()	1392
10.398.2 Member Function Documentation	1393
10.398.2.1 New()	1393
10.398.2.2 PrintSelf()	1393
10.398.2.3 ThreadedExecute()	1393
10.398.2.4 vtkTypeMacro()	1393
10.399 vtkLookupTable16 Class Reference	1393
10.399.1 Constructor & Destructor Documentation	1394
10.399.1.1 vtkLookupTable16()	1394
10.399.1.2 ~vtkLookupTable16()	1395
10.399.2 Member Function Documentation	1395
10.399.2.1 Build()	1395
10.399.2.2 GetPointer()	1395
10.399.2.3 MapScalarsThroughTable2()	1395
10.399.2.4 New()	1395
10.399.2.5 PrintSelf()	1395
10.399.2.6 SetNumberOfTableValues()	1395

10.399.2.7 vtkTypeMacro()	1396
10.399.2.8 WritePointer()	1396
10.399.3 Member Data Documentation	1396
10.399.3.1 Table16	1396
10.400 vtkRTStructSetProperties Class Reference	1396
10.400.1 Detailed Description	1398
10.400.2 Constructor & Destructor Documentation	1398
10.400.2.1 vtkRTStructSetProperties()	1398
10.400.2.2 ~vtkRTStructSetProperties()	1399
10.400.3 Member Function Documentation	1399
10.400.3.1 AddContourReferencedFrameOfReference()	1399
10.400.3.2 AddReferencedFrameOfReference()	1399
10.400.3.3 AddStructureSetROI()	1399
10.400.3.4 AddStructureSetROIObservation()	1399
10.400.3.5 Clear()	1399
10.400.3.6 DeepCopy()	1400
10.400.3.7 GetContourReferencedFrameOfReferenceClassUID()	1400
10.400.3.8 GetContourReferencedFrameOfReferenceInstanceUID()	1400
10.400.3.9 GetNumberOfContourReferencedFrameOfReferences() [1/2]	1400
10.400.3.10 GetNumberOfContourReferencedFrameOfReferences() [2/2]	1400
10.400.3.11 GetNumberOfReferencedFrameOfReferences()	1400
10.400.3.12 GetNumberOfStructureSetROIs()	1400
10.400.3.13 GetReferencedFrameOfReferenceClassUID()	1400
10.400.3.14 GetReferencedFrameOfReferenceInstanceUID()	1401
10.400.3.15 GetStructureSetObservationNumber()	1401
10.400.3.16 GetStructureSetROIDescription()	1401
10.400.3.17 GetStructureSetROIGenerationAlgorithm()	1401
10.400.3.18 GetStructureSetROIName()	1401
10.400.3.19 GetStructureSetROINumber()	1401
10.400.3.20 GetStructureSetROIObservationLabel()	1401
10.400.3.21 GetStructureSetROIRefFrameRefUID()	1401
10.400.3.22 GetStructureSetRTROIInterpretedType()	1402
10.400.3.23 New()	1402
10.400.3.24 PrintSelf()	1402
10.400.3.25 vtkGetStringMacro() [1/9]	1402
10.400.3.26 vtkGetStringMacro() [2/9]	1402
10.400.3.27 vtkGetStringMacro() [3/9]	1402
10.400.3.28 vtkGetStringMacro() [4/9]	1402
10.400.3.29 vtkGetStringMacro() [5/9]	1403



10.400.3.30 vtkGetStringMacro() [6/9]	1403
10.400.3.31 vtkGetStringMacro() [7/9]	1403
10.400.3.32 vtkGetStringMacro() [8/9]	1403
10.400.3.33 vtkGetStringMacro() [9/9]	1403
10.400.3.34 vtkSetStringMacro() [1/9]	1403
10.400.3.35 vtkSetStringMacro() [2/9]	1403
10.400.3.36 vtkSetStringMacro() [3/9]	1403
10.400.3.37 vtkSetStringMacro() [4/9]	1404
10.400.3.38 vtkSetStringMacro() [5/9]	1404
10.400.3.39 vtkSetStringMacro() [6/9]	1404
10.400.3.40 vtkSetStringMacro() [7/9]	1404
10.400.3.41 vtkSetStringMacro() [8/9]	1404
10.400.3.42 vtkSetStringMacro() [9/9]	1404
10.400.3.43 vtkTypeMacro()	1404
10.400.4 Member Data Documentation	1404
10.400.4.1 Internals	1404
10.400.4.2 ReferenceFrameOfReferenceUID	1405
10.400.4.3 ReferenceSeriesInstanceUID	1405
10.400.4.4 SeriesInstanceUID	1405
10.400.4.5 SOPInstanceUID	1405
10.400.4.6 StructureSetDate	1405
10.400.4.7 StructureSetLabel	1405
10.400.4.8 StructureSetName	1405
10.400.4.9 StructureSetTime	1405
10.400.4.10 StudyInstanceUID	1405
10.401 gdcmm::Waveform Class Reference	1406
10.401.1 Detailed Description	1406
10.401.2 Constructor & Destructor Documentation	1406
10.401.2.1 Waveform()	1406
10.402 gdcmm::WLMFindQuery Class Reference	1406
10.402.1 Detailed Description	1409
10.402.2 Constructor & Destructor Documentation	1409
10.402.2.1 WLMFindQuery()	1409
10.402.3 Member Function Documentation	1409
10.402.3.1 GetAbstractSyntaxUID()	1409
10.402.3.2 GetTagListByLevel()	1410
10.402.3.3 GetValidDataSet()	1410
10.402.3.4 InitializeDataSet()	1410
10.402.3.5 ValidateQuery()	1410

10.402.4 Friends And Related Symbol Documentation . . . . .	1410
10.402.4.1 QueryFactory . . . . .	1410
10.403 gdcm::Writer Class Reference . . . . .	1411
10.403.1 Detailed Description . . . . .	1413
10.403.2 Constructor & Destructor Documentation . . . . .	1414
10.403.2.1 Writer() . . . . .	1414
10.403.2.2 ~Writer() . . . . .	1414
10.403.3 Member Function Documentation . . . . .	1414
10.403.3.1 CheckFileMetaInformationOff() . . . . .	1414
10.403.3.2 CheckFileMetaInformationOn() . . . . .	1414
10.403.3.3 GetCheckFileMetaInformation() . . . . .	1414
10.403.3.4 GetFile() . . . . .	1414
10.403.3.5 GetStreamPtr() . . . . .	1414
10.403.3.6 SetCheckFileMetaInformation() . . . . .	1415
10.403.3.7 SetFile() . . . . .	1415
10.403.3.8 SetFileName() . . . . .	1415
10.403.3.9 SetStream() . . . . .	1416
10.403.3.10 SetWriteDataSetOnly() . . . . .	1416
10.403.3.11 Write() . . . . .	1416
10.403.4 Friends And Related Symbol Documentation . . . . .	1416
10.403.4.1 StreamImageWriter . . . . .	1416
10.403.5 Member Data Documentation . . . . .	1416
10.403.5.1 Ofstream . . . . .	1416
10.403.5.2 Stream . . . . .	1417
10.404 gdcm::XMLDictReader Class Reference . . . . .	1417
10.404.1 Detailed Description . . . . .	1418
10.404.2 Constructor & Destructor Documentation . . . . .	1418
10.404.2.1 XMLDictReader() . . . . .	1418
10.404.2.2 ~XMLDictReader() . . . . .	1419
10.404.3 Member Function Documentation . . . . .	1419
10.404.3.1 CharacterDataHandler() . . . . .	1419
10.404.3.2 EndElement() . . . . .	1419
10.404.3.3 GetDict() . . . . .	1419
10.404.3.4 HandleDescription() . . . . .	1419
10.404.3.5 HandleEntry() . . . . .	1419
10.404.3.6 StartElement() . . . . .	1419
10.405 gdcm::XMLPrinter Class Reference . . . . .	1420
10.405.1 Member Enumeration Documentation . . . . .	1421
10.405.1.1 PrintStyles . . . . .	1421

10.405.2 Constructor & Destructor Documentation	1421
10.405.2.1 XMLPrinter()	1421
10.405.2.2 ~XMLPrinter()	1421
10.405.3 Member Function Documentation	1421
10.405.3.1 GetPrintStyle()	1421
10.405.3.2 HandleBulkData()	1421
10.405.3.3 Print()	1422
10.405.3.4 PrintDataElement()	1422
10.405.3.5 PrintDataSet()	1422
10.405.3.6 PrintSQ()	1422
10.405.3.7 SetFile()	1422
10.405.3.8 SetStyle()	1422
10.405.4 Member Data Documentation	1422
10.405.4.1 F	1422
10.405.4.2 PrintStyle	1423
10.406 gdcm::XMLPrivateDictReader Class Reference	1423
10.406.1 Detailed Description	1424
10.406.2 Constructor & Destructor Documentation	1424
10.406.2.1 XMLPrivateDictReader()	1424
10.406.2.2 ~XMLPrivateDictReader()	1425
10.406.3 Member Function Documentation	1425
10.406.3.1 CharacterDataHandler()	1425
10.406.3.2 EndElement()	1425
10.406.3.3 GetPrivateDict()	1425
10.406.3.4 HandleDescription()	1425
10.406.3.5 HandleEntry()	1425
10.406.3.6 StartElement()	1425
<b>11 File Documentation</b>	<b>1427</b>
11.1 README.txt File Reference	1427
11.2 TestsList.txt File Reference	1427
11.3 gdcmASN1.h File Reference	1427
11.4 gdcmASN1.h	1428
11.5 gdcmBase64.h File Reference	1429
11.6 gdcmBase64.h	1429
11.7 gdcmBoxRegion.h File Reference	1430
11.8 gdcmBoxRegion.h	1431
11.9 gdcmByteSwap.h File Reference	1431
11.10 gdcmByteSwap.h	1432

11.11 gdcmlCAPICryptoFactory.h File Reference . . . . .	1433
11.12 gdcmlCAPICryptoFactory.h . . . . .	1434
11.13 gdcmlCAPICryptographicMessageSyntax.h File Reference . . . . .	1434
11.14 gdcmlCAPICryptographicMessageSyntax.h . . . . .	1435
11.15 gdcmlCommand.h File Reference . . . . .	1436
11.16 gdcmlCommand.h . . . . .	1437
11.17 gdcmlCryptoFactory.h File Reference . . . . .	1439
11.18 gdcmlCryptoFactory.h . . . . .	1440
11.19 gdcmlCryptographicMessageSyntax.h File Reference . . . . .	1441
11.20 gdcmlCryptographicMessageSyntax.h . . . . .	1441
11.21 gdcmlDataEvent.h File Reference . . . . .	1443
11.22 gdcmlDataEvent.h . . . . .	1444
11.23 gdcmlDeflateStream.h File Reference . . . . .	1445
11.24 gdcmlDeflateStream.h . . . . .	1445
11.25 gdcmlDirectory.h File Reference . . . . .	1445
11.26 gdcmlDirectory.h . . . . .	1446
11.27 gdcmlDummyValueGenerator.h File Reference . . . . .	1448
11.28 gdcmlDummyValueGenerator.h . . . . .	1448
11.29 gdcmlEvent.h File Reference . . . . .	1449
11.29.1 Macro Definition Documentation . . . . .	1450
11.29.1.1 gdcmlEventMacro . . . . .	1450
11.30 gdcmlEvent.h . . . . .	1451
11.31 gdcmlException.h File Reference . . . . .	1452
11.32 gdcmlException.h . . . . .	1453
11.33 gdcmlFilename.h File Reference . . . . .	1454
11.34 gdcmlFilename.h . . . . .	1455
11.35 gdcmlFileNameEvent.h File Reference . . . . .	1455
11.36 gdcmlFileNameEvent.h . . . . .	1456
11.37 gdcmlFilenameGenerator.h File Reference . . . . .	1457
11.38 gdcmlFilenameGenerator.h . . . . .	1458
11.39 gdcmlLegacyMacro.h File Reference . . . . .	1458
11.39.1 Macro Definition Documentation . . . . .	1459
11.39.1.1 GDCM_LEGACY . . . . .	1459
11.39.1.2 GDCM_LEGACY_BODY . . . . .	1459
11.39.1.3 GDCM_LEGACY_REPLACED_BODY . . . . .	1460
11.39.1.4 GDCM_NOOP_STATEMENT . . . . .	1460
11.40 gdcmlLegacyMacro.h . . . . .	1460
11.41 gdcmlMD5.h File Reference . . . . .	1461
11.42 gdcmlMD5.h . . . . .	1462

---

11.43 gdcObject.h File Reference . . . . .	1462
11.44 gdcObject.h . . . . .	1463
11.45 gdcOpenSSLCryptoFactory.h File Reference . . . . .	1464
11.46 gdcOpenSSLCryptoFactory.h . . . . .	1465
11.47 gdcOpenSSLCryptographicMessageSyntax.h File Reference . . . . .	1466
11.48 gdcOpenSSLCryptographicMessageSyntax.h . . . . .	1467
11.49 gdcOpenSSLP7CryptoFactory.h File Reference . . . . .	1468
11.50 gdcOpenSSLP7CryptoFactory.h . . . . .	1469
11.51 gdcOpenSSLP7CryptographicMessageSyntax.h File Reference . . . . .	1469
11.52 gdcOpenSSLP7CryptographicMessageSyntax.h . . . . .	1471
11.53 gdcProgressEvent.h File Reference . . . . .	1471
11.54 gdcProgressEvent.h . . . . .	1472
11.55 gdcRegion.h File Reference . . . . .	1473
11.56 gdcRegion.h . . . . .	1474
11.57 gdcSHA1.h File Reference . . . . .	1475
11.58 gdcSHA1.h . . . . .	1476
11.59 gdcSmartPointer.h File Reference . . . . .	1476
11.60 gdcSmartPointer.h . . . . .	1477
11.61 gdcStaticAssert.h File Reference . . . . .	1478
11.61.1 Macro Definition Documentation . . . . .	1479
11.61.1.1 GDCM_DO_JOIN . . . . .	1479
11.61.1.2 GDCM_DO_JOIN2 . . . . .	1479
11.61.1.3 GDCM_JOIN . . . . .	1479
11.61.1.4 GDCM_STATIC_ASSERT . . . . .	1479
11.62 gdcStaticAssert.h . . . . .	1480
11.63 gdcString.h File Reference . . . . .	1480
11.64 gdcString.h . . . . .	1481
11.65 gdcSubject.h File Reference . . . . .	1483
11.66 gdcSubject.h . . . . .	1484
11.67 gdcSwapCode.h File Reference . . . . .	1485
11.68 gdcSwapCode.h . . . . .	1486
11.69 gdcSwapper.h File Reference . . . . .	1486
11.70 gdcSwapper.h . . . . .	1487
11.71 gdcSystem.h File Reference . . . . .	1489
11.72 gdcSystem.h . . . . .	1489
11.73 gdcTerminal.h File Reference . . . . .	1491
11.74 gdcTerminal.h . . . . .	1492
11.75 gdcTestDriver.h File Reference . . . . .	1493
11.76 gdcTestDriver.h . . . . .	1493

11.77 gdcTesting.h File Reference . . . . .	1494
11.78 gdcTesting.h . . . . .	1494
11.79 gdcTrace.h File Reference . . . . .	1495
11.79.1 Macro Definition Documentation . . . . .	1497
11.79.1.1 GDCM_FUNCTION . . . . .	1497
11.79.1.2 gdcAssertAlwaysMacro . . . . .	1497
11.79.1.3 gdcAssertMacro . . . . .	1497
11.79.1.4 gdcDebugMacro . . . . .	1497
11.79.1.5 gdcErrorMacro . . . . .	1498
11.79.1.6 gdcWarningMacro . . . . .	1498
11.80 gdcTrace.h . . . . .	1499
11.81 gdcTypes.h File Reference . . . . .	1501
11.82 gdcTypes.h . . . . .	1502
11.83 gdcUnpacker12Bits.h File Reference . . . . .	1502
11.84 gdcUnpacker12Bits.h . . . . .	1503
11.85 gdcVersion.h File Reference . . . . .	1503
11.86 gdcVersion.h . . . . .	1504
11.87 gdcWin32.h File Reference . . . . .	1504
11.87.1 Macro Definition Documentation . . . . .	1505
11.87.1.1 GDCM_EXPORT . . . . .	1505
11.88 gdcWin32.h . . . . .	1505
11.89 gdcCSAHeaderDict.h File Reference . . . . .	1506
11.90 gdcCSAHeaderDict.h . . . . .	1507
11.91 gdcCSAHeaderDictEntry.h File Reference . . . . .	1509
11.92 gdcCSAHeaderDictEntry.h . . . . .	1510
11.93 gdcDict.h File Reference . . . . .	1512
11.94 gdcDict.h . . . . .	1513
11.95 gdcDictConverter.h File Reference . . . . .	1517
11.96 gdcDictConverter.h . . . . .	1518
11.97 gdcDictEntry.h File Reference . . . . .	1519
11.98 gdcDictEntry.h . . . . .	1520
11.99 gdcDicts.h File Reference . . . . .	1521
11.100 gdcDicts.h . . . . .	1523
11.101 gdcGlobal.h File Reference . . . . .	1524
11.102 gdcGlobal.h . . . . .	1525
11.103 gdcGroupDict.h File Reference . . . . .	1526
11.104 gdcGroupDict.h . . . . .	1527
11.105 gdcSOPClassUIDToIOD.h File Reference . . . . .	1528
11.106 gdcSOPClassUIDToIOD.h . . . . .	1528

---

11.107 gdcmlUIDs.h File Reference . . . . .	1529
11.108 gdcmlUIDs.h . . . . .	1530
11.109 gdcmlAttribute.h File Reference . . . . .	1543
11.110 gdcmlAttribute.h . . . . .	1544
11.111 gdcmlBasicOffsetTable.h File Reference . . . . .	1557
11.112 gdcmlBasicOffsetTable.h . . . . .	1558
11.113 gdcmlByteBuffer.h File Reference . . . . .	1560
11.114 gdcmlByteBuffer.h . . . . .	1561
11.115 gdcmlByteSwapFilter.h File Reference . . . . .	1563
11.116 gdcmlByteSwapFilter.h . . . . .	1563
11.117 gdcmlByteValue.h File Reference . . . . .	1564
11.118 gdcmlByteValue.h . . . . .	1565
11.119 gdcmlCodeString.h File Reference . . . . .	1568
11.120 gdcmlCodeString.h . . . . .	1569
11.121 gdcmlCP246ExplicitDataElement.h File Reference . . . . .	1570
11.122 gdcmlCP246ExplicitDataElement.h . . . . .	1570
11.123 gdcmlCSAElement.h File Reference . . . . .	1571
11.124 gdcmlCSAElement.h . . . . .	1572
11.125 gdcmlCSAHeader.h File Reference . . . . .	1574
11.126 gdcmlCSAHeader.h . . . . .	1575
11.127 gdcmlDataElement.h File Reference . . . . .	1577
11.128 gdcmlDataElement.h . . . . .	1578
11.129 gdcmlDataSet.h File Reference . . . . .	1580
11.130 gdcmlDataSet.h . . . . .	1581
11.131 gdcmlDataSetEvent.h File Reference . . . . .	1585
11.132 gdcmlDataSetEvent.h . . . . .	1586
11.133 gdcmlElement.h File Reference . . . . .	1587
11.134 gdcmlElement.h . . . . .	1588
11.135 gdcmlExplicitDataElement.h File Reference . . . . .	1599
11.136 gdcmlExplicitDataElement.h . . . . .	1600
11.137 gdcmlExplicitImplicitDataElement.h File Reference . . . . .	1601
11.138 gdcmlExplicitImplicitDataElement.h . . . . .	1602
11.139 gdcmlFile.h File Reference . . . . .	1602
11.140 gdcmlFile.h . . . . .	1603
11.141 gdcmlFileMetaInformation.h File Reference . . . . .	1604
11.142 gdcmlFileMetaInformation.h . . . . .	1605
11.143 gdcmlFileSet.h File Reference . . . . .	1607
11.144 gdcmlFileSet.h . . . . .	1608
11.145 gdcmlFragment.h File Reference . . . . .	1609

11.146 gdcmlFragment.h . . . . .	1610
11.147 gdcmlImplicitDataElement.h File Reference . . . . .	1613
11.148 gdcmlImplicitDataElement.h . . . . .	1614
11.149 gdcmlItem.h File Reference . . . . .	1615
11.150 gdcmlItem.h . . . . .	1616
11.151 gdcmlLO.h File Reference . . . . .	1621
11.152 gdcmlLO.h . . . . .	1621
11.153 gdcmlMediaStorage.h File Reference . . . . .	1622
11.154 gdcmlMediaStorage.h . . . . .	1623
11.155 gdcmlMrProtocol.h File Reference . . . . .	1626
11.156 gdcmlMrProtocol.h . . . . .	1627
11.157 gdcmlParseException.h File Reference . . . . .	1628
11.158 gdcmlParseException.h . . . . .	1629
11.159 gdcmlParser.h File Reference . . . . .	1630
11.160 gdcmlParser.h . . . . .	1631
11.161 gdcmlPDBelement.h File Reference . . . . .	1633
11.162 gdcmlPDBelement.h . . . . .	1634
11.163 gdcmlPDBHeader.h File Reference . . . . .	1635
11.164 gdcmlPDBHeader.h . . . . .	1635
11.165 gdcmlPreamble.h File Reference . . . . .	1636
11.166 gdcmlPreamble.h . . . . .	1638
11.167 gdcmlPrivateTag.h File Reference . . . . .	1639
11.168 gdcmlPrivateTag.h . . . . .	1640
11.169 gdcmlReader.h File Reference . . . . .	1641
11.170 gdcmlReader.h . . . . .	1642
11.171 gdcmlSequenceOfFragments.h File Reference . . . . .	1643
11.172 gdcmlSequenceOfFragments.h . . . . .	1644
11.173 gdcmlSequenceOfItems.h File Reference . . . . .	1648
11.174 gdcmlSequenceOfItems.h . . . . .	1649
11.175 gdcmlTag.h File Reference . . . . .	1652
11.176 gdcmlTag.h . . . . .	1653
11.177 gdcmlTagToVR.h File Reference . . . . .	1657
11.178 gdcmlTagToVR.h . . . . .	1657
11.179 gdcmlTransferSyntax.h File Reference . . . . .	1658
11.180 gdcmlTransferSyntax.h . . . . .	1659
11.181 gdcmlUNExplicitDataElement.h File Reference . . . . .	1660
11.182 gdcmlUNExplicitDataElement.h . . . . .	1661
11.183 gdcmlUNExplicitImplicitDataElement.h File Reference . . . . .	1662
11.184 gdcmlUNExplicitImplicitDataElement.h . . . . .	1663



11.185 gdcmlValue.h File Reference . . . . .	1663
11.186 gdcmlValue.h . . . . .	1664
11.187 gdcmlValueIO.h File Reference . . . . .	1665
11.188 gdcmlValueIO.h . . . . .	1666
11.189 gdcmlVL.h File Reference . . . . .	1666
11.190 gdcmlVL.h . . . . .	1667
11.191 gdcmlVM.h File Reference . . . . .	1669
11.191.1 Macro Definition Documentation . . . . .	1670
11.191.1.1 TYPETOLENGTH . . . . .	1670
11.192 gdcmlVM.h . . . . .	1670
11.193 gdcmlVR.h File Reference . . . . .	1672
11.193.1 Macro Definition Documentation . . . . .	1673
11.193.1.1 TYPETOENCODING . . . . .	1673
11.193.1.2 VRTypeTemplateCase . . . . .	1674
11.194 gdcmlVR.h . . . . .	1674
11.195 gdcmlVR16ExplicitDataElement.h File Reference . . . . .	1678
11.196 gdcmlVR16ExplicitDataElement.h . . . . .	1679
11.197 gdcmlWriter.h File Reference . . . . .	1680
11.198 gdcmlWriter.h . . . . .	1681
11.199 gdcmlDefinedTerms.h File Reference . . . . .	1682
11.200 gdcmlDefinedTerms.h . . . . .	1683
11.201 gdcmlDefs.h File Reference . . . . .	1683
11.202 gdcmlDefs.h . . . . .	1684
11.203 gdcmlEnumeratedValues.h File Reference . . . . .	1685
11.204 gdcmlEnumeratedValues.h . . . . .	1686
11.205 gdcmlIOD.h File Reference . . . . .	1686
11.206 gdcmlIOD.h . . . . .	1688
11.207 gdcmlIODEntry.h File Reference . . . . .	1689
11.208 gdcmlIODEntry.h . . . . .	1691
11.209 gdcmlIODs.h File Reference . . . . .	1691
11.210 gdcmlIODs.h . . . . .	1693
11.211 gdcmlMacro.h File Reference . . . . .	1694
11.212 gdcmlMacro.h . . . . .	1695
11.213 gdcmlMacroEntry.h File Reference . . . . .	1697
11.213.1 Macro Definition Documentation . . . . .	1698
11.213.1.1 GDCMMACROENTRY_H . . . . .	1698
11.214 gdcmlMacroEntry.h . . . . .	1698
11.215 gdcmlMacros.h File Reference . . . . .	1700
11.216 gdcmlMacros.h . . . . .	1701

11.217 <a href="#">gdcmModule.h File Reference</a> . . . . .	1702
11.218 <a href="#">gdcmModule.h</a> . . . . .	1703
11.219 <a href="#">gdcmModuleEntry.h File Reference</a> . . . . .	1705
11.220 <a href="#">gdcmModuleEntry.h</a> . . . . .	1707
11.221 <a href="#">gdcmModules.h File Reference</a> . . . . .	1708
11.222 <a href="#">gdcmModules.h</a> . . . . .	1709
11.223 <a href="#">gdcmNestedModuleEntries.h File Reference</a> . . . . .	1710
11.224 <a href="#">gdcmNestedModuleEntries.h</a> . . . . .	1711
11.225 <a href="#">gdcmPatient.h File Reference</a> . . . . .	1712
11.226 <a href="#">gdcmPatient.h</a> . . . . .	1712
11.227 <a href="#">gdcmSeries.h File Reference</a> . . . . .	1713
11.228 <a href="#">gdcmSeries.h</a> . . . . .	1714
11.229 <a href="#">gdcmStudy.h File Reference</a> . . . . .	1715
11.230 <a href="#">gdcmStudy.h</a> . . . . .	1716
11.231 <a href="#">gdcmTable.h File Reference</a> . . . . .	1716
11.232 <a href="#">gdcmTable.h</a> . . . . .	1717
11.233 <a href="#">gdcmTableEntry.h File Reference</a> . . . . .	1718
11.234 <a href="#">gdcmTableEntry.h</a> . . . . .	1719
11.235 <a href="#">gdcmTableReader.h File Reference</a> . . . . .	1720
11.236 <a href="#">gdcmTableReader.h</a> . . . . .	1721
11.237 <a href="#">gdcmType.h File Reference</a> . . . . .	1722
11.238 <a href="#">gdcmType.h</a> . . . . .	1723
11.239 <a href="#">gdcmUsage.h File Reference</a> . . . . .	1724
11.240 <a href="#">gdcmUsage.h</a> . . . . .	1726
11.241 <a href="#">gdcmXMLDictReader.h File Reference</a> . . . . .	1727
11.242 <a href="#">gdcmXMLDictReader.h</a> . . . . .	1727
11.243 <a href="#">gdcmXMLPrivateDictReader.h File Reference</a> . . . . .	1728
11.244 <a href="#">gdcmXMLPrivateDictReader.h</a> . . . . .	1729
11.245 <a href="#">gdcmAnonymizeEvent.h File Reference</a> . . . . .	1729
11.246 <a href="#">gdcmAnonymizeEvent.h</a> . . . . .	1731
11.247 <a href="#">gdcmAnonymizer.h File Reference</a> . . . . .	1731
11.248 <a href="#">gdcmAnonymizer.h</a> . . . . .	1732
11.249 <a href="#">gdcmApplicationEntity.h File Reference</a> . . . . .	1733
11.250 <a href="#">gdcmApplicationEntity.h</a> . . . . .	1734
11.251 <a href="#">gdcmAudioCodec.h File Reference</a> . . . . .	1735
11.252 <a href="#">gdcmAudioCodec.h</a> . . . . .	1736
11.253 <a href="#">gdcmBitmap.h File Reference</a> . . . . .	1736
11.254 <a href="#">gdcmBitmap.h</a> . . . . .	1737
11.255 <a href="#">gdcmBitmapToBitmapFilter.h File Reference</a> . . . . .	1740

11.256 gdcmapBitmapToBitmapFilter.h . . . . .	1740
11.257 gdcmapCleaner.h File Reference . . . . .	1741
11.258 gdcmapCleaner.h . . . . .	1742
11.259 gdcmapCodec.h File Reference . . . . .	1743
11.260 gdcmapCodec.h . . . . .	1744
11.261 gdcmapCoder.h File Reference . . . . .	1744
11.262 gdcmapCoder.h . . . . .	1745
11.263 gdcmapConstCharWrapper.h File Reference . . . . .	1746
11.264 gdcmapConstCharWrapper.h . . . . .	1746
11.265 gdcmapCurve.h File Reference . . . . .	1747
11.266 gdcmapCurve.h . . . . .	1748
11.267 gdcmapDataSetHelper.h File Reference . . . . .	1749
11.268 gdcmapDataSetHelper.h . . . . .	1749
11.269 gdcmapDecoder.h File Reference . . . . .	1750
11.270 gdcmapDecoder.h . . . . .	1751
11.271 gdcmapDeltaEncodingCodec.h File Reference . . . . .	1752
11.272 gdcmapDeltaEncodingCodec.h . . . . .	1752
11.273 gdcmapDICOMDIR.h File Reference . . . . .	1753
11.274 gdcmapDICOMDIR.h . . . . .	1754
11.275 gdcmapDICOMDIRGenerator.h File Reference . . . . .	1754
11.276 gdcmapDICOMDIRGenerator.h . . . . .	1755
11.277 gdcmapDictPrinter.h File Reference . . . . .	1756
11.278 gdcmapDictPrinter.h . . . . .	1757
11.279 gdcmapDirectionCosines.h File Reference . . . . .	1757
11.280 gdcmapDirectionCosines.h . . . . .	1758
11.281 gdcmapDirectoryHelper.h File Reference . . . . .	1759
11.282 gdcmapDirectoryHelper.h . . . . .	1759
11.283 gdcmapDPath.h File Reference . . . . .	1760
11.284 gdcmapDPath.h . . . . .	1761
11.285 gdcmapDumper.h File Reference . . . . .	1762
11.286 gdcmapDumper.h . . . . .	1763
11.287 gdcmapEmptyMaskGenerator.h File Reference . . . . .	1763
11.288 gdcmapEmptyMaskGenerator.h . . . . .	1764
11.289 gdcmapEncapsulatedDocument.h File Reference . . . . .	1765
11.290 gdcmapEncapsulatedDocument.h . . . . .	1765
11.291 gdcmapEquipmentManufacturer.h File Reference . . . . .	1766
11.292 gdcmapEquipmentManufacturer.h . . . . .	1766
11.293 gdcmapFiducials.h File Reference . . . . .	1767
11.294 gdcmapFiducials.h . . . . .	1768

11.295 gdcmlFileAnonymizer.h File Reference	1768
11.296 gdcmlFileAnonymizer.h	1769
11.297 gdcmlFileChangeTransferSyntax.h File Reference	1770
11.298 gdcmlFileChangeTransferSyntax.h	1771
11.299 gdcmlFileDecompressLookupTable.h File Reference	1772
11.300 gdcmlFileDecompressLookupTable.h	1772
11.301 gdcmlFileDerivation.h File Reference	1773
11.302 gdcmlFileDerivation.h	1774
11.303 gdcmlFileExplicitFilter.h File Reference	1775
11.304 gdcmlFileExplicitFilter.h	1775
11.305 gdcmlFileStreamer.h File Reference	1776
11.306 gdcmlFileStreamer.h	1777
11.307 gdcmlIconImage.h File Reference	1778
11.308 gdcmlIconImage.h	1779
11.309 gdcmlIconImageFilter.h File Reference	1780
11.310 gdcmlIconImageFilter.h	1780
11.311 gdcmlIconImageGenerator.h File Reference	1781
11.312 gdcmlIconImageGenerator.h	1782
11.313 gdcmlImage.h File Reference	1783
11.314 gdcmlImage.h	1784
11.315 gdcmlImageApplyLookupTable.h File Reference	1785
11.316 gdcmlImageApplyLookupTable.h	1785
11.317 gdcmlImageChangePhotometricInterpretation.h File Reference	1786
11.318 gdcmlImageChangePhotometricInterpretation.h	1787
11.319 gdcmlImageChangePlanarConfiguration.h File Reference	1788
11.320 gdcmlImageChangePlanarConfiguration.h	1789
11.321 gdcmlImageChangeTransferSyntax.h File Reference	1790
11.322 gdcmlImageChangeTransferSyntax.h	1791
11.323 gdcmlImageCodec.h File Reference	1791
11.324 gdcmlImageCodec.h	1792
11.325 gdcmlImageConverter.h File Reference	1795
11.326 gdcmlImageConverter.h	1795
11.327 gdcmlImageFragmentSplitter.h File Reference	1796
11.328 gdcmlImageFragmentSplitter.h	1797
11.329 gdcmlImageHelper.h File Reference	1797
11.330 gdcmlImageHelper.h	1798
11.331 gdcmlImageReader.h File Reference	1800
11.332 gdcmlImageReader.h	1801
11.333 gdcmlImageRegionReader.h File Reference	1801

11.334 gdcmlImageRegionReader.h . . . . .	1802
11.335 gdcmlImageToImageFilter.h File Reference . . . . .	1803
11.336 gdcmlImageToImageFilter.h . . . . .	1804
11.337 gdcmlImageWriter.h File Reference . . . . .	1804
11.338 gdcmlImageWriter.h . . . . .	1805
11.339 gdcmlPPSorter.h File Reference . . . . .	1806
11.340 gdcmlPPSorter.h . . . . .	1807
11.341 gdcmlJPEG12Codec.h File Reference . . . . .	1808
11.342 gdcmlJPEG12Codec.h . . . . .	1808
11.343 gdcmlJPEG16Codec.h File Reference . . . . .	1809
11.344 gdcmlJPEG16Codec.h . . . . .	1810
11.345 gdcmlJPEG2000Codec.h File Reference . . . . .	1811
11.346 gdcmlJPEG2000Codec.h . . . . .	1811
11.347 gdcmlJPEG8Codec.h File Reference . . . . .	1813
11.348 gdcmlJPEG8Codec.h . . . . .	1813
11.349 gdcmlJPEGCodec.h File Reference . . . . .	1814
11.350 gdcmlJPEGCodec.h . . . . .	1815
11.351 gdcmlJPEGLSCodec.h File Reference . . . . .	1817
11.352 gdcmlJPEGLSCodec.h . . . . .	1817
11.353 gdcmlJSON.h File Reference . . . . .	1818
11.354 gdcmlJSON.h . . . . .	1819
11.355 gdcmlKAKADUCodec.h File Reference . . . . .	1820
11.356 gdcmlKAKADUCodec.h . . . . .	1821
11.357 gdcmlLookupTable.h File Reference . . . . .	1821
11.358 gdcmlLookupTable.h . . . . .	1822
11.359 gdcmlMEC_MR3.h File Reference . . . . .	1824
11.360 gdcmlMEC_MR3.h . . . . .	1824
11.361 gdcmlMeshPrimitive.h File Reference . . . . .	1825
11.362 gdcmlMeshPrimitive.h . . . . .	1826
11.363 gdcmlOrientation.h File Reference . . . . .	1828
11.364 gdcmlOrientation.h . . . . .	1828
11.365 gdcmlOverlay.h File Reference . . . . .	1829
11.366 gdcmlOverlay.h . . . . .	1830
11.367 gdcmlPDFCodec.h File Reference . . . . .	1832
11.368 gdcmlPDFCodec.h . . . . .	1832
11.369 gdcmlPersonName.h File Reference . . . . .	1833
11.370 gdcmlPersonName.h . . . . .	1834
11.371 gdcmlPGXCodec.h File Reference . . . . .	1835
11.372 gdcmlPGXCodec.h . . . . .	1835

11.373 <a href="#">gdcmPhotometricInterpretation.h File Reference</a>	1836
11.374 <a href="#">gdcmPhotometricInterpretation.h</a>	1837
11.375 <a href="#">gdcmPixelFormat.h File Reference</a>	1838
11.376 <a href="#">gdcmPixelFormat.h</a>	1840
11.377 <a href="#">gdcmPixmap.h File Reference</a>	1842
11.378 <a href="#">gdcmPixmap.h</a>	1843
11.379 <a href="#">gdcmPixmapReader.h File Reference</a>	1844
11.380 <a href="#">gdcmPixmapReader.h</a>	1846
11.381 <a href="#">gdcmPixmapToPixmapFilter.h File Reference</a>	1847
11.382 <a href="#">gdcmPixmapToPixmapFilter.h</a>	1847
11.383 <a href="#">gdcmPixmapWriter.h File Reference</a>	1848
11.384 <a href="#">gdcmPixmapWriter.h</a>	1849
11.385 <a href="#">gdcmPNMCodec.h File Reference</a>	1850
11.386 <a href="#">gdcmPNMCodec.h</a>	1851
11.387 <a href="#">gdcmPrinter.h File Reference</a>	1851
11.388 <a href="#">gdcmPrinter.h</a>	1853
11.389 <a href="#">gdcmPVRGCodec.h File Reference</a>	1854
11.390 <a href="#">gdcmPVRGCodec.h</a>	1855
11.391 <a href="#">gdcmRAWCodec.h File Reference</a>	1855
11.392 <a href="#">gdcmRAWCodec.h</a>	1856
11.393 <a href="#">gdcmRescaler.h File Reference</a>	1857
11.394 <a href="#">gdcmRescaler.h</a>	1857
11.395 <a href="#">gdcmRLECodec.h File Reference</a>	1859
11.396 <a href="#">gdcmRLECodec.h</a>	1859
11.397 <a href="#">gdcmScanner.h File Reference</a>	1860
11.398 <a href="#">gdcmScanner.h</a>	1861
11.399 <a href="#">gdcmScanner2.h File Reference</a>	1863
11.400 <a href="#">gdcmScanner2.h</a>	1864
11.401 <a href="#">gdcmSegment.h File Reference</a>	1866
11.402 <a href="#">gdcmSegment.h</a>	1868
11.403 <a href="#">gdcmSegmentedPaletteColorLookupTable.h File Reference</a>	1870
11.404 <a href="#">gdcmSegmentedPaletteColorLookupTable.h</a>	1870
11.405 <a href="#">gdcmSegmentHelper.h File Reference</a>	1871
11.406 <a href="#">gdcmSegmentHelper.h</a>	1872
11.407 <a href="#">gdcmSegmentReader.h File Reference</a>	1873
11.408 <a href="#">gdcmSegmentReader.h</a>	1875
11.409 <a href="#">gdcmSegmentWriter.h File Reference</a>	1875
11.410 <a href="#">gdcmSegmentWriter.h</a>	1877
11.411 <a href="#">gdcmSerieHelper.h File Reference</a>	1877

11.412 gdcSerieHelper.h . . . . .	1879
11.413 gdcSimpleSubjectWatcher.h File Reference . . . . .	1880
11.414 gdcSimpleSubjectWatcher.h . . . . .	1881
11.415 gdcSorter.h File Reference . . . . .	1882
11.416 gdcSorter.h . . . . .	1884
11.417 gdcSpacing.h File Reference . . . . .	1885
11.418 gdcSpacing.h . . . . .	1885
11.419 gdcSpectroscopy.h File Reference . . . . .	1886
11.420 gdcSpectroscopy.h . . . . .	1887
11.421 gdcSplitMosaicFilter.h File Reference . . . . .	1887
11.422 gdcSplitMosaicFilter.h . . . . .	1888
11.423 gdcStreamImageReader.h File Reference . . . . .	1889
11.424 gdcStreamImageReader.h . . . . .	1890
11.425 gdcStreamImageWriter.h File Reference . . . . .	1891
11.426 gdcStreamImageWriter.h . . . . .	1892
11.427 gdcStrictScanner.h File Reference . . . . .	1893
11.428 gdcStrictScanner.h . . . . .	1894
11.429 gdcStrictScanner2.h File Reference . . . . .	1895
11.430 gdcStrictScanner2.h . . . . .	1896
11.431 gdcStringFilter.h File Reference . . . . .	1898
11.432 gdcStringFilter.h . . . . .	1899
11.433 gdcSurface.h File Reference . . . . .	1900
11.434 gdcSurface.h . . . . .	1901
11.435 gdcSurfaceHelper.h File Reference . . . . .	1904
11.436 gdcSurfaceHelper.h . . . . .	1905
11.437 gdcSurfaceReader.h File Reference . . . . .	1907
11.438 gdcSurfaceReader.h . . . . .	1908
11.439 gdcSurfaceWriter.h File Reference . . . . .	1909
11.440 gdcSurfaceWriter.h . . . . .	1910
11.441 gdcTagPath.h File Reference . . . . .	1910
11.442 gdcTagPath.h . . . . .	1911
11.443 gdcUIDGenerator.h File Reference . . . . .	1912
11.444 gdcUIDGenerator.h . . . . .	1913
11.445 gdcUUIDGenerator.h File Reference . . . . .	1914
11.446 gdcUUIDGenerator.h . . . . .	1914
11.447 gdcValidate.h File Reference . . . . .	1915
11.448 gdcValidate.h . . . . .	1915
11.449 gdcWaveform.h File Reference . . . . .	1916
11.450 gdcWaveform.h . . . . .	1917

11.451 gdcXMLPrinter.h File Reference . . . . .	1917
11.452 gdcXMLPrinter.h . . . . .	1918
11.453 gdcMAbortPDU.h File Reference . . . . .	1920
11.454 gdcMAbortPDU.h . . . . .	1920
11.455 gdcMAAssociateACPDU.h File Reference . . . . .	1921
11.456 gdcMAAssociateACPDU.h . . . . .	1922
11.457 gdcMAAssociateRJPDU.h File Reference . . . . .	1923
11.458 gdcMAAssociateRJPDU.h . . . . .	1924
11.459 gdcMAAssociateRQPDU.h File Reference . . . . .	1925
11.460 gdcMAAssociateRQPDU.h . . . . .	1925
11.461 gdcMAbstractSyntax.h File Reference . . . . .	1927
11.462 gdcMAbstractSyntax.h . . . . .	1928
11.463 gdcMApplicationContext.h File Reference . . . . .	1929
11.464 gdcMApplicationContext.h . . . . .	1930
11.465 gdcMAReleaseRPPDU.h File Reference . . . . .	1931
11.466 gdcMAReleaseRPPDU.h . . . . .	1932
11.467 gdcMAReleaseRQPDU.h File Reference . . . . .	1932
11.468 gdcMAReleaseRQPDU.h . . . . .	1933
11.469 gdcMARTIMTimer.h File Reference . . . . .	1934
11.470 gdcMARTIMTimer.h . . . . .	1935
11.471 gdcMAynchronousOperationsWindowSub.h File Reference . . . . .	1936
11.472 gdcMAynchronousOperationsWindowSub.h . . . . .	1936
11.473 gdcMBaseCompositeMessage.h File Reference . . . . .	1937
11.474 gdcMBaseCompositeMessage.h . . . . .	1938
11.475 gdcMBaseNormalizedMessage.h File Reference . . . . .	1939
11.476 gdcMBaseNormalizedMessage.h . . . . .	1939
11.477 gdcMBasePDU.h File Reference . . . . .	1940
11.478 gdcMBasePDU.h . . . . .	1941
11.479 gdcMBaseQuery.h File Reference . . . . .	1942
11.480 gdcMBaseQuery.h . . . . .	1943
11.481 gdcMBaseRootQuery.h File Reference . . . . .	1944
11.482 gdcMBaseRootQuery.h . . . . .	1945
11.483 gdcMCEchoMessages.h File Reference . . . . .	1946
11.484 gdcMCEchoMessages.h . . . . .	1947
11.485 gdcMCFindMessages.h File Reference . . . . .	1947
11.486 gdcMCFindMessages.h . . . . .	1948
11.487 gdcMCMoveMessages.h File Reference . . . . .	1949
11.488 gdcMCMoveMessages.h . . . . .	1950
11.489 gdcMCommandDataSet.h File Reference . . . . .	1950



11.490 gdcCommandDataSet.h . . . . .	1951
11.491 gdcCompositeMessageFactory.h File Reference . . . . .	1952
11.492 gdcCompositeMessageFactory.h . . . . .	1953
11.493 gdcCompositeNetworkFunctions.h File Reference . . . . .	1953
11.494 gdcCompositeNetworkFunctions.h . . . . .	1954
11.495 gdcCStoreMessages.h File Reference . . . . .	1955
11.496 gdcCStoreMessages.h . . . . .	1956
11.497 gdcDIMSE.h File Reference . . . . .	1957
11.498 gdcDIMSE.h . . . . .	1957
11.499 gdcFindPatientRootQuery.h File Reference . . . . .	1959
11.500 gdcFindPatientRootQuery.h . . . . .	1960
11.501 gdcFindStudyRootQuery.h File Reference . . . . .	1961
11.502 gdcFindStudyRootQuery.h . . . . .	1961
11.503 gdcImplementationClassUIDSub.h File Reference . . . . .	1962
11.504 gdcImplementationClassUIDSub.h . . . . .	1963
11.505 gdcImplementationUIDSub.h File Reference . . . . .	1964
11.506 gdcImplementationUIDSub.h . . . . .	1964
11.507 gdcImplementationVersionNameSub.h File Reference . . . . .	1965
11.508 gdcImplementationVersionNameSub.h . . . . .	1966
11.509 gdcMaximumLengthSub.h File Reference . . . . .	1967
11.510 gdcMaximumLengthSub.h . . . . .	1968
11.511 gdcModalityPerformedProcedureStepCreateQuery.h File Reference . . . . .	1969
11.512 gdcModalityPerformedProcedureStepCreateQuery.h . . . . .	1969
11.513 gdcModalityPerformedProcedureStepSetQuery.h File Reference . . . . .	1970
11.514 gdcModalityPerformedProcedureStepSetQuery.h . . . . .	1971
11.515 gdcMovePatientRootQuery.h File Reference . . . . .	1971
11.516 gdcMovePatientRootQuery.h . . . . .	1972
11.517 gdcMoveStudyRootQuery.h File Reference . . . . .	1973
11.518 gdcMoveStudyRootQuery.h . . . . .	1973
11.519 gdcNActionMessages.h File Reference . . . . .	1974
11.520 gdcNActionMessages.h . . . . .	1975
11.521 gdcNCreateMessages.h File Reference . . . . .	1975
11.522 gdcNCreateMessages.h . . . . .	1976
11.523 gdcNDeleteMessages.h File Reference . . . . .	1977
11.524 gdcNDeleteMessages.h . . . . .	1977
11.525 gdcNetworkEvents.h File Reference . . . . .	1978
11.526 gdcNetworkEvents.h . . . . .	1979
11.527 gdcNetworkStateID.h File Reference . . . . .	1980
11.528 gdcNetworkStateID.h . . . . .	1981

11.529 gdcmlNEventReportMessages.h File Reference	1982
11.530 gdcmlNEventReportMessages.h	1983
11.531 gdcmlNGetMessages.h File Reference	1983
11.532 gdcmlNGetMessages.h	1984
11.533 gdcmlNormalizedMessageFactory.h File Reference	1984
11.534 gdcmlNormalizedMessageFactory.h	1985
11.535 gdcmlNormalizedNetworkFunctions.h File Reference	1986
11.536 gdcmlNormalizedNetworkFunctions.h	1987
11.537 gdcmlNSetMessages.h File Reference	1988
11.538 gdcmlNSetMessages.h	1988
11.539 gdcmlPDataTFPDU.h File Reference	1989
11.540 gdcmlPDataTFPDU.h	1990
11.541 gdcmlPDUFactory.h File Reference	1991
11.542 gdcmlPDUFactory.h	1991
11.543 gdcmlPresentationContext.h File Reference	1992
11.544 gdcmlPresentationContext.h	1993
11.545 gdcmlPresentationContextAC.h File Reference	1994
11.546 gdcmlPresentationContextAC.h	1996
11.547 gdcmlPresentationContextGenerator.h File Reference	1996
11.548 gdcmlPresentationContextGenerator.h	1997
11.549 gdcmlPresentationContextRQ.h File Reference	1998
11.550 gdcmlPresentationContextRQ.h	1999
11.551 gdcmlPresentationDataValue.h File Reference	2000
11.552 gdcmlPresentationDataValue.h	2001
11.553 gdcmlQueryBase.h File Reference	2002
11.554 gdcmlQueryBase.h	2004
11.555 gdcmlQueryFactory.h File Reference	2005
11.556 gdcmlQueryFactory.h	2006
11.557 gdcmlQueryImage.h File Reference	2006
11.558 gdcmlQueryImage.h	2007
11.559 gdcmlQueryPatient.h File Reference	2008
11.560 gdcmlQueryPatient.h	2009
11.561 gdcmlQuerySeries.h File Reference	2010
11.562 gdcmlQuerySeries.h	2010
11.563 gdcmlQueryStudy.h File Reference	2011
11.564 gdcmlQueryStudy.h	2012
11.565 gdcmlRoleSelectionSub.h File Reference	2013
11.566 gdcmlRoleSelectionSub.h	2013
11.567 gdcmlServiceClassApplicationInformation.h File Reference	2014

11.568 gdcmserviceClassApplicationInformation.h . . . . .	2015
11.569 gdcmserviceClassUser.h File Reference . . . . .	2016
11.570 gdcmserviceClassUser.h . . . . .	2017
11.571 gdcmSOPClassExtendedNegociationSub.h File Reference . . . . .	2018
11.572 gdcmSOPClassExtendedNegociationSub.h . . . . .	2019
11.573 gdcmTransferSyntaxSub.h File Reference . . . . .	2019
11.574 gdcmTransferSyntaxSub.h . . . . .	2021
11.575 gdcmULAction.h File Reference . . . . .	2021
11.576 gdcmULAction.h . . . . .	2022
11.577 gdcmULActionAA.h File Reference . . . . .	2023
11.578 gdcmULActionAA.h . . . . .	2024
11.579 gdcmULActionAE.h File Reference . . . . .	2025
11.580 gdcmULActionAE.h . . . . .	2026
11.581 gdcmULActionAR.h File Reference . . . . .	2027
11.582 gdcmULActionAR.h . . . . .	2028
11.583 gdcmULActionDT.h File Reference . . . . .	2030
11.584 gdcmULActionDT.h . . . . .	2030
11.585 gdcmULBasicCallback.h File Reference . . . . .	2031
11.586 gdcmULBasicCallback.h . . . . .	2032
11.587 gdcmULConnection.h File Reference . . . . .	2032
11.588 gdcmULConnection.h . . . . .	2033
11.589 gdcmULConnectionCallback.h File Reference . . . . .	2035
11.590 gdcmULConnectionCallback.h . . . . .	2036
11.591 gdcmULConnectionInfo.h File Reference . . . . .	2036
11.592 gdcmULConnectionInfo.h . . . . .	2038
11.593 gdcmULConnectionManager.h File Reference . . . . .	2038
11.594 gdcmULConnectionManager.h . . . . .	2039
11.595 gdcmULEvent.h File Reference . . . . .	2041
11.596 gdcmULEvent.h . . . . .	2042
11.597 gdcmULTransitionTable.h File Reference . . . . .	2043
11.598 gdcmULTransitionTable.h . . . . .	2044
11.599 gdcmULWritingCallback.h File Reference . . . . .	2046
11.600 gdcmULWritingCallback.h . . . . .	2046
11.601 gdcmUserInformation.h File Reference . . . . .	2047
11.602 gdcmUserInformation.h . . . . .	2048
11.603 gdcmWLMFindQuery.h File Reference . . . . .	2049
11.604 gdcmWLMFindQuery.h . . . . .	2050
11.605 vtkGDCMImageReader.h File Reference . . . . .	2050
11.605.1 Macro Definition Documentation . . . . .	2051

11.605.1.1 VTK_CMYK . . . . .	2051
11.605.1.2 VTK_INVERSE_LUMINANCE . . . . .	2052
11.605.1.3 VTK_LOOKUP_TABLE . . . . .	2052
11.605.1.4 VTK_YBR . . . . .	2052
11.606 vtkGDCMImageReader.h . . . . .	2052
11.607 vtkGDCMImageReader2.h File Reference . . . . .	2056
11.607.1 Macro Definition Documentation . . . . .	2057
11.607.1.1 VTK_CMYK . . . . .	2057
11.607.1.2 VTK_INVERSE_LUMINANCE . . . . .	2057
11.607.1.3 VTK_LOOKUP_TABLE . . . . .	2057
11.607.1.4 VTK_YBR . . . . .	2057
11.608 vtkGDCMImageReader2.h . . . . .	2057
11.609 vtkGDCMImageWriter.h File Reference . . . . .	2061
11.610 vtkGDCMImageWriter.h . . . . .	2061
11.611 vtkGDCMMedicalImageProperties.h File Reference . . . . .	2064
11.612 vtkGDCMMedicalImageProperties.h . . . . .	2064
11.613 vtkGDCMPolyDataReader.h File Reference . . . . .	2069
11.614 vtkGDCMPolyDataReader.h . . . . .	2070
11.615 vtkGDCMPolyDataWriter.h File Reference . . . . .	2071
11.616 vtkGDCMPolyDataWriter.h . . . . .	2072
11.617 vtkGDCMTesting.h File Reference . . . . .	2073
11.618 vtkGDCMTesting.h . . . . .	2073
11.619 vtkGDCMThreadedImageReader.h File Reference . . . . .	2074
11.620 vtkGDCMThreadedImageReader.h . . . . .	2075
11.621 vtkGDCMThreadedImageReader2.h File Reference . . . . .	2076
11.622 vtkGDCMThreadedImageReader2.h . . . . .	2076
11.623 vtkImageColorViewer.h File Reference . . . . .	2078
11.624 vtkImageColorViewer.h . . . . .	2079
11.625 vtkImageMapToColors16.h File Reference . . . . .	2082
11.626 vtkImageMapToColors16.h . . . . .	2083
11.627 vtkImageMapToWindowLevelColors2.h File Reference . . . . .	2084
11.628 vtkImageMapToWindowLevelColors2.h . . . . .	2085
11.629 vtkImagePlanarComponentsToComponents.h File Reference . . . . .	2086
11.630 vtkImagePlanarComponentsToComponents.h . . . . .	2086
11.631 vtkImageRGBToYBR.h File Reference . . . . .	2088
11.632 vtkImageRGBToYBR.h . . . . .	2088
11.633 vtkImageYBRToRGB.h File Reference . . . . .	2089
11.634 vtkImageYBRToRGB.h . . . . .	2090
11.635 vtkLookupTable16.h File Reference . . . . .	2090

11.636 vtkLookupTable16.h . . . . .	2091
11.637 vtkRTStructSetProperties.h File Reference . . . . .	2092
11.638 vtkRTStructSetProperties.h . . . . .	2093
11.639 gdcmPythonFilter.h File Reference . . . . .	2094
11.640 gdcmPythonFilter.h . . . . .	2095
<b>12 Examples</b>	<b>2097</b>
12.1 TestByteSwap.cxx . . . . .	2097
12.2 PatchFile.cxx . . . . .	2099
12.3 SimplePrint.cs . . . . .	2100
12.4 TestReader.cxx . . . . .	2101
12.5 TestReader.py . . . . .	2103
12.6 DecompressJPEGFile.cs . . . . .	2103
12.7 ManipulateFile.cs . . . . .	2104
12.8 ClinicalTrialIdentificationWorkflow.cs . . . . .	2105
12.9 GenerateDICOMDIR.cs . . . . .	2108
12.10 GenFakelImage.cxx . . . . .	2109
12.11 ReformatFile.cs . . . . .	2111
12.12 DecompressImage.cs . . . . .	2112
12.13 StandardizeFiles.cs . . . . .	2113
12.14 ScanDirectory.cs . . . . .	2115
12.15 BasicAnonymizer.cs . . . . .	2116
12.16 BasicImageAnonymizer.cs . . . . .	2117
12.17 Cleaner.cs . . . . .	2118
12.18 CompressLossyJPEG.cs . . . . .	2120
12.19 DecompressImageMultiframe.cs . . . . .	2121
12.20 DumpCSA.cs . . . . .	2123
12.21 ExtractEncapsulatedFile.cs . . . . .	2124
12.22 ExtractImageRegion.cs . . . . .	2125
12.23 ExtractImageRegionWithLUT.cs . . . . .	2126
12.24 ExtractOneFrame.cs . . . . .	2128
12.25 FileAnonymize.cs . . . . .	2129
12.26 FileChangeTS.cs . . . . .	2130
12.27 FileChangeTSLossy.cs . . . . .	2132
12.28 FileStreaming.cs . . . . .	2134
12.29 GetArray.cs . . . . .	2135
12.30 MpegVideoInfo.cs . . . . .	2136
12.31 NewSequence.cs . . . . .	2141
12.32 RescaleImage.cs . . . . .	2142

12.33 SendFileSCU.cs	2143
12.34 SimplePrintPatientName.cs	2143
12.35 SortImage2.cs	2144
12.36 CStoreQtProgress.cxx	2145
12.37 ChangePrivateTags.cxx	2147
12.38 ChangeSequenceUltrasound.cxx	2148
12.39 CheckBigEndianBug.cxx	2149
12.40 ClinicalTrialAnnotate.cxx	2151
12.41 CompressImage.cxx	2152
12.42 ConvertToQImage.cxx	2153
12.43 CreateARGBImage.cxx	2155
12.44 CreateCMYKImage.cxx	2156
12.45 CreateJPIPDataSet.cxx	2157
12.46 DeriveSeries.cxx	2158
12.47 DiffFile.cxx	2159
12.48 DiscriminateVolume.cxx	2160
12.49 DumpADAC.cxx	2164
12.50 DumpExamCard.cxx	2168
12.51 DumpGEMSMovieGroup.cxx	2177
12.52 DumpImageHeaderInfo.cxx	2183
12.53 DumpPhilipsECHO.cxx	2185
12.54 DumpSiemensBase64.cxx	2190
12.55 DumpToSQLITE3.cxx	2191
12.56 DumpToshibaDTI.cxx	2193
12.57 DumpToshibaDTI2.cxx	2195
12.58 DumpVisusChange.cxx	2196
12.59 DuplicatePCDE.cxx	2198
12.60 ELSCINT1WaveToText.cxx	2201
12.61 EmptyMask.cxx	2203
12.62 EncapsulateFileInRawData.cxx	2203
12.63 ExtractEncryptedContent.cxx	2204
12.64 ExtractIconFromFile.cxx	2205
12.65 Extracting_All_Resolution.cxx	2207
12.66 Fake_Image_Using_Stream_Image_Writer.cxx	2212
12.67 FixBrokenJ2K.cxx	2215
12.68 FixJAIBugJPEGLS.cxx	2217
12.69 FixOrientation.cxx	2220
12.70 GenAllVR.cxx	2221
12.71 GenFakeIdentifyFile.cxx	2223

12.72 GenLongSeqs.cxx	2226
12.73 GenSeqs.cxx	2227
12.74 GenerateStandardSOPClasses.cxx	2228
12.75 GetJPEGSamplePrecision.cxx	2229
12.76 GetSequenceUltrasound.cxx	2231
12.77 GetSubSequenceData.cxx	2233
12.78 HelloVizWorld.cxx	2235
12.79 HelloWorld.cxx	2236
12.80 LargeVRDSExplicit.cxx	2237
12.81 MakeTemplate.cxx	2239
12.82 MergeTwoFiles.cxx	2240
12.83 MrProtocol.cxx	2242
12.84 PrintLUT.cxx	2248
12.85 PublicDict.cxx	2249
12.86 QIDO-RS.cxx	2250
12.87 ReadAndDumpDICOMDIR.cxx	2251
12.88 ReadAndDumpDICOMDIR2.cxx	2253
12.89 ReadAndPrintAttributes.cxx	2258
12.90 ReadExplicitLengthSQIVR.cxx	2260
12.91 ReadGEMSSDO.cxx	2261
12.92 ReadMultiTimesException.cxx	2263
12.93 ReadUTF8QtDir.cxx	2264
12.94 SimpleScanner.cxx	2265
12.95 SortImage.cxx	2267
12.96 StreamImageReaderTest.cxx	2268
12.97 TemplateEmptyImage.cxx	2272
12.98 TraverseModules.cxx	2273
12.99 VolumeSorter.cxx	2274
12.100 csa2img.cxx	2277
12.101 iU22tomultisc.cxx	2279
12.102 pmsct_rgb1.cxx	2280
12.103 rle2img.cxx	2284
12.104 uid_unique.cxx	2286
12.105 DecompressImage.java	2287
12.106 DecompressPixmap.java	2288
12.107 ExtractImageRegion.java	2289
12.108 FileAnonymize.java	2290
12.109 HelloSimple.java	2291
12.110 ReadFiles.java	2291

12.111 ScanDirectory.java . . . . .	2293
12.112 SimplePrint.java . . . . .	2296
12.113 AddPrivateAttribute.py . . . . .	2297
12.114 ConvertMPL.py . . . . .	2298
12.115 ConvertNumpy.py . . . . .	2299
12.116 ConvertPIL.py . . . . .	2300
12.117 CreateRAWStorage.py . . . . .	2301
12.118 DecompressImage.py . . . . .	2303
12.119 DumbAnonymizer.py . . . . .	2304
12.120 ExtractImageRegion.py . . . . .	2305
12.121 FindAllPatientName.py . . . . .	2306
12.122 FixCommaBug.py . . . . .	2307
12.123 GetPortionCSAHeader.py . . . . .	2308
12.124 HelloWorld.py . . . . .	2308
12.125 ManipulateFile.py . . . . .	2309
12.126 ManipulateSequence.py . . . . .	2310
12.127 MergeFile.py . . . . .	2311
12.128 NewSequence.py . . . . .	2312
12.129 PhilipsPrivateRescaleInterceptSlope.py . . . . .	2313
12.130 PlaySound.py . . . . .	2313
12.131 PrivateDict.py . . . . .	2314
12.132 ReWriteSCAsMR.py . . . . .	2315
12.133 ReadAndDumpDICOMDIR.py . . . . .	2316
12.134 RemovePrivateTags.py . . . . .	2318
12.135 ScanDirectory.py . . . . .	2318
12.136 SortImage.py . . . . .	2319
12.137 WriteBuffer.py . . . . .	2320
12.138 HelloActiviz.cs . . . . .	2321
12.139 HelloActiviz2.cs . . . . .	2322
12.140 HelloActiviz3.cs . . . . .	2323
12.141 HelloActiviz4.cs . . . . .	2324
12.142 HelloActiviz5.cs . . . . .	2325
12.143 HelloVTKWorld.cs . . . . .	2326
12.144 HelloVTKWorld2.cs . . . . .	2327
12.145 MetalImageMD5Activiz.cs . . . . .	2327
12.146 RefCounting.cs . . . . .	2329
12.147 Compute3DSpacing.cxx . . . . .	2330
12.148 Convert16BitsTo8Bits.cxx . . . . .	2331
12.149 ConvertMultiFrameToSingleFrame.cxx . . . . .	2332



12.150 ConvertRGBToLuminance.cxx . . . . .	2333
12.151 ConvertSingleBitTo8Bits.cxx . . . . .	2334
12.152 CreateFakePET.cxx . . . . .	2335
12.153 CreateFakeRTDOSE.cxx . . . . .	2337
12.154 GenerateRTSTRUCT.cxx . . . . .	2338
12.155 MagnifyFile.cxx . . . . .	2341
12.156 gdcmmorthoplanes.cxx . . . . .	2342
12.157 gdcmmreslice.cxx . . . . .	2348
12.158 gdcmmrtionplan.cxx . . . . .	2350
12.159 gdcmmrtplan.cxx . . . . .	2354
12.160 gdcmmscene.cxx . . . . .	2358
12.161 gdcmmtexture.cxx . . . . .	2360
12.162 gdcmmvolume.cxx . . . . .	2362
12.163 offscreenimage.cxx . . . . .	2363
12.164 reslicesphere.cxx . . . . .	2364
12.165 rtstructapp.cxx . . . . .	2372
12.166 threadgdcmm.cxx . . . . .	2374
12.167 AWTMedical3.java . . . . .	2377
12.168 HelloVTKWorld.java . . . . .	2381
12.169 MIPViewer.java . . . . .	2383
12.170 MPRViewer.java . . . . .	2385
12.171 MPRViewer2.java . . . . .	2387
12.172 ReadSeriesIntoVTK.java . . . . .	2391
12.173 CastConvertPhilips.py . . . . .	2393
12.174 headsq2dcm.py . . . . .	2395

**Index****2397**



# Chapter 1

## GDCM Documentation

This is the developers documentation.

A PDF version of this doxygen documentation can be found here:

`http://gdcm.sourceforge.net/3.0/gdcm-3.0.24.pdf`

A tarball version of this HTML doxygen documentation can be found here:

`http://gdcm.sourceforge.net/3.0/gdcm-3.0.24-doc.tar.gz`

**Author**

Mathieu Malaterre



# Chapter 2

## Todo List

### Class `gdcm::CSAHeader`

MrEvaProtocol in 29,1020 contains ^M that would be nice to get rid of on UNIX system...

### Class `gdcm::network::ApplicationContext`

Looks like Application Context can only be 64 bytes at max (see Figure 9-1 / PS 3.8 - 2009 )

### Class `gdcm::Overlay`

Is there actually any way to recognize an overlay ? On images with multiple overlay I do not see any way to differentiate them (other than the group tag).

### Class `gdcm::SequenceOfFragments`

I do not enforce that Sequence of Fragments ends with a SQ end del

### Class `gdcm::TransferSyntax`

: The implementation is completely retarded -> see `gdcm::UIDs` for a replacement We need: IsSupported We need preprocess of raw/xml file We need GetFullName()

### Member `gdcm::UIDGenerator::IsValid` (`const char *uid`)

: Move that in DataStructureAndEncoding (see `FileMetaInformation::CheckFileMetaInformation`)



## Chapter 3

# Deprecated List

Member [gdcm::CompositeNetworkFunctions::ConstructQuery](#) (ERootType inRootType, EQueryLevel inQueryLevel, const KeyValuePairArrayType &keys, EQueryType queryType=eFind)

Member [gdcm::FileSet::AddFile](#) (File const &)

. Does nothing

Member [gdcm::TransferSyntax::GetSwapCode](#) () const

Return the [SwapCode](#) associated with the Transfer Syntax. Be careful with the special GE private syntax the [DataSet](#) is written in little endian but the Pixel Data is in Big Endian.





## Chapter 4

# Bug List

### Class `gdcm::DICOMDIRGenerator`

: There is a current limitation of not handling Referenced SOP Class UID / Referenced SOP Instance UID simply because the `Scanner` does not allow us See PS 3.11 / [Table D.3-2 STD-GEN Additional DICOMDIR Keys](#)

### Member `gdcm::FileStreamer::StartGroupDataElement` (`const PrivateTag &pt, size_t maxsize=0, uint8_t startoffset=0`)

`maxsize` should be a value lower than the actual total size of the buffer to be copied

### Class `gdcm::IPPSorter`

There are currently a couple of bugs in this implementation:



## Chapter 5

# Namespace Index

### 5.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">gdc</a>	43
<a href="#">gdc::network</a>	75
<a href="#">gdc::SegmentHelper</a>	81
<a href="#">gdc::terminal</a>	
Class for Terminal	81



## Chapter 6

# Hierarchical Index

### 6.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

gdcmm::network::AbstractSyntax . . . . .	100
gdcmm::network::ApplicationContext . . . . .	116
gdcmm::ApplicationEntity . . . . .	118
gdcmm::network::ARTIMTimer . . . . .	125
gdcmm::ASN1 . . . . .	126
gdcmm::network::AsynchronousOperationsWindowSub . . . . .	128
gdcmm::Attribute< Group, Element, TVR, TVM > . . . . .	130
gdcmm::Attribute< Group, Element, TVR, VM::VM1 > . . . . .	138
gdcmm::Attribute< Group, Element, TVR, VM::VM1_n > . . . . .	148
gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 > . . . . .	144
gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 > . . . . .	146
gdcmm::Attribute< Group, Element, TVR, VM::VM2_n > . . . . .	156
gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n > . . . . .	153
gdcmm::Attribute< Group, Element, TVR, VM::VM3_n > . . . . .	161
gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n > . . . . .	158
gdcmm::Base64 . . . . .	166
gdcmm::network::BaseCompositeMessage . . . . .	169
gdcmm::network::CEchoRQ . . . . .	231
gdcmm::network::CEchoRSP . . . . .	233
gdcmm::network::CFindCancelRQ . . . . .	234
gdcmm::network::CFindRQ . . . . .	236
gdcmm::network::CFindRSP . . . . .	237
gdcmm::network::CMoveCancelRq . . . . .	246
gdcmm::network::CMoveRQ . . . . .	247
gdcmm::network::CMoveRSP . . . . .	248
gdcmm::network::CStoreRQ . . . . .	300
gdcmm::network::CStoreRSP . . . . .	301
gdcmm::network::BaseNormalizedMessage . . . . .	170
gdcmm::network::NActionRQ . . . . .	737
gdcmm::network::NActionRSP . . . . .	738

gdcmm::network::NCreateRQ . . . . .	740
gdcmm::network::NCreateRSP . . . . .	741
gdcmm::network::NDeleteRQ . . . . .	743
gdcmm::network::NDeleteRSP . . . . .	744
gdcmm::network::NEventReportRQ . . . . .	749
gdcmm::network::NEventReportRSP . . . . .	751
gdcmm::network::NGetRQ . . . . .	752
gdcmm::network::NGetRSP . . . . .	754
gdcmm::network::NSetRQ . . . . .	760
gdcmm::network::NSetRSP . . . . .	761
gdcmm::network::BasePDU . . . . .	173
gdcmm::network::AAabortPDU . . . . .	85
gdcmm::network::AAAssociateACPDU . . . . .	88
gdcmm::network::AAAssociateRJPDU . . . . .	92
gdcmm::network::AAAssociateRQPDU . . . . .	94
gdcmm::network::AReleaseRPPDU . . . . .	120
gdcmm::network::AReleaseRQPDU . . . . .	123
gdcmm::network::PDataTFPDU . . . . .	795
std::basic_string< Char >	
std::string	
gdcmm::String< TDelimiter, TMaxLength, TPadChar > . . . . .	1067
gdcmm::SegmentHelper::BasicCodedEntry . . . . .	185
gdcmm::BitmapToBitmapFilter . . . . .	206
gdcmm::PixmapToPixmapFilter . . . . .	839
gdcmm::ImageToImageFilter . . . . .	587
gdcmm::ImageApplyLookupTable . . . . .	536
gdcmm::ImageChangePhotometricInterpretation . . . . .	539
gdcmm::ImageChangePlanarConfiguration . . . . .	543
gdcmm::ImageChangeTransferSyntax . . . . .	548
gdcmm::ImageFragmentSplitter . . . . .	567
gdcmm::ByteBuffer . . . . .	213
gdcmm::ByteSwap< T > . . . . .	214
gdcmm::ByteSwapFilter . . . . .	216
gdcmm::network::CFind . . . . .	234
gdcmm::Coder . . . . .	251
gdcmm::Codec . . . . .	250
gdcmm::AudioCodec . . . . .	163
gdcmm::ImageCodec . . . . .	554
gdcmm::DeltaEncodingCodec . . . . .	348
gdcmm::JPEG2000Codec . . . . .	633
gdcmm::JPEGCodec . . . . .	646
gdcmm::JPEG12Codec . . . . .	623
gdcmm::JPEG16Codec . . . . .	628
gdcmm::JPEG8Codec . . . . .	641
gdcmm::JPEGLSCCodec . . . . .	655
gdcmm::KAKADUCodec . . . . .	664
gdcmm::PGXCodec . . . . .	812
gdcmm::PNMCodec . . . . .	847
gdcmm::PVRGCodec . . . . .	886
gdcmm::RAWCodec . . . . .	906
gdcmm::RLECodec . . . . .	926
gdcmm::PDFCodec . . . . .	803
gdcmm::CodeString . . . . .	253

gdcm::network::CompositeMessageFactory . . . . .	264
gdcm::CompositeNetworkFunctions . . . . .	266
gdcm::ConstCharWrapper . . . . .	271
gdcm::CryptoFactory . . . . .	275
gdcm::CAPICryptoFactory . . . . .	226
gdcm::OpenSSLCryptoFactory . . . . .	766
gdcm::OpenSSLP7CryptoFactory . . . . .	771
gdcm::CryptographicMessageSyntax . . . . .	278
gdcm::CAPICryptographicMessageSyntax . . . . .	227
gdcm::OpenSSLCryptographicMessageSyntax . . . . .	768
gdcm::OpenSSLP7CryptographicMessageSyntax . . . . .	773
gdcm::CSAElement . . . . .	281
gdcm::CSAHeader . . . . .	288
gdcm::CSAHeaderDict . . . . .	293
gdcm::CSAHeaderDictEntry . . . . .	296
gdcm::DataElement . . . . .	308
gdcm::CP246ExplicitDataElement . . . . .	272
gdcm::ExplicitDataElement . . . . .	439
gdcm::ExplicitImplicitDataElement . . . . .	443
gdcm::Fragment . . . . .	510
gdcm::BasicOffsetTable . . . . .	188
gdcm::ImplicitDataElement . . . . .	598
gdcm::Item . . . . .	616
gdcm::UNExplicitDataElement . . . . .	1262
gdcm::UNExplicitImplicitDataElement . . . . .	1265
gdcm::VR16ExplicitDataElement . . . . .	1301
gdcm::DataSet . . . . .	325
gdcm::CommandDataSet . . . . .	260
gdcm::FileMetaInformation . . . . .	471
gdcm::DataSetHelper . . . . .	340
gdcm::Decoder . . . . .	341
gdcm::Codec . . . . .	250
gdcm::DefinedTerms . . . . .	343
gdcm::Defs . . . . .	344
gdcm::DICOMDIR . . . . .	351
gdcm::DICOMDIRGenerator . . . . .	352
gdcm::Dict . . . . .	356
gdcm::DictConverter . . . . .	360
gdcm::DictEntry . . . . .	364
gdcm::Dicts . . . . .	371
gdcm::network::DIMSE . . . . .	375
gdcm::DirectionCosines . . . . .	376
gdcm::Directory . . . . .	380
gdcm::DirectoryHelper . . . . .	383
gdcm::DPath . . . . .	385
gdcm::DummyValueGenerator . . . . .	388
gdcm::Element< TVR, TVM > . . . . .	391
gdcm::Element< TVR, VM::VM1_n > . . . . .	399
gdcm::Element< TVR, VM::VM1_2 > . . . . .	397
gdcm::Element< TVR, VM::VM2_n > . . . . .	406
gdcm::Element< TVR, VM::VM2_2n > . . . . .	404
gdcm::Element< TVR, VM::VM3_4 > . . . . .	411

gdcmm::Element< TVR, VM::VM3_n > . . . . .	413
gdcmm::Element< TVR, VM::VM3_3n > . . . . .	408
gdcmm::Element< VR::AS, VM::VM5 > . . . . .	416
gdcmm::Element< VR::OB, VM::VM1_n > . . . . .	391
gdcmm::Element< VR::OB, VM::VM1 > . . . . .	417
gdcmm::Element< VR::OW, VM::VM1_n > . . . . .	391
gdcmm::Element< VR::OW, VM::VM1 > . . . . .	419
gdcmm::ElementDisableCombinations< TVR, TVM > . . . . .	421
gdcmm::ElementDisableCombinations< VR::OB, VM::VM1_n > . . . . .	422
gdcmm::ElementDisableCombinations< VR::OW, VM::VM1_n > . . . . .	422
gdcmm::EmptyMaskGenerator . . . . .	422
gdcmm::EncapsulatedDocument . . . . .	425
gdcmm::EncodingImplementation< T > . . . . .	425
gdcmm::EncodingImplementation< VR::VRASCII > . . . . .	426
gdcmm::EncodingImplementation< VR::VRBINARY > . . . . .	427
gdcmm::EnumeratedValues . . . . .	430
gdcmm::EquipmentManufacturer . . . . .	431
gdcmm::Event . . . . .	432
gdcmm::AnyEvent . . . . .	115
gdcmm::AbortEvent . . . . .	99
gdcmm::AnonymizeEvent . . . . .	103
gdcmm::DataEvent . . . . .	321
gdcmm::DataSetEvent . . . . .	337
gdcmm::EndEvent . . . . .	429
gdcmm::ExitEvent . . . . .	438
gdcmm::FileNameEvent . . . . .	484
gdcmm::InitializeEvent . . . . .	602
gdcmm::IterationEvent . . . . .	622
gdcmm::ModifiedEvent . . . . .	717
gdcmm::ProgressEvent . . . . .	883
gdcmm::StartEvent . . . . .	1035
gdcmm::UserEvent . . . . .	1273
gdcmm::NoEvent . . . . .	755
std::exception	
gdcmm::CSAHeaderDictException . . . . .	299
gdcmm::DataElementException . . . . .	321
gdcmm::Exception . . . . .	435
gdcmm::ParseException . . . . .	788
gdcmm::Fiducials . . . . .	447
gdcmm::FileDerivation . . . . .	464
gdcmm::FileExplicitFilter . . . . .	468
gdcmm::Filename . . . . .	481
gdcmm::FilenameGenerator . . . . .	487
gdcmm::FileSet . . . . .	491
gdcmm::Global . . . . .	514
gdcmm::GroupDict . . . . .	518
gdcmm::IconImageFilter . . . . .	520
gdcmm::IconImageGenerator . . . . .	523
gdcmm::ignore_char . . . . .	526
gdcmm::ImageConverter . . . . .	566
gdcmm::ImageHelper . . . . .	570
gdcmm::network::ImplementationClassUIDSub . . . . .	594
gdcmm::network::ImplementationUIDSub . . . . .	596



gdcm::network::ImplementationVersionNameSub	596
gdcm::IOD	603
gdcm::IODEntry	606
gdcm::IODs	608
gdcm::JSON	662
gdcm::Scanner2::ltstr	679
gdcm::Scanner::ltstr	680
gdcm::StrictScanner2::ltstr	680
gdcm::StrictScanner::ltstr	681
gdcm::Macro	681
gdcm::Macros	684
gdcm::network::MaximumLengthSub	686
gdcm::MD5	687
gdcm::MEC_MR3	688
gdcm::MediaStorage	689
gdcm::Module	718
gdcm::ModuleEntry	721
gdcm::NestedModuleEntries	746
gdcm::Modules	725
gdcm::MrProtocol	735
gdcm::network::NormalizedMessageFactory	756
gdcm::NormalizedNetworkFunctions	757
gdcm::Object	763
gdcm::BaseQuery	175
gdcm::BaseRootQuery	180
gdcm::FindPatientRootQuery	502
gdcm::FindStudyRootQuery	506
gdcm::MovePatientRootQuery	727
gdcm::MoveStudyRootQuery	731
gdcm::WLMFindQuery	1406
gdcm::ModalityPerformedProcedureStepCreateQuery	710
gdcm::ModalityPerformedProcedureStepSetQuery	713
gdcm::Bitmap	192
gdcm::Pixmap	828
gdcm::Image	527
gdcm::Curve	303
gdcm::File	448
gdcm::FileWithName	499
gdcm::LookupTable	671
gdcm::SegmentedPaletteColorLookupTable	963
gdcm::MeshPrimitive	705
gdcm::Overlay	779
gdcm::Segment	954
gdcm::Subject	1077
gdcm::Anonymizer	106
gdcm::Cleaner	239
gdcm::Command	257
gdcm::MemberCommand< T >	699
gdcm::SimpleMemberCommand< T >	1007
gdcm::FileAnonymizer	453
gdcm::FileChangeTransferSyntax	457
gdcm::FileDecompressLookupTable	461
gdcm::FileStreamer	493

gdcmm::Scanner	934
gdcmm::Scanner2	943
gdcmm::ServiceClassUser	997
gdcmm::StrictScanner	1048
gdcmm::StrictScanner2	1057
gdcmm::network::ULConnectionManager	1249
gdcmm::Surface	1081
gdcmm::Value	1279
gdcmm::ByteValue	218
gdcmm::SequenceOfFragments	974
gdcmm::SequenceOfItems	982
gdcmm::Orientation	776
gdcmm::Parser	790
gdcmm::Patient	794
gdcmm::PDBelement	798
gdcmm::PDBHeader	800
gdcmm::network::PDUFactory	806
gdcmm::PersonName	809
gdcmm::PhotometricInterpretation	816
gdcmm::PixelFormat	820
gdcmm::Preamble	852
gdcmm::PresentationContext	855
gdcmm::network::PresentationContextAC	859
gdcmm::PresentationContextGenerator	861
gdcmm::network::PresentationContextRQ	864
gdcmm::network::PresentationDataValue	867
gdcmm::Printer	871
gdcmm::DictPrinter	368
gdcmm::Dumper	389
gdcmm::PrivateDict	875
gdcmm::PythonFilter	890
gdcmm::QueryBase	892
gdcmm::QueryImage	896
gdcmm::QueryPatient	899
gdcmm::QuerySeries	901
gdcmm::QueryStudy	904
gdcmm::QueryFactory	895
gdcmm::Reader	911
gdcmm::PixmapReader	835
gdcmm::ImageReader	577
gdcmm::ImageRegionReader	582
gdcmm::SegmentReader	966
gdcmm::SurfaceReader	1096
gdcmm::RealWorldValueMappingContent	918
gdcmm::Region	919
gdcmm::BoxRegion	208
gdcmm::Rescaler	921
gdcmm::network::RoleSelectionSub	933
gdcmm::SerieHelper	990
gdcmm::Series	995
gdcmm::network::ServiceClassApplicationInformation	996
gdcmm::SHA1	1005
gdcmm::SimpleSubjectWatcher	1012

gdcm::MrProtocol::Slice	1015
gdcm::MrProtocol::SliceArray	1016
gdcm::SmartPointer< ObjectType >	1017
gdcm::SmartPointer< gdcm::Bitmap >	1017
gdcm::SmartPointer< gdcm::File >	1017
gdcm::SmartPointer< gdcm::Image >	1017
gdcm::SmartPointer< gdcm::MemberCommand >	1017
gdcm::SmartPointer< gdcm::MeshPrimitive >	1017
gdcm::SmartPointer< gdcm::Pixmap >	1017
gdcm::SmartPointer< gdcm::SimpleMemberCommand >	1017
gdcm::SmartPointer< gdcm::Subject >	1017
gdcm::SmartPointer< LookupTable >	1017
gdcm::SmartPointer< Segment >	1017
gdcm::SmartPointer< Surface >	1017
gdcm::SmartPointer< Value >	1017
gdcm::network::SOPClassExtendedNegociationSub	1021
gdcm::SOPClassUIDToIOD	1022
gdcm::Sorter	1024
gdcm::IPPSorter	611
gdcm::Spacing	1029
gdcm::Spectroscopy	1031
gdcm::SplitMosaicFilter	1032
gdcm::static_assert_test< x >	1036
gdcm::STATIC_ASSERTION_FAILURE< x >	1037
gdcm::STATIC_ASSERTION_FAILURE< true >	1037
gdcm::StreamImageReader	1037
gdcm::StreamImageWriter	1041
String<'\', 64 >	
gdcm::LO	668
gdcm::StringFilter	1072
gdcm::Study	1076
gdcm::SurfaceHelper	1093
gdcm::SwapCode	1104
gdcm::SwapperDoOp	1106
gdcm::SwapperNoOp	1107
gdcm::System	1107
gdcm::Table	1114
gdcm::TableEntry	1117
gdcm::TableReader	1118
gdcm::XMLDictReader	1417
gdcm::XMLPrivateDictReader	1423
gdcm::network::TableRow	1122
gdcm::Tag	1123
gdcm::PrivateTag	877
gdcm::TagPath	1132
gdcm::Testing	1134
gdcm::Trace	1141
gdcm::TransferSyntax	1146
gdcm::network::TransferSyntaxSub	1152
gdcm::network::Transition	1154
gdcm::Type	1156
gdcm::UI	1158
gdcm::UIDGenerator	1158

gdcM::UIDs	1161
gdcM::network::ULAction	1197
gdcM::network::ULActionAA1	1200
gdcM::network::ULActionAA2	1202
gdcM::network::ULActionAA3	1203
gdcM::network::ULActionAA4	1204
gdcM::network::ULActionAA5	1206
gdcM::network::ULActionAA6	1207
gdcM::network::ULActionAA7	1208
gdcM::network::ULActionAA8	1210
gdcM::network::ULActionAE1	1211
gdcM::network::ULActionAE2	1212
gdcM::network::ULActionAE3	1214
gdcM::network::ULActionAE4	1215
gdcM::network::ULActionAE5	1216
gdcM::network::ULActionAE6	1218
gdcM::network::ULActionAE7	1219
gdcM::network::ULActionAE8	1220
gdcM::network::ULActionAR1	1222
gdcM::network::ULActionAR10	1223
gdcM::network::ULActionAR2	1224
gdcM::network::ULActionAR3	1226
gdcM::network::ULActionAR4	1227
gdcM::network::ULActionAR5	1228
gdcM::network::ULActionAR6	1230
gdcM::network::ULActionAR7	1231
gdcM::network::ULActionAR8	1232
gdcM::network::ULActionAR9	1234
gdcM::network::ULActionDT1	1235
gdcM::network::ULActionDT2	1236
gdcM::network::ULConnection	1240
gdcM::network::ULConnectionCallback	1245
gdcM::network::ULBasicCallback	1238
gdcM::network::ULWritingCallback	1259
gdcM::network::ULConnectionInfo	1247
gdcM::network::ULEvent	1256
gdcM::network::ULTransitionTable	1258
gdcM::Unpacker12Bits	1269
gdcM::Usage	1270
gdcM::network::UserInformation	1274
gdcM::UUIDGenerator	1276
gdcM::Validate	1277
gdcM::ValueIO< TDE, TSwap, TType >	1282
gdcM::MrProtocol::Vector3	1283
gdcM::Version	1284
gdcM::VL	1285
gdcM::VM	1289
gdcM::VMToLength< T >	1294
gdcM::VR	1294
gdcM::VRToEncoding< T >	1304
gdcM::VRToType< T >	1305
gdcM::VRToType< TagToType< Group, Element >::VRType >	1305
gdcM::VRToType< TVR >	1305
gdcM::VRVLSIZE< T >	1305

gdcm::VRVLSize< 0 > . . . . .	.1305
gdcm::VRVLSize< 1 > . . . . .	.1306
vtkImageAlgorithm	
vtkImagePlanarComponentsToComponents . . . . .	.1387
vtkImageMapToColors	
vtkImageMapToWindowLevelColors2 . . . . .	.1384
vtkImageWriter	
vtkGDCMImageWriter . . . . .	.1332
vtkLookupTable	
vtkLookupTable16 . . . . .	.1393
vtkMedicalImageProperties	
vtkGDCMMedicalImageProperties . . . . .	.1340
vtkMedicalImageReader2	
vtkGDCMImageReader . . . . .	.1307
vtkGDCMThreadedImageReader . . . . .	.1354
vtkGDCMImageReader2 . . . . .	.1320
vtkObject	
vtkGDCMTesting . . . . .	.1351
vtkImageColorViewer . . . . .	.1366
vtkRTStructSetProperties . . . . .	.1396
vtkPolyDataAlgorithm	
vtkGDCMPolyDataReader . . . . .	.1343
vtkPolyDataWriter	
vtkGDCMPolyDataWriter . . . . .	.1347
vtkThreadedImageAlgorithm	
vtkGDCMThreadedImageReader2 . . . . .	.1359
vtkImageMapToColors16 . . . . .	.1378
vtkImageRGBToYBR . . . . .	.1389
vtkImageYBRToRGB . . . . .	.1391
gdcm::Waveform . . . . .	.1406
gdcm::Writer . . . . .	.1411
gdcm::PixmapWriter . . . . .	.842
gdcm::ImageWriter . . . . .	.590
gdcm::SegmentWriter . . . . .	.970
gdcm::SurfaceWriter . . . . .	.1100
gdcm::XMLPrinter . . . . .	.1420



# Chapter 7

## Class Index

### 7.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">gdcn::network::AAabortPDU</a>	
<a href="#">AAabortPDU</a>	85
<a href="#">gdcn::network::AAAssociateACPDU</a>	
<a href="#">AAAssociateACPDU</a>	88
<a href="#">gdcn::network::AAAssociateRJPDU</a>	
<a href="#">AAAssociateRJPDU</a>	92
<a href="#">gdcn::network::AAAssociateRQPDU</a>	
<a href="#">AAAssociateRQPDU</a>	94
<a href="#">gdcn::AbortEvent</a>	99
<a href="#">gdcn::network::AbstractSyntax</a>	
<a href="#">AbstractSyntax</a>	100
<a href="#">gdcn::AnonymizeEvent</a>	
<a href="#">AnonymizeEvent</a>	103
<a href="#">gdcn::Anonymizer</a>	
<a href="#">Anonymizer</a>	106
<a href="#">gdcn::AnyEvent</a>	115
<a href="#">gdcn::network::ApplicationContext</a>	
<a href="#">ApplicationContext</a>	116
<a href="#">gdcn::ApplicationEntity</a>	
<a href="#">ApplicationEntity</a>	118
<a href="#">gdcn::network::AReleaseRPPDU</a>	
<a href="#">AReleaseRPPDU</a>	120
<a href="#">gdcn::network::AReleaseRQPDU</a>	
<a href="#">AReleaseRQPDU</a>	123
<a href="#">gdcn::network::ARTIMTimer</a>	
<a href="#">ARTIMTimer</a>	125
<a href="#">gdcn::ASN1</a>	
<a href="#">Class for ASN1</a>	126
<a href="#">gdcn::network::AsynchronousOperationsWindowSub</a>	
<a href="#">AsynchronousOperationsWindowSub</a>	128

gdcmm::Attribute< Group, Element, TVR, TVM >	
Attribute class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary	130
gdcmm::Attribute< Group, Element, TVR, VM::VM1 >	138
gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >	144
gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >	146
gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >	148
gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >	153
gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >	156
gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >	158
gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >	161
gdcmm::AudioCodec	
AudioCodec	163
gdcmm::Base64	
Class for Base64	166
gdcmm::network::BaseCompositeMessage	
BaseCompositeMessage	169
gdcmm::network::BaseNormalizedMessage	
BaseNormalizedMessage	170
gdcmm::network::BasePDU	
BasePDU	173
gdcmm::BaseQuery	
BaseQuery	175
gdcmm::BaseRootQuery	
BaseRootQuery	180
gdcmm::SegmentHelper::BasicCodedEntry	
This structure defines a basic coded entry with all of its attributes	185
gdcmm::BasicOffsetTable	
Class to represent a BasicOffsetTable	188
gdcmm::Bitmap	
Bitmap class	192
gdcmm::BitmapToBitmapFilter	
BitmapToBitmapFilter class	206
gdcmm::BoxRegion	
Class for manipulation box region	208
gdcmm::ByteBuffer	
ByteBuffer	213
gdcmm::ByteSwap< T >	
ByteSwap	214
gdcmm::ByteSwapFilter	
ByteSwapFilter	216
gdcmm::ByteValue	
Class to represent binary value (array of bytes)	218
gdcmm::CAPICryptoFactory	226
gdcmm::CAPICryptographicMessageSyntax	227
gdcmm::network::CEchoRQ	
CEchoRQ	231
gdcmm::network::CEchoRSP	
CEchoRSP this file defines the messages for the cecho action	233
gdcmm::network::CFind	234
gdcmm::network::CFindCancelRQ	
CFindCancelRQ this file defines the messages for the cfind action	234
gdcmm::network::CFindRQ	
CFindRQ	236



gdcm::network::CFindRSP	
CFindRSP this file defines the messages for the cfind action	237
gdcm::Cleaner	
Cleaner	239
gdcm::network::CMoveCancelRq	246
gdcm::network::CMoveRQ	
CMoveRQ	247
gdcm::network::CMoveRSP	
CMoveRSP this file defines the messages for the cmove action	248
gdcm::Codec	
Codec class	250
gdcm::Coder	
Coder	251
gdcm::CodeString	
CodeString	253
gdcm::Command	
Command superclass for callback/observer methods	257
gdcm::CommandDataSet	
Class to represent a <a href="#">Command DataSet</a>	260
gdcm::network::CompositeMessageFactory	
CompositeMessageFactory	264
gdcm::CompositeNetworkFunctions	
Composite Network Functions	266
gdcm::ConstCharWrapper	
Do not use me	271
gdcm::CP246ExplicitDataElement	
Class to read/write a <a href="#">DataElement</a> as CP246Explicit Data <a href="#">Element</a>	272
gdcm::CryptoFactory	
Class to do handle the crypto factory	275
gdcm::CryptographicMessageSyntax	278
gdcm::CSAElement	
Class to represent a CSA <a href="#">Element</a>	281
gdcm::CSAHeader	
Class for <a href="#">CSAHeader</a>	288
gdcm::CSAHeaderDict	
Class to represent a map of <a href="#">CSAHeaderDictEntry</a>	293
gdcm::CSAHeaderDictEntry	
Class to represent an Entry in the <a href="#">Dict</a>	296
gdcm::CSAHeaderDictException	299
gdcm::network::CStoreRQ	
CStoreRQ	300
gdcm::network::CStoreRSP	
CStoreRSP this file defines the messages for the cecho action	301
gdcm::Curve	
Curve class to handle element 50xx,3000 <a href="#">Curve</a> Data	303
gdcm::DataElement	
Class to represent a Data <a href="#">Element</a> either Implicit or Explicit	308
gdcm::DataElementException	321
gdcm::DataEvent	
DataEvent	321
gdcm::DataSet	
Class to represent a Data Set (which contains Data Elements)	325
gdcm::DataSetEvent	
DataSetEvent	337

gdcm::DataSetHelper	
DataSetHelper (internal class, not intended for user level)	340
gdcm::Decoder	
Decoder	341
gdcm::DefinedTerms	
Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data Element with Defined Terms that does not contain a Value equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation Type ID (4008,0210) is an example of a Data Element having Defined Terms. It is defined to have a Value that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data Element has Defined Terms other Interpretation Type IDs may be defined by the implementor	343
gdcm::Defs	
FIXME I do not like the name 'Defs'	344
gdcm::DeltaEncodingCodec	
DeltaEncodingCodec compression used by some private vendor	348
gdcm::DICOMDIR	
DICOMDIR class	351
gdcm::DICOMDIRGenerator	
DICOMDIRGenerator class	352
gdcm::Dict	
Class to represent a map of DictEntry	356
gdcm::DictConverter	
Class to convert a .dic file into something else:	360
gdcm::DictEntry	
Class to represent an Entry in the Dict	364
gdcm::DictPrinter	
DictPrinter class	368
gdcm::Dicts	
Class to manipulate the sum of knowledge (all the dict user load)	371
gdcm::network::DIMSE	
DIMSE	375
gdcm::DirectionCosines	
Class to handle DirectionCosines	376
gdcm::Directory	
Class for manipulation directories	380
gdcm::DirectoryHelper	
DirectoryHelper	383
gdcm::DPath	
Class to handle a DICOM path While supp 118 did introduced a notion of XPath for XML Native model this convention is too XML-centric. Instead prefer DCMTK style notation <a href="https://groups.google.com/g/comp.protocols.dicom/c/IyIH0IOBMPA">https://groups.google.com/g/comp.protocols.dicom/c/IyIH0IOBMPA</a>	385
gdcm::DummyValueGenerator	
Class for generating dummy value	388
gdcm::Dumper	
Codec class	389
gdcm::Element< TVR, TVM >	
Element class	391
gdcm::Element< TVR, VM::VM1_2 >	397
gdcm::Element< TVR, VM::VM1_n >	399
gdcm::Element< TVR, VM::VM2_2n >	404
gdcm::Element< TVR, VM::VM2_n >	406

gdcm::Element< TVR, VM::VM3_3n > . . . . .	408
gdcm::Element< TVR, VM::VM3_4 > . . . . .	411
gdcm::Element< TVR, VM::VM3_n > . . . . .	413
gdcm::Element< VR::AS, VM::VM5 > . . . . .	416
gdcm::Element< VR::OB, VM::VM1 > . . . . .	417
gdcm::Element< VR::OW, VM::VM1 > . . . . .	419
gdcm::ElementDisableCombinations< TVR, TVM > . . . . .	
A class which is used to produce compile errors for an invalid combination of template parameters . . . . .	421
gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n > . . . . .	422
gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n > . . . . .	422
gdcm::EmptyMaskGenerator . . . . .	
EmptyMaskGenerator Main class to generate a Empty Mask <a href="#">Series</a> from an input <a href="#">Series</a> . This class takes an input folder and generates a series of DICOM files in the specified output directory. This class handles multiples DICOM <a href="#">Series</a> within the same input directory . . . . .	422
gdcm::EncapsulatedDocument . . . . .	
EncapsulatedDocument . . . . .	425
gdcm::EncodingImplementation< T > . . . . .	
EncodingImplementation . . . . .	425
gdcm::EncodingImplementation< VR::VRASCII > . . . . .	426
gdcm::EncodingImplementation< VR::VRBINARY > . . . . .	427
gdcm::EndEvent . . . . .	429
gdcm::EnumeratedValues . . . . .	
Element. A Data <a href="#">Element</a> with Enumerated Values that does not have a <a href="#">Value</a> equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note: . . . . .	430
gdcm::EquipmentManufacturer . . . . .	431
gdcm::Event . . . . .	
Superclass for callback/observer methods . . . . .	432
gdcm::Exception . . . . .	
Exception . . . . .	435
gdcm::ExitEvent . . . . .	438
gdcm::ExplicitDataElement . . . . .	
Class to read/write a <a href="#">DataElement</a> as Explicit Data <a href="#">Element</a> . . . . .	439
gdcm::ExplicitImplicitDataElement . . . . .	
Class to read/write a <a href="#">DataElement</a> as ExplicitImplicit Data <a href="#">Element</a> . . . . .	443
gdcm::Fiducials . . . . .	
Fiducials . . . . .	447
gdcm::File . . . . .	
DICOM <a href="#">File</a> . . . . .	448
gdcm::FileAnonymizer . . . . .	
FileAnonymizer . . . . .	453
gdcm::FileChangeTransferSyntax . . . . .	
FileChangeTransferSyntax . . . . .	457
gdcm::FileDecompressLookupTable . . . . .	
FileDecompressLookupTable class . . . . .	461
gdcm::FileDerivation . . . . .	
FileDerivation class . . . . .	464
gdcm::FileExplicitFilter . . . . .	
FileExplicitFilter class . . . . .	468
gdcm::FileMetaInformation . . . . .	
Class to represent a <a href="#">File</a> Meta Information . . . . .	471
gdcm::Filename . . . . .	
Class to manipulate file name's . . . . .	481

gdcm::FileNameEvent	
FileNameEvent	484
gdcm::FilenameGenerator	
FilenameGenerator	487
gdcm::FileSet	491
gdcm::FileStreamer	
FileStreamer	493
gdcm::FileWithName	
FileWithName	499
gdcm::FindPatientRootQuery	
PatientRootQuery	502
gdcm::FindStudyRootQuery	
FindStudyRootQuery	506
gdcm::Fragment	
Class to represent a <a href="#">Fragment</a>	510
gdcm::Global	
Global	514
gdcm::GroupDict	
Class to represent the mapping from group number to its abbreviation and name	518
gdcm::IconImageFilter	
IconImageFilter	520
gdcm::IconImageGenerator	
IconImageGenerator	523
gdcm::ignore_char	526
gdcm::Image	
Image	527
gdcm::ImageApplyLookupTable	
ImageApplyLookupTable class	536
gdcm::ImageChangePhotometricInterpretation	
ImageChangePhotometricInterpretation class	539
gdcm::ImageChangePlanarConfiguration	
ImageChangePlanarConfiguration class	543
gdcm::ImageChangeTransferSyntax	
ImageChangeTransferSyntax class	548
gdcm::ImageCodec	
ImageCodec	554
gdcm::ImageConverter	
Image Converter	566
gdcm::ImageFragmentSplitter	
ImageFragmentSplitter class	567
gdcm::ImageHelper	
ImageHelper (internal class, not intended for user level)	570
gdcm::ImageReader	
ImageReader	577
gdcm::ImageRegionReader	
ImageRegionReader	582
gdcm::ImageToImageFilter	
ImageToImageFilter class	587
gdcm::ImageWriter	
ImageWriter	590
gdcm::network::ImplementationClassUIDSub	
ImplementationClassUIDSub	594
gdcm::network::ImplementationUIDSub	
ImplementationUIDSub	596

gdcm::network::ImplementationVersionNameSub	
ImplementationVersionNameSub	596
gdcm::ImplicitDataElement	
Class to represent an <i>Implicit VR</i> Data Element	598
gdcm::InitializeEvent	602
gdcm::IOD	
Class for representing a IOD	603
gdcm::IODEntry	
Class for representing a IODEntry	606
gdcm::IODs	
Class for representing a IODs	608
gdcm::IPPSorter	
IPPSorter	611
gdcm::Item	
Class to represent an Item	616
gdcm::IterationEvent	622
gdcm::JPEG12Codec	
Class to do JPEG 12bits (lossy & lossless)	623
gdcm::JPEG16Codec	
Class to do JPEG 16bits (lossless)	628
gdcm::JPEG2000Codec	
Class to do JPEG 2000	633
gdcm::JPEG8Codec	
Class to do JPEG 8bits (lossy & lossless)	641
gdcm::JPEGCodec	
JPEG codec	646
gdcm::JPEGLSCodec	
JPEG-LS	655
gdcm::JSON	662
gdcm::KAKADUCodec	
KAKADUCodec	664
gdcm::LO	
LO	668
gdcm::LookupTable	
LookupTable class	671
gdcm::Scanner2::Itstr	679
gdcm::Scanner::Itstr	680
gdcm::StrictScanner2::Itstr	680
gdcm::StrictScanner::Itstr	681
gdcm::Macro	
Class for representing a Macro	681
gdcm::Macros	
Class for representing a Modules	684
gdcm::network::MaximumLengthSub	
MaximumLengthSub	686
gdcm::MD5	
Class for MD5	687
gdcm::MEC_MR3	
Class for MEC_MR3	688
gdcm::MediaStorage	
MediaStorage	689
gdcm::MemberCommand< T >	
Command subclass that calls a pointer to a member function	699

<a href="#">gdcmmesh::MeshPrimitive</a>	
This class defines surface mesh primitives	705
<a href="#">gdcmmesh::ModalityPerformedProcedureStepCreateQuery</a>	
ModalityPerformedProcedureStepCreateQuery	710
<a href="#">gdcmmesh::ModalityPerformedProcedureStepSetQuery</a>	
ModalityPerformedProcedureStepSetQuery	713
<a href="#">gdcmmesh::ModifiedEvent</a>	717
<a href="#">gdcmmesh::Module</a>	
Class for representing a <a href="#">Module</a>	718
<a href="#">gdcmmesh::ModuleEntry</a>	
Class for representing a <a href="#">ModuleEntry</a>	721
<a href="#">gdcmmesh::Modules</a>	
Class for representing a <a href="#">Modules</a>	725
<a href="#">gdcmmesh::MovePatientRootQuery</a>	
MovePatientRootQuery	727
<a href="#">gdcmmesh::MoveStudyRootQuery</a>	
MoveStudyRootQuery	731
<a href="#">gdcmmesh::MrProtocol</a>	
Class for <a href="#">MrProtocol</a>	735
<a href="#">gdcmmesh::network::NActionRQ</a>	
NActionRQ	737
<a href="#">gdcmmesh::network::NActionRSP</a>	
NActionRSP this file defines the messages for the NAction action	738
<a href="#">gdcmmesh::network::NCreateRQ</a>	
NCreateRQ	740
<a href="#">gdcmmesh::network::NCreateRSP</a>	
NCreateRSP this file defines the messages for the ncreate action	741
<a href="#">gdcmmesh::network::NDeleteRQ</a>	
NDeleteRQ	743
<a href="#">gdcmmesh::network::NDeleteRSP</a>	
NDeleteRSP this file defines the messages for the ndelete action	744
<a href="#">gdcmmesh::NestedModuleEntries</a>	
Class for representing a <a href="#">NestedModuleEntries</a>	746
<a href="#">gdcmmesh::network::NEventReportRQ</a>	
NEventReportRQ	749
<a href="#">gdcmmesh::network::NEventReportRSP</a>	
NEventReportRSP this file defines the messages for the neventreport action	751
<a href="#">gdcmmesh::network::NGetRQ</a>	
NGetRQ	752
<a href="#">gdcmmesh::network::NGetRSP</a>	
NGetRSP this file defines the messages for the nget action	754
<a href="#">gdcmmesh::NoEvent</a>	755
<a href="#">gdcmmesh::network::NormalizedMessageFactory</a>	756
<a href="#">gdcmmesh::NormalizedNetworkFunctions</a>	
Normalized Network Functions	757
<a href="#">gdcmmesh::network::NSetRQ</a>	
NSetRQ	760
<a href="#">gdcmmesh::network::NSetRSP</a>	
NSetRSP this file defines the messages for the nset action	761
<a href="#">gdcmmesh::Object</a>	
Object	763
<a href="#">gdcmmesh::OpenSSLCryptoFactory</a>	766
<a href="#">gdcmmesh::OpenSSLCryptographicMessageSyntax</a>	768
<a href="#">gdcmmesh::OpenSSLP7CryptoFactory</a>	771

gdcm::OpenSSLP7CryptographicMessageSyntax	773
gdcm::Orientation	
Class to handle <a href="#">Orientation</a>	776
gdcm::Overlay	
Overlay class	779
gdcm::ParseException	
ParseException Standard exception handling object	788
gdcm::Parser	
Parser ala XML_Parser from expat (SAX)	790
gdcm::Patient	
See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54	794
gdcm::network::PDataTFPDU	
PDataTFPDU	795
gdcm::PDBelement	
Class to represent a PDB <a href="#">Element</a>	798
gdcm::PDBHeader	
Class for <a href="#">PDBHeader</a>	800
gdcm::PDFCodec	
PDFCodec class	803
gdcm::network::PDUFactory	
PDUFactory basically, given an initial byte, construct the	806
gdcm::PersonName	
PersonName class	809
gdcm::PGXCodec	
Class to do PGX	812
gdcm::PhotometricInterpretation	
Class to represent an <a href="#">PhotometricInterpretation</a>	816
gdcm::PixelFormat	
PixelFormat	820
gdcm::Pixmap	
Pixmap class	828
gdcm::PixmapReader	
PixmapReader	835
gdcm::PixmapToPixmapFilter	
PixmapToPixmapFilter class	839
gdcm::PixmapWriter	
PixmapWriter	842
gdcm::PNMCodec	
Class to do PNM	847
gdcm::Preamble	
DICOM <a href="#">Preamble</a> (Part 10)	852
gdcm::PresentationContext	
PresentationContext	855
gdcm::network::PresentationContextAC	
PresentationContextAC	859
gdcm::PresentationContextGenerator	
PresentationContextGenerator	861
gdcm::network::PresentationContextRQ	
PresentationContextRQ	864
gdcm::network::PresentationDataValue	
PresentationDataValue	867
gdcm::Printer	
Printer class	871

<a href="#">gdcmm::PrivateDict</a>	
Private Dict . . . . .	875
<a href="#">gdcmm::PrivateTag</a>	
Class to represent a Private DICOM Data <a href="#">Element</a> ( <a href="#">Attribute</a> ) <a href="#">Tag</a> (Group, <a href="#">Element</a> , Owner) . . . . .	877
<a href="#">gdcmm::ProgressEvent</a>	
ProgressEvent . . . . .	883
<a href="#">gdcmm::PVRGCodec</a>	
PVRGCodec . . . . .	886
<a href="#">gdcmm::PythonFilter</a>	
PythonFilter <a href="#">PythonFilter</a> is the class that make gdcmm2.x looks more like gdcmm1 and transform the binary blob contained in a <a href="#">DataElement</a> into a string, typically this is a nice feature to have for wrapped language . . . . .	890
<a href="#">gdcmm::QueryBase</a>	
QueryBase . . . . .	892
<a href="#">gdcmm::QueryFactory</a>	
QueryFactory.h . . . . .	895
<a href="#">gdcmm::QueryImage</a>	
QueryImage . . . . .	896
<a href="#">gdcmm::QueryPatient</a>	
QueryPatient . . . . .	899
<a href="#">gdcmm::QuerySeries</a>	
QuerySeries . . . . .	901
<a href="#">gdcmm::QueryStudy</a>	
QueryStudy.h . . . . .	904
<a href="#">gdcmm::RAWCodec</a>	
RAWCodec class . . . . .	906
<a href="#">gdcmm::Reader</a>	
Reader ala DOM (Document <a href="#">Object</a> Model) . . . . .	911
<a href="#">gdcmm::RealWorldValueMappingContent</a>	918
<a href="#">gdcmm::Region</a>	
Class for manipulation region . . . . .	919
<a href="#">gdcmm::Rescaler</a>	
Rescale class . . . . .	921
<a href="#">gdcmm::RLECodec</a>	
Class to do RLE . . . . .	926
<a href="#">gdcmm::network::RoleSelectionSub</a>	
RoleSelectionSub . . . . .	933
<a href="#">gdcmm::Scanner</a>	
Scanner . . . . .	934
<a href="#">gdcmm::Scanner2</a>	
Scanner2 . . . . .	943
<a href="#">gdcmm::Segment</a>	
This class defines a segment . . . . .	954
<a href="#">gdcmm::SegmentedPaletteColorLookupTable</a>	
SegmentedPaletteColorLookupTable class . . . . .	963
<a href="#">gdcmm::SegmentReader</a>	
This class defines a segment reader . . . . .	966
<a href="#">gdcmm::SegmentWriter</a>	
This class defines a segment writer . . . . .	970
<a href="#">gdcmm::SequenceOfFragments</a>	
Class to represent a Sequence Of Fragments . . . . .	974
<a href="#">gdcmm::SequenceOfItems</a>	
Class to represent a Sequence Of Items . . . . .	982



gdcm::SerieHelper	
SerieHelper	DO NOT USE this class, it is only a temporary solution for ITK migration from GDCM 1.x to GDCM 2.x It will disappear soon, you've been warned . . . . .
	990
gdcm::Series	
Series	. . . . .
	995
gdcm::network::ServiceClassApplicationInformation	. . . . .
	996
gdcm::ServiceClassUser	
ServiceClassUser	. . . . .
	997
gdcm::SHA1	
Class for SHA1	. . . . .
	1005
gdcm::SimpleMemberCommand< T >	
Command	subclass that calls a pointer to a member function . . . . .
	1007
gdcm::SimpleSubjectWatcher	
SimpleSubjectWatcher	. . . . .
	1012
gdcm::MrProtocol::Slice	. . . . .
	1015
gdcm::MrProtocol::SliceArray	. . . . .
	1016
gdcm::SmartPointer< ObjectType >	
Class for Smart Pointer	. . . . .
	1017
gdcm::network::SOPClassExtendedNegociationSub	
SOPClassExtendedNegociationSub	. . . . .
	1021
gdcm::SOPClassUIDToIOD	
Class convert a class SOP Class UID into IOD	. . . . .
	1022
gdcm::Sorter	
Sorter	. . . . .
	1024
gdcm::Spacing	
Class for Spacing	. . . . .
	1029
gdcm::Spectroscopy	
Spectroscopy class	. . . . .
	1031
gdcm::SplitMosaicFilter	
SplitMosaicFilter class	. . . . .
	1032
gdcm::StartEvent	. . . . .
	1035
gdcm::static_assert_test< x >	. . . . .
	1036
gdcm::STATIC_ASSERTION_FAILURE< x >	. . . . .
	1037
gdcm::STATIC_ASSERTION_FAILURE< true >	. . . . .
	1037
gdcm::StreamImageReader	
StreamImageReader	. . . . .
	1037
gdcm::StreamImageWriter	
StreamImageReader	. . . . .
	1041
gdcm::StrictScanner	
StrictScanner	. . . . .
	1048
gdcm::StrictScanner2	
StrictScanner2	. . . . .
	1057
gdcm::String< TDelimiter, TMaxLength, TPadChar >	
String	. . . . .
	1067
gdcm::StringFilter	
StringFilter	. . . . .
	1072
gdcm::Study	
Study	. . . . .
	1076
gdcm::Subject	
Subject	. . . . .
	1077
gdcm::Surface	
This class defines a SURFACE IE	. . . . .
	1081
gdcm::SurfaceHelper	
SurfaceHelper	. . . . .
	1093

<a href="#">gdcmm::SurfaceReader</a>	
This class defines a SURFACE IE reader	1096
<a href="#">gdcmm::SurfaceWriter</a>	
This class defines a SURFACE IE writer	1100
<a href="#">gdcmm::SwapCode</a>	
SwapCode representation	1104
<a href="#">gdcmm::SwapperDoOp</a>	1106
<a href="#">gdcmm::SwapperNoOp</a>	1107
<a href="#">gdcmm::System</a>	
Class to do system operation	1107
<a href="#">gdcmm::Table</a>	
Table	1114
<a href="#">gdcmm::TableEntry</a>	
TableEntry	1117
<a href="#">gdcmm::TableReader</a>	
Class for representing a <a href="#">TableReader</a>	1118
<a href="#">gdcmm::network::TableRow</a>	1122
<a href="#">gdcmm::Tag</a>	
Class to represent a DICOM Data <a href="#">Element</a> ( <a href="#">Attribute</a> ) <a href="#">Tag</a> (Group, <a href="#">Element</a> )	1123
<a href="#">gdcmm::TagPath</a>	
Class to handle a path of tag	1132
<a href="#">gdcmm::Testing</a>	
Class for testing	1134
<a href="#">gdcmm::Trace</a>	
Trace	1141
<a href="#">gdcmm::TransferSyntax</a>	
Class to manipulate Transfer Syntax	1146
<a href="#">gdcmm::network::TransferSyntaxSub</a>	
TransferSyntaxSub	1152
<a href="#">gdcmm::network::Transition</a>	1154
<a href="#">gdcmm::Type</a>	
Type	1156
<a href="#">gdcmm::UI</a>	1158
<a href="#">gdcmm::UIDGenerator</a>	
Class for generating unique UID	1158
<a href="#">gdcmm::UIDs</a>	
All known uids	1161
<a href="#">gdcmm::network::ULAction</a>	
ULAction	1197
<a href="#">gdcmm::network::ULActionAA1</a>	1200
<a href="#">gdcmm::network::ULActionAA2</a>	1202
<a href="#">gdcmm::network::ULActionAA3</a>	1203
<a href="#">gdcmm::network::ULActionAA4</a>	1204
<a href="#">gdcmm::network::ULActionAA5</a>	1206
<a href="#">gdcmm::network::ULActionAA6</a>	1207
<a href="#">gdcmm::network::ULActionAA7</a>	1208
<a href="#">gdcmm::network::ULActionAA8</a>	1210
<a href="#">gdcmm::network::ULActionAE1</a>	1211
<a href="#">gdcmm::network::ULActionAE2</a>	1212
<a href="#">gdcmm::network::ULActionAE3</a>	1214
<a href="#">gdcmm::network::ULActionAE4</a>	1215
<a href="#">gdcmm::network::ULActionAE5</a>	1216
<a href="#">gdcmm::network::ULActionAE6</a>	1218
<a href="#">gdcmm::network::ULActionAE7</a>	1219

gdcm::network::ULActionAE8	1220
gdcm::network::ULActionAR1	1222
gdcm::network::ULActionAR10	1223
gdcm::network::ULActionAR2	1224
gdcm::network::ULActionAR3	1226
gdcm::network::ULActionAR4	1227
gdcm::network::ULActionAR5	1228
gdcm::network::ULActionAR6	1230
gdcm::network::ULActionAR7	1231
gdcm::network::ULActionAR8	1232
gdcm::network::ULActionAR9	1234
gdcm::network::ULActionDT1	1235
gdcm::network::ULActionDT2	1236
gdcm::network::ULBasicCallback	
ULBasicCallback	1238
gdcm::network::ULConnection	
ULConnection	1240
gdcm::network::ULConnectionCallback	1245
gdcm::network::ULConnectionInfo	
ULConnectionInfo	1247
gdcm::network::ULConnectionManager	
ULConnectionManager	1249
gdcm::network::ULEvent	
ULEvent	1256
gdcm::network::ULTransitionTable	
ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates	1258
gdcm::network::ULWritingCallback	1259
gdcm::UNExplicitDataElement	
Class to read/write a <a href="#">DataElement</a> as UNExplicit Data <a href="#">Element</a>	1262
gdcm::UNExplicitImplicitDataElement	
Class to read/write a <a href="#">DataElement</a> as ExplicitImplicit Data <a href="#">Element</a>	1265
gdcm::Unpacker12Bits	
Pack/Unpack 12 bits pixel into 16bits	1269
gdcm::Usage	
Usage	1270
gdcm::UserEvent	1273
gdcm::network::UserInformation	
UserInformation	1274
gdcm::UUIDGenerator	
Class for generating unique UUID	1276
gdcm::Validate	
Validate class	1277
gdcm::Value	
Class to represent the value of a Data <a href="#">Element</a>	1279
gdcm::ValueIO< TDE, TSwap, TType >	
Class to dispatch template calls	1282
gdcm::MrProtocol::Vector3	1283
gdcm::Version	
Major/minor and build version	1284
gdcm::VL	
Value Length	1285
gdcm::VM	
Value Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n	1289

gdcm::VMToLength< T > . . . . .	1294
gdcm::VR	
VR class . . . . .	1294
gdcm::VR16ExplicitDataElement	
Class to read/write a <a href="#">DataElement</a> as Explicit Data <a href="#">Element</a> . . . . .	1301
gdcm::VRToEncoding< T > . . . . .	1304
gdcm::VRToType< T > . . . . .	1305
gdcm::VRVLSize< T > . . . . .	1305
gdcm::VRVLSize< 0 > . . . . .	1305
gdcm::VRVLSize< 1 > . . . . .	1306
vtkGDCMImageReader . . . . .	1307
vtkGDCMImageReader2 . . . . .	1320
vtkGDCMImageWriter . . . . .	1332
vtkGDCMMedicalImageProperties . . . . .	1340
vtkGDCMPolyDataReader . . . . .	1343
vtkGDCMPolyDataWriter . . . . .	1347
vtkGDCMTesting . . . . .	1351
vtkGDCMThreadedImageReader . . . . .	1354
vtkGDCMThreadedImageReader2 . . . . .	1359
vtkImageColorViewer . . . . .	1366
vtkImageMapToColors16 . . . . .	1378
vtkImageMapToWindowLevelColors2 . . . . .	1384
vtkImagePlanarComponentsToComponents . . . . .	1387
vtkImageRGBToYBR . . . . .	1389
vtkImageYBRToRGB . . . . .	1391
vtkLookupTable16 . . . . .	1393
vtkRTStructSetProperties . . . . .	1396
gdcm::Waveform	
Waveform class . . . . .	1406
gdcm::WLMFindQuery	
PatientRootQuery . . . . .	1406
gdcm::Writer	
Writer ala DOM (Document <a href="#">Object</a> Model) . . . . .	1411
gdcm::XMLDictReader	
Class for representing a <a href="#">XMLDictReader</a> . . . . .	1417
gdcm::XMLPrinter . . . . .	1420
gdcm::XMLPrivateDictReader	
Class for representing a <a href="#">XMLPrivateDictReader</a> . . . . .	1423

# Chapter 8

## File Index

### 8.1 File List

Here is a list of all files with brief descriptions:

<a href="#">gdc ASN1.h</a>	1427
<a href="#">gdc Base64.h</a>	1429
<a href="#">gdc BoxRegion.h</a>	1430
<a href="#">gdc ByteSwap.h</a>	1431
<a href="#">gdc CAPICryptoFactory.h</a>	1433
<a href="#">gdc CAPICryptographicMessageSyntax.h</a>	1434
<a href="#">gdc Command.h</a>	1436
<a href="#">gdc CryptoFactory.h</a>	1439
<a href="#">gdc CryptographicMessageSyntax.h</a>	1441
<a href="#">gdc DataEvent.h</a>	1443
<a href="#">gdc DeflateStream.h</a>	1445
<a href="#">gdc Directory.h</a>	1445
<a href="#">gdc DummyValueGenerator.h</a>	1448
<a href="#">gdc Event.h</a>	1449
<a href="#">gdc Exception.h</a>	1452
<a href="#">gdc Filename.h</a>	1454
<a href="#">gdc FileNameEvent.h</a>	1455
<a href="#">gdc FilenameGenerator.h</a>	1457
<a href="#">gdc LegacyMacro.h</a>	1458
<a href="#">gdc MD5.h</a>	1461
<a href="#">gdc Object.h</a>	1462
<a href="#">gdc OpenSSLCryptoFactory.h</a>	1464
<a href="#">gdc OpenSSLCryptographicMessageSyntax.h</a>	1466
<a href="#">gdc OpenSSLP7CryptoFactory.h</a>	1468
<a href="#">gdc OpenSSLP7CryptographicMessageSyntax.h</a>	1469
<a href="#">gdc ProgressEvent.h</a>	1471
<a href="#">gdc Region.h</a>	1473
<a href="#">gdc SHA1.h</a>	1475
<a href="#">gdc SmartPointer.h</a>	1476
<a href="#">gdc StaticAssert.h</a>	1478
<a href="#">gdc String.h</a>	1480

<a href="#">gdcmSubject.h</a>	1483
<a href="#">gdcmSwapCode.h</a>	1485
<a href="#">gdcmSwapper.h</a>	1486
<a href="#">gdcmSystem.h</a>	1489
<a href="#">gdcmTerminal.h</a>	1491
<a href="#">gdcmTestDriver.h</a>	1493
<a href="#">gdcmTesting.h</a>	1494
<a href="#">gdcmTrace.h</a>	1495
<a href="#">gdcmTypes.h</a>	1501
<a href="#">gdcmUnpacker12Bits.h</a>	1502
<a href="#">gdcmVersion.h</a>	1503
<a href="#">gdcmWin32.h</a>	1504
<a href="#">gdcmCSAHeaderDict.h</a>	1506
<a href="#">gdcmCSAHeaderDictEntry.h</a>	1509
<a href="#">gdcmDict.h</a>	1512
<a href="#">gdcmDictConverter.h</a>	1517
<a href="#">gdcmDictEntry.h</a>	1519
<a href="#">gdcmDicts.h</a>	1521
<a href="#">gdcmGlobal.h</a>	1524
<a href="#">gdcmGroupDict.h</a>	1526
<a href="#">gdcmSOPClassUIDToIOD.h</a>	1528
<a href="#">gdcmUIDs.h</a>	1529
<a href="#">gdcmAttribute.h</a>	1543
<a href="#">gdcmBasicOffsetTable.h</a>	1557
<a href="#">gdcmByteBuffer.h</a>	1560
<a href="#">gdcmByteSwapFilter.h</a>	1563
<a href="#">gdcmByteValue.h</a>	1564
<a href="#">gdcmCodeString.h</a>	1568
<a href="#">gdcmCP246ExplicitDataElement.h</a>	1570
<a href="#">gdcmCSAElement.h</a>	1571
<a href="#">gdcmCSAHeader.h</a>	1574
<a href="#">gdcmDataElement.h</a>	1577
<a href="#">gdcmDataSet.h</a>	1580
<a href="#">gdcmDataSetEvent.h</a>	1585
<a href="#">gdcmElement.h</a>	1587
<a href="#">gdcmExplicitDataElement.h</a>	1599
<a href="#">gdcmExplicitImplicitDataElement.h</a>	1601
<a href="#">gdcmFile.h</a>	1602
<a href="#">gdcmFileMetaInformation.h</a>	1604
<a href="#">gdcmFileSet.h</a>	1607
<a href="#">gdcmFragment.h</a>	1609
<a href="#">gdcmImplicitDataElement.h</a>	1613
<a href="#">gdcmItem.h</a>	1615
<a href="#">gdcmLO.h</a>	1621
<a href="#">gdcmMediaStorage.h</a>	1622
<a href="#">gdcmMrProtocol.h</a>	1626
<a href="#">gdcmParseException.h</a>	1628
<a href="#">gdcmParser.h</a>	1630
<a href="#">gdcmPDBElement.h</a>	1633
<a href="#">gdcmPDBHeader.h</a>	1635
<a href="#">gdcmPreamble.h</a>	1636
<a href="#">gdcmPrivateTag.h</a>	1639
<a href="#">gdcmReader.h</a>	1641
<a href="#">gdcmSequenceOfFragments.h</a>	1643

gdcmSequenceOfItems.h	1648
gdcmTag.h	1652
gdcmTagToVR.h	1657
gdcmTransferSyntax.h	1658
gdcmUNExplicitDataElement.h	1660
gdcmUNExplicitImplicitDataElement.h	1662
gdcmValue.h	1663
gdcmValueIO.h	1665
gdcmVL.h	1666
gdcmVM.h	1669
gdcmVR.h	1672
gdcmVR16ExplicitDataElement.h	1678
gdcmWriter.h	1680
gdcmDefinedTerms.h	1682
gdcmDefs.h	1683
gdcmEnumeratedValues.h	1685
gdcmIOD.h	1686
gdcmIODEntry.h	1689
gdcmIODs.h	1691
gdcmMacro.h	1694
gdcmMacroEntry.h	1697
gdcmMacros.h	1700
gdcmModule.h	1702
gdcmModuleEntry.h	1705
gdcmModules.h	1708
gdcmNestedModuleEntries.h	1710
gdcmPatient.h	1712
gdcmSeries.h	1713
gdcmStudy.h	1715
gdcmTable.h	1716
gdcmTableEntry.h	1718
gdcmTableReader.h	1720
gdcmType.h	1722
gdcmUsage.h	1724
gdcmXMLDictReader.h	1727
gdcmXMLPrivateDictReader.h	1728
gdcmAnonymizeEvent.h	1729
gdcmAnonymizer.h	1731
gdcmApplicationEntity.h	1733
gdcmAudioCodec.h	1735
gdcmBitmap.h	1736
gdcmBitmapToBitmapFilter.h	1740
gdcmCleaner.h	1741
gdcmCodec.h	1743
gdcmCoder.h	1744
gdcmConstCharWrapper.h	1746
gdcmCurve.h	1747
gdcmDataSetHelper.h	1749
gdcmDecoder.h	1750
gdcmDeltaEncodingCodec.h	1752
gdcmDICOMDIR.h	1753
gdcmDICOMDIRGenerator.h	1754
gdcmDictPrinter.h	1756
gdcmDirectionCosines.h	1757

<a href="#">gdcmDirectoryHelper.h</a>	1759
<a href="#">gdcmDPath.h</a>	1760
<a href="#">gdcmDumper.h</a>	1762
<a href="#">gdcmEmptyMaskGenerator.h</a>	1763
<a href="#">gdcmEncapsulatedDocument.h</a>	1765
<a href="#">gdcmEquipmentManufacturer.h</a>	1766
<a href="#">gdcmFiducials.h</a>	1767
<a href="#">gdcmFileAnonymizer.h</a>	1768
<a href="#">gdcmFileChangeTransferSyntax.h</a>	1770
<a href="#">gdcmFileDecompressLookupTable.h</a>	1772
<a href="#">gdcmFileDerivation.h</a>	1773
<a href="#">gdcmFileExplicitFilter.h</a>	1775
<a href="#">gdcmFileStreamer.h</a>	1776
<a href="#">gdcmIconImage.h</a>	1778
<a href="#">gdcmIconImageFilter.h</a>	1780
<a href="#">gdcmIconImageGenerator.h</a>	1781
<a href="#">gdcmImage.h</a>	1783
<a href="#">gdcmImageApplyLookupTable.h</a>	1785
<a href="#">gdcmImageChangePhotometricInterpretation.h</a>	1786
<a href="#">gdcmImageChangePlanarConfiguration.h</a>	1788
<a href="#">gdcmImageChangeTransferSyntax.h</a>	1790
<a href="#">gdcmImageCodec.h</a>	1791
<a href="#">gdcmImageConverter.h</a>	1795
<a href="#">gdcmImageFragmentSplitter.h</a>	1796
<a href="#">gdcmImageHelper.h</a>	1797
<a href="#">gdcmImageReader.h</a>	1800
<a href="#">gdcmImageRegionReader.h</a>	1801
<a href="#">gdcmImageToImageFilter.h</a>	1803
<a href="#">gdcmImageWriter.h</a>	1804
<a href="#">gdcmIPPSorter.h</a>	1806
<a href="#">gdcmJPEG12Codec.h</a>	1808
<a href="#">gdcmJPEG16Codec.h</a>	1809
<a href="#">gdcmJPEG2000Codec.h</a>	1811
<a href="#">gdcmJPEG8Codec.h</a>	1813
<a href="#">gdcmJPEGCodec.h</a>	1814
<a href="#">gdcmJPEGLSCodec.h</a>	1817
<a href="#">gdcmJSON.h</a>	1818
<a href="#">gdcmKAKADUCodec.h</a>	1820
<a href="#">gdcmLookupTable.h</a>	1821
<a href="#">gdcmMEC_MR3.h</a>	1824
<a href="#">gdcmMeshPrimitive.h</a>	1825
<a href="#">gdcmOrientation.h</a>	1828
<a href="#">gdcmOverlay.h</a>	1829
<a href="#">gdcmPDFCodec.h</a>	1832
<a href="#">gdcmPersonName.h</a>	1833
<a href="#">gdcmPGXCodec.h</a>	1835
<a href="#">gdcmPhotometricInterpretation.h</a>	1836
<a href="#">gdcmPixelFormat.h</a>	1838
<a href="#">gdcmPixmap.h</a>	1842
<a href="#">gdcmPixmapReader.h</a>	1844
<a href="#">gdcmPixmapToPixmapFilter.h</a>	1847
<a href="#">gdcmPixmapWriter.h</a>	1848
<a href="#">gdcmPNMCodec.h</a>	1850
<a href="#">gdcmPrinter.h</a>	1851



gdcmPVRGCodec.h	1854
gdcmRAWCodec.h	1855
gdcmRescaler.h	1857
gdcmRLECodec.h	1859
gdcmScanner.h	1860
gdcmScanner2.h	1863
gdcmSegment.h	1866
gdcmSegmentedPaletteColorLookupTable.h	1870
gdcmSegmentHelper.h	1871
gdcmSegmentReader.h	1873
gdcmSegmentWriter.h	1875
gdcmSerieHelper.h	1877
gdcmSimpleSubjectWatcher.h	1880
gdcmSorter.h	1882
gdcmSpacing.h	1885
gdcmSpectroscopy.h	1886
gdcmSplitMosaicFilter.h	1887
gdcmStreamImageReader.h	1889
gdcmStreamImageWriter.h	1891
gdcmStrictScanner.h	1893
gdcmStrictScanner2.h	1895
gdcmStringFilter.h	1898
gdcmSurface.h	1900
gdcmSurfaceHelper.h	1904
gdcmSurfaceReader.h	1907
gdcmSurfaceWriter.h	1909
gdcmTagPath.h	1910
gdcmUIDGenerator.h	1912
gdcmUUIDGenerator.h	1914
gdcmValidate.h	1915
gdcmWaveform.h	1916
gdcmXMLPrinter.h	1917
gdcmAAbortPDU.h	1920
gdcmAAssociateACPDU.h	1921
gdcmAAssociateRJPDU.h	1923
gdcmAAssociateRQPDU.h	1925
gdcmAbstractSyntax.h	1927
gdcmApplicationContext.h	1929
gdcmAReleaseRPPDU.h	1931
gdcmAReleaseRQPDU.h	1932
gdcmARTIMTimer.h	1934
gdcmAsynchronousOperationsWindowSub.h	1936
gdcmBaseCompositeMessage.h	1937
gdcmBaseNormalizedMessage.h	1939
gdcmBasePDU.h	1940
gdcmBaseQuery.h	1942
gdcmBaseRootQuery.h	1944
gdcmCEchoMessages.h	1946
gdcmCFindMessages.h	1947
gdcmCMoveMessages.h	1949
gdcmCommandDataSet.h	1950
gdcmCompositeMessageFactory.h	1952
gdcmCompositeNetworkFunctions.h	1953
gdcmCStoreMessages.h	1955

gdcmDIMSE.h	1957
gdcmFindPatientRootQuery.h	1959
gdcmFindStudyRootQuery.h	1961
gdcmImplementationClassUIDSub.h	1962
gdcmImplementationUIDSub.h	1964
gdcmImplementationVersionNameSub.h	1965
gdcmMaximumLengthSub.h	1967
gdcmModalityPerformedProcedureStepCreateQuery.h	1969
gdcmModalityPerformedProcedureStepSetQuery.h	1970
gdcmMovePatientRootQuery.h	1971
gdcmMoveStudyRootQuery.h	1973
gdcmNActionMessages.h	1974
gdcmNCreateMessages.h	1975
gdcmNDeleteMessages.h	1977
gdcmNetworkEvents.h	1978
gdcmNetworkStateID.h	1980
gdcmNEventReportMessages.h	1982
gdcmNGetMessages.h	1983
gdcmNormalizedMessageFactory.h	1984
gdcmNormalizedNetworkFunctions.h	1986
gdcmNSetMessages.h	1988
gdcmPDataTFPDU.h	1989
gdcmPDUFactory.h	1991
gdcmPresentationContext.h	1992
gdcmPresentationContextAC.h	1994
gdcmPresentationContextGenerator.h	1996
gdcmPresentationContextRQ.h	1998
gdcmPresentationDataValue.h	2000
gdcmQueryBase.h	2002
gdcmQueryFactory.h	2005
gdcmQueryImage.h	2006
gdcmQueryPatient.h	2008
gdcmQuerySeries.h	2010
gdcmQueryStudy.h	2011
gdcmRoleSelectionSub.h	2013
gdcmServiceClassApplicationInformation.h	2014
gdcmServiceClassUser.h	2016
gdcmSOPClassExtendedNegociationSub.h	2018
gdcmTransferSyntaxSub.h	2019
gdcmULAction.h	2021
gdcmULActionAA.h	2023
gdcmULActionAE.h	2025
gdcmULActionAR.h	2027
gdcmULActionDT.h	2030
gdcmULBasicCallback.h	2031
gdcmULConnection.h	2032
gdcmULConnectionCallback.h	2035
gdcmULConnectionInfo.h	2036
gdcmULConnectionManager.h	2038
gdcmULEvent.h	2041
gdcmULTransitionTable.h	2043
gdcmULWritingCallback.h	2046
gdcmUserInformation.h	2047
gdcmWLMFindQuery.h	2049

<a href="#">vtkGDCMImageReader.h</a>	2050
<a href="#">vtkGDCMImageReader2.h</a>	2056
<a href="#">vtkGDCMImageWriter.h</a>	2061
<a href="#">vtkGDCMMedicalImageProperties.h</a>	2064
<a href="#">vtkGDCMPolyDataReader.h</a>	2069
<a href="#">vtkGDCMPolyDataWriter.h</a>	2071
<a href="#">vtkGDCMTesting.h</a>	2073
<a href="#">vtkGDCMThreadedImageReader.h</a>	2074
<a href="#">vtkGDCMThreadedImageReader2.h</a>	2076
<a href="#">vtkImageColorViewer.h</a>	2078
<a href="#">vtkImageMapToColors16.h</a>	2082
<a href="#">vtkImageMapToWindowLevelColors2.h</a>	2084
<a href="#">vtkImagePlanarComponentsToComponents.h</a>	2086
<a href="#">vtkImageRGBToYBR.h</a>	2088
<a href="#">vtkImageYBRToRGB.h</a>	2089
<a href="#">vtkLookupTable16.h</a>	2090
<a href="#">vtkRTStructSetProperties.h</a>	2092
<a href="#">gdcmPythonFilter.h</a>	2094



## Chapter 9

# Namespace Documentation

### 9.1 gdcM Namespace Reference

#### Namespaces

- namespace [network](#)
- namespace [SegmentHelper](#)
- namespace [terminal](#)

*Class for Terminal.*

#### Classes

- class [AbortEvent](#)
- class [AnonymizeEvent](#)  
*AnonymizeEvent.*
- class [Anonymizer](#)  
*Anonymizer.*
- class [AnyEvent](#)
- class [ApplicationEntity](#)  
*ApplicationEntity.*
- class [ASN1](#)  
*Class for ASN1.*
- class [Attribute](#)  
*Attribute class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary.*
- class [Attribute< Group, Element, TVR, VM::VM1 >](#)
- class [Attribute< Group, Element, TVR, VM::VM1\\_3 >](#)
- class [Attribute< Group, Element, TVR, VM::VM1\\_8 >](#)
- class [Attribute< Group, Element, TVR, VM::VM1\\_n >](#)
- class [Attribute< Group, Element, TVR, VM::VM2\\_2n >](#)
- class [Attribute< Group, Element, TVR, VM::VM2\\_n >](#)
- class [Attribute< Group, Element, TVR, VM::VM3\\_3n >](#)
- class [Attribute< Group, Element, TVR, VM::VM3\\_n >](#)

- class [AudioCodec](#)  
*AudioCodec.*
- class [Base64](#)  
*Class for Base64.*
- class [BaseQuery](#)  
*BaseQuery.*
- class [BaseRootQuery](#)  
*BaseRootQuery.*
- class [BasicOffsetTable](#)  
*Class to represent a BasicOffsetTable.*
- class [Bitmap](#)  
*Bitmap class.*
- class [BitmapToBitmapFilter](#)  
*BitmapToBitmapFilter class.*
- class [BoxRegion](#)  
*Class for manipulation box region.*
- class [ByteBuffer](#)  
*ByteBuffer.*
- class [ByteSwap](#)  
*ByteSwap.*
- class [ByteSwapFilter](#)  
*ByteSwapFilter.*
- class [ByteValue](#)  
*Class to represent binary value (array of bytes)*
- class [CAPICryptoFactory](#)
- class [CAPICryptographicMessageSyntax](#)
- class [Cleaner](#)  
*Cleaner.*
- class [Codec](#)  
*Codec class.*
- class [Coder](#)  
*Coder.*
- class [CodeString](#)  
*CodeString.*
- class [Command](#)  
*Command superclass for callback/observer methods.*
- class [CommandDataSet](#)  
*Class to represent a Command DataSet.*
- class [CompositeNetworkFunctions](#)  
*Composite Network Functions.*
- class [ConstCharWrapper](#)  
*Do not use me.*
- class [CP246ExplicitDataElement](#)  
*Class to read/write a DataElement as CP246Explicit Data Element.*
- class [CryptoFactory](#)  
*Class to do handle the crypto factory.*
- class [CryptographicMessageSyntax](#)

- class [CSAElement](#)  
*Class to represent a CSA [Element](#).*
- class [CSAHeader](#)  
*Class for [CSAHeader](#).*
- class [CSAHeaderDict](#)  
*Class to represent a map of [CSAHeaderDictEntry](#).*
- class [CSAHeaderDictEntry](#)  
*Class to represent an Entry in the [Dict](#).*
- class [CSAHeaderDictException](#)
- class [Curve](#)  
*[Curve](#) class to handle element 50xx,3000 [Curve](#) Data.*
- class [DataElement](#)  
*Class to represent a Data [Element](#) either Implicit or Explicit.*
- class [DataElementException](#)
- class [DataEvent](#)  
*[DataEvent](#).*
- class [DataSet](#)  
*Class to represent a Data Set (which contains Data Elements)*
- class [DataSetEvent](#)  
*[DataSetEvent](#).*
- class [DataSetHelper](#)  
*[DataSetHelper](#) (internal class, not intended for user level)*
- class [Decoder](#)  
*[Decoder](#).*
- class [DefinedTerms](#)  
*Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data [Element](#) with Defined Terms that does not contain a [Value](#) equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation [Type](#) ID (4008,0210) is an example of a Data [Element](#) having Defined Terms. It is defined to have a [Value](#) that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data [Element](#) has Defined Terms other Interpretation [Type](#) IDs may be defined by the implementor.*
- class [Defs](#)  
*FIXME I do not like the name 'Defs'.*
- class [DeltaEncodingCodec](#)  
*[DeltaEncodingCodec](#) compression used by some private vendor.*
- class [DICOMDIR](#)  
*[DICOMDIR](#) class.*
- class [DICOMDIRGenerator](#)  
*[DICOMDIRGenerator](#) class.*
- class [Dict](#)  
*Class to represent a map of [DictEntry](#).*
- class [DictConverter](#)  
*Class to convert a .dic file into something else:*
- class [DictEntry](#)  
*Class to represent an Entry in the [Dict](#).*
- class [DictPrinter](#)  
*[DictPrinter](#) class.*

- class [Dicts](#)  
*Class to manipulate the sum of knowledge (all the dict user load)*
- class [DirectionCosines](#)  
*class to handle [DirectionCosines](#)*
- class [Directory](#)  
*Class for manipulation directories.*
- class [DirectoryHelper](#)  
*[DirectoryHelper](#).*
- class [DPath](#)  
*class to handle a DICOM path While supp 118 did introduced a notion of XPath for XML Native model this convention is too XML-centric. Instead prefer DCMTK style notation <https://groups.google.com/g/comp.protocols>.↔  
[dicom/c/IyIH0IOBMPA](#)*
- class [DummyValueGenerator](#)  
*Class for generating dummy value.*
- class [Dumper](#)  
*[Codec](#) class.*
- class [Element](#)  
*[Element](#) class.*
- class [Element< TVR, VM::VM1\\_2 >](#)
- class [Element< TVR, VM::VM1\\_n >](#)
- class [Element< TVR, VM::VM2\\_2n >](#)
- class [Element< TVR, VM::VM2\\_n >](#)
- class [Element< TVR, VM::VM3\\_3n >](#)
- class [Element< TVR, VM::VM3\\_4 >](#)
- class [Element< TVR, VM::VM3\\_n >](#)
- class [Element< VR::AS, VM::VM5 >](#)
- class [Element< VR::OB, VM::VM1 >](#)
- class [Element< VR::OW, VM::VM1 >](#)
- class [ElementDisableCombinations](#)  
*A class which is used to produce compile errors for an invalid combination of template parameters.*
- class [ElementDisableCombinations< VR::OB, VM::VM1\\_n >](#)
- class [ElementDisableCombinations< VR::OW, VM::VM1\\_n >](#)
- class [EmptyMaskGenerator](#)  
*[EmptyMaskGenerator](#) Main class to generate a Empty Mask [Series](#) from an input [Series](#). This class takes an input folder and generates a series of DICOM files in the specified output directory. This class handles multiples DICOM [Series](#) within the same input directory.*
- class [EncapsulatedDocument](#)  
*[EncapsulatedDocument](#).*
- class [EncodingImplementation](#)  
*[EncodingImplementation](#).*
- class [EncodingImplementation< VR::VRASCII >](#)
- class [EncodingImplementation< VR::VRBINARY >](#)
- class [EndEvent](#)
- class [EnumeratedValues](#)  
*[Element](#). A Data [Element](#) with Enumerated Values that does not have a [Value](#) equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:*
- class [EquipmentManufacturer](#)
- class [Event](#)  
*superclass for callback/observer methods*



- class [Exception](#)  
*Exception.*
- class [ExitEvent](#)
- class [ExplicitDataElement](#)  
*Class to read/write a [DataElement](#) as Explicit Data [Element](#).*
- class [ExplicitImplicitDataElement](#)  
*Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#).*
- class [Fiducials](#)  
*Fiducials.*
- class [File](#)  
*a DICOM File*
- class [FileAnonymizer](#)  
*FileAnonymizer.*
- class [FileChangeTransferSyntax](#)  
*FileChangeTransferSyntax.*
- class [FileDecompressLookupTable](#)  
*FileDecompressLookupTable class.*
- class [FileDerivation](#)  
*FileDerivation class.*
- class [FileExplicitFilter](#)  
*FileExplicitFilter class.*
- class [FileMetaInformation](#)  
*Class to represent a [File](#) Meta Information.*
- class [Filename](#)  
*Class to manipulate file name's.*
- class [FileNameEvent](#)  
*FileNameEvent.*
- class [FilenameGenerator](#)  
*FilenameGenerator.*
- class [FileSet](#)
- class [FileStreamer](#)  
*FileStreamer.*
- class [FileWithName](#)  
*FileWithName.*
- class [FindPatientRootQuery](#)  
*PatientRootQuery.*
- class [FindStudyRootQuery](#)  
*FindStudyRootQuery.*
- class [Fragment](#)  
*Class to represent a [Fragment](#).*
- class [Global](#)  
*Global.*
- class [GroupDict](#)  
*Class to represent the mapping from group number to its abbreviation and name.*
- class [IconImageFilter](#)  
*IconImageFilter.*
- class [IconImageGenerator](#)

- [\*IconImageGenerator.\*](#)
- struct [ignore\\_char](#)
- class [Image](#)
  - [\*Image.\*](#)
- class [ImageApplyLookupTable](#)
  - [\*ImageApplyLookupTable\* class.](#)
- class [ImageChangePhotometricInterpretation](#)
  - [\*ImageChangePhotometricInterpretation\* class.](#)
- class [ImageChangePlanarConfiguration](#)
  - [\*ImageChangePlanarConfiguration\* class.](#)
- class [ImageChangeTransferSyntax](#)
  - [\*ImageChangeTransferSyntax\* class.](#)
- class [ImageCodec](#)
  - [\*ImageCodec.\*](#)
- class [ImageConverter](#)
  - [\*Image\* Converter.](#)
- class [ImageFragmentSplitter](#)
  - [\*ImageFragmentSplitter\* class.](#)
- class [ImageHelper](#)
  - [\*ImageHelper\* \(internal class, not intended for user level\)](#)
- class [ImageReader](#)
  - [\*ImageReader.\*](#)
- class [ImageRegionReader](#)
  - [\*ImageRegionReader.\*](#)
- class [ImageToImageFilter](#)
  - [\*ImageToImageFilter\* class.](#)
- class [ImageWriter](#)
  - [\*ImageWriter.\*](#)
- class [ImplicitDataElement](#)
  - [\*Class to represent an Implicit VR Data Element.\*](#)
- class [InitializeEvent](#)
- class [IOD](#)
  - [\*Class for representing a IOD.\*](#)
- class [IODEntry](#)
  - [\*Class for representing a IODEntry.\*](#)
- class [IODs](#)
  - [\*Class for representing a IODs.\*](#)
- class [IPPSorter](#)
  - [\*IPPSorter.\*](#)
- class [Item](#)
  - [\*Class to represent an Item.\*](#)
- class [IterationEvent](#)
- class [JPEG12Codec](#)
  - [\*Class to do JPEG 12bits \(lossy & lossless\)\*](#)
- class [JPEG16Codec](#)
  - [\*Class to do JPEG 16bits \(lossless\)\*](#)
- class [JPEG2000Codec](#)

- Class to do JPEG 2000.*
- class [JPEG8Codec](#)
  - Class to do JPEG 8bits (lossy & lossless)*
- class [JPEGCodec](#)
  - JPEG codec.*
- class [JPEGLSCodec](#)
  - JPEG-LS.*
- class [JSON](#)
- class [KAKADUCodec](#)
  - KAKADUCodec.*
- class [LO](#)
  - LO.*
- class [LookupTable](#)
  - LookupTable class.*
- class [Macro](#)
  - Class for representing a [Macro](#).*
- class [Macros](#)
  - Class for representing a [Modules](#).*
- class [MD5](#)
  - Class for [MD5](#).*
- class [MEC\\_MR3](#)
  - Class for [MEC\\_MR3](#).*
- class [MediaStorage](#)
  - MediaStorage.*
- class [MemberCommand](#)
  - Command subclass that calls a pointer to a member function.*
- class [MeshPrimitive](#)
  - This class defines surface mesh primitives.*
- class [ModalityPerformedProcedureStepCreateQuery](#)
  - ModalityPerformedProcedureStepCreateQuery.*
- class [ModalityPerformedProcedureStepSetQuery](#)
  - ModalityPerformedProcedureStepSetQuery.*
- class [ModifiedEvent](#)
- class [Module](#)
  - Class for representing a [Module](#).*
- class [ModuleEntry](#)
  - Class for representing a [ModuleEntry](#).*
- class [Modules](#)
  - Class for representing a [Modules](#).*
- class [MovePatientRootQuery](#)
  - MovePatientRootQuery.*
- class [MoveStudyRootQuery](#)
  - MoveStudyRootQuery.*
- class [MrProtocol](#)
  - Class for [MrProtocol](#).*
- class [NestedModuleEntries](#)
  - Class for representing a [NestedModuleEntries](#).*

- class [NoEvent](#)
- class [NormalizedNetworkFunctions](#)  
*Normalized Network Functions.*
- class [Object](#)  
*Object.*
- class [OpenSSLCryptoFactory](#)
- class [OpenSSLCryptographicMessageSyntax](#)
- class [OpenSSLP7CryptoFactory](#)
- class [OpenSSLP7CryptographicMessageSyntax](#)
- class [Orientation](#)  
*class to handle [Orientation](#)*
- class [Overlay](#)  
*Overlay class.*
- class [ParseException](#)  
*[ParseException](#) Standard exception handling object.*
- class [Parser](#)  
*[Parser](#) ala XML\_Parser from expat (SAX)*
- class [Patient](#)  
*See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.*
- class [PDBElement](#)  
*Class to represent a PDB [Element](#).*
- class [PDBHeader](#)  
*Class for [PDBHeader](#).*
- class [PDFCodec](#)  
*[PDFCodec](#) class.*
- class [PersonName](#)  
*[PersonName](#) class.*
- class [PGXCodec](#)  
*Class to do PGX.*
- class [PhotometricInterpretation](#)  
*Class to represent an [PhotometricInterpretation](#).*
- class [PixelFormat](#)  
*[PixelFormat](#).*
- class [Pixmap](#)  
*[Pixmap](#) class.*
- class [PixmapReader](#)  
*[PixmapReader](#).*
- class [PixmapToPixmapFilter](#)  
*[PixmapToPixmapFilter](#) class.*
- class [PixmapWriter](#)  
*[PixmapWriter](#).*
- class [PNMCodec](#)  
*Class to do PNM.*
- class [Preamble](#)  
*DICOM [Preamble](#) (Part 10)*
- class [PresentationContext](#)  
*[PresentationContext](#).*

- class [PresentationContextGenerator](#)  
*PresentationContextGenerator.*
- class [Printer](#)  
*Printer class.*
- class [PrivateDict](#)  
*Private Dict.*
- class [PrivateTag](#)  
*Class to represent a Private DICOM Data [Element](#) ([Attribute](#)) [Tag](#) (Group, [Element](#), Owner)*
- class [ProgressEvent](#)  
*ProgressEvent.*
- class [PVRGCodec](#)  
*PVRGCodec.*
- class [PythonFilter](#)  
*PythonFilter [PythonFilter](#) is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a [DataElement](#) into a string, typically this is a nice feature to have for wrapped language.*
- class [QueryBase](#)  
*QueryBase.*
- class [QueryFactory](#)  
*QueryFactory.h.*
- class [QueryImage](#)  
*QueryImage.*
- class [QueryPatient](#)  
*QueryPatient.*
- class [QuerySeries](#)  
*QuerySeries.*
- class [QueryStudy](#)  
*QueryStudy.h.*
- class [RAWCodec](#)  
*RAWCodec class.*
- class [Reader](#)  
*Reader ala DOM (Document [Object](#) Model)*
- struct [RealWorldValueMappingContent](#)
- class [Region](#)  
*Class for manipulation region.*
- class [Rescaler](#)  
*Rescale class.*
- class [RLECodec](#)  
*Class to do RLE.*
- class [Scanner](#)  
*Scanner.*
- class [Scanner2](#)  
*Scanner2.*
- class [Segment](#)  
*This class defines a segment.*
- class [SegmentedPaletteColorLookupTable](#)  
*SegmentedPaletteColorLookupTable class.*
- class [SegmentReader](#)

*This class defines a segment reader.*

- class [SegmentWriter](#)

*This class defines a segment writer.*

- class [SequenceOfFragments](#)

*Class to represent a Sequence Of Fragments.*

- class [SequenceOfItems](#)

*Class to represent a Sequence Of Items.*

- class [SerieHelper](#)

*[SerieHelper](#) DO NOT USE this class, it is only a temporary solution for ITK migration from GDCM 1.x to GDCM 2.x It will disappear soon, you've been warned.*

- class [Series](#)

*Series.*

- class [ServiceClassUser](#)

*ServiceClassUser.*

- class [SHA1](#)

*Class for [SHA1](#).*

- class [SimpleMemberCommand](#)

*Command subclass that calls a pointer to a member function.*

- class [SimpleSubjectWatcher](#)

*SimpleSubjectWatcher.*

- class [SmartPointer](#)

*Class for Smart Pointer.*

- class [SOPClassUIDToIOD](#)

*Class convert a class SOP Class UID into [IOD](#).*

- class [Sorter](#)

*Sorter.*

- class [Spacing](#)

*Class for [Spacing](#).*

- class [Spectroscopy](#)

*Spectroscopy class.*

- class [SplitMosaicFilter](#)

*SplitMosaicFilter class.*

- class [StartEvent](#)

- struct [static\\_assert\\_test](#)

- struct [STATIC\\_ASSERTION\\_FAILURE](#)

- struct [STATIC\\_ASSERTION\\_FAILURE< true >](#)

- class [StreamImageReader](#)

*StreamImageReader.*

- class [StreamImageWriter](#)

*StreamImageReader.*

- class [StrictScanner](#)

*StrictScanner.*

- class [StrictScanner2](#)

*StrictScanner2.*

- class [String](#)

*String.*

- class [StringFilter](#)

- StringFilter.*
- class [Study](#)
  - Study.*
- class [Subject](#)
  - Subject.*
- class [Surface](#)
  - This class defines a SURFACE IE.*
- class [SurfaceHelper](#)
  - SurfaceHelper.*
- class [SurfaceReader](#)
  - This class defines a SURFACE IE reader.*
- class [SurfaceWriter](#)
  - This class defines a SURFACE IE writer.*
- class [SwapCode](#)
  - SwapCode representation.*
- class [SwapperDoOp](#)
- class [SwapperNoOp](#)
- class [System](#)
  - Class to do system operation.*
- class [Table](#)
  - Table.*
- class [TableEntry](#)
  - TableEntry.*
- class [TableReader](#)
  - Class for representing a [TableReader](#).*
- class [Tag](#)
  - Class to represent a DICOM Data [Element](#) ([Attribute](#)) [Tag](#) (Group, [Element](#)).*
- class [TagPath](#)
  - class to handle a path of tag.*
- class [Testing](#)
  - class for testing*
- class [Trace](#)
  - Trace.*
- class [TransferSyntax](#)
  - Class to manipulate Transfer Syntax.*
- class [Type](#)
  - Type.*
- struct [UI](#)
- class [UIDGenerator](#)
  - Class for generating unique UID.*
- class [UIDs](#)
  - all known uids*
- class [UNExplicitDataElement](#)
  - Class to read/write a [DataElement](#) as UNExplicit Data [Element](#).*
- class [UNExplicitImplicitDataElement](#)
  - Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#).*
- class [Unpacker12Bits](#)

- Pack/Unpack 12 bits pixel into 16bits.*
- class [Usage](#)
  - Usage.*
- class [UserEvent](#)
- class [UUIDGenerator](#)
  - Class for generating unique UUID.*
- class [Validate](#)
  - Validate class.*
- class [Value](#)
  - Class to represent the value of a Data [Element](#).*
- class [ValueIO](#)
  - Class to dispatch template calls.*
- class [Version](#)
  - major/minor and build version*
- class [VL](#)
  - Value Length.*
- class [VM](#)
  - Value Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n.*
- struct [VMToLength](#)
- class [VR](#)
  - VR class.*
- class [VR16ExplicitDataElement](#)
  - Class to read/write a [DataElement](#) as Explicit Data [Element](#).*
- struct [VRToEncoding](#)
- struct [VRToType](#)
- class [VRVLSize](#)
- class [VRVLSize< 0 >](#)
- class [VRVLSize< 1 >](#)
- class [Waveform](#)
  - Waveform class.*
- class [WLMFindQuery](#)
  - PatientRootQuery.*
- class [Writer](#)
  - Writer ala DOM (Document [Object Model](#))*
- class [XMLDictReader](#)
  - Class for representing a [XMLDictReader](#).*
- class [XMLPrinter](#)
- class [XMLPrivateDictReader](#)
  - Class for representing a [XMLPrivateDictReader](#).*

## Typedefs

- typedef [String](#)<"\", 16 > [AECComp](#)
- typedef [String](#)<"\", 64 > [ASComp](#)
- typedef bool(\* [BOOL\\_FUNCTION\\_PFILE\\_PFILE\\_POINTER](#)) ([File](#) \*, [File](#) \*)
- typedef [String](#)<"\", 16 > [CSCComp](#)
- typedef [String](#)<"\", 64 > [DACComp](#)



- typedef [String](#)<"\", 64 > [DTComp](#)
- typedef [std::vector](#)< [SmartPointer](#)< [FileWithName](#) > > [FileList](#)
- typedef [Bitmap](#) [IconImage](#)
- typedef [String](#)<"\", 64 > [LOComp](#)
- typedef [String](#)<"\", 64 > [LTComp](#)
- typedef [ModuleEntry](#) [MacroEntry](#)
- typedef [NestedModuleEntries](#) [NestedMacroEntries](#)
- typedef [String](#)<"\", 64 > [PNComp](#)
- typedef [String](#)<"\", 64 > [SHComp](#)
- typedef [String](#)<"\", 64 > [STComp](#)
- typedef [String](#)<"\", 16 > [TMComp](#)
- typedef [String](#)<"\", 4294967294 > [UCComp](#)
- typedef [String](#)<"\", 64, 0 > [UIComp](#)
- typedef [String](#)<"\", 4294967294 > [URComp](#)
- typedef [String](#)<"\", 64 > [UTComp](#)

## Enumerations

- enum [CompOperators](#) {  
[GDCM\\_EQUAL](#) = 0 ,  
[GDCM\\_DIFFERENT](#) ,  
[GDCM\\_GREATER](#) ,  
[GDCM\\_GREATEROREQUAL](#) ,  
[GDCM\\_LESS](#) ,  
[GDCM\\_LESSEOREQUAL](#) }
- enum [ECharSet](#) {  
[eLatin1](#) = 0 ,  
[eLatin2](#) ,  
[eLatin3](#) ,  
[eLatin4](#) ,  
[eCyrillic](#) ,  
[eArabic](#) ,  
[eGreek](#) ,  
[eHebrew](#) ,  
[eLatin5](#) ,  
[eJapanese](#) ,  
[eThai](#) ,  
[eJapaneseKanjiMultibyte](#) ,  
[eJapaneseSupplementaryKanjiMultibyte](#) ,  
[eKoreanHangulHanjaMultibyte](#) ,  
[eUTF8](#) ,  
[eGB18030](#) }
- enum [ENQueryType](#) {  
[eCreateMMPS](#) = 0 ,  
[eSetMMPS](#) }
- enum [EQueryLevel](#) {  
[ePatient](#) = 0 ,  
[eStudy](#) = 1 ,  
[eSeries](#) = 2 ,  
[eImage](#) = 3 }

- enum [EQueryType](#) {  
    [eFind](#) = 0 ,  
    [eMove](#) ,  
    [eWLMFind](#) }
- enum [ERootType](#) {  
    [ePatientRootType](#) ,  
    [eStudyRootType](#) }
- enum [LodModeType](#) {  
    [LD\\_ALL](#) = 0x00000000 ,  
    [LD\\_NOSEQ](#) = 0x00000001 ,  
    [LD\\_NOSHADOW](#) = 0x00000002 ,  
    [LD\\_NOSHADOWSEQ](#) = 0x00000004 }

## Functions

- static int [add1](#) (char \*buf, int n)
- [ignore\\_char](#) const [backslash](#) ("\\")
- template<typename T >  
    static T [Clamp](#) (int v)
- static void [clean](#) (char \*mant)
- static int [doround](#) (char \*buf, unsigned int n)
- [VR::VRType](#) [GetVRFromTag](#) ([Tag](#) const &tag)
- bool [operator!=](#) (const [CodeString](#) &ref, const [CodeString](#) &cs)
- bool [operator!=](#) (const [DataElement](#) &lhs, const [DataElement](#) &rhs)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [GroupDict](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [IOD](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [IODEntry](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [IODs](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [Macro](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [Macros](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [MediaStorage](#) &ms)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [Module](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [ModuleEntry](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [Modules](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [NestedModuleEntries](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [Tag](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [TransferSyntax](#) &ts)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [Type](#) &val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [UI](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [UIDs](#) &uid)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [Usage](#) &val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [VM](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [VR](#) &val)
- std::ostream & [operator<<](#) (std::ostream &os, const [BasicOffsetTable](#) &val)
- std::ostream & [operator<<](#) (std::ostream &os, const [CodeString](#) &str)
- std::ostream & [operator<<](#) (std::ostream &os, const [CommandDataSet](#) &val)
- std::ostream & [operator<<](#) (std::ostream &os, const [CSAElement](#) &val)
- std::ostream & [operator<<](#) (std::ostream &os, const [CSAHeader](#) &d)
- std::ostream & [operator<<](#) (std::ostream &os, const [CSAHeaderDict](#) &val)
- std::ostream & [operator<<](#) (std::ostream &os, const [CSAHeaderDictEntry](#) &val)

- `std::ostream & operator<< (std::ostream &os, const DataElement &val)`
- `std::ostream & operator<< (std::ostream &os, const DataSet &val)`
- `std::ostream & operator<< (std::ostream &os, const Dict &val)`
- `std::ostream & operator<< (std::ostream &os, const DictEntry &val)`
- `std::ostream & operator<< (std::ostream &os, const Dicts &d)`
- `std::ostream & operator<< (std::ostream &os, const Directory &d)`
- `std::ostream & operator<< (std::ostream &os, const DPath &val)`
- `std::ostream & operator<< (std::ostream &os, const Event &e)`

*Generic inserter operator for [Event](#) and its subclasses.*

- `std::ostream & operator<< (std::ostream &os, const File &val)`
- `std::ostream & operator<< (std::ostream &os, const FileMetalInformation &val)`
- `std::ostream & operator<< (std::ostream &os, const FileSet &f)`
- `std::ostream & operator<< (std::ostream &os, const Fragment &val)`
- `std::ostream & operator<< (std::ostream &os, const Global &g)`
- `std::ostream & operator<< (std::ostream &os, const Item &val)`
- `std::ostream & operator<< (std::ostream &os, const MrProtocol &d)`
- `std::ostream & operator<< (std::ostream &os, const Object &obj)`
- `std::ostream & operator<< (std::ostream &os, const Orientation &o)`
- `std::ostream & operator<< (std::ostream &os, const PDElement &val)`
- `std::ostream & operator<< (std::ostream &os, const PDBHeader &d)`
- `std::ostream & operator<< (std::ostream &os, const PhotometricInterpretation &val)`
- `std::ostream & operator<< (std::ostream &os, const PixelFormat &pf)`
- `std::ostream & operator<< (std::ostream &os, const Preamble &val)`
- `std::ostream & operator<< (std::ostream &os, const PrivateDict &val)`
- `std::ostream & operator<< (std::ostream &os, const PrivateTag &val)`
- `std::ostream & operator<< (std::ostream &os, const Region &r)`
- `std::ostream & operator<< (std::ostream &os, const Scanner &s)`
- `std::ostream & operator<< (std::ostream &os, const Scanner2 &s)`
- `std::ostream & operator<< (std::ostream &os, const Sorter &s)`
- `std::ostream & operator<< (std::ostream &os, const StrictScanner &s)`
- `std::ostream & operator<< (std::ostream &os, const StrictScanner2 &s)`
- `std::ostream & operator<< (std::ostream &os, const SwapCode &sc)`
- `std::ostream & operator<< (std::ostream &os, const Version &v)`
- `std::ostream & operator<< (std::ostream &os, const VL &val)`
- `bool operator== (const CodeString &ref, const CodeString &cs)`
- `std::istream & operator>> (std::istream &_is, Tag &_val)`
- `std::istream & operator>> (std::istream &in, ignore\_char const &ic)`
- `template<char TDelimiter, unsigned int TMaxLength, char TPadChar>  
std::istream & operator>> (std::istream &is, String< TDelimiter, TMaxLength, TPadChar > &ms)`
- `template<typename T >  
static int Round (T x)`
- `static int roundat (char *buf, size_t bufLen, unsigned int i, int iexp)`
- `template<typename Float >  
static void x16printf (char *buf, int size, Float f)`

## Variables

- static [Global GlobalInstance](#)

### 9.1.1 Detailed Description

This header defines the classes for the AA Actions, Association Abort Related Actions ([Table 9-9](#) of ps 3.8-2009).

Since each class is essentially a placeholder for a function pointer, I'm breaking with having each class have its own file for the sake of brevity of the number of files.

This header defines the classes for the AE Actions, Association Establishment Related Actions ([Table 9-6](#) of ps 3.8-2009).

Since each class is essentially a placeholder for a function pointer, I'm breaking with having each class have its own file for the sake of brevity of the number of files.

This header defines the classes for the AR Actions, Association Release Related Actions ([Table 9-8](#) of ps 3.8-2009).

Since each class is essentially a placeholder for a function pointer, I'm breaking with having each class have its own file for the sake of brevity of the number of files.

This header defines the classes for the DT Actions, Data Transfer Related Actions ([Table 9-8](#) of ps 3.8-2009).

Since each class is essentially a placeholder for a function pointer, I'm breaking with having each class have its own file for the sake of brevity of the number of files.

### 9.1.2 Typedef Documentation

#### 9.1.2.1 AEComp

```
typedef String<'\\',16> gdcm::AEComp
```

#### 9.1.2.2 ASComp

```
typedef String<'\\',64> gdcm::ASComp
```

#### 9.1.2.3 BOOL\_FUNCTION\_PFILE\_PFILE\_POINTER

```
typedef bool(* gdcm::BOOL_FUNCTION_PFILE_PFILE_POINTER) (File *, File *)
```

#### 9.1.2.4 CSComp

```
typedef String<'\\',16> gdcm::CSComp
```

#### 9.1.2.5 DComp

```
typedef String<'\\',64> gdcm::DComp
```

### 9.1.2.6 DTComp

```
typedef String<'\\', 64> gdcm::DTComp
```

### 9.1.2.7 FileList

```
typedef std::vector< SmartPointer<FileWithName> > gdcm::FileList
```

### 9.1.2.8 IconImage

```
typedef Bitmap gdcm::IconImage
```

### 9.1.2.9 LOComp

```
typedef String<'\\', 64> gdcm::LOComp
```

### 9.1.2.10 LTComp

```
typedef String<'\\', 64> gdcm::LTComp
```

### 9.1.2.11 MacroEntry

```
typedef ModuleEntry gdcm::MacroEntry
```

### 9.1.2.12 NestedMacroEntries

```
typedef NestedModuleEntries gdcm::NestedMacroEntries
```

### 9.1.2.13 PNComp

```
typedef String<'\\', 64> gdcm::PNComp
```

### 9.1.2.14 SHComp

```
typedef String<'\\', 64> gdcm::SHComp
```

### 9.1.2.15 STComp

```
typedef String<'\\', 64> gdcm::STComp
```

### 9.1.2.16 TComp

```
typedef String<'\\',16> gdcM::TComp
```

### 9.1.2.17 UComp

```
typedef String<'\\',4294967294> gdcM::UComp
```

### 9.1.2.18 UIComp

```
typedef String<'\\',64,0> gdcM::UIComp
```

### 9.1.2.19 URComp

```
typedef String<'\\',4294967294> gdcM::URComp
```

### 9.1.2.20 UComp

```
typedef String<'\\',64> gdcM::UComp
```

## 9.1.3 Enumeration Type Documentation

### 9.1.3.1 CompOperators

```
enum gdcM::CompOperators
```

Enumerator

GDCM_EQUAL	
GDCM_DIFFERENT	
GDCM_GREATER	
GDCM_GREATEROREQUAL	
GDCM_LESS	
GDCM_LESOREQUAL	

### 9.1.3.2 ECharSet

```
enum gdcM::ECharSet
```

The character sets enumerated in PS 3.3 2009 Annex C, section C.12.1.1.2 The resulting character set is stored in 0008,0005 The conversion to the data element is performed by the [QueryFactory](#) itself

**Enumerator**

eLatin1	
eLatin2	
eLatin3	
eLatin4	
eCyrillic	
eArabic	
eGreek	
eHebrew	
eLatin5	
eJapanese	
eThai	
eJapaneseKanjiMultibyte	
eJapaneseSupplementaryKanjiMultibyte	
eKoreanHangulHanjaMultibyte	
eUTF8	
eGB18030	

**9.1.3.3 ENQueryType**

enum `gdcm::ENQueryType`

**Enumerator**

eCreateMMPS	
eSetMMPS	

**9.1.3.4 EQueryLevel**

enum `gdcm::EQueryLevel`

**Enumerator**

ePatient	
eStudy	
eSeries	
eImage	

**9.1.3.5 EQueryType**

enum `gdcm::EQueryType`



## Enumerator

eFind	
eMove	
eWLMFind	

## 9.1.3.6 ERootType

enum [gdcM::ERootType](#)

## Enumerator

ePatientRootType	
eStudyRootType	

## 9.1.3.7 LodModeType

enum [gdcM::LodModeType](#)

## Enumerator

LD_ALL	
LD_NOSEQ	
LD_NOSHADOW	
LD_NOSHADOWSEQ	

## 9.1.4 Function Documentation

## 9.1.4.1 add1()

```
static int gdcM::add1 (
    char * buf,
    int n ) [static]
```

References [add1\(\)](#).

Referenced by [add1\(\)](#), and [doround\(\)](#).

## 9.1.4.2 backslash()

```
ignore_char const gdcM::backslash (
    '\\ ' )
```

References [backslash\(\)](#).

Referenced by [backslash\(\)](#), and [gdcM::EncodingImplementation< VR::VRASCII >::ReadComputeLength\(\)](#).

#### 9.1.4.3 Clamp()

```
template<typename T >
static T gdcM::Clamp (
    int v ) [inline], [static]
```

#### 9.1.4.4 clean()

```
static void gdcM::clean (
    char * mant ) [inline], [static]
```

References [clean\(\)](#).

Referenced by [clean\(\)](#), and [x16printf\(\)](#).

#### 9.1.4.5 doround()

```
static int gdcM::doround (
    char * buf,
    unsigned int n ) [static]
```

References [add1\(\)](#), and [doround\(\)](#).

Referenced by [doround\(\)](#), and [roundat\(\)](#).

#### 9.1.4.6 GetVRFromTag()

```
VR::VRType gdcM::GetVRFromTag (
    Tag const & tag )
```

#### 9.1.4.7 operator"!="() [1/2]

```
bool gdcM::operator!= (
    const CodeString & ref,
    const CodeString & cs ) [inline]
```

Referenced by [operator!=\(\)](#).

#### 9.1.4.8 operator"!="() [2/2]

```
bool gdcM::operator!= (
    const DataElement & lhs,
    const DataElement & rhs ) [inline]
```

References [operator!=\(\)](#).

**9.1.4.9 operator<<() [1/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & _os,
    const GroupDict & _val ) [inline]
```

**9.1.4.10 operator<<() [2/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & _os,
    const IOD & _val ) [inline]
```

**9.1.4.11 operator<<() [3/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & _os,
    const IOEntry & _val ) [inline]
```

**9.1.4.12 operator<<() [4/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & _os,
    const IODs & _val ) [inline]
```

**9.1.4.13 operator<<() [5/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & _os,
    const Macro & _val ) [inline]
```

**9.1.4.14 operator<<() [6/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & _os,
    const Macros & _val ) [inline]
```

**9.1.4.15 operator<<() [7/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & _os,
    const MediaStorage & ms ) [inline]
```

**9.1.4.16 operator<<() [8/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & _os,
    const Module & _val ) [inline]
```

**9.1.4.17 operator<<() [9/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & _os,
    const ModuleEntry & _val ) [inline]
```

**9.1.4.18 operator<<() [10/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & _os,
    const Modules & _val ) [inline]
```

**9.1.4.19 operator<<() [11/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & _os,
    const NestedModuleEntries & _val ) [inline]
```

**9.1.4.20 operator<<() [12/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & _os,
    const Tag & _val ) [inline]
```

**9.1.4.21 operator<<() [13/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & _os,
    const TransferSyntax & ts ) [inline]
```

**9.1.4.22 operator<<() [14/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & _os,
    const Type & val ) [inline]
```

**9.1.4.23 operator<<() [15/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & _os,
    const UI & _val ) [inline]
```

**9.1.4.24 operator<<() [16/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & _os,
    const UIDs & uid ) [inline]
```

References [gdcm::UIDs::GetName\(\)](#), and [gdcm::UIDs::GetString\(\)](#).

**9.1.4.25 operator<<() [17/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & _os,
    const Usage & val ) [inline]
```

**9.1.4.26 operator<<() [18/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & _os,
    const VM & _val ) [inline]
```

**9.1.4.27 operator<<() [19/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & _os,
    const VR & val ) [inline]
```

**9.1.4.28 operator<<() [20/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & os,
    const BasicOffsetTable & val ) [inline]
```

**9.1.4.29 operator<<() [21/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & os,
    const CodeString & str ) [inline]
```

**9.1.4.30 operator<<() [22/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const CommandDataSet & val ) [inline]
```

**9.1.4.31 operator<<() [23/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const CSAElement & val ) [inline]
```

**9.1.4.32 operator<<() [24/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const CSAHeader & d ) [inline]
```

**9.1.4.33 operator<<() [25/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const CSAHeaderDict & val ) [inline]
```

**9.1.4.34 operator<<() [26/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const CSAHeaderDictEntry & val ) [inline]
```

**9.1.4.35 operator<<() [27/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const DataElement & val ) [inline]
```

**9.1.4.36 operator<<() [28/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const DataSet & val ) [inline]
```

**9.1.4.37 operator<<() [29/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & os,
    const Dict & val ) [inline]
```

**9.1.4.38 operator<<() [30/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & os,
    const DictEntry & val ) [inline]
```

**9.1.4.39 operator<<() [31/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & os,
    const Dicts & d ) [inline]
```

**9.1.4.40 operator<<() [32/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & os,
    const Directory & d ) [inline]
```

Referenced by [gdcm::CSAHeaderDict::CSAHeaderDict\(\)](#), [gdcm::Dict::Dict\(\)](#), [gdcm::IOD::IOD\(\)](#), [gdcm::IODs::IODs\(\)](#), [gdcm::Macro::Macro\(\)](#), [gdcm::Macros::Macros\(\)](#), [gdcm::Module::Module\(\)](#), [gdcm::Modules::Modules\(\)](#), [gdcm::PDBElement::PDBElement\(\)](#), [gdcm::CommandDataSet::~~CommandDataSet\(\)](#), [gdcm::File::~~File\(\)](#), [gdcm::FileMetaInformation::~~FileMetaInformation\(\)](#), [gdcm::GroupDict::~~GroupDict\(\)](#), [gdcm::ModuleEntry::~~ModuleEntry\(\)](#), [gdcm::Preamble::~~Preamble\(\)](#), [gdcm::DataSet::ComputeDataElement\(\)](#), [gdcm::VM::GetLength\(\)](#), [gdcm::PhotometricInterpretation::GetSamplesPerPixel\(\)](#), [gdcm::SwapCode::GetSwapCodeString\(\)](#), and [gdcm::MediaStorage::GuessFromModality\(\)](#).

**9.1.4.41 operator<<() [33/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & os,
    const DPath & val ) [inline]
```

**9.1.4.42 operator<<() [34/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & os,
    const Event & e ) [inline]
```

Generic inserter operator for [Event](#) and its subclasses.

References [gdcm::Event::Print\(\)](#).

**9.1.4.43 operator<<() [35/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const File & val ) [inline]
```

**9.1.4.44 operator<<() [36/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const FileMetaInformation & val ) [inline]
```

**9.1.4.45 operator<<() [37/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const FileSet & f ) [inline]
```

**9.1.4.46 operator<<() [38/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const Fragment & val ) [inline]
```

**9.1.4.47 operator<<() [39/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const Global & g ) [inline]
```

**9.1.4.48 operator<<() [40/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const Item & val ) [inline]
```

**9.1.4.49 operator<<() [41/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const MrProtocol & d ) [inline]
```



**9.1.4.50 operator<<() [42/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & os,
    const Object & obj ) [inline]
```

**9.1.4.51 operator<<() [43/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & os,
    const Orientation & o ) [inline]
```

**9.1.4.52 operator<<() [44/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & os,
    const PDBelement & val ) [inline]
```

**9.1.4.53 operator<<() [45/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & os,
    const PDBHeader & d ) [inline]
```

**9.1.4.54 operator<<() [46/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & os,
    const PhotometricInterpretation & val ) [inline]
```

**9.1.4.55 operator<<() [47/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & os,
    const PixelFormat & pf ) [inline]
```

**9.1.4.56 operator<<() [48/59]**

```
std::ostream & gdcm::operator<< (
    std::ostream & os,
    const Preamble & val ) [inline]
```

**9.1.4.57 operator<<() [49/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const PrivateDict & val ) [inline]
```

**9.1.4.58 operator<<() [50/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const PrivateTag & val ) [inline]
```

**9.1.4.59 operator<<() [51/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const Region & r ) [inline]
```

References [gdcmm::Region::Print\(\)](#).

**9.1.4.60 operator<<() [52/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const Scanner & s ) [inline]
```

**9.1.4.61 operator<<() [53/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const Scanner2 & s ) [inline]
```

**9.1.4.62 operator<<() [54/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const Sorter & s ) [inline]
```

**9.1.4.63 operator<<() [55/59]**

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const StrictScanner & s ) [inline]
```

**9.1.4.64 operator<<()** [56/59]

```
std::ostream & gdcm::operator<< (
    std::ostream & os,
    const StrictScanner2 & s ) [inline]
```

**9.1.4.65 operator<<()** [57/59]

```
std::ostream & gdcm::operator<< (
    std::ostream & os,
    const SwapCode & sc ) [inline]
```

**9.1.4.66 operator<<()** [58/59]

```
std::ostream & gdcm::operator<< (
    std::ostream & os,
    const Version & v ) [inline]
```

**9.1.4.67 operator<<()** [59/59]

```
std::ostream & gdcm::operator<< (
    std::ostream & os,
    const VL & val ) [inline]
```

**9.1.4.68 operator==( )**

```
bool gdcm::operator==(
    const CodeString & ref,
    const CodeString & cs ) [inline]
```

**9.1.4.69 operator>>()** [1/3]

```
std::istream & gdcm::operator>> (
    std::istream & _is,
    Tag & _val ) [inline]
```

**9.1.4.70 operator>>()** [2/3]

```
std::istream & gdcm::operator>> (
    std::istream & in,
    ignore\_char const & ic ) [inline]
```

References [gdcm::ignore\\_char::m\\_char](#).

#### 9.1.4.71 operator>>() [3/3]

```
template<char TDelimiter, unsigned int TMaxLength, char TPadChar>
std::istream & gdc::operator>> (
    std::istream & is,
    String< TDelimiter, TMaxLength, TPadChar > & ms ) [inline]
```

#### 9.1.4.72 Round()

```
template<typename T >
static int gdc::Round (
    T x ) [inline], [static]
```

Referenced by [gdc::ImageChangePhotometricInterpretation::RGB2YBR\(\)](#), and [gdc::ImageChangePhotometricInterpretation::YBR2RGB\(\)](#).

#### 9.1.4.73 roundat()

```
static int gdc::roundat (
    char * buf,
    size_t bufLen,
    unsigned int i,
    int iexp ) [static]
```

References [doround\(\)](#), and [roundat\(\)](#).

Referenced by [roundat\(\)](#), and [x16printf\(\)](#).

#### 9.1.4.74 x16printf()

```
template<typename Float >
static void gdc::x16printf (
    char * buf,
    int size,
    Float f ) [static]
```

References [clean\(\)](#), [roundat\(\)](#), and [x16printf\(\)](#).

Referenced by [gdc::EncodingImplementation< VR::VRASCII >::Write\(\)](#), and [x16printf\(\)](#).

### 9.1.5 Variable Documentation

#### 9.1.5.1 GlobalInstance

```
Global gdc::GlobalInstance [static]
```

## 9.2 gdcm::network Namespace Reference

### Classes

- class [AAbortPDU](#)  
*AAbortPDU.*
- class [AAssociateACPDU](#)  
*AAssociateACPDU.*
- class [AAssociateRJPDU](#)  
*AAssociateRJPDU.*
- class [AAssociateRQPDU](#)  
*AAssociateRQPDU.*
- class [AbstractSyntax](#)  
*AbstractSyntax.*
- class [ApplicationContext](#)  
*ApplicationContext.*
- class [AReleaseRPPDU](#)  
*AReleaseRPPDU.*
- class [AReleaseRQPDU](#)  
*AReleaseRQPDU.*
- class [ARTIMTimer](#)  
*ARTIMTimer.*
- class [AsynchronousOperationsWindowSub](#)  
*AsynchronousOperationsWindowSub.*
- class [BaseCompositeMessage](#)  
*BaseCompositeMessage.*
- class [BaseNormalizedMessage](#)  
*BaseNormalizedMessage.*
- class [BasePDU](#)  
*BasePDU.*
- class [CEchoRQ](#)  
*CEchoRQ.*
- class [CEchoRSP](#)  
*CEchoRSP this file defines the messages for the cecho action.*
- class [CFind](#)
- class [CFindCancelRQ](#)  
*CFindCancelRQ this file defines the messages for the cfind action.*
- class [CFindRQ](#)  
*CFindRQ.*
- class [CFindRSP](#)  
*CFindRSP this file defines the messages for the cfind action.*
- class [CMoveCancelRq](#)
- class [CMoveRQ](#)  
*CMoveRQ.*
- class [CMoveRSP](#)  
*CMoveRSP this file defines the messages for the cmove action.*
- class [CompositeMessageFactory](#)

- CompositeMessageFactory.*
- class [CStoreRQ](#)
  - CStoreRQ.*
- class [CStoreRSP](#)
  - CStoreRSP* this file defines the messages for the cecho action.
- class [DIMSE](#)
  - DIMSE.*
- class [ImplementationClassUIDSub](#)
  - ImplementationClassUIDSub.*
- class [ImplementationUIDSub](#)
  - ImplementationUIDSub.*
- class [ImplementationVersionNameSub](#)
  - ImplementationVersionNameSub.*
- class [MaximumLengthSub](#)
  - MaximumLengthSub.*
- class [NActionRQ](#)
  - NActionRQ.*
- class [NActionRSP](#)
  - NActionRSP* this file defines the messages for the NAction action.
- class [NCreateRQ](#)
  - NCreateRQ.*
- class [NCreateRSP](#)
  - NCreateRSP* this file defines the messages for the ncreate action.
- class [NDeleteRQ](#)
  - NDeleteRQ.*
- class [NDeleteRSP](#)
  - NDeleteRSP* this file defines the messages for the ndelete action.
- class [NEventReportRQ](#)
  - NEventReportRQ.*
- class [NEventReportRSP](#)
  - NEventReportRSP* this file defines the messages for the neventreport action.
- class [NGetRQ](#)
  - NGetRQ.*
- class [NGetRSP](#)
  - NGetRSP* this file defines the messages for the nget action.
- class [NormalizedMessageFactory](#)
- class [NSetRQ](#)
  - NSetRQ.*
- class [NSetRSP](#)
  - NSetRSP* this file defines the messages for the nset action.
- class [PDataTFPDU](#)
  - PDataTFPDU.*
- class [PDUFactory](#)
  - PDUFactory* basically, given an initial byte, construct the.
- class [PresentationContextAC](#)
  - PresentationContextAC.*
- class [PresentationContextRQ](#)

- PresentationContextRQ.*
- class [PresentationDataValue](#)
  - PresentationDataValue.*
- class [RoleSelectionSub](#)
  - RoleSelectionSub.*
- class [ServiceClassApplicationInformation](#)
- class [SOPClassExtendedNegociationSub](#)
  - SOPClassExtendedNegociationSub.*
- class [TableRow](#)
- class [TransferSyntaxSub](#)
  - TransferSyntaxSub.*
- struct [Transition](#)
- class [ULAction](#)
  - ULAction.*
- class [ULActionAA1](#)
- class [ULActionAA2](#)
- class [ULActionAA3](#)
- class [ULActionAA4](#)
- class [ULActionAA5](#)
- class [ULActionAA6](#)
- class [ULActionAA7](#)
- class [ULActionAA8](#)
- class [ULActionAE1](#)
- class [ULActionAE2](#)
- class [ULActionAE3](#)
- class [ULActionAE4](#)
- class [ULActionAE5](#)
- class [ULActionAE6](#)
- class [ULActionAE7](#)
- class [ULActionAE8](#)
- class [ULActionAR1](#)
- class [ULActionAR10](#)
- class [ULActionAR2](#)
- class [ULActionAR3](#)
- class [ULActionAR4](#)
- class [ULActionAR5](#)
- class [ULActionAR6](#)
- class [ULActionAR7](#)
- class [ULActionAR8](#)
- class [ULActionAR9](#)
- class [ULActionDT1](#)
- class [ULActionDT2](#)
- class [ULBasicCallback](#)
  - ULBasicCallback.*
- class [ULConnection](#)
  - ULConnection.*
- class [ULConnectionCallback](#)
- class [ULConnectionInfo](#)
  - ULConnectionInfo.*

- class [ULConnectionManager](#)  
*ULConnectionManager.*
- class [ULEvent](#)  
*ULEvent.*
- class [ULTransitionTable](#)  
*ULTransitionTable* The transition table of all the ULEvents, new ULActions, and ULStates.
- class [ULWritingCallback](#)
- class [UserInformation](#)  
*UserInformation.*

## Enumerations

- enum [EEventID](#) {  
[eAASSOCIATERequestLocalUser](#) = 0 ,  
[eTransportConnConfirmLocal](#) ,  
[eASSOCIATE\\_ACPDUreceived](#) ,  
[eASSOCIATE\\_RJPDUreceived](#) ,  
[eTransportConnIndicLocal](#) ,  
[eAASSOCIATE\\_RQPDUreceived](#) ,  
[eAASSOCIATEResponseAccept](#) ,  
[eAASSOCIATEResponseReject](#) ,  
[ePDATArequest](#) ,  
[ePDATATFPDU](#) ,  
[eARELEASERequest](#) ,  
[eARELEASE\\_RQPDUReceivedOpen](#) ,  
[eARELEASE\\_RPPDUReceived](#) ,  
[eARELEASEResponse](#) ,  
[eAABORTRequest](#) ,  
[eAABORTPDUReceivedOpen](#) ,  
[eTransportConnectionClosed](#) ,  
[eARTIMTimerExpired](#) ,  
[eUnrecognizedPDUReceived](#) ,  
[eEventDoesNotExist](#) }
- enum [EStateID](#) {  
[eStaDoesNotExist](#) = 0 ,  
[eSta1Idle](#) = 1 ,  
[eSta2Open](#) = 2 ,  
[eSta3WaitLocalAssoc](#) = 4 ,  
[eSta4LocalAssocDone](#) = 8 ,  
[eSta5WaitRemoteAssoc](#) = 16 ,  
[eSta6TransferReady](#) = 32 ,  
[eSta7WaitRelease](#) = 64 ,  
[eSta8WaitLocalRelease](#) = 128 ,  
[eSta9ReleaseCollisionRqLocal](#) = 256 ,  
[eSta10ReleaseCollisionAc](#) = 512 ,  
[eSta11ReleaseCollisionRq](#) = 1024 ,  
[eSta12ReleaseCollisionAcLocal](#) = 2048 ,  
[eSta13AwaitingClose](#) = 4096 }

## Functions

- int [GetStateIndex](#) ([EStateID](#) inState)



## Variables

- const int `cMaxEventID` = `eEventDoesNotExist`
- const int `cMaxStateID` = 13

## 9.2.1 Enumeration Type Documentation

### 9.2.1.1 EEventID

```
enum gdcmm::network::EEventID
```

#### Enumerator

<code>eAASSOCIATERequestLocalUser</code>	
<code>eTransportConnConfirmLocal</code>	
<code>eASSOCIATE_ACPDUreceived</code>	
<code>eASSOCIATE_RJPDUreceived</code>	
<code>eTransportConnIndicLocal</code>	
<code>eAASSOCIATE_RQPDUreceived</code>	
<code>eAASSOCIATEresponseAccept</code>	
<code>eAASSOCIATEresponseReject</code>	
<code>ePDATArequest</code>	
<code>ePDATATFPDU</code>	
<code>eARELEASERequest</code>	
<code>eARELEASE_RQPDUReceivedOpen</code>	
<code>eARELEASE_RPPDUReceived</code>	
<code>eARELEASEResponse</code>	
<code>eAABORTRequest</code>	
<code>eAABORTPDUReceivedOpen</code>	
<code>eTransportConnectionClosed</code>	
<code>eARTIMTimerExpired</code>	
<code>eUnrecognizedPDUReceived</code>	
<code>eEventDoesNotExist</code>	

### 9.2.1.2 EStateID

```
enum gdcmm::network::EStateID
```

Each network connection will be in a particular state at any given time. Those states have IDs as described in the standard ps3.8-2009, roughly 1-13. This enumeration lists those states. The actual ULState class will contain more information about transitions to other states.

name and date: 16 sept 2010 mmr

### Enumerator

eStaDoesNotExist	
eSta1Idle	
eSta2Open	
eSta3WaitLocalAssoc	
eSta4LocalAssocDone	
eSta5WaitRemoteAssoc	
eSta6TransferReady	
eSta7WaitRelease	
eSta8WaitLocalRelease	
eSta9ReleaseCollisionRqLocal	
eSta10ReleaseCollisionAc	
eSta11ReleaseCollisionRq	
eSta12ReleaseCollisionAcLocal	
eSta13AwaitingClose	

## 9.2.2 Function Documentation

### 9.2.2.1 GetStateIndex()

```
int gdcn::network::GetStateIndex (
    EStateID inState ) [inline]
```

References [eSta10ReleaseCollisionAc](#), [eSta11ReleaseCollisionRq](#), [eSta12ReleaseCollisionAcLocal](#), [eSta13AwaitingClose](#), [eSta1Idle](#), [eSta2Open](#), [eSta3WaitLocalAssoc](#), [eSta4LocalAssocDone](#), [eSta5WaitRemoteAssoc](#), [eSta6TransferReady](#), [eSta7WaitRelease](#), [eSta8WaitLocalRelease](#), [eSta9ReleaseCollisionRqLocal](#), and [eStaDoesNotExist](#).

## 9.2.3 Variable Documentation

### 9.2.3.1 cMaxEventID

```
const int gdcn::network::cMaxEventID = eEventDoesNotExist
```

### 9.2.3.2 cMaxStateID

```
const int gdcn::network::cMaxStateID = 13
```

Referenced by [gdcn::network::TableRow::TableRow\(\)](#), and [gdcn::network::TableRow::~~TableRow\(\)](#).

## 9.3 gdcmm::SegmentHelper Namespace Reference

### Classes

- struct [BasicCodedEntry](#)

*This structure defines a basic coded entry with all of its attributes.*

## 9.4 gdcmm::terminal Namespace Reference

Class for Terminal.

### Enumerations

- enum [Attribute](#) {  
    [reset](#) = 0 ,  
    [bright](#) = 1 ,  
    [dim](#) = 2 ,  
    [underline](#) = 3 ,  
    [blink](#) = 5 ,  
    [reverse](#) = 7 ,  
    [hidden](#) = 8 }
- enum [Color](#) {  
    [black](#) = 0 ,  
    [red](#) ,  
    [green](#) ,  
    [yellow](#) ,  
    [blue](#) ,  
    [magenta](#) ,  
    [cyan](#) ,  
    [white](#) }
- enum [Mode](#) {  
    [CONSOLE](#) = 0 ,  
    [VT100](#) }

### Functions

- [GDCM\\_EXPORT](#) std::string [setattribute](#) ([Attribute](#) att)
- [GDCM\\_EXPORT](#) std::string [setbgcolor](#) ([Color](#) c)
- [GDCM\\_EXPORT](#) std::string [setfgcolor](#) ([Color](#) c)
- [GDCM\\_EXPORT](#) void [setmode](#) ([Mode](#) m)

### 9.4.1 Detailed Description

Class for Terminal.

Allow one to print in color in a shell

- support VT100 compatible shell
- win32 console

## 9.4.2 Enumeration Type Documentation

### 9.4.2.1 Attribute

enum `gdc::terminal::Attribute`

#### Enumerator

reset	
bright	
dim	
underline	
blink	
reverse	
hidden	

### 9.4.2.2 Color

enum `gdc::terminal::Color`

#### Enumerator

black	
red	
green	
yellow	
blue	
magenta	
cyan	
white	

### 9.4.2.3 Mode

enum `gdc::terminal::Mode`

#### Enumerator

CONSOLE	
VT100	

## 9.4.3 Function Documentation

### 9.4.3.1 setattribute()

```
GDCM_EXPORT std::string gdcm::terminal::setattribute (
    Attribute att )
```

### 9.4.3.2 setbgcolor()

```
GDCM_EXPORT std::string gdcm::terminal::setbgcolor (
    Color c )
```

### 9.4.3.3 setfgcolor()

```
GDCM_EXPORT std::string gdcm::terminal::setfgcolor (
    Color c )
```

### 9.4.3.4 setmode()

```
GDCM_EXPORT void gdcm::terminal::setmode (
    Mode m )
```



## Chapter 10

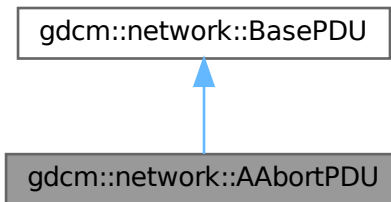
# Class Documentation

### 10.1 gdcmm::network::AAabortPDU Class Reference

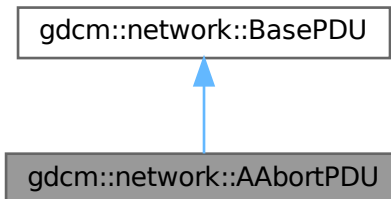
[AAabortPDU](#).

```
#include <gdcmmAAabortPDU.h>
```

Inheritance diagram for gdcmm::network::AAabortPDU:



Collaboration diagram for gdcmm::network::AAabortPDU:



## Public Member Functions

- [AAbortPDU](#) ()
- bool [IsLastFragment](#) () const override
- void [Print](#) (std::ostream &os) const override
- std::istream & [Read](#) (std::istream &is) override
- void [SetReason](#) (const uint8\_t r)
- void [SetSource](#) (const uint8\_t s)
- size\_t [Size](#) () const override
- const std::ostream & [Write](#) (std::ostream &os) const override

## Public Member Functions inherited from [gdcm::network::BasePDU](#)

- virtual [~BasePDU](#) ()=default

### 10.1.1 Detailed Description

[AAbortPDU](#).

[Table 9-26](#) A-ABORT PDU FIELDS

### 10.1.2 Constructor & Destructor Documentation

#### 10.1.2.1 AAbortPDU()

```
gdcm::network::AAbortPDU::AAbortPDU ( )
```

### 10.1.3 Member Function Documentation

#### 10.1.3.1 IsLastFragment()

```
bool gdcm::network::AAbortPDU::IsLastFragment ( ) const [inline], [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

#### 10.1.3.2 Print()

```
void gdcm::network::AAbortPDU::Print (
    std::ostream & os ) const [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).



### 10.1.3.3 Read()

```
std::istream & gdcm::network::AAabortPDU::Read (
    std::istream & is ) [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

### 10.1.3.4 SetReason()

```
void gdcm::network::AAabortPDU::SetReason (
    const uint8_t r )
```

### 10.1.3.5 SetSource()

```
void gdcm::network::AAabortPDU::SetSource (
    const uint8_t s )
```

### 10.1.3.6 Size()

```
size_t gdcm::network::AAabortPDU::Size ( ) const [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

### 10.1.3.7 Write()

```
const std::ostream & gdcm::network::AAabortPDU::Write (
    std::ostream & os ) const [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

The documentation for this class was generated from the following file:

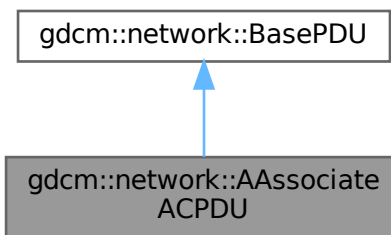
- [gdcmAAabortPDU.h](#)

## 10.2 gdcm::network::AAssociateACPDU Class Reference

[AAssociateACPDU](#).

```
#include <gdcmAAssociateACPDU.h>
```

Inheritance diagram for gdcm::network::AAssociateACPDU:



Collaboration diagram for gdcm::network::AAssociateACPDU:



### Public Types

- typedef std::vector< [PresentationContextAC](#) >::size\_type [SizeType](#)

**Public Member Functions**

- [AAssociateACPDU](#) ()
- void [AddPresentationContextAC](#) ([PresentationContextAC](#) const &pcac)
- [SizeType](#) [GetNumberOfPresentationContextAC](#) () const
- const [PresentationContextAC](#) & [GetPresentationContextAC](#) ([SizeType](#) i)
- const [UserInformation](#) & [GetUserInformation](#) () const
- void [InitFromRQ](#) ([AAssociateRQPDU](#) const &rqpdu)
- bool [IsLastFragment](#) () const override
- void [Print](#) (std::ostream &os) const override
- std::istream & [Read](#) (std::istream &is) override
- [SizeType](#) [Size](#) () const override
- const std::ostream & [Write](#) (std::ostream &os) const override

**Public Member Functions inherited from [gdcm::network::BasePDU](#)**

- virtual [~BasePDU](#) ()=default

**Protected Member Functions**

- void [SetCalledAETitle](#) (const char calledaetitle[16])
- void [SetCallingAETitle](#) (const char callingaetitle[16])

**Friends**

- class [AAssociateRQPDU](#)

**10.2.1 Detailed Description**

[AAssociateACPDU](#).

[Table](#) 9-17 ASSOCIATE-AC PDU fields

**10.2.2 Member Typedef Documentation****10.2.2.1 SizeType**

```
typedef std::vector<PresentationContextAC>::size_type gdcm::network::AAssociateACPDU::SizeType
```

**10.2.3 Constructor & Destructor Documentation****10.2.3.1 AAssociateACPDU()**

```
gdcm::network::AAssociateACPDU::AAssociateACPDU ( )
```

## 10.2.4 Member Function Documentation

### 10.2.4.1 AddPresentationContextAC()

```
void gdcn::network::AAssociateACPDU::AddPresentationContextAC (
    PresentationContextAC const & pcac )
```

### 10.2.4.2 GetNumberOfPresentationContextAC()

```
SizeType gdcn::network::AAssociateACPDU::GetNumberOfPresentationContextAC ( ) const [inline]
```

### 10.2.4.3 GetPresentationContextAC()

```
const PresentationContextAC & gdcn::network::AAssociateACPDU::GetPresentationContextAC (
    SizeType i ) [inline]
```

### 10.2.4.4 GetUserInfoInformation()

```
const UserInformation & gdcn::network::AAssociateACPDU::GetUserInfoInformation ( ) const [inline]
```

### 10.2.4.5 InitFromRQ()

```
void gdcn::network::AAssociateACPDU::InitFromRQ (
    AAssociateRQPDU const & rqpdu )
```

### 10.2.4.6 IsLastFragment()

```
bool gdcn::network::AAssociateACPDU::IsLastFragment ( ) const [inline], [override], [virtual]
```

Implements [gdcn::network::BasePDU](#).

### 10.2.4.7 Print()

```
void gdcn::network::AAssociateACPDU::Print (
    std::ostream & os ) const [override], [virtual]
```

Implements [gdcn::network::BasePDU](#).

#### 10.2.4.8 Read()

```
std::istream & gdcm::network::AAssociateACPDU::Read (
    std::istream & is ) [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

#### 10.2.4.9 SetCalledAETitle()

```
void gdcm::network::AAssociateACPDU::SetCalledAETitle (
    const char calledaetitle[16] ) [protected]
```

#### 10.2.4.10 SetCallingAETitle()

```
void gdcm::network::AAssociateACPDU::SetCallingAETitle (
    const char callingaetitle[16] ) [protected]
```

#### 10.2.4.11 Size()

```
SizeType gdcm::network::AAssociateACPDU::Size ( ) const [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

#### 10.2.4.12 Write()

```
const std::ostream & gdcm::network::AAssociateACPDU::Write (
    std::ostream & os ) const [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

### 10.2.5 Friends And Related Symbol Documentation

#### 10.2.5.1 AAssociateRQPDU

```
friend class AAssociateRQPDU [friend]
```

The documentation for this class was generated from the following file:

- [gdcmAAssociateACPDU.h](#)

## 10.3 gdcmm::network::AAssociateRJPDU Class Reference

[AAssociateRJPDU](#).

```
#include <gdcmAAssociateRJPDU.h>
```

Inheritance diagram for gdcmm::network::AAssociateRJPDU:



Collaboration diagram for gdcmm::network::AAssociateRJPDU:



### Public Member Functions

- [AAssociateRJPDU](#) ()
- bool [IsLastFragment](#) () const override
- void [Print](#) (std::ostream &os) const override
- std::istream & [Read](#) (std::istream &is) override
- size\_t [Size](#) () const override
- const std::ostream & [Write](#) (std::ostream &os) const override

## Public Member Functions inherited from [gdcm::network::BasePDU](#)

- virtual [~BasePDU](#) ()=default

### 10.3.1 Detailed Description

[AAssociateRJPDU](#).

Table 9-21 ASSOCIATE-RJ PDU FIELDS

### 10.3.2 Constructor & Destructor Documentation

#### 10.3.2.1 AAssociateRJPDU()

```
gdcm::network::AAssociateRJPDU::AAssociateRJPDU ( )
```

### 10.3.3 Member Function Documentation

#### 10.3.3.1 IsLastFragment()

```
bool gdcm::network::AAssociateRJPDU::IsLastFragment ( ) const [inline], [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

#### 10.3.3.2 Print()

```
void gdcm::network::AAssociateRJPDU::Print (
    std::ostream & os ) const [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

#### 10.3.3.3 Read()

```
std::istream & gdcm::network::AAssociateRJPDU::Read (
    std::istream & is ) [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

#### 10.3.3.4 Size()

```
size_t gdcm::network::AAssociateRJPDU::Size ( ) const [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

### 10.3.3.5 Write()

```
const std::ostream & gdcm::network::AAssociateRJPDU::Write (
    std::ostream & os ) const [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

The documentation for this class was generated from the following file:

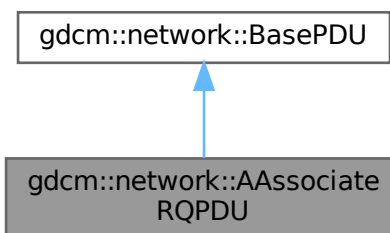
- [gdcmAAssociateRJPDU.h](#)

## 10.4 gdcm::network::AAssociateRQPDU Class Reference

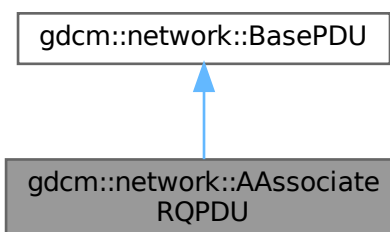
[AAssociateRQPDU](#).

```
#include <gdcmAAssociateRQPDU.h>
```

Inheritance diagram for `gdcm::network::AAssociateRQPDU`:



Collaboration diagram for `gdcm::network::AAssociateRQPDU`:





## Public Types

- typedef std::vector< [PresentationContextRQ](#) > [PresentationContextArrayType](#)
- typedef std::vector< [PresentationContextRQ](#) >::size\_type [SizeType](#)

## Public Member Functions

- [AAssociateRQPDU](#) ()
- [AAssociateRQPDU](#) (const [AAssociateRQPDU](#) &pdu)
- void [AddPresentationContext](#) ([PresentationContextRQ](#) const &pc)
- std::string [GetCalledAETitle](#) () const
- std::string [GetCallingAETitle](#) () const
- [SizeType](#) [GetNumberOfPresentationContext](#) () const
- [PresentationContextRQ](#) const & [GetPresentationContext](#) ([SizeType](#) i) const
- const [PresentationContextRQ](#) \* [GetPresentationContextByAbstractSyntax](#) ([AbstractSyntax](#) const &absyn) const
- const [PresentationContextRQ](#) \* [GetPresentationContextByID](#) (uint8\_t i) const
- [PresentationContextArrayType](#) const & [GetPresentationContexts](#) ()
- const [UserInformation](#) & [GetUserInformation](#) () const
- bool [IsLastFragment](#) () const override
- void [Print](#) (std::ostream &os) const override
- std::istream & [Read](#) (std::istream &is) override
- void [SetCalledAETitle](#) (const char calledaetitle[16])  
*Set the Called AE Title.*
- void [SetCallingAETitle](#) (const char callingaetitle[16])  
*Set the Calling AE Title.*
- void [SetUserInformation](#) ([UserInformation](#) const &ui)
- size\_t [Size](#) () const override
- const std::ostream & [Write](#) (std::ostream &os) const override

## Public Member Functions inherited from [gdcm::network::BasePDU](#)

- virtual [~BasePDU](#) ()=default

## Static Public Member Functions

- static bool [IsAETitleValid](#) (const char title[16])  
*Check whether or not the.*

## Protected Member Functions

- std::string [GetReserved43\\_74](#) () const

## Friends

- class [AAssociateACPDU](#)

### 10.4.1 Detailed Description

[AAssociateRQPDU](#).

[Table 9-11](#) ASSOCIATE-RQ PDU fields

### 10.4.2 Member Typedef Documentation

#### 10.4.2.1 PresentationContextArrayType

```
typedef std::vector<PresentationContextRQ> gdcn::network::AAssociateRQPDU::PresentationContextArrayType
```

#### 10.4.2.2 SizeType

```
typedef std::vector<PresentationContextRQ>::size_type gdcn::network::AAssociateRQPDU::SizeType
```

### 10.4.3 Constructor & Destructor Documentation

#### 10.4.3.1 AAssociateRQPDU() [1/2]

```
gdcn::network::AAssociateRQPDU::AAssociateRQPDU ( )
```

#### 10.4.3.2 AAssociateRQPDU() [2/2]

```
gdcn::network::AAssociateRQPDU::AAssociateRQPDU (
    const AAssociateRQPDU & pdu ) [inline]
```

### 10.4.4 Member Function Documentation

#### 10.4.4.1 AddPresentationContext()

```
void gdcn::network::AAssociateRQPDU::AddPresentationContext (
    PresentationContextRQ const & pc )
```

#### 10.4.4.2 GetCalledAETitle()

```
std::string gdcn::network::AAssociateRQPDU::GetCalledAETitle ( ) const [inline]
```

**10.4.4.3 GetCallingAETitle()**

```
std::string gdcm::network::AAssociateRQPDU::GetCallingAETitle ( ) const [inline]
```

**10.4.4.4 GetNumberOfPresentationContext()**

```
SizeType gdcm::network::AAssociateRQPDU::GetNumberOfPresentationContext ( ) const [inline]
```

**10.4.4.5 GetPresentationContext()**

```
PresentationContextRQ const & gdcm::network::AAssociateRQPDU::GetPresentationContext (
    SizeType i ) const [inline]
```

**10.4.4.6 GetPresentationContextByAbstractSyntax()**

```
const PresentationContextRQ * gdcm::network::AAssociateRQPDU::GetPresentationContextByAbstract←
Syntax (
    AbstractSyntax const & absyn ) const
```

**10.4.4.7 GetPresentationContextByID()**

```
const PresentationContextRQ * gdcm::network::AAssociateRQPDU::GetPresentationContextByID (
    uint8_t i ) const
```

**10.4.4.8 GetPresentationContexts()**

```
PresentationContextArrayType const & gdcm::network::AAssociateRQPDU::GetPresentationContexts ( )
[inline]
```

**10.4.4.9 GetReserved43\_74()**

```
std::string gdcm::network::AAssociateRQPDU::GetReserved43_74 ( ) const [protected]
```

**10.4.4.10 GetUserInfoInformation()**

```
const UserInformation & gdcm::network::AAssociateRQPDU::GetUserInfoInformation ( ) const [inline]
```

**10.4.4.11 IsAETitleValid()**

```
static bool gdcm::network::AAssociateRQPDU::IsAETitleValid (
    const char title[16] ) [static]
```

Check whether or not the.

### Parameters

<i>title</i>	is a valid AE title
--------------	---------------------

#### 10.4.4.12 IsLastFragment()

```
bool gdcm::network::AAssociateRQPDU::IsLastFragment ( ) const [inline], [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

#### 10.4.4.13 Print()

```
void gdcm::network::AAssociateRQPDU::Print (
    std::ostream & os ) const [override], [virtual]
```

This function will initialize an [AAssociateACPDU](#) from the fields in the [AAssociateRQPDU](#) structure

Implements [gdcm::network::BasePDU](#).

#### 10.4.4.14 Read()

```
std::istream & gdcm::network::AAssociateRQPDU::Read (
    std::istream & is ) [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

#### 10.4.4.15 SetCalledAETitle()

```
void gdcm::network::AAssociateRQPDU::SetCalledAETitle (
    const char calledaetitle[16] )
```

Set the Called AE Title.

#### 10.4.4.16 SetCallingAETitle()

```
void gdcm::network::AAssociateRQPDU::SetCallingAETitle (
    const char callingaetitle[16] )
```

Set the Calling AE Title.

#### 10.4.4.17 SetUserInformation()

```
void gdcm::network::AAssociateRQPDU::SetUserInformation (
    UserInformation const & ui )
```

#### 10.4.4.18 Size()

```
size_t gdcm::network::AAssociateRQPDU::Size ( ) const [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

#### 10.4.4.19 Write()

```
const std::ostream & gdcm::network::AAssociateRQPDU::Write (
    std::ostream & os ) const [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

### 10.4.5 Friends And Related Symbol Documentation

#### 10.4.5.1 AAssociateACPDU

```
friend class AAssociateACPDU [friend]
```

The documentation for this class was generated from the following file:

- [gdcmAAssociateRQPDU.h](#)

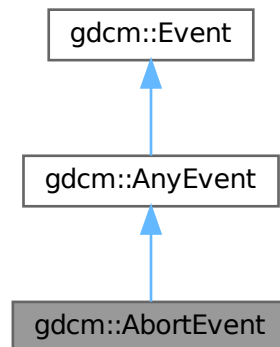
## 10.5 gdcm::AbortEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::AbortEvent:



Collaboration diagram for `gdcm::AbortEvent`:



#### Additional Inherited Members

#### Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()
- [Event](#) (const [Event](#) &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) \*) const =0
- virtual const char \* [GetEventName](#) () const =0
- virtual [Event](#) \* [MakeObject](#) () const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

The documentation for this class was generated from the following file:

- [gdcmEvent.h](#)

## 10.6 `gdcm::network::AbstractSyntax` Class Reference

[AbstractSyntax](#).

```
#include <gdcmAbstractSyntax.h>
```

## Public Member Functions

- [AbstractSyntax](#) ()
- [DataElement GetAsDataElement](#) () const
- const char \* [GetName](#) () const
- bool [operator==](#) (const [AbstractSyntax](#) &as) const
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [SetName](#) (const char \*name)
- void [SetNameFromUID](#) (UIDs::TSName tsname)
- size\_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

### 10.6.1 Detailed Description

[AbstractSyntax](#).

[Table](#) 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS

### 10.6.2 Constructor & Destructor Documentation

#### 10.6.2.1 AbstractSyntax()

```
gdcm::network::AbstractSyntax::AbstractSyntax ( )
```

### 10.6.3 Member Function Documentation

#### 10.6.3.1 GetAsDataElement()

```
DataElement gdcm::network::AbstractSyntax::GetAsDataElement ( ) const
```

#### 10.6.3.2 GetName()

```
const char * gdcm::network::AbstractSyntax::GetName ( ) const [inline]
```

#### 10.6.3.3 operator==( )

```
bool gdcm::network::AbstractSyntax::operator== (
    const AbstractSyntax & as ) const [inline]
```

#### 10.6.3.4 Print()

```
void gdcmm::network::AbstractSyntax::Print (
    std::ostream & os ) const
```

#### 10.6.3.5 Read()

```
std::istream & gdcmm::network::AbstractSyntax::Read (
    std::istream & is )
```

#### 10.6.3.6 SetName()

```
void gdcmm::network::AbstractSyntax::SetName (
    const char * name ) [inline]
```

#### 10.6.3.7 SetNameFromUID()

```
void gdcmm::network::AbstractSyntax::SetNameFromUID (
    UIDs::TSName tname )
```

#### 10.6.3.8 Size()

```
size_t gdcmm::network::AbstractSyntax::Size ( ) const
```

#### 10.6.3.9 Write()

```
const std::ostream & gdcmm::network::AbstractSyntax::Write (
    std::ostream & os ) const
```

The documentation for this class was generated from the following file:

- [gdcmmAbstractSyntax.h](#)



## 10.7 gdcm::AnonymizeEvent Class Reference

[AnonymizeEvent](#).

```
#include <gdcmAnonymizeEvent.h>
```

Inheritance diagram for gdcm::AnonymizeEvent:



Collaboration diagram for gdcm::AnonymizeEvent:



### Public Types

- typedef [AnonymizeEvent](#) Self
- typedef [AnyEvent](#) Superclass

## Public Member Functions

- [AnonymizeEvent](#) (const [Self](#) &s)
- [AnonymizeEvent](#) ([Tag](#) const &tag=0)
- [~AnonymizeEvent](#) () override=default
- bool [CheckEvent](#) (const [::gdcm::Event](#) \*e) const override
- const char \* [GetEventName](#) () const override
- [Tag](#) const & [GetTag](#) () const
- [::gdcm::Event](#) \* [MakeObject](#) () const override
- void [operator=](#) (const [Self](#) &)=delete
- void [SetTag](#) (const [Tag](#) &t)

## Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()
- [Event](#) (const [Event](#) &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) \*) const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

### 10.7.1 Detailed Description

[AnonymizeEvent](#).

Special type of event triggered during the Anonymization process

See also

[Anonymizer](#)

Examples

[BasicAnonymizer.cs](#), [Cleaner.cs](#), and [ClinicalTrialIdentificationWorkflow.cs](#).

### 10.7.2 Member Typedef Documentation

#### 10.7.2.1 Self

```
typedef AnonymizeEvent gdcm::AnonymizeEvent::Self
```

#### 10.7.2.2 Superclass

```
typedef AnyEvent gdcm::AnonymizeEvent::Superclass
```

## 10.7.3 Constructor & Destructor Documentation

### 10.7.3.1 AnonymizeEvent() [1/2]

```
gdcm::AnonymizeEvent::AnonymizeEvent (
    Tag const & tag = 0 ) [inline]
```

### 10.7.3.2 ~AnonymizeEvent()

```
gdcm::AnonymizeEvent::~~AnonymizeEvent ( ) [override], [default]
```

### 10.7.3.3 AnonymizeEvent() [2/2]

```
gdcm::AnonymizeEvent::AnonymizeEvent (
    const Self & s ) [inline]
```

## 10.7.4 Member Function Documentation

### 10.7.4.1 CheckEvent()

```
bool gdcm::AnonymizeEvent::CheckEvent (
    const ::gdcm::Event * e ) const [inline], [override]
```

### 10.7.4.2 GetEventName()

```
const char * gdcm::AnonymizeEvent::GetEventName ( ) const [inline], [override], [virtual]
```

Return the StringName associated with the event.

Implements [gdcm::Event](#).

### 10.7.4.3 GetTag()

```
Tag const & gdcm::AnonymizeEvent::GetTag ( ) const [inline]
```

#### Examples

[BasicAnonymizer.cs](#), [Cleaner.cs](#), and [ClinicalTrialIdentificationWorkflow.cs](#).

#### 10.7.4.4 MakeObject()

```
::gdcM::Event * gdcM::AnonymizeEvent::MakeObject ( ) const [inline], [override], [virtual]
```

Create an [Event](#) of this type This method work as a Factory for creating events of each particular type.

Implements [gdcM::Event](#).

#### 10.7.4.5 operator=()

```
void gdcM::AnonymizeEvent::operator= (
    const Self & ) [delete]
```

#### 10.7.4.6 SetTag()

```
void gdcM::AnonymizeEvent::SetTag (
    const Tag & t ) [inline]
```

The documentation for this class was generated from the following file:

- [gdcMAnonymizeEvent.h](#)

## 10.8 gdcM::Anonymizer Class Reference

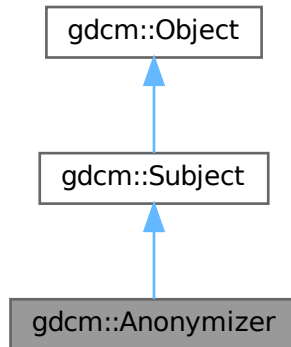
[Anonymizer](#).

```
#include <gdcMAnonymizer.h>
```

Inheritance diagram for gdcM::Anonymizer:



Collaboration diagram for gdcm::Anonymizer:



### Public Member Functions

- [Anonymizer](#) ()
- [~Anonymizer](#) () override
- bool [BasicApplicationLevelConfidentialityProfile](#) (bool deidentify=true)
- bool [Clear](#) ([PrivateTag](#) const &pt)
- bool [Clear](#) ([Tag](#) const &t)  
*Identical to 'Empty' except no action is done when tag is not present.*
- bool [Empty](#) ([PrivateTag](#) const &pt)
- bool [Empty](#) ([Tag](#) const &t)  
*Make [Tag](#) t empty (if not found tag will be created)*
- const [CryptographicMessageSyntax](#) \* [GetCryptographicMessageSyntax](#) () const
- [File](#) & [GetFile](#) ()
- bool [Remove](#) ([PrivateTag](#) const &pt)
- bool [Remove](#) ([Tag](#) const &t)  
*remove a tag (even a SQ can be removed)*
- bool [RemoveGroupLength](#) ()  
*Main function that loop over all elements and remove group length.*
- bool [RemovePrivateTags](#) ()  
*Main function that loop over all elements and remove private tags.*
- bool [RemoveRetired](#) ()  
*Main function that loop over all elements and remove retired element.*
- bool [Replace](#) ([PrivateTag](#) const &t, const char \*value)
- bool [Replace](#) ([PrivateTag](#) const &t, const char \*value, [VL](#) const &vl)
- bool [Replace](#) ([Tag](#) const &t, const char \*value)
- bool [Replace](#) ([Tag](#) const &t, const char \*value, [VL](#) const &vl)
- void [SetCryptographicMessageSyntax](#) ([CryptographicMessageSyntax](#) \*cms)  
*Set/Get CMS key that will be used to encrypt the dataset within BasicApplicationLevelConfidentialityProfile.*
- void [SetFile](#) (const [File](#) &f)  
*Set/Get [File](#).*

## Public Member Functions inherited from [gdcm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*) const
- [Command](#) \* [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
- *Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

## Static Public Member Functions

- static void [ClearInternalUIDs](#) ()
- static std::vector< [Tag](#) > [GetBasicApplicationLevelConfidentialityProfileAttributes](#) ()
- *Return the list of [Tag](#) that will be considered when anonymizing a DICOM file.*
- static [SmartPointer](#)< [Anonymizer](#) > [New](#) ()
- *for wrapped language: instantiate a reference counted object*

## Protected Member Functions

- bool [BALCPPProtect](#) ([DataSet](#) &ds, [Tag](#) const &tag, const [IOD](#) &iod)
- bool [CanEmptyTag](#) ([Tag](#) const &tag, const [IOD](#) &iod) const
- void [RecurseDataSet](#) ([DataSet](#) &ds)

## Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

## 10.8.1 Detailed Description

### Anonymizer.

This class is a multi purpose anonymizer. It can work in 2 mode:

- Full (irreversible) anonymizer (aka dumb mode)
- reversible de-identifier/re-identifier (aka smart mode). This implements the Basic Application Level Confidentiality Profile, DICOM PS 3.15-2009

1. dumb mode This is a dumb anonymizer implementation. All it allows user is simple operation such as:

[Tag](#) based functions:

- complete removal of DICOM attribute (Remove)
- make a tag empty, ie make it's length 0 (Empty)
- replace with another string-based value (Replace)

[DataSet](#) based functions:

- Remove all group length attribute from a DICOM dataset (Group Length element are deprecated, DICOM 2008)
- Remove all private attributes
- Remove all retired attributes

All function calls actually execute the user specified request. Previous implementation were calling a general Anonymize function but traversing a `std::set` is  $O(n)$  operation, while a simple user specified request is  $O(\log(n))$  operation. So 'm' user interaction is  $O(m \cdot \log(n))$  which is  $< O(n)$  complexity.

1. smart mode this mode implements the Basic Application Level Confidentiality Profile (DICOM PS 3.15-2008) In this case, it is extremely important to use the same [Anonymizer](#) class when anonymizing a [FileSet](#). Once the [Anonymizer](#) is destroyed its memory of known (already processed) [UIDs](#) will be lost. which will make the anonymizer behaves incorrectly for attributes such as [Series](#) [UID](#) [Study](#) [UID](#) where user want some consistency. When attribute is [Type](#) 1 / [Type](#) 1C, a dummy generator will take in the existing value and produce a dummy value (a sha1 representation). sha1 algorithm is considered to be cryptographically strong (compared to md5sum) so that we meet the following two conditions:

- Produce the same dummy value for the same input value
- do not provide an easy way to retrieve the original value from the sha1 generated value

This class implement the Subject/Observer pattern trigger the following event:

- [AnonymizeEvent](#)
- [IterationEvent](#)
- [StartEvent](#)
- [EndEvent](#)

See also

[CryptographicMessageSyntax](#)

Examples

[BasicAnonymizer.cs](#), [ClinicalTrialAnnotate.cxx](#), [ClinicalTrialIdentificationWorkflow.cs](#), [CreateJPIPDataSet.cxx](#), [EncapsulateFileInRawData.cxx](#), [ManipulateFile.cs](#), and [MpegVideoInfo.cs](#).

## 10.8.2 Constructor & Destructor Documentation

### 10.8.2.1 Anonymizer()

```
gdcm::Anonymizer::Anonymizer ( ) [inline]
```

### 10.8.2.2 ~Anonymizer()

```
gdcm::Anonymizer::~~Anonymizer ( ) [override]
```

## 10.8.3 Member Function Documentation

### 10.8.3.1 BALCPPProtect()

```
bool gdcm::Anonymizer::BALCPPProtect (
    DataSet & ds,
    Tag const & tag,
    const IOD & iod ) [protected]
```

### 10.8.3.2 BasicApplicationLevelConfidentialityProfile()

```
bool gdcm::Anonymizer::BasicApplicationLevelConfidentialityProfile (
    bool deidentify = true )
```

PS 3.15 / E.1.1 De-Identifier An Application may claim conformance to the Basic Application Level Confidentiality Profile as a deidentifier if it protects all Attributes that might be used by unauthorized entities to identify the patient. NOT THREAD SAFE

#### Examples

[BasicAnonymizer.cs](#).

### 10.8.3.3 CanEmptyTag()

```
bool gdcm::Anonymizer::CanEmptyTag (
    Tag const & tag,
    const IOD & iod ) const [protected]
```

### 10.8.3.4 Clear() [1/2]

```
bool gdcm::Anonymizer::Clear (
    PrivateTag const & pt )
```



### 10.8.3.5 Clear() [2/2]

```
bool gdcm::Anonymizer::Clear (
    Tag const & t )
```

Identical to 'Empty' except no action is done when tag is not present.

### 10.8.3.6 ClearInternalUIDs()

```
static void gdcm::Anonymizer::ClearInternalUIDs ( ) [static]
```

Clear the internal mapping of real [UIDs](#) to generated [UIDs](#)

#### Warning

the mapping is definitely lost

### 10.8.3.7 Empty() [1/2]

```
bool gdcm::Anonymizer::Empty (
    PrivateTag const & pt )
```

Make [PrivateTag](#) pt empty (if not found tag will be created) Pay special attention that this code must be done before any call to Empty/Remove of the associated Private Creator, but before any call to Replace.

### 10.8.3.8 Empty() [2/2]

```
bool gdcm::Anonymizer::Empty (
    Tag const & t )
```

Make [Tag](#) t empty (if not found tag will be created)

#### Examples

[CreateJPIPDataSet.cxx](#).

### 10.8.3.9 GetBasicApplicationLevelConfidentialityProfileAttributes()

```
static std::vector< Tag > gdcm::Anonymizer::GetBasicApplicationLevelConfidentialityProfile↵
Attributes ( ) [static]
```

Return the list of [Tag](#) that will be considered when anonymizing a DICOM file.

#### Examples

[GenFakeIdentifyFile.cxx](#), and [TraverseModules.cxx](#).

### 10.8.3.10 GetCryptographicMessageSyntax()

```
const CryptographicMessageSyntax * gdcM::Anonymizer::GetCryptographicMessageSyntax ( ) const
```

### 10.8.3.11 GetFile()

```
File & gdcM::Anonymizer::GetFile ( ) [inline]
```

#### Examples

[BasicAnonymizer.cs](#), and [ManipulateFile.cs](#).

### 10.8.3.12 New()

```
static SmartPointer< Anonymizer > gdcM::Anonymizer::New ( ) [inline], [static]
```

for wrapped language: instantiate a reference counted object

#### Examples

[BasicAnonymizer.cs](#), and [ClinicalTrialIdentificationWorkflow.cs](#).

### 10.8.3.13 RecurseDataSet()

```
void gdcM::Anonymizer::RecurseDataSet (
    DataSet & ds ) [protected]
```

### 10.8.3.14 Remove() [1/2]

```
bool gdcM::Anonymizer::Remove (
    PrivateTag const & pt )
```

remove a private tag (even a SQ can be removed) Pay special attention that this code must be done before any call to Empty/Remove of the associated Private Creator, but before any call to Replace. When the private reservation becomes empty, no check is done to automatically remove the private creator

### 10.8.3.15 Remove() [2/2]

```
bool gdcM::Anonymizer::Remove (
    Tag const & t )
```

remove a tag (even a SQ can be removed)

### 10.8.3.16 RemoveGroupLength()

```
bool gdcm::Anonymizer::RemoveGroupLength ( )
```

Main function that loop over all elements and remove group length.

#### Examples

[ClinicalTrialAnnotate.cxx](#), and [ManipulateFile.cs](#).

### 10.8.3.17 RemovePrivateTags()

```
bool gdcm::Anonymizer::RemovePrivateTags ( )
```

Main function that loop over all elements and remove private tags.

#### Examples

[ClinicalTrialAnnotate.cxx](#), and [ManipulateFile.cs](#).

### 10.8.3.18 RemoveRetired()

```
bool gdcm::Anonymizer::RemoveRetired ( )
```

Main function that loop over all elements and remove retired element.

### 10.8.3.19 Replace() [1/4]

```
bool gdcm::Anonymizer::Replace (
    PrivateTag const & t,
    const char * value )
```

### 10.8.3.20 Replace() [2/4]

```
bool gdcm::Anonymizer::Replace (
    PrivateTag const & t,
    const char * value,
    VL const & vl )
```

### 10.8.3.21 Replace() [3/4]

```
bool gdcM::Anonymizer::Replace (
    Tag const & t,
    const char * value )
```

Replace tag with another value, if tag is not found it will be created: WARNING: this function can only execute if tag is a VRASCI

#### Examples

[ClinicalTrialAnnotate.cxx](#), [CreateJPIPDataSet.cxx](#), [EncapsulateFileInRawData.cxx](#), [ManipulateFile.cs](#), and [MpegVideoInfo.cs](#).

### 10.8.3.22 Replace() [4/4]

```
bool gdcM::Anonymizer::Replace (
    Tag const & t,
    const char * value,
    VL const & vl )
```

when the value contains \0, it is a good idea to specify the length. This function is required when dealing with VRBINARY tag

### 10.8.3.23 SetCryptographicMessageSyntax()

```
void gdcM::Anonymizer::SetCryptographicMessageSyntax (
    CryptographicMessageSyntax * cms )
```

Set/Get CMS key that will be used to encrypt the dataset within BasicApplicationLevelConfidentialityProfile.

#### Examples

[BasicAnonymizer.cs](#), and [ClinicalTrialIdentificationWorkflow.cs](#).

### 10.8.3.24 SetFile()

```
void gdcM::Anonymizer::SetFile (
    const File & f ) [inline]
```

Set/Get [File](#).

#### Examples

[BasicAnonymizer.cs](#), [ClinicalTrialAnnotate.cxx](#), [CreateJPIPDataSet.cxx](#), [EncapsulateFileInRawData.cxx](#), [ManipulateFile.cs](#), and [MpegVideoInfo.cs](#).

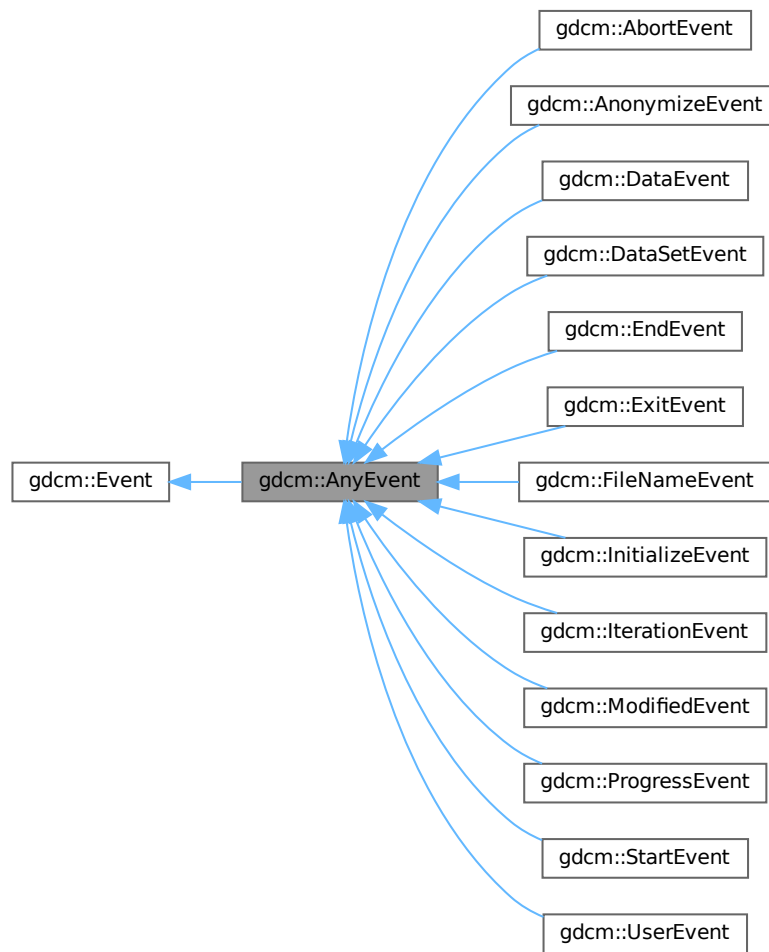
The documentation for this class was generated from the following file:

- [gdcMAnonymizer.h](#)

## 10.9 gdcmm::AnyEvent Class Reference

```
#include <gdcmmEvent.h>
```

Inheritance diagram for gdcmm::AnyEvent:



Collaboration diagram for `gdcm::AnyEvent`:



### Additional Inherited Members

### Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()
- [Event](#) (const [Event](#) &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) \*) const =0
- virtual const char \* [GetEventName](#) () const =0
- virtual [Event](#) \* [MakeObject](#) () const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

The documentation for this class was generated from the following file:

- [gdcmEvent.h](#)

## 10.10 `gdcm::network::ApplicationContext` Class Reference

[ApplicationContext](#).

```
#include <gdcmApplicationContext.h>
```

### Public Member Functions

- [ApplicationContext](#) ()
- const char \* [GetName](#) () const
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [SetName](#) (const char \*name)
- size\_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

## 10.10.1 Detailed Description

[ApplicationContext](#).

Table 9-12 APPLICATION CONTEXT ITEM FIELDS

**Todo** Looks like Application Context can only be 64 bytes at max (see Figure 9-1 / PS 3.8 - 2009 )

## 10.10.2 Constructor & Destructor Documentation

### 10.10.2.1 ApplicationContext()

```
gdcm::network::ApplicationContext::ApplicationContext ( )
```

## 10.10.3 Member Function Documentation

### 10.10.3.1 GetName()

```
const char * gdcm::network::ApplicationContext::GetName ( ) const [inline]
```

### 10.10.3.2 Print()

```
void gdcm::network::ApplicationContext::Print (
    std::ostream & os ) const
```

### 10.10.3.3 Read()

```
std::istream & gdcm::network::ApplicationContext::Read (
    std::istream & is )
```

### 10.10.3.4 SetName()

```
void gdcm::network::ApplicationContext::SetName (
    const char * name ) [inline]
```

### 10.10.3.5 Size()

```
size_t gdcm::network::ApplicationContext::Size ( ) const
```

### 10.10.3.6 Write()

```
const std::ostream & gdcm::network::ApplicationContext::Write (
    std::ostream & os ) const
```

The documentation for this class was generated from the following file:

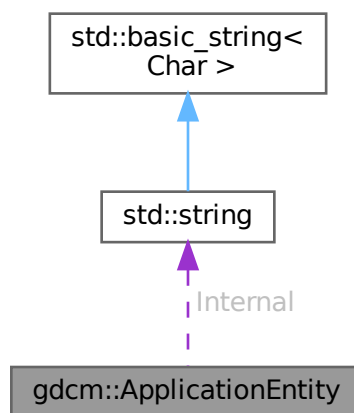
- [gdcmApplicationContext.h](#)

## 10.11 gdcm::ApplicationEntity Class Reference

[ApplicationEntity](#).

```
#include <gdcmApplicationEntity.h>
```

Collaboration diagram for gdcm::ApplicationEntity:



### Public Member Functions

- bool [IsValid](#) () const
- void [Print](#) (std::ostream &os) const
- void [SetBlob](#) (const std::vector< char > &v)
- void [Squeeze](#) ()

### Public Attributes

- std::string [Internal](#)



## Static Public Attributes

- static const unsigned int [MaxLength](#) = 16
- static const unsigned int [MaxNumberOfComponents](#) = 1
- static const char [Padding](#) = ''
- static const char [Separator](#) = ''

### 10.11.1 Detailed Description

[ApplicationEntity](#).

- AE Application Entity
- A string of characters that identifies an Application Entity with leading and trailing spaces (20H) being non-significant. A value consisting solely of spaces shall not be used.
- Default Character Repertoire excluding character code 5CH (the BACKSLASH \ in ISO-IR 6), and control characters LF, FF, CR and ESC.
- 16 bytes maximum

### 10.11.2 Member Function Documentation

#### 10.11.2.1 IsValid()

```
bool gdcmm::ApplicationEntity::IsValid ( ) const [inline]
```

#### 10.11.2.2 Print()

```
void gdcmm::ApplicationEntity::Print (
    std::ostream & os ) const [inline]
```

#### 10.11.2.3 SetBlob()

```
void gdcmm::ApplicationEntity::SetBlob (
    const std::vector< char > & v ) [inline]
```

#### 10.11.2.4 Squeeze()

```
void gdcmm::ApplicationEntity::Squeeze ( ) [inline]
```

### 10.11.3 Member Data Documentation

#### 10.11.3.1 Internal

```
std::string gdcM::ApplicationEntity::Internal
```

#### 10.11.3.2 MaxLength

```
const unsigned int gdcM::ApplicationEntity::MaxLength = 16 [static]
```

#### 10.11.3.3 MaxNumberOfComponents

```
const unsigned int gdcM::ApplicationEntity::MaxNumberOfComponents = 1 [static]
```

#### 10.11.3.4 Padding

```
const char gdcM::ApplicationEntity::Padding = ' ' [static]
```

#### 10.11.3.5 Separator

```
const char gdcM::ApplicationEntity::Separator = ' ' [static]
```

The documentation for this class was generated from the following file:

- [gdcMApplicationEntity.h](#)

## 10.12 gdcM::network::AReleaseRPPDU Class Reference

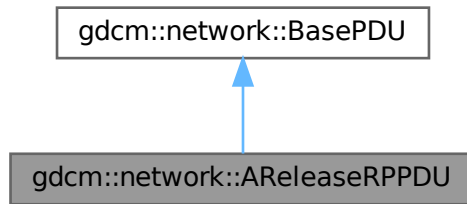
[AReleaseRPPDU](#).

```
#include <gdcMAReleaseRPPDU.h>
```

Inheritance diagram for gdcM::network::AReleaseRPPDU:



Collaboration diagram for gdcmm::network::AReleaseRPPDU:



### Public Member Functions

- [AReleaseRPPDU](#) ()
- bool [IsLastFragment](#) () const override
- void [Print](#) (std::ostream &os) const override
- std::istream & [Read](#) (std::istream &is) override
- size\_t [Size](#) () const override
- const std::ostream & [Write](#) (std::ostream &os) const override

### Public Member Functions inherited from [gdcmm::network::BasePDU](#)

- virtual [~BasePDU](#) ()=default

## 10.12.1 Detailed Description

[AReleaseRPPDU](#).

[Table 9-25](#) A-RELEASE-RP PDU fields

## 10.12.2 Constructor & Destructor Documentation

### 10.12.2.1 AReleaseRPPDU()

```
gdcmm::network::AReleaseRPPDU::AReleaseRPPDU ( )
```

## 10.12.3 Member Function Documentation

### 10.12.3.1 IsLastFragment()

```
bool gdcn::network::AReleaseRPPDU::IsLastFragment ( ) const [inline], [override], [virtual]
```

Implements [gdcn::network::BasePDU](#).

### 10.12.3.2 Print()

```
void gdcn::network::AReleaseRPPDU::Print (
    std::ostream & os ) const [override], [virtual]
```

Implements [gdcn::network::BasePDU](#).

### 10.12.3.3 Read()

```
std::istream & gdcn::network::AReleaseRPPDU::Read (
    std::istream & is ) [override], [virtual]
```

Implements [gdcn::network::BasePDU](#).

### 10.12.3.4 Size()

```
size_t gdcn::network::AReleaseRPPDU::Size ( ) const [override], [virtual]
```

Implements [gdcn::network::BasePDU](#).

### 10.12.3.5 Write()

```
const std::ostream & gdcn::network::AReleaseRPPDU::Write (
    std::ostream & os ) const [override], [virtual]
```

Implements [gdcn::network::BasePDU](#).

The documentation for this class was generated from the following file:

- [gdcnAReleaseRPPDU.h](#)

## 10.13 gdcmm::network::AReleaseRQPDU Class Reference

[AReleaseRQPDU](#).

```
#include <gdcmmAReleaseRQPDU.h>
```

Inheritance diagram for gdcmm::network::AReleaseRQPDU:



Collaboration diagram for gdcmm::network::AReleaseRQPDU:



### Public Member Functions

- [AReleaseRQPDU](#) ()
- bool [IsLastFragment](#) () const override
- void [Print](#) (std::ostream &os) const override
- std::istream & [Read](#) (std::istream &is) override
- size\_t [Size](#) () const override
- const std::ostream & [Write](#) (std::ostream &os) const override

### Public Member Functions inherited from [gdcmm::network::BasePDU](#)

- virtual [~BasePDU](#) ()=default

### 10.13.1 Detailed Description

[AReleaseRQPDU](#).

[Table 9-24](#) A-RELEASE-RQ PDU FIELDS

### 10.13.2 Constructor & Destructor Documentation

#### 10.13.2.1 AReleaseRQPDU()

```
gdcm::network::AReleaseRQPDU::AReleaseRQPDU ( )
```

### 10.13.3 Member Function Documentation

#### 10.13.3.1 IsLastFragment()

```
bool gdcm::network::AReleaseRQPDU::IsLastFragment ( ) const [inline], [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

#### 10.13.3.2 Print()

```
void gdcm::network::AReleaseRQPDU::Print (
    std::ostream & os ) const [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

#### 10.13.3.3 Read()

```
std::istream & gdcm::network::AReleaseRQPDU::Read (
    std::istream & is ) [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

#### 10.13.3.4 Size()

```
size_t gdcm::network::AReleaseRQPDU::Size ( ) const [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

### 10.13.3.5 Write()

```
const std::ostream & gdcm::network::AReleaseRQPDU::Write (
    std::ostream & os ) const [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

The documentation for this class was generated from the following file:

- [gdcmAReleaseRQPDU.h](#)

## 10.14 gdcm::network::ARTIMTimer Class Reference

[ARTIMTimer](#).

```
#include <gdcmARTIMTimer.h>
```

### Public Member Functions

- [ARTIMTimer](#) ()
- double [GetElapsedTime](#) () const
- bool [GetHasExpired](#) () const
- double [GetTimeout](#) () const
- void [SetTimeout](#) (double inTimeout)
- void [Start](#) ()
- void [Stop](#) ()

### 10.14.1 Detailed Description

[ARTIMTimer](#).

This file contains the code for the ARTIM timer.

Basically, the ARTIM timer will just get the wall time when it's started, and then can be queried for the current time, and then can be stopped (ie, the start time reset).

Because we're trying to do this without threading, we should be able to 'start' the ARTIM timer by this mechanism, and then when waiting for a particular response, tight loop that with sleep calls and determinations of when the ARTIM timer has reached its peak. As such, this isn't a strict 'timer' in the traditional sense of the word, but more of a time keeper.

There can be only one ARTIM timer per connection.

### 10.14.2 Constructor & Destructor Documentation

#### 10.14.2.1 ARTIMTimer()

```
gdcm::network::ARTIMTimer::ARTIMTimer ( )
```

### 10.14.3 Member Function Documentation

#### 10.14.3.1 GetElapsedTime()

```
double gdcm::network::ARTIMTimer::GetElapsedTime ( ) const
```

#### 10.14.3.2 GetHasExpired()

```
bool gdcm::network::ARTIMTimer::GetHasExpired ( ) const
```

#### 10.14.3.3 GetTimeout()

```
double gdcm::network::ARTIMTimer::GetTimeout ( ) const
```

#### 10.14.3.4 SetTimeout()

```
void gdcm::network::ARTIMTimer::SetTimeout (
    double inTimeout )
```

#### 10.14.3.5 Start()

```
void gdcm::network::ARTIMTimer::Start ( )
```

#### 10.14.3.6 Stop()

```
void gdcm::network::ARTIMTimer::Stop ( )
```

The documentation for this class was generated from the following file:

- [gdcmARTIMTimer.h](#)

## 10.15 gdcm::ASN1 Class Reference

Class for [ASN1](#).

```
#include <gdcmASN1.h>
```



## Public Member Functions

- [ASN1](#) ()
- [ASN1](#) (const [ASN1](#) &)=delete
- [~ASN1](#) ()
- void [operator=](#) (const [ASN1](#) &)=delete

## Static Public Member Functions

- static bool [ParseDump](#) (const char \*array, size\_t length)
- static bool [ParseDumpFile](#) (const char \*filename)

## Protected Member Functions

- int [TestPBKDF2](#) ()

### 10.15.1 Detailed Description

Class for [ASN1](#).

### 10.15.2 Constructor & Destructor Documentation

#### 10.15.2.1 [ASN1\(\)](#) [1/2]

```
gdcn::ASN1::ASN1 ( )
```

#### 10.15.2.2 [~ASN1\(\)](#)

```
gdcn::ASN1::~~ASN1 ( )
```

#### 10.15.2.3 [ASN1\(\)](#) [2/2]

```
gdcn::ASN1::ASN1 (
    const ASN1 & ) [delete]
```

### 10.15.3 Member Function Documentation

#### 10.15.3.1 [operator=\(\)](#)

```
void gdcn::ASN1::operator= (
    const ASN1 & ) [delete]
```

### 10.15.3.2 ParseDump()

```
static bool gdcm::ASN1::ParseDump (
    const char * array,
    size_t length ) [static]
```

### 10.15.3.3 ParseDumpFile()

```
static bool gdcm::ASN1::ParseDumpFile (
    const char * filename ) [static]
```

### 10.15.3.4 TestPBKDF2()

```
int gdcm::ASN1::TestPBKDF2 ( ) [protected]
```

The documentation for this class was generated from the following file:

- [gdcmASN1.h](#)

## 10.16 gdcm::network::AsynchronousOperationsWindowSub Class Reference

[AsynchronousOperationsWindowSub.](#)

```
#include <gdcmAsynchronousOperationsWindowSub.h>
```

### Public Member Functions

- [AsynchronousOperationsWindowSub](#) ()
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- size\_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

### 10.16.1 Detailed Description

[AsynchronousOperationsWindowSub.](#)

PS 3.7 [Table](#) D.3-7 ASYNCHRONOUS OPERATIONS WINDOW SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

## 10.16.2 Constructor & Destructor Documentation

### 10.16.2.1 AsynchronousOperationsWindowSub()

```
gdcm::network::AsynchronousOperationsWindowSub::AsynchronousOperationsWindowSub ( )
```

## 10.16.3 Member Function Documentation

### 10.16.3.1 Print()

```
void gdcm::network::AsynchronousOperationsWindowSub::Print (
    std::ostream & os ) const
```

### 10.16.3.2 Read()

```
std::istream & gdcm::network::AsynchronousOperationsWindowSub::Read (
    std::istream & is )
```

### 10.16.3.3 Size()

```
size_t gdcm::network::AsynchronousOperationsWindowSub::Size ( ) const
```

### 10.16.3.4 Write()

```
const std::ostream & gdcm::network::AsynchronousOperationsWindowSub::Write (
    std::ostream & os ) const
```

The documentation for this class was generated from the following file:

- [gdcmAsynchronousOperationsWindowSub.h](#)

## 10.17 gdcm::Attribute< Group, Element, TVR, TVM > Class Template Reference

[Attribute](#) class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary.

```
#include <gdcmAttribute.h>
```

Collaboration diagram for gdcm::Attribute< Group, Element, TVR, TVM >:



### Public Types

- enum { `VMType` = `VMToLength<TVM>::Length` }
- typedef `VRToType< TVR >::Type` `ArrayType`

### Public Member Functions

- `GDCM_STATIC_ASSERT` (((((`VR::VRType`) `TVR` & `VR::VR_VM1`) & ((`VM::VMType`) `TVM` == `VM::VM1`)) || !((`VR::VRType`) `TVR` & `VR::VR_VM1`)))
- `GDCM_STATIC_ASSERT` (((`VM::VMType`) `TVM` & (`VM::VMType`) (`TagToType< Group, Element >::VMType`)))
- `GDCM_STATIC_ASSERT` (((`VR::VRType`) `TVR` & (`VR::VRType`) (`TagToType< Group, Element >::VRType`)))
- `DataElement` `GetAsDataElement` () const
- unsigned int `GetNumberOfValues` () const
- `ArrayType` & `GetValue` (unsigned int idx=0)
- `ArrayType` const & `GetValue` (unsigned int idx=0) const
- const `ArrayType` \* `GetValues` () const

- bool [operator!=](#) (const [Attribute](#) &att) const
- bool [operator<](#) (const [Attribute](#) &att) const
- bool [operator==](#) (const [Attribute](#) &att) const
- [ArrayType](#) & [operator\[\]](#) (unsigned int idx)
- [ArrayType](#) const & [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &os) const
- void [Set](#) ([DataSet](#) const &ds)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetFromDataSet](#) ([DataSet](#) const &ds)
- void [SetValue](#) ([ArrayType](#) v, unsigned int idx=0)
- void [SetValues](#) (const [ArrayType](#) \*array, unsigned int numel=[VMType](#))

### Static Public Member Functions

- static [VM GetDictVM](#) ()
- static [VR GetDictVR](#) ()
- static [Tag GetTag](#) ()
- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

### Public Attributes

- [ArrayType Internal](#) [[VMToLength](#)< TVM >::Length]

### Protected Member Functions

- void [SetByteValue](#) (const [ByteValue](#) \*bv)
- void [SetByteValueNoSwap](#) (const [ByteValue](#) \*bv)

## 10.17.1 Detailed Description

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM
= TagToType<Group, Element>::VMType>
class gdcm::Attribute< Group, Element, TVR, TVM >
```

[Attribute](#) class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary.

Typical example that compile is: `Attribute<0x0008,0x9007> a = {"ORIGINAL","PRIMARY","T1","NONE"};`

Examples that will NOT compile are:

`Attribute<0x0018,0x1182, VR::IS, VM::VM1> fd1 = {};` // not enough parameters  
`Attribute<0x0018,0x1182, VR::IS, VM::VM2> fd2 = {0,1,2};` // too many initializers  
`Attribute<0x0018,0x1182, VR::IS, VM::VM3> fd3 = {0,1,2};` // VM3 is not valid  
`Attribute<0x0018,0x1182, VR::UL, VM::VM2> fd3 = {0,1};` // UL is not valid [VR](#)

### Examples

[CreateFakeRTDOSE.cxx](#), [CreateJIPIDataSet.cxx](#), [DeriveSeries.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_In](#), [FixOrientation.cxx](#), [GenFakeIdentifyFile.cxx](#), [GetSequenceUltrasound.cxx](#), [HelloWorld.cxx](#), [LargeVRDSEExplicit.cxx](#), [PatchFile.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [SortImage.cxx](#), [StreamImageReaderTest.cxx](#), [VolumeSorter.cxx](#), [gdcmrtpionplan.cxx](#), [gdcmrtpplan.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

## 10.17.2 Member Typedef Documentation

### 10.17.2.1 ArrayType

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, TVM >::ArrayType
```

## 10.17.3 Member Enumeration Documentation

### 10.17.3.1 anonymous enum

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
anonymous enum
```

#### Enumerator

VMType	
--------	--

## 10.17.4 Member Function Documentation

### 10.17.4.1 GDCM\_STATIC\_ASSERT() [1/3]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
gdcm::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (
    (((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TVM==VM::VM1))||!((VR::VRType) TVR
&VR::VR_VM1)) )
```

### 10.17.4.2 GDCM\_STATIC\_ASSERT() [2/3]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
gdcm::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (
    ((VM::VMType) TVM &(VM::VMType) (TagToType< Group, Element >::VMType)) )
```

### 10.17.4.3 GDCM\_STATIC\_ASSERT() [3/3]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
gdcm::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (
    ((VR::VRType) TVR &(VR::VRType) (TagToType< Group, Element >::VRType)) )
```

#### 10.17.4.4 GetAsDataElement()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
DataElement gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement ( ) const [inline]
```

##### Examples

[CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FixOrientation.cxx](#), [GenFakeIdentifyFile.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [PatchFile.cxx](#), and [StreamImageReaderTest.cxx](#).

References [gdcm::DataElement::GetVR\(\)](#), [gdcm::DataElement::SetByteValue\(\)](#), and [gdcm::DataElement::SetVR\(\)](#).

#### 10.17.4.5 GetDictVM()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
static VM gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVM ( ) [inline], [static]
```

#### 10.17.4.6 GetDictVR()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
static VR gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVR ( ) [inline], [static]
```

#### 10.17.4.7 GetNumberOfValues()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
unsigned int gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues ( ) const [inline]
```

##### Examples

[LargeVRDSExplicit.cxx](#).

Referenced by [gdcm::Attribute< Group, Element, TVR, TVM >::operator<\(\)>](#), and [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator<\(\)>](#).

#### 10.17.4.8 GetTag()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
static Tag gdcm::Attribute< Group, Element, TVR, TVM >::GetTag ( ) [inline], [static]
```

##### Examples

[PatchFile.cxx](#), [ReadAndPrintAttributes.cxx](#), [gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

**10.17.4.9 GetValue() [1/2]**

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
ArrayType & gdcM::Attribute< Group, Element, TVR, TVM >::GetValue (
    unsigned int idx = 0 ) [inline]
```

**Examples**

[DeriveSeries.cxx](#), [FixOrientation.cxx](#), [GetSequenceUltrasound.cxx](#), [PatchFile.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#),  
[ReadAndPrintAttributes.cxx](#), [gdcMrtionplan.cxx](#), [gdcMrtplan.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

**10.17.4.10 GetValue() [2/2]**

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
ArrayType const & gdcM::Attribute< Group, Element, TVR, TVM >::GetValue (
    unsigned int idx = 0 ) const [inline]
```

**10.17.4.11 GetValues()**

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
const ArrayType * gdcM::Attribute< Group, Element, TVR, TVM >::GetValues ( ) const [inline]
```

**Examples**

[FixOrientation.cxx](#), [LargeVRDSExplicit.cxx](#), [gdcMrtionplan.cxx](#), and [gdcMrtplan.cxx](#).

Referenced by [gdcM::Attribute< Group, Element, TVR, TVM >::operator!=\(\)](#), [gdcM::Attribute< Group, Element, TVR, VM::VM1 >::operator!=\(\)](#), [gdcM::Attribute< Group, Element, TVR, TVM >::operator<\(\)](#), [gdcM::Attribute< Group, Element, TVR, VM::VM1 >::operator<\(\)](#), [gdcM::Attribute< Group, Element, TVR, TVM >::operator==\(\)](#), and [gdcM::Attribute< Group, Element, TVR, VM::VM1 >::operator==\(\)](#).

**10.17.4.12 GetVM()**

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
static VM gdcM::Attribute< Group, Element, TVR, TVM >::GetVM ( ) [inline], [static]
```

**10.17.4.13 GetVR()**

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
static VR gdcM::Attribute< Group, Element, TVR, TVM >::GetVR ( ) [inline], [static]
```



**10.17.4.14 operator"!=(**

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
bool gdcmm::Attribute< Group, Element, TVR, TVM >::operator!=(
    const Attribute< Group, Element, TVR, TVM > & att ) const [inline]
```

References [gdcmm::Attribute< Group, Element, TVR, TVM >::GetValues\(\)](#).

**10.17.4.15 operator<(**

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
bool gdcmm::Attribute< Group, Element, TVR, TVM >::operator< (
    const Attribute< Group, Element, TVR, TVM > & att ) const [inline]
```

References [gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues\(\)](#), and [gdcmm::Attribute< Group, Element, TVR, TVM >](#)

**10.17.4.16 operator==(**

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
bool gdcmm::Attribute< Group, Element, TVR, TVM >::operator==(
    const Attribute< Group, Element, TVR, TVM > & att ) const [inline]
```

References [gdcmm::Attribute< Group, Element, TVR, TVM >::GetValues\(\)](#).

**10.17.4.17 operator[]()** [1/2]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
ArrayType & gdcmm::Attribute< Group, Element, TVR, TVM >::operator[] (
    unsigned int idx ) [inline]
```

**10.17.4.18 operator[]()** [2/2]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
ArrayType const & gdcmm::Attribute< Group, Element, TVR, TVM >::operator[] (
    unsigned int idx ) const [inline]
```

**10.17.4.19 Print()**

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcmm::Attribute< Group, Element, TVR, TVM >::Print (
    std::ostream & os ) const [inline]
```

**10.17.4.20 Set()**

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::Set (
    DataSet const & ds ) [inline]
```

**Examples**

[LargeVRDSExplicit.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).

References [gdcm::DataSet::GetDataElement\(\)](#).

**10.17.4.21 SetByteValue()**

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue (
    const ByteValue * bv ) [inline], [protected]
```

References [gdcm::ByteValue::GetLength\(\)](#), and [gdcm::ByteValue::GetPointer\(\)](#).

**10.17.4.22 SetByteValueNoSwap()**

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap (
    const ByteValue * bv ) [inline], [protected]
```

References [gdcm::ByteValue::GetLength\(\)](#), and [gdcm::ByteValue::GetPointer\(\)](#).

**10.17.4.23 SetFromDataElement()**

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement (
    DataElement const & de ) [inline]
```

**Examples**

[GetSequenceUltrasound.cxx](#), [LargeVRDSExplicit.cxx](#), [PatchFile.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [gdcmrtionplan.cxx](#),  
and [gdcmrtplan.cxx](#).

References [gdcm::DataElement::GetByteValue\(\)](#), [gdcm::DataElement::GetTag\(\)](#), [gdcm::DataElement::GetVR\(\)](#), and  
[gdcm::DataElement::IsEmpty\(\)](#).

#### 10.17.4.24 SetFromDataSet()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet (
    DataSet const & ds ) [inline]
```

##### Examples

[DeriveSeries.cxx](#), [FixOrientation.cxx](#), [ReadAndPrintAttributes.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

References [gdcm::DataSet::FindDataElement\(\)](#), and [gdcm::DataSet::GetDataElement\(\)](#).

#### 10.17.4.25 SetValue()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::SetValue (
    ArrayType v,
    unsigned int idx = 0 ) [inline]
```

##### Examples

[CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [FixOrientation.cxx](#), [HelloWorld.cxx](#), [LargeVRDSEExplicit.cxx](#), and [PatchFile.cxx](#).

#### 10.17.4.26 SetValues()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::SetValues (
    const ArrayType * array,
    unsigned int numel = VMType ) [inline]
```

##### Examples

[FixOrientation.cxx](#), and [LargeVRDSEExplicit.cxx](#).

### 10.17.5 Member Data Documentation

#### 10.17.5.1 Internal

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
ArrayType gdcm::Attribute< Group, Element, TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

## 10.18 gdcm::Attribute< Group, Element, TVR, VM::VM1 > Class Template Reference

```
#include <gdcmAttribute.h>
```

Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1 >:



### Public Types

- enum { `VMType` = `VMToLength<VM::VM1>::Length` }
- typedef `VRTToType< TVR >::Type` `ArrayType`

### Public Member Functions

- `GDCM_STATIC_ASSERT` (((((VR::VRTType) TVR &VR::VR\_VM1) &&((VM::VMType) VM::VM1==VM::VM1))||!((VR::VRTType) TVR &VR::VR\_VM1)))
- `GDCM_STATIC_ASSERT` (((VM::VMType) VM::VM1 &(VM::VMType)(TagToType< Group, Element >::VMType)))
- `GDCM_STATIC_ASSERT` (((VR::VRTType) TVR &(VR::VRTType)(TagToType< Group, Element >::VRTType)))
- `GDCM_STATIC_ASSERT` (VMToLength< VM::VM1 >::Length==1)
- `DataElement` `GetAsDataElement` () const
- unsigned int `GetNumberOfValues` () const
- `ArrayType` & `GetValue` ()
- `ArrayType` const & `GetValue` () const
- const `ArrayType` \* `GetValues` () const
- bool `operator!=` (const `Attribute` &att) const
- bool `operator<` (const `Attribute` &att) const
- bool `operator==` (const `Attribute` &att) const
- void `Print` (std::ostream &os) const
- void `Set` (`DataSet` const &ds)
- void `SetFromDataElement` (`DataElement` const &de)
- void `SetFromDataSet` (`DataSet` const &ds)
- void `SetValue` (`ArrayType` v)

### Static Public Member Functions

- static [VM GetDictVM](#) ()
- static [VR GetDictVR](#) ()
- static [Tag GetTag](#) ()
- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

### Public Attributes

- [ArrayType](#) Internal

### Protected Member Functions

- void [SetByteValue](#) (const [ByteValue](#) \*bv)
- void [SetByteValueNoSwap](#) (const [ByteValue](#) \*bv)

## 10.18.1 Member Typedef Documentation

### 10.18.1.1 ArrayType

```
template<uint16_t Group, uint16_t Element, long long TVR>
typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, VM::VM1 >::ArrayType
```

## 10.18.2 Member Enumeration Documentation

### 10.18.2.1 anonymous enum

```
template<uint16_t Group, uint16_t Element, long long TVR>
anonymous enum
```

#### Enumerator

<a href="#">VMType</a>	
------------------------	--

## 10.18.3 Member Function Documentation

### 10.18.3.1 GDCM\_STATIC\_ASSERT() [1/4]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (
    (((VR::VRType) TVR &VR::VR\_VM1) && ((VM::VMType) VM::VM1==VM::VM1)) || !((VR::VRType)
TVR &VR::VR\_VM1)) )
```

**10.18.3.2 GDCM\_STATIC\_ASSERT() [2/4]**

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (
    ((VM::VMType) VM::VM1 & (VM::VMType) (TagToType< Group, Element >::VMType)) )
```

**10.18.3.3 GDCM\_STATIC\_ASSERT() [3/4]**

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (
    ((VR::VRType) TVR & (VR::VRType) (TagToType< Group, Element >::VRType)) )
```

**10.18.3.4 GDCM\_STATIC\_ASSERT() [4/4]**

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (
    VMToLength< VM::VM1 >::Length == 1 )
```

**10.18.3.5 GetAsDataElement()**

```
template<uint16_t Group, uint16_t Element, long long TVR>
DataElement gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement ( ) const [inline]
```

References [gdcmm::DataElement::GetVR\(\)](#), [gdcmm::DataElement::SetByteValue\(\)](#), and [gdcmm::DataElement::SetVR\(\)](#).

**10.18.3.6 GetDictVM()**

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VM gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetDictVM ( ) [inline], [static]
```

**10.18.3.7 GetDictVR()**

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VR gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetDictVR ( ) [inline], [static]
```

**10.18.3.8 GetNumberOfValues()**

```
template<uint16_t Group, uint16_t Element, long long TVR>
unsigned int gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetNumberOfValues ( ) const [inline]
```

### 10.18.3.9 GetTag()

```
template<uint16_t Group, uint16_t Element, long long TVR>
static Tag gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetTag ( ) [inline], [static]
```

### 10.18.3.10 GetValue() [1/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType & gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetValue ( ) [inline]
```

### 10.18.3.11 GetValue() [2/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType const & gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetValue ( ) const [inline]
```

### 10.18.3.12 GetValues()

```
template<uint16_t Group, uint16_t Element, long long TVR>
const ArrayType * gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetValues ( ) const [inline]
```

### 10.18.3.13 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VM gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetVM ( ) [inline], [static]
```

### 10.18.3.14 GetVR()

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VR gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetVR ( ) [inline], [static]
```

### 10.18.3.15 operator"!="()

```
template<uint16_t Group, uint16_t Element, long long TVR>
bool gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator!= (
    const Attribute< Group, Element, TVR, VM::VM1 > & att ) const [inline]
```

References [gdcmm::Attribute< Group, Element, TVR, TVM >::GetValues\(\)](#).

**10.18.3.16 operator<()**

```
template<uint16_t Group, uint16_t Element, long long TVR>
bool gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator< (
    const Attribute< Group, Element, TVR, VM::VM1 > & att ) const [inline]
```

References [gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues\(\)](#), and [gdcm::Attribute< Group, Element, TVR, TVM >](#)

**10.18.3.17 operator==(**

```
template<uint16_t Group, uint16_t Element, long long TVR>
bool gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator==( (
    const Attribute< Group, Element, TVR, VM::VM1 > & att ) const [inline]
```

References [gdcm::Attribute< Group, Element, TVR, TVM >::GetValues\(\)](#).

**10.18.3.18 Print()**

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Print (
    std::ostream & os ) const [inline]
```

**10.18.3.19 Set()**

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Set (
    DataSet const & ds ) [inline]
```

References [gdcm::DataSet::GetDataElement\(\)](#).

**10.18.3.20 SetByteValue()**

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue (
    const ByteValue * bv ) [inline], [protected]
```

References [gdcm::ByteValue::GetLength\(\)](#), and [gdcm::ByteValue::GetPointer\(\)](#).

**10.18.3.21 SetByteValueNoSwap()**

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap (
    const ByteValue * bv ) [inline], [protected]
```

References [gdcm::ByteValue::GetLength\(\)](#), and [gdcm::ByteValue::GetPointer\(\)](#).



### 10.18.3.22 SetFromDataElement()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement (
    DataElement const & de ) [inline]
```

References [gdcm::DataElement::GetByteValue\(\)](#), [gdcm::DataElement::GetTag\(\)](#), [gdcm::DataElement::GetVR\(\)](#), and [gdcm::DataElement::IsEmpty\(\)](#).

### 10.18.3.23 SetFromDataSet()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet (
    DataSet const & ds ) [inline]
```

References [gdcm::DataSet::FindDataElement\(\)](#), and [gdcm::DataSet::GetDataElement\(\)](#).

### 10.18.3.24 SetValue()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetValue (
    ArrayType v ) [inline]
```

## 10.18.4 Member Data Documentation

### 10.18.4.1 Internal

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Internal
```

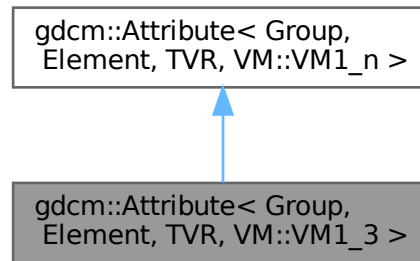
The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

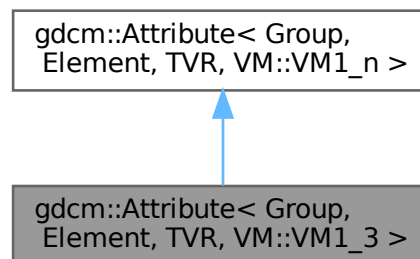
## 10.19 `gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >` Class Template Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for `gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >`:



Collaboration diagram for `gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >`:



### Public Member Functions

- [VM GetVM](#) () const

## Public Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >

- [Attribute](#) ()
- [~Attribute](#) ()
- [GDCM\\_STATIC\\_ASSERT](#) (((((VR::VRType) TVR &VR::VR\_VM1) &&((VM::VMType) TagToType< Group, [Element](#) >::VMType==VM::VM1))||!((VR::VRType) TVR &VR::VR\_VM1)))
- [GDCM\\_STATIC\\_ASSERT](#) (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, [Element](#) >::VRType)))
- [GDCM\\_STATIC\\_ASSERT](#) ((VM::VM1\_n &(VM::VMType)(TagToType< Group, [Element](#) >::VMType)))
- [DataElement](#) [GetAsDataElement](#) () const
- unsigned int [GetNumberOfValues](#) () const
- [ArrayType](#) & [GetValue](#) (unsigned int idx=0)
- [ArrayType](#) const & [GetValue](#) (unsigned int idx=0) const
- const [ArrayType](#) \* [GetValues](#) () const
- [ArrayType](#) & [operator\[\]](#) (unsigned int idx)
- [ArrayType](#) const & [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &os) const
- void [Set](#) ([DataSet](#) const &ds)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetFromDataSet](#) ([DataSet](#) const &ds)
- void [SetNumberOfValues](#) (unsigned int numel)
- void [SetValue](#) ([ArrayType](#) v)
- void [SetValue](#) (unsigned int idx, [ArrayType](#) v)
- void [SetValues](#) (const [ArrayType](#) \*array, unsigned int numel, bool own=false)

## Additional Inherited Members

## Public Types inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >

- typedef [VRToType](#)< TVR >::Type [ArrayType](#)

## Static Public Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >

- static [VM](#) [GetDictVM](#) ()
- static [VR](#) [GetDictVR](#) ()
- static [Tag](#) [GetTag](#) ()
- static [VM](#) [GetVM](#) ()
- static [VR](#) [GetVR](#) ()

## Protected Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >

- void [SetByteValue](#) (const [ByteValue](#) \*bv)

## 10.19.1 Member Function Documentation

### 10.19.1.1 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VM gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >::GetVM ( ) const [inline]
```

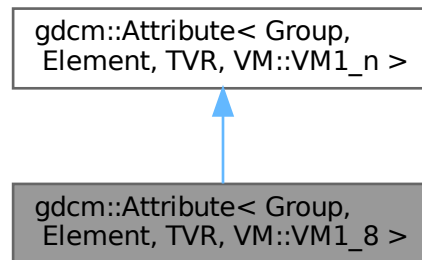
The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

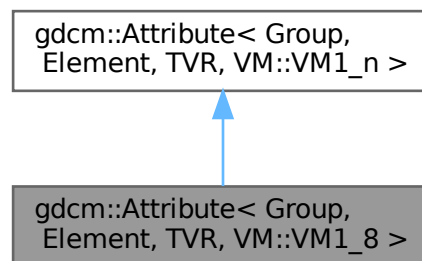
## 10.20 gdcm::Attribute< Group, Element, TVR, VM::VM1\_8 > Class Template Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1\_8 >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1\_8 >:



**Public Member Functions**

- [VM GetVM](#) () const

**Public Member Functions inherited from****[gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >](#)**

- [Attribute](#) ()
- [~Attribute](#) ()
- [GDCM\\_STATIC\\_ASSERT](#) (((((VR::VRType) TVR &VR::VR\_VM1) &&((VM::VMType) TagToType< Group, [Element](#) >::VMType==VM::VM1))||!((VR::VRType) TVR &VR::VR\_VM1)))
- [GDCM\\_STATIC\\_ASSERT](#) (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, [Element](#) >::VRType)))
- [GDCM\\_STATIC\\_ASSERT](#) ((VM::VM1\_n &(VM::VMType)(TagToType< Group, [Element](#) >::VMType)))
- [DataElement GetAsDataElement](#) () const
- unsigned int [GetNumberOfValues](#) () const
- [ArrayType](#) & [GetValue](#) (unsigned int idx=0)
- [ArrayType](#) const & [GetValue](#) (unsigned int idx=0) const
- const [ArrayType](#) \* [GetValues](#) () const
- [ArrayType](#) & [operator\[\]](#) (unsigned int idx)
- [ArrayType](#) const & [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &os) const
- void [Set](#) ([DataSet](#) const &ds)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetFromDataSet](#) ([DataSet](#) const &ds)
- void [SetNumberOfValues](#) (unsigned int numel)
- void [SetValue](#) ([ArrayType](#) v)
- void [SetValue](#) (unsigned int idx, [ArrayType](#) v)
- void [SetValues](#) (const [ArrayType](#) \*array, unsigned int numel, bool own=false)

**Additional Inherited Members****Public Types inherited from [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >](#)**

- typedef [VRToType](#)< TVR >::Type [ArrayType](#)

**Static Public Member Functions inherited from****[gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >](#)**

- static [VM GetDictVM](#) ()
- static [VR GetDictVR](#) ()
- static [Tag GetTag](#) ()
- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

**Protected Member Functions inherited from****[gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >](#)**

- void [SetByteValue](#) (const [ByteValue](#) \*bv)

## 10.20.1 Member Function Documentation

### 10.20.1.1 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VM gdcM::Attribute< Group, Element, TVR, VM::VM1_8 >::GetVM ( ) const [inline]
```

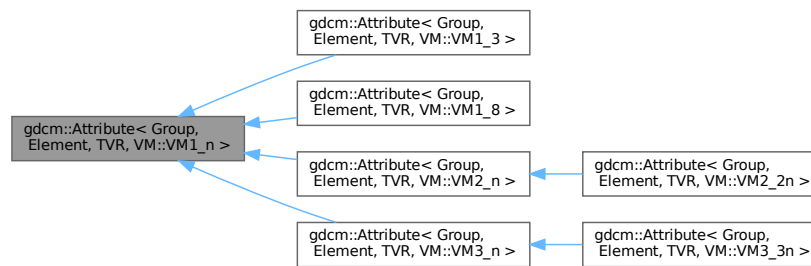
The documentation for this class was generated from the following file:

- [gdcMAttribute.h](#)

## 10.21 gdcM::Attribute< Group, Element, TVR, VM::VM1\_n > Class Template Reference

```
#include <gdcMAttribute.h>
```

Inheritance diagram for gdcM::Attribute< Group, Element, TVR, VM::VM1\_n >:



### Public Types

- typedef `VRToType< TVR >::Type` `ArrayType`

### Public Member Functions

- `Attribute ( )`
- `~Attribute ( )`
- `GDCM_STATIC_ASSERT (((((VR::VRType) TVR & VR::VR_VM1) && ((VM::VMType) TagToType< Group, Element >::VMType==VM::VM1)) || !((VR::VRType) TVR & VR::VR_VM1)))`
- `GDCM_STATIC_ASSERT (((VR::VRType) TVR & (VR::VRType)(TagToType< Group, Element >::VRType)))`
- `GDCM_STATIC_ASSERT ((VM::VM1_n & (VM::VMType)(TagToType< Group, Element >::VMType)))`
- `DataElement GetAsDataElement ( ) const`
- `unsigned int GetNumberOfValues ( ) const`
- `ArrayType & GetValue (unsigned int idx=0)`

- [ArrayType](#) const & [GetValue](#) (unsigned int idx=0) const
- const [ArrayType](#) \* [GetValues](#) () const
- [ArrayType](#) & [operator\[\]](#) (unsigned int idx)
- [ArrayType](#) const & [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &os) const
- void [Set](#) ([DataSet](#) const &ds)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetFromDataSet](#) ([DataSet](#) const &ds)
- void [SetNumberOfValues](#) (unsigned int numel)
- void [SetValue](#) ([ArrayType](#) v)
- void [SetValue](#) (unsigned int idx, [ArrayType](#) v)
- void [SetValues](#) (const [ArrayType](#) \*array, unsigned int numel, bool own=false)

### Static Public Member Functions

- static [VM](#) [GetDictVM](#) ()
- static [VR](#) [GetDictVR](#) ()
- static [Tag](#) [GetTag](#) ()
- static [VM](#) [GetVM](#) ()
- static [VR](#) [GetVR](#) ()

### Protected Member Functions

- void [SetByteValue](#) (const [ByteValue](#) \*bv)

## 10.21.1 Member Typedef Documentation

### 10.21.1.1 ArrayType

```
template<uint16_t Group, uint16_t Element, long long TVR>
typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >::ArrayType
```

## 10.21.2 Constructor & Destructor Documentation

### 10.21.2.1 Attribute()

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >::Attribute ( ) [inline], [explicit]
```

### 10.21.2.2 ~Attribute()

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >::~~Attribute ( ) [inline]
```

## 10.21.3 Member Function Documentation

### 10.21.3.1 GDCM\_STATIC\_ASSERT() [1/3]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GDCM_STATIC_ASSERT (
    (((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TagToType< Group, Element >::VMType==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)) )
```

### 10.21.3.2 GDCM\_STATIC\_ASSERT() [2/3]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GDCM_STATIC_ASSERT (
    ((VR::VRType) TVR &(VR::VRType) (TagToType< Group, Element >::VRType)) )
```

### 10.21.3.3 GDCM\_STATIC\_ASSERT() [3/3]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GDCM_STATIC_ASSERT (
    (VM::VM1_n &(VM::VMType) (TagToType< Group, Element >::VMType)) )
```

### 10.21.3.4 GetAsDataElement()

```
template<uint16_t Group, uint16_t Element, long long TVR>
DataElement gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement ( ) const [inline]
```

References [gdcmm::DataElement::GetVR\(\)](#), [gdcmm::DataElement::SetByteValue\(\)](#), and [gdcmm::DataElement::SetVR\(\)](#).

### 10.21.3.5 GetDictVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VM gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVM ( ) [inline], [static]
```

### 10.21.3.6 GetDictVR()

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VR gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVR ( ) [inline], [static]
```

### 10.21.3.7 GetNumberOfValues()

```
template<uint16_t Group, uint16_t Element, long long TVR>
unsigned int gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetNumberOfValues ( ) const [inline]
```



**10.21.3.8 GetTag()**

```
template<uint16_t Group, uint16_t Element, long long TVR>
static Tag gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetTag ( ) [inline], [static]
```

**10.21.3.9 GetValue() [1/2]**

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType & gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue (
    unsigned int idx = 0 ) [inline]
```

**10.21.3.10 GetValue() [2/2]**

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType const & gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue (
    unsigned int idx = 0 ) const [inline]
```

**10.21.3.11 GetValues()**

```
template<uint16_t Group, uint16_t Element, long long TVR>
const ArrayType * gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValues ( ) const [inline]
```

**10.21.3.12 GetVM()**

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VM gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetVM ( ) [inline], [static]
```

**10.21.3.13 GetVR()**

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VR gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetVR ( ) [inline], [static]
```

**10.21.3.14 operator[]() [1/2]**

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType & gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::operator[] (
    unsigned int idx ) [inline]
```

**10.21.3.15 operator[]() [2/2]**

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType const & gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::operator[] (
    unsigned int idx ) const [inline]
```

**10.21.3.16 Print()**

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Print (
    std::ostream & os ) const [inline]
```

**10.21.3.17 Set()**

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Set (
    DataSet const & ds ) [inline]
```

References [gdcm::DataSet::GetDataElement\(\)](#).

**10.21.3.18 SetByteValue()**

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue (
    const ByteValue * bv ) [inline], [protected]
```

References [gdcm::ByteValue::GetLength\(\)](#), and [gdcm::ByteValue::GetPointer\(\)](#).

**10.21.3.19 SetFromDataElement()**

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement (
    DataElement const & de ) [inline]
```

References [gdcm::DataElement::GetByteValue\(\)](#), [gdcm::DataElement::GetTag\(\)](#), [gdcm::DataElement::GetVR\(\)](#), and [gdcm::DataElement::IsEmpty\(\)](#).

**10.21.3.20 SetFromDataSet()**

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet (
    DataSet const & ds ) [inline]
```

References [gdcm::DataSet::FindDataElement\(\)](#), and [gdcm::DataSet::GetDataElement\(\)](#).

**10.21.3.21 SetNumberOfValues()**

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetNumberOfValues (
    unsigned int numel ) [inline]
```

### 10.21.3.22 SetValue() [1/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue (
    ArrayType v ) [inline]
```

References [SetValue\(\)](#).

Referenced by [SetValue\(\)](#).

### 10.21.3.23 SetValue() [2/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue (
    unsigned int idx,
    ArrayType v ) [inline]
```

### 10.21.3.24 SetValues()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::SetValues (
    const ArrayType * array,
    unsigned int numel,
    bool own = false ) [inline]
```

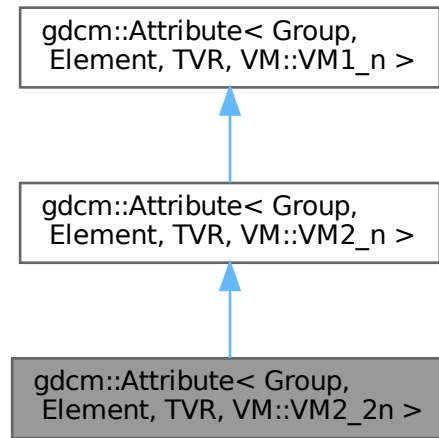
The documentation for this class was generated from the following file:

- [gdcMAttribute.h](#)

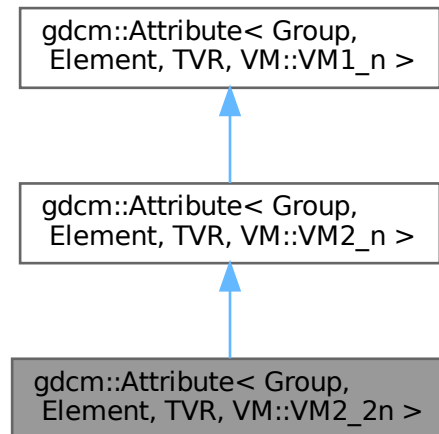
## 10.22 gdcM::Attribute< Group, Element, TVR, VM::VM2\_2n > Class Template Reference

```
#include <gdcMAttribute.h>
```

Inheritance diagram for `gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >`:



Collaboration diagram for `gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >`:



#### Static Public Member Functions

- static `VM GetVM ()`

## Static Public Member Functions inherited from [gdcmm::Attribute< Group, Element, TVR, VM::VM1\\_n >](#)

- static [VM GetDictVM](#) ()
- static [VR GetDictVR](#) ()
- static [Tag GetTag](#) ()
- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

## Additional Inherited Members

## Public Types inherited from [gdcmm::Attribute< Group, Element, TVR, VM::VM1\\_n >](#)

- typedef [VRToType< TVR >::Type](#) [ArrayType](#)

## Public Member Functions inherited from [gdcmm::Attribute< Group, Element, TVR, VM::VM2\\_n >](#)

- [VM GetVM](#) () const

## Public Member Functions inherited from [gdcmm::Attribute< Group, Element, TVR, VM::VM1\\_n >](#)

- [Attribute](#) ()
- [~Attribute](#) ()
- [GDCM\\_STATIC\\_ASSERT](#) (((((VR::VRType) TVR &VR::VR\_VM1) &&((VM::VMType) TagToType< Group, [Element](#) >::VMType==VM::VM1))||!((VR::VRType) TVR &VR::VR\_VM1)))
- [GDCM\\_STATIC\\_ASSERT](#) (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, [Element](#) >::VRType)))
- [GDCM\\_STATIC\\_ASSERT](#) ((VM::VM1\_n &(VM::VMType)(TagToType< Group, [Element](#) >::VMType)))
- [DataElement GetAsDataElement](#) () const
- unsigned int [GetNumberOfValues](#) () const
- [ArrayType](#) & [GetValue](#) (unsigned int idx=0)
- [ArrayType](#) const & [GetValue](#) (unsigned int idx=0) const
- const [ArrayType](#) \* [GetValues](#) () const
- [ArrayType](#) & [operator\[\]](#) (unsigned int idx)
- [ArrayType](#) const & [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &os) const
- void [Set](#) ([DataSet](#) const &ds)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetFromDataSet](#) ([DataSet](#) const &ds)
- void [SetNumberOfValues](#) (unsigned int numel)
- void [SetValue](#) ([ArrayType](#) v)
- void [SetValue](#) (unsigned int idx, [ArrayType](#) v)
- void [SetValues](#) (const [ArrayType](#) \*array, unsigned int numel, bool own=false)

## Protected Member Functions inherited from [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >](#)

- void [SetByteValue](#) (const [ByteValue](#) \*bv)

### 10.22.1 Member Function Documentation

#### 10.22.1.1 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VM gdcm::Attribute< Group, Element, TVR, VM::VM2\_2n >::GetVM ( ) [inline], [static]
```

The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

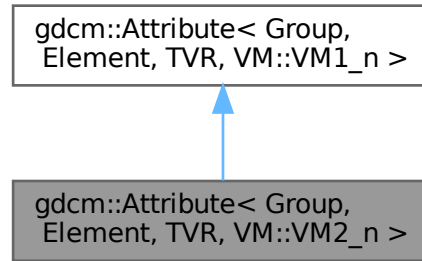
## 10.23 [gdcm::Attribute< Group, Element, TVR, VM::VM2\\_n >](#) Class Template Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for [gdcm::Attribute< Group, Element, TVR, VM::VM2\\_n >](#):



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM2\_n >:



### Public Member Functions

- [VM GetVM](#) () const

### Public Member Functions inherited from

[gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >](#)

- [Attribute](#) ()
- [~Attribute](#) ()
- [GDCM\\_STATIC\\_ASSERT](#) ((((((VR::VRType) TVR &VR::VR\_VM1) &&((VM::VMType) TagToType< Group, [Element](#) >::VMType==VM::VM1))||!((VR::VRType) TVR &VR::VR\_VM1))))
- [GDCM\\_STATIC\\_ASSERT](#) (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, [Element](#) >::VRType)))
- [GDCM\\_STATIC\\_ASSERT](#) ((VM::VM1\_n &(VM::VMType)(TagToType< Group, [Element](#) >::VMType)))
- [DataElement GetAsDataElement](#) () const
- unsigned int [GetNumberOfValues](#) () const
- [ArrayType](#) & [GetValue](#) (unsigned int idx=0)
- [ArrayType](#) const & [GetValue](#) (unsigned int idx=0) const
- const [ArrayType](#) \* [GetValues](#) () const
- [ArrayType](#) & [operator\[\]](#) (unsigned int idx)
- [ArrayType](#) const & [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &os) const
- void [Set](#) ([DataSet](#) const &ds)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetFromDataSet](#) ([DataSet](#) const &ds)
- void [SetNumberOfValues](#) (unsigned int numel)
- void [SetValue](#) ([ArrayType](#) v)
- void [SetValue](#) (unsigned int idx, [ArrayType](#) v)
- void [SetValues](#) (const [ArrayType](#) \*array, unsigned int numel, bool own=false)

### Additional Inherited Members

#### Public Types inherited from [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >](#)

- typedef [VRToType< TVR >::Type](#) [ArrayType](#)

#### Static Public Member Functions inherited from [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >](#)

- static [VM GetDictVM](#) ()
- static [VR GetDictVR](#) ()
- static [Tag GetTag](#) ()
- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

#### Protected Member Functions inherited from [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >](#)

- void [SetByteValue](#) (const [ByteValue](#) \*bv)

## 10.23.1 Member Function Documentation

### 10.23.1.1 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VM gdcm::Attribute< Group, Element, TVR, VM::VM2\_n >::GetVM ( ) const [inline]
```

The documentation for this class was generated from the following file:

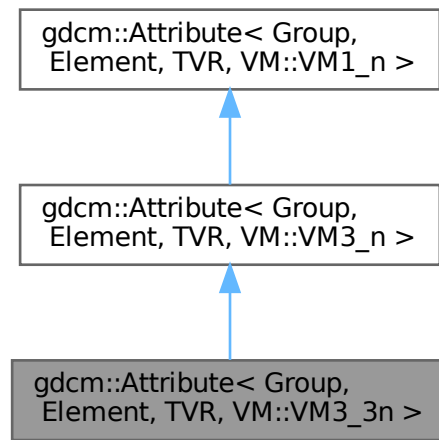
- [gdcmAttribute.h](#)

## 10.24 [gdcm::Attribute< Group, Element, TVR, VM::VM3\\_3n >](#) Class Template Reference

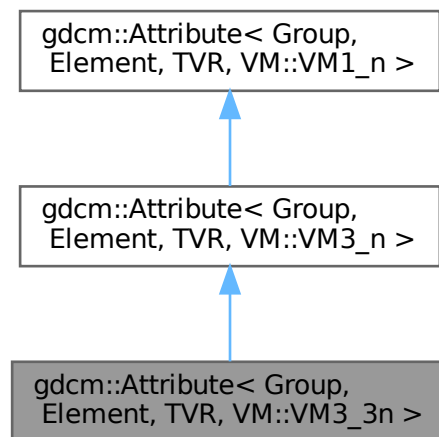
```
#include <gdcmAttribute.h>
```



Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM3\_3n >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM3\_3n >:



#### Static Public Member Functions

- static `VM GetVM ()`

### Static Public Member Functions inherited from [gdcm::Attribute< Group, Element, TVR, VM::VM3\\_n >](#)

- static [VM GetVM](#) ()

### Static Public Member Functions inherited from [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >](#)

- static [VM GetDictVM](#) ()
- static [VR GetDictVR](#) ()
- static [Tag GetTag](#) ()
- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

### Additional Inherited Members

### Public Types inherited from [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >](#)

- typedef [VRToType< TVR >::Type](#) [ArrayType](#)

### Public Member Functions inherited from [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >](#)

- [Attribute](#) ()
- [~Attribute](#) ()
- [GDCM\\_STATIC\\_ASSERT](#) (((((VR::VRType) TVR &VR::VR\_VM1) &&((VM::VMType) TagToType< Group, [Element](#) >::VMType==VM::VM1))||!((VR::VRType) TVR &VR::VR\_VM1)))
- [GDCM\\_STATIC\\_ASSERT](#) (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, [Element](#) >::VRType)))
- [GDCM\\_STATIC\\_ASSERT](#) ((VM::VM1\_n &(VM::VMType)(TagToType< Group, [Element](#) >::VMType)))
- [DataElement GetAsDataElement](#) () const
- unsigned int [GetNumberOfValues](#) () const
- [ArrayType](#) & [GetValue](#) (unsigned int idx=0)
- [ArrayType](#) const & [GetValue](#) (unsigned int idx=0) const
- const [ArrayType](#) \* [GetValues](#) () const
- [ArrayType](#) & [operator\[\]](#) (unsigned int idx)
- [ArrayType](#) const & [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &os) const
- void [Set](#) ([DataSet](#) const &ds)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetFromDataSet](#) ([DataSet](#) const &ds)
- void [SetNumberOfValues](#) (unsigned int numel)
- void [SetValue](#) ([ArrayType](#) v)
- void [SetValue](#) (unsigned int idx, [ArrayType](#) v)
- void [SetValues](#) (const [ArrayType](#) \*array, unsigned int numel, bool own=false)

## Protected Member Functions inherited from [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >](#)

- void [SetByteValue](#) (const [ByteValue](#) \*bv)

### 10.24.1 Member Function Documentation

#### 10.24.1.1 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VM gdcm::Attribute< Group, Element, TVR, VM::VM3\_3n >::GetVM ( ) [inline], [static]
```

The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

## 10.25 [gdcm::Attribute< Group, Element, TVR, VM::VM3\\_n >](#) Class Template Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for [gdcm::Attribute< Group, Element, TVR, VM::VM3\\_n >](#):



Collaboration diagram for `gdcm::Attribute< Group, Element, TVR, VM::VM3_n >`:



#### Static Public Member Functions

- static [VM GetVM](#) ()

#### Static Public Member Functions inherited from [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >](#)

- static [VM GetDictVM](#) ()
- static [VR GetDictVR](#) ()
- static [Tag GetTag](#) ()
- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

#### Additional Inherited Members

#### Public Types inherited from [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >](#)

- typedef [VRToType< TVR >::Type](#) [ArrayType](#)

#### Public Member Functions inherited from [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >](#)

- [Attribute](#) ()
- [~Attribute](#) ()
- [GDCM\\_STATIC\\_ASSERT](#) (((((VR::VRType) TVR &VR::VR\_VM1) &&((VM::VMType) TagToType< Group, [Element](#) >::VMType==VM::VM1))||!((VR::VRType) TVR &VR::VR\_VM1)))
- [GDCM\\_STATIC\\_ASSERT](#) (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, [Element](#) >::VRType)))
- [GDCM\\_STATIC\\_ASSERT](#) ((VM::VM1\_n &(VM::VMType)(TagToType< Group, [Element](#) >::VMType)))
- [DataElement GetAsDataElement](#) () const

- unsigned int [GetNumberOfValues](#) () const
- [ArrayType](#) & [GetValue](#) (unsigned int idx=0)
- [ArrayType](#) const & [GetValue](#) (unsigned int idx=0) const
- const [ArrayType](#) \* [GetValues](#) () const
- [ArrayType](#) & [operator\[\]](#) (unsigned int idx)
- [ArrayType](#) const & [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &os) const
- void [Set](#) ([DataSet](#) const &ds)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetFromDataSet](#) ([DataSet](#) const &ds)
- void [SetNumberOfValues](#) (unsigned int numel)
- void [SetValue](#) ([ArrayType](#) v)
- void [SetValue](#) (unsigned int idx, [ArrayType](#) v)
- void [SetValues](#) (const [ArrayType](#) \*array, unsigned int numel, bool own=false)

### Protected Member Functions inherited from [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >](#)

- void [SetByteValue](#) (const [ByteValue](#) \*bv)

## 10.25.1 Member Function Documentation

### 10.25.1.1 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VM gdcm::Attribute< Group, Element, TVR, VM::VM3_n >::GetVM ( ) [inline], [static]
```

The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

## 10.26 gdcm::AudioCodec Class Reference

[AudioCodec](#).

```
#include <gdcmAudioCodec.h>
```

Inheritance diagram for `gdcm::AudioCodec`:



Collaboration diagram for `gdcm::AudioCodec`:



## Public Member Functions

- [AudioCodec](#) ()
- [~AudioCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &) const override  
*Return whether this coder support this transfer syntax (can code it)*
- bool [CanDecode](#) ([TransferSyntax](#) const &) const override  
*Return whether this decoder support this transfer syntax (can decode it)*
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override  
*Decode.*

**Public Member Functions inherited from [gdcm::Coder](#)**

- virtual [~Coder](#) ()=default
- virtual bool [Code](#) ([DataElement](#) const &in\_, [DataElement](#) &out\_)  
*Code.*

**Public Member Functions inherited from [gdcm::Decoder](#)**

- virtual [~Decoder](#) ()=default

**Additional Inherited Members****Protected Member Functions inherited from [gdcm::Coder](#)**

- virtual bool [InternalCode](#) (const char \*bv, unsigned long len, std::ostream &os)

**Protected Member Functions inherited from [gdcm::Decoder](#)**

- virtual bool [DecodeByStreams](#) (std::istream &, std::ostream &)

**10.26.1 Detailed Description**

[AudioCodec](#).

**10.26.2 Constructor & Destructor Documentation****10.26.2.1 [AudioCodec\(\)](#)**

```
gdcm::AudioCodec::AudioCodec ( )
```

**10.26.2.2 [~AudioCodec\(\)](#)**

```
gdcm::AudioCodec::~~AudioCodec ( ) [override]
```

**10.26.3 Member Function Documentation****10.26.3.1 [CanCode\(\)](#)**

```
bool gdcm::AudioCodec::CanCode (
    TransferSyntax const & ) const [inline], [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Implements [gdcm::Coder](#).

### 10.26.3.2 CanDecode()

```
bool gdcm::AudioCodec::CanDecode (
    TransferSyntax const & ) const [inline], [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Implements [gdcm::Decoder](#).

### 10.26.3.3 Decode()

```
bool gdcm::AudioCodec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcm::Decoder](#).

The documentation for this class was generated from the following file:

- [gdcmAudioCodec.h](#)

## 10.27 gdcm::Base64 Class Reference

Class for [Base64](#).

```
#include <gdcmBase64.h>
```

### Public Member Functions

- [Base64](#) (const [Base64](#) &)=delete
- void [operator=](#) (const [Base64](#) &)=delete

### Static Public Member Functions

- static size\_t [Decode](#) (char \*dst, size\_t dlen, const char \*src, size\_t slen)  
*Decode a base64-formatted buffer.*
- static size\_t [Encode](#) (char \*dst, size\_t dlen, const char \*src, size\_t slen)  
*Encode a buffer into base64 format.*
- static size\_t [GetDecodeLength](#) (const char \*src, size\_t len)
- static size\_t [GetEncodeLength](#) (const char \*src, size\_t srclen)



## 10.27.1 Detailed Description

Class for [Base64](#).

## 10.27.2 Constructor & Destructor Documentation

### 10.27.2.1 Base64()

```
gdcmm::Base64::Base64 (
    const Base64 & ) [delete]
```

## 10.27.3 Member Function Documentation

### 10.27.3.1 Decode()

```
static size_t gdcmm::Base64::Decode (
    char * dst,
    size_t dlen,
    const char * src,
    size_t slen ) [static]
```

Decode a base64-formatted buffer.

#### Parameters

<i>dst</i>	destination buffer
<i>dlen</i>	size of the buffer
<i>src</i>	source buffer
<i>slen</i>	amount of data to be decoded

#### Returns

0 if not successful, size of decoded otherwise

#### Examples

[DumpExamCard.cxx](#), and [DumpSiemensBase64.cxx](#).

### 10.27.3.2 Encode()

```
static size_t gdcmm::Base64::Encode (
    char * dst,
    size_t dlen,
    const char * src,
    size_t slen ) [static]
```

Encode a buffer into base64 format.

**Parameters**

<i>dst</i>	destination buffer
<i>dlen</i>	size of the buffer
<i>src</i>	source buffer
<i>slen</i>	amount of data to be encoded

**Returns**

0 if not successful, size of encoded otherwise

**10.27.3.3 GetDecodeLength()**

```
static size_t gdcM::Base64::GetDecodeLength (
    const char * src,
    size_t len ) [static]
```

Call this function to obtain the required buffer size

**Examples**

[DumpExamCard.cxx](#), and [DumpSiemensBase64.cxx](#).

**10.27.3.4 GetEncodeLength()**

```
static size_t gdcM::Base64::GetEncodeLength (
    const char * src,
    size_t srclen ) [static]
```

Call this function to obtain the required buffer size

**10.27.3.5 operator=()**

```
void gdcM::Base64::operator= (
    const Base64 & ) [delete]
```

The documentation for this class was generated from the following file:

- [gdcMBase64.h](#)

## 10.28 gdcmm::network::BaseCompositeMessage Class Reference

[BaseCompositeMessage.](#)

```
#include <gdcmmBaseCompositeMessage.h>
```

Inheritance diagram for gdcmm::network::BaseCompositeMessage:



### Public Member Functions

- virtual [~BaseCompositeMessage](#) ()=default
- virtual std::vector< [PresentationDataValue](#) > [ConstructPDV](#) (const [ULConnection](#) &inConnection, const [BaseRootQuery](#) \*inRootQuery)=0

### 10.28.1 Detailed Description

[BaseCompositeMessage.](#)

The Composite events described in section 3.7-2009 of the DICOM standard all use their own messages. These messages are constructed using Presentation Data Values, from section 3.8-2009 of the standard, and then fill in appropriate values in their datasets.

So, for the five composites:

- C-ECHO
- C-FIND
- C-MOVE
- C-GET
- C-STORE there are a series of messages. However, all of these messages are obtained as part of a PDataPDU, and all have to be placed there. Therefore, since they all have shared functionality and construction tropes, that will be put into a base class. Further, the base class will be then returned by the factory class, `gdcmCompositePDUFactory`.

This is an abstract class. It cannot be instantiated on its own.

## 10.28.2 Constructor & Destructor Documentation

### 10.28.2.1 `~BaseCompositeMessage()`

```
virtual gdcm::network::BaseCompositeMessage::~BaseCompositeMessage ( ) [virtual], [default]
```

## 10.28.3 Member Function Documentation

### 10.28.3.1 `ConstructPDV()`

```
virtual std::vector< PresentationDataValue > gdcm::network::BaseCompositeMessage::ConstructPDV (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery ) [pure virtual]
```

Implemented in `gdcm::network::CEchoRQ`, `gdcm::network::CFindRQ`, and `gdcm::network::CMoveRQ`.

The documentation for this class was generated from the following file:

- `gdcmBaseCompositeMessage.h`

## 10.29 `gdcm::network::BaseNormalizedMessage` Class Reference

`BaseNormalizedMessage`.

```
#include <gdcmBaseNormalizedMessage.h>
```

Inheritance diagram for gdcmm::network::BaseNormalizedMessage:



## Public Member Functions

- virtual `~BaseNormalizedMessage()`=default
- virtual `std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery)=0`

### 10.29.1 Detailed Description

[BaseNormalizedMessage](#).

The Normalized events described in section 3.7-2011 of the DICOM standard all use their own messages. These messages are constructed using Presentation Data Values, from section 3.8-2011 of the standard, and then fill in appropriate values in their datasets.

So, for the five normalized:

- N-ACTION
- N-CREATE
- N-DELETE
- N-EVENT
- N-GET
- N-SET there are a series of messages. However, all of these messages are obtained as part of a PData←PDU, and all have to be placed there. Therefore, since they all have shared functionality and construction tropes, that will be put into a base class. Further, the base class will be then returned by the factory class, [gdcmNormalizedMessageFactory.h](#).

This is an abstract class. It cannot be instantiated on its own.

## 10.29.2 Constructor & Destructor Documentation

### 10.29.2.1 ~BaseNormalizedMessage()

```
virtual gdcm::network::BaseNormalizedMessage::~~BaseNormalizedMessage ( ) [virtual], [default]
```

## 10.29.3 Member Function Documentation

### 10.29.3.1 ConstructPDV()

```
virtual std::vector< PresentationDataValue > gdcm::network::BaseNormalizedMessage::ConstructPDV (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [pure virtual]
```

Implemented in [gdcm::network::NActionRQ](#), [gdcm::network::NCreateRQ](#), [gdcm::network::NDeleteRQ](#), [gdcm::network::NEventReportRQ](#), [gdcm::network::NGetRQ](#), and [gdcm::network::NSetRQ](#).

The documentation for this class was generated from the following file:

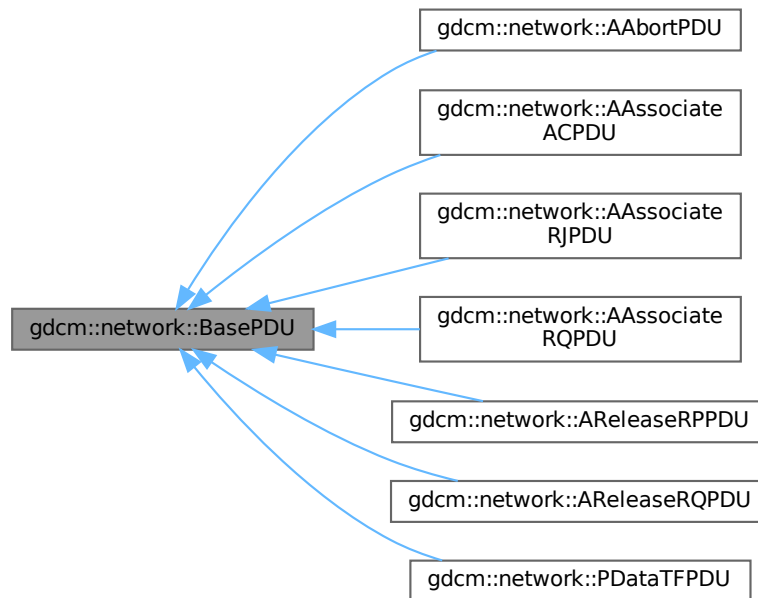
- [gdcmBaseNormalizedMessage.h](#)

## 10.30 gdcmm::network::BasePDU Class Reference

[BasePDU](#).

```
#include <gdcmmBasePDU.h>
```

Inheritance diagram for gdcmm::network::BasePDU:



### Public Member Functions

- virtual [~BasePDU](#) ()=default
- virtual bool [IsLastFragment](#) () const =0
- virtual void [Print](#) (std::ostream &os) const =0
- virtual std::istream & [Read](#) (std::istream &is)=0
- virtual size\_t [Size](#) () const =0
- virtual const std::ostream & [Write](#) (std::ostream &os) const =0

### 10.30.1 Detailed Description

[BasePDU](#).

base class for PDUs

all PDUs start with the first ten bytes as specified: 01 PDU type 02 reserved 3-6 PDU Length (unsigned) 7-10 variable

on some, 7-10 are split (7-8 as protocol version in Associate-RQ, for instance, while associate-rj splits those four bytes differently).

Also common to all the PDUs is their ability to read and write to a stream.

So, let's just get them all bunched together into one (abstract) class, shall we?

Why? 1) so that the [ULEvent](#) can have the PDU stored in it, since the event takes PDUs and not other class structures (other class structures get converted into PDUs) 2) to make reading PDUs in the event loop cleaner

## 10.30.2 Constructor & Destructor Documentation

### 10.30.2.1 ~BasePDU()

```
virtual gdc::network::BasePDU::~~BasePDU ( ) [virtual], [default]
```

## 10.30.3 Member Function Documentation

### 10.30.3.1 IsLastFragment()

```
virtual bool gdc::network::BasePDU::IsLastFragment ( ) const [pure virtual]
```

Implemented in [gdc::network::AAbortPDU](#), [gdc::network::AAssociateACPDU](#), [gdc::network::AAssociateRJPDU](#), [gdc::network::AAssociateRQPDU](#), [gdc::network::AReleaseRPPDU](#), [gdc::network::AReleaseRQPDU](#), and [gdc::network::PDataTFPDU](#).

### 10.30.3.2 Print()

```
virtual void gdc::network::BasePDU::Print (
    std::ostream & os ) const [pure virtual]
```

Implemented in [gdc::network::AAbortPDU](#), [gdc::network::AAssociateACPDU](#), [gdc::network::AAssociateRJPDU](#), [gdc::network::AAssociateRQPDU](#), [gdc::network::AReleaseRPPDU](#), [gdc::network::AReleaseRQPDU](#), and [gdc::network::PDataTFPDU](#).

### 10.30.3.3 Read()

```
virtual std::istream & gdc::network::BasePDU::Read (
    std::istream & is ) [pure virtual]
```

Implemented in [gdc::network::AAbortPDU](#), [gdc::network::AAssociateACPDU](#), [gdc::network::AAssociateRJPDU](#), [gdc::network::AAssociateRQPDU](#), [gdc::network::AReleaseRPPDU](#), [gdc::network::AReleaseRQPDU](#), and [gdc::network::PDataTFPDU](#).



### 10.30.3.4 Size()

```
virtual size_t gdcm::network::BasePDU::Size ( ) const [pure virtual]
```

Implemented in [gdcm::network::AAbortPDU](#), [gdcm::network::AAssociateACPDU](#), [gdcm::network::AAssociateRJPDU](#), [gdcm::network::AAssociateRQPDU](#), [gdcm::network::AReleaseRPPDU](#), [gdcm::network::AReleaseRQPDU](#), and [gdcm::network::PDataTFPDU](#).

### 10.30.3.5 Write()

```
virtual const std::ostream & gdcm::network::BasePDU::Write (
    std::ostream & os ) const [pure virtual]
```

Implemented in [gdcm::network::AAbortPDU](#), [gdcm::network::AAssociateACPDU](#), [gdcm::network::AAssociateRJPDU](#), [gdcm::network::AAssociateRQPDU](#), [gdcm::network::AReleaseRPPDU](#), [gdcm::network::AReleaseRQPDU](#), and [gdcm::network::PDataTFPDU](#).

The documentation for this class was generated from the following file:

- [gdcmBasePDU.h](#)

## 10.31 gdcm::BaseQuery Class Reference

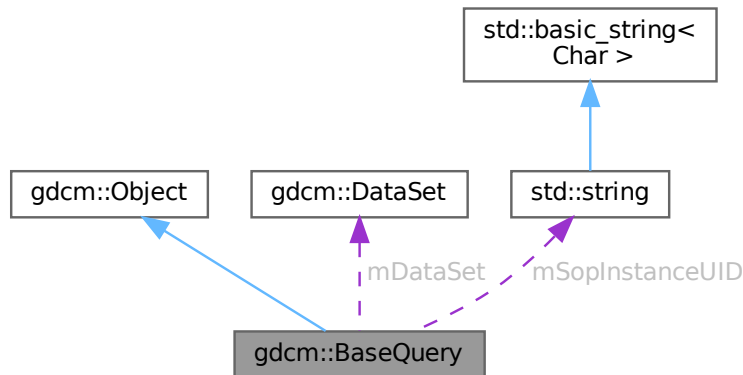
[BaseQuery](#).

```
#include <gdcmBaseQuery.h>
```

Inheritance diagram for `gdcm::BaseQuery`:



Collaboration diagram for `gdcm::BaseQuery`:



### Public Member Functions

- `~BaseQuery` () override
- void `AddQueryDataSet` (const `DataSet` &ds)
- virtual `UIDs::TSName GetAbstractSyntaxUID` () const =0
- `DataSet` & `GetQueryDataSet` ()
- `DataSet` const & `GetQueryDataSet` () const
- *Set/Get the internal representation of the query as a `DataSet`.*
- `std::string GetSOPInstanceUID` () const
- void `Print` (std::ostream &os) const override
- void `SetSearchParameter` (const std::string &inKeyword, const std::string &inValue)
- void `SetSearchParameter` (const `Tag` &inTag, const std::string &inValue)
- void `SetSOPInstanceUID` (const std::string &iSopInstanceUID)
- virtual bool `ValidateQuery` (bool inStrict=true) const =0
- const std::ostream & `WriteHelpFile` (std::ostream &os)
- bool `WriteQuery` (const std::string &inFileName)

### Public Member Functions inherited from `gdcm::Object`

- `Object` ()
- `Object` (const `Object` &)
- *Special requirement for copy/cstor, assignment operator.*
- virtual `~Object` ()
- void `operator=` (const `Object` &)

### Protected Member Functions

- `BaseQuery` ()
- void `SetSearchParameter` (const `Tag` &inTag, const `DictEntry` &inDictEntry, const std::string &inValue)
- bool `ValidDataSet` (const `DataSet` &dataSetToValid, const `DataSet` &dataSetReference) const

## Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

## Protected Attributes

- [DataSet](#) `mDataSet`
- `std::string` `mSopInstanceUID`

## Friends

- class [QueryFactory](#)

### 10.31.1 Detailed Description

[BaseQuery](#).

contains: a baseclass which will produce a dataset for all dimse messages

### 10.31.2 Constructor & Destructor Documentation

#### 10.31.2.1 BaseQuery()

```
gdcm::BaseQuery::BaseQuery ( ) [protected]
```

#### 10.31.2.2 ~BaseQuery()

```
gdcm::BaseQuery::~~BaseQuery ( ) [override]
```

### 10.31.3 Member Function Documentation

#### 10.31.3.1 AddQueryDataSet()

```
void gdcm::BaseQuery::AddQueryDataSet (
    const DataSet & ds )
```

#### 10.31.3.2 GetAbstractSyntaxUID()

```
virtual UIDs::TSName gdcm::BaseQuery::GetAbstractSyntaxUID ( ) const [pure virtual]
```

Implemented in [gdcm::FindPatientRootQuery](#), [gdcm::FindStudyRootQuery](#), [gdcm::ModalityPerformedProcedureStepCreateQuery](#), [gdcm::ModalityPerformedProcedureStepSetQuery](#), [gdcm::MovePatientRootQuery](#), [gdcm::MoveStudyRootQuery](#), and [gdcm::WLMFindQuery](#).

#### 10.31.3.3 GetQueryDataSet() [1/2]

```
DataSet & gdcm::BaseQuery::GetQueryDataSet ( )
```

#### 10.31.3.4 GetQueryDataSet() [2/2]

```
DataSet const & gdcm::BaseQuery::GetQueryDataSet ( ) const
```

Set/Get the internal representation of the query as a [DataSet](#).

#### 10.31.3.5 GetSOPInstanceUID()

```
std::string gdcm::BaseQuery::GetSOPInstanceUID ( ) const [inline]
```

#### 10.31.3.6 Print()

```
void gdcm::BaseQuery::Print (
    std::ostream & os ) const [override], [virtual]
```

Reimplemented from [gdcm::Object](#).

#### 10.31.3.7 SetSearchParameter() [1/3]

```
void gdcm::BaseQuery::SetSearchParameter (
    const std::string & inKeyword,
    const std::string & inValue )
```

#### 10.31.3.8 SetSearchParameter() [2/3]

```
void gdcm::BaseQuery::SetSearchParameter (
    const Tag & inTag,
    const DictEntry & inDictEntry,
    const std::string & inValue ) [protected]
```

#### 10.31.3.9 SetSearchParameter() [3/3]

```
void gdcm::BaseQuery::SetSearchParameter (
    const Tag & inTag,
    const std::string & inValue )
```

#### 10.31.3.10 SetSOPInstanceUID()

```
void gdcm::BaseQuery::SetSOPInstanceUID (
    const std::string & iSopInstanceUID ) [inline]
```

#### 10.31.3.11 ValidateQuery()

```
virtual bool gdcm::BaseQuery::ValidateQuery (
    bool inStrict = true ) const [pure virtual]
```

Implemented in [gdcm::FindPatientRootQuery](#), [gdcm::FindStudyRootQuery](#), [gdcm::ModalityPerformedProcedureStepCreateQuery](#), [gdcm::ModalityPerformedProcedureStepSetQuery](#), [gdcm::MovePatientRootQuery](#), [gdcm::MoveStudyRootQuery](#), [gdcm::WLMFindQuery](#), and [gdcm::BaseRootQuery](#).

#### 10.31.3.12 ValidDataSet()

```
bool gdcm::BaseQuery::ValidDataSet (
    const DataSet & dataSetToValid,
    const DataSet & dataSetReference ) const [protected]
```

#### 10.31.3.13 WriteHelpFile()

```
const std::ostream & gdcm::BaseQuery::WriteHelpFile (
    std::ostream & os )
```

#### 10.31.3.14 WriteQuery()

```
bool gdcm::BaseQuery::WriteQuery (
    const std::string & inFileName )
```

### 10.31.4 Friends And Related Symbol Documentation

#### 10.31.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

### 10.31.5 Member Data Documentation

#### 10.31.5.1 mDataSet

```
DataSet gdcm::BaseQuery::mDataSet [protected]
```

### 10.31.5.2 mSopInstanceUID

```
std::string gdcM::BaseQuery::mSopInstanceUID [protected]
```

The documentation for this class was generated from the following file:

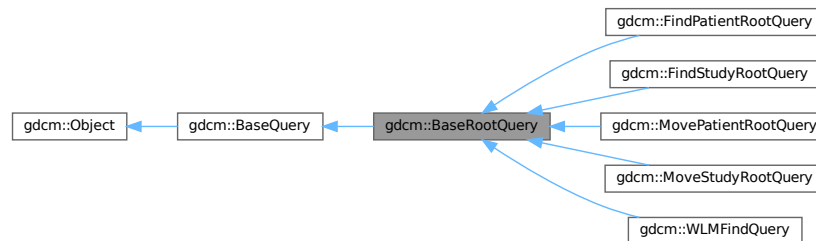
- [gdcMBaseQuery.h](#)

## 10.32 gdcM::BaseRootQuery Class Reference

[BaseRootQuery](#).

```
#include <gdcMBaseRootQuery.h>
```

Inheritance diagram for gdcM::BaseRootQuery:



Collaboration diagram for gdcM::BaseRootQuery:



**Public Member Functions**

- [~BaseRootQuery](#) () override=default
- [EQueryLevel](#) [GetQueryLevelFromQueryRoot](#) ([ERootType](#) roottype)
- virtual [std::vector< Tag >](#) [GetTagListByLevel](#) (const [EQueryLevel](#) &inQueryLevel)=0
- virtual void [InitializeDataSet](#) (const [EQueryLevel](#) &inQueryLevel)=0
- bool [ValidateQuery](#) (bool inStrict=true) const override=0

**Public Member Functions inherited from [gdcm::BaseQuery](#)**

- [~BaseQuery](#) () override
- void [AddQueryDataSet](#) (const [DataSet](#) &ds)
- virtual [UIDs::TSName](#) [GetAbstractSyntaxUID](#) () const =0
- [DataSet](#) & [GetQueryDataSet](#) ()
- [DataSet](#) const & [GetQueryDataSet](#) () const  
*Set/Get the internal representation of the query as a [DataSet](#).*
- [std::string](#) [GetSOPInstanceUID](#) () const
- void [Print](#) ([std::ostream](#) &os) const override
- void [SetSearchParameter](#) (const [std::string](#) &inKeyword, const [std::string](#) &inValue)
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const [std::string](#) &inValue)
- void [SetSOPInstanceUID](#) (const [std::string](#) &iSopInstanceUID)
- const [std::ostream](#) & [WriteHelpFile](#) ([std::ostream](#) &os)
- bool [WriteQuery](#) (const [std::string](#) &inFileName)

**Public Member Functions inherited from [gdcm::Object](#)**

- [Object](#) ()
- [Object](#) (const [Object](#) &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

**Static Public Member Functions**

- static [QueryBase](#) \* [Construct](#) ([ERootType](#) inRootType, [EQueryLevel](#) qllevel)
- static int [GetQueryLevelFromString](#) (const char \*str)
- static const char \* [GetQueryLevelString](#) ([EQueryLevel](#) ql)

**Protected Member Functions**

- [BaseRootQuery](#) ()

**Protected Member Functions inherited from [gdcm::BaseQuery](#)**

- [BaseQuery](#) ()
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const [DictEntry](#) &inDictEntry, const [std::string](#) &inValue)
- bool [ValidDataSet](#) (const [DataSet](#) &dataSetToValid, const [DataSet](#) &dataSetReference) const

## Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

## Protected Attributes

- std::string [mHelpDescription](#)
- [QueryImage](#) [mImage](#)
- [QueryPatient](#) [mPatient](#)
- [ERootType](#) [mRootType](#)
- [QuerySeries](#) [mSeries](#)
- [QueryStudy](#) [mStudy](#)

## Protected Attributes inherited from [gdcm::BaseQuery](#)

- [DataSet](#) [mDataSet](#)
- std::string [mSopInstanceUID](#)

## Friends

- class [QueryFactory](#)

## 10.32.1 Detailed Description

[BaseRootQuery](#).

contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root

This class contains the functionality used in patient c-find and c-move queries. [PatientRootQuery](#) and [StudyRootQuery](#) derive from this class.

Namely: 1) list all tags associated with a particular query type 2) produce a query dataset via tag association

Eventually, it can be used to validate a particular dataset type.

The dataset held by this object (or, really, one of its derivatives) should be passed to a c-find or c-move query.

## 10.32.2 Constructor & Destructor Documentation

### 10.32.2.1 [BaseRootQuery](#)()

```
gdcm::BaseRootQuery::BaseRootQuery ( ) [protected]
```



### 10.32.2.2 ~BaseRootQuery()

```
gdcm::BaseRootQuery::~~BaseRootQuery ( ) [override], [default]
```

## 10.32.3 Member Function Documentation

### 10.32.3.1 Construct()

```
static QueryBase * gdcm::BaseRootQuery::Construct (
    ERootType inRootType,
    EQueryLevel qllevel ) [static]
```

### 10.32.3.2 GetQueryLevelFromQueryRoot()

```
EQueryLevel gdcm::BaseRootQuery::GetQueryLevelFromQueryRoot (
    ERootType roottype )
```

### 10.32.3.3 GetQueryLevelFromString()

```
static int gdcm::BaseRootQuery::GetQueryLevelFromString (
    const char * str ) [static]
```

### 10.32.3.4 GetQueryLevelString()

```
static const char * gdcm::BaseRootQuery::GetQueryLevelString (
    EQueryLevel ql ) [static]
```

### 10.32.3.5 GetTagListByLevel()

```
virtual std::vector< Tag > gdcm::BaseRootQuery::GetTagListByLevel (
    const EQueryLevel & inQueryLevel ) [pure virtual]
```

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implemented in [gdcm::FindPatientRootQuery](#), [gdcm::FindStudyRootQuery](#), [gdcm::MovePatientRootQuery](#), [gdcm::MoveStudyRootQuery](#), and [gdcm::WLMFindQuery](#).

### 10.32.3.6 InitializeDataSet()

```
virtual void gdcm::BaseRootQuery::InitializeDataSet (
    const EQueryLevel & inQueryLevel ) [pure virtual]
```

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmTk

Implemented in [gdcm::FindPatientRootQuery](#), [gdcm::FindStudyRootQuery](#), [gdcm::MovePatientRootQuery](#), [gdcm::MoveStudyRootQuery](#), and [gdcm::WLMFindQuery](#).

### 10.32.3.7 ValidateQuery()

```
bool gdcm::BaseRootQuery::ValidateQuery (
    bool inStrict = true ) const [override], [pure virtual]
```

have to be able to ensure that 0x8,0x52 is set (which will be true if InitializeDataSet is called...) that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional) by default, this function checks to see if the query is for finding, which is more permissive than for moving. For moving, only the unique tags are allowed. 10 Jan 2011: adding in the 'strict' mode. according to the standard (at least, how I've read it), only tags for a particular level should be allowed in a particular query (ie, just series level tags in a series level query). However, it seems that dcm4chee doesn't share that interpretation. So, if 'inStrict' is false, then tags from the current level and all higher levels are now considered valid. So, if you're doing a non-strict series-level query, tags from the patient and study level can be passed along as well.

Implements [gdcm::BaseQuery](#).

Implemented in [gdcm::FindPatientRootQuery](#), [gdcm::FindStudyRootQuery](#), [gdcm::MovePatientRootQuery](#), [gdcm::MoveStudyRootQuery](#), and [gdcm::WLMFindQuery](#).

## 10.32.4 Friends And Related Symbol Documentation

### 10.32.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

## 10.32.5 Member Data Documentation

### 10.32.5.1 mHelpDescription

```
std::string gdcm::BaseRootQuery::mHelpDescription [protected]
```

### 10.32.5.2 mImage

```
QueryImage gdcm::BaseRootQuery::mImage [protected]
```

### 10.32.5.3 mPatient

`QueryPatient` `gdcm::BaseRootQuery::mPatient` [protected]

### 10.32.5.4 mRootType

`ERootType` `gdcm::BaseRootQuery::mRootType` [protected]

### 10.32.5.5 mSeries

`QuerySeries` `gdcm::BaseRootQuery::mSeries` [protected]

### 10.32.5.6 mStudy

`QueryStudy` `gdcm::BaseRootQuery::mStudy` [protected]

The documentation for this class was generated from the following file:

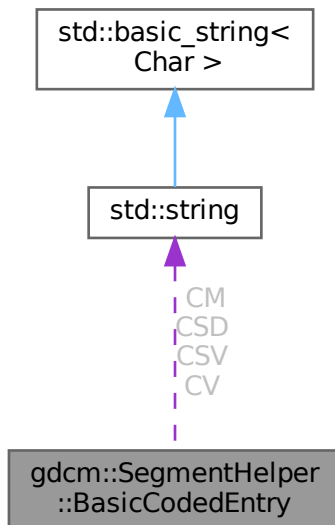
- [gdcmBaseRootQuery.h](#)

## 10.33 gdcm::SegmentHelper::BasicCodedEntry Struct Reference

This structure defines a basic coded entry with all of its attributes.

```
#include <gdcmSegmentHelper.h>
```

Collaboration diagram for `gdcm::SegmentHelper::BasicCodedEntry`:



## Public Member Functions

- [BasicCodedEntry](#) ()  
*Constructor.*
- [BasicCodedEntry](#) (const char \*a\_CV, const char \*a\_CSD, const char \*a\_CM)  
*constructor which defines type 1 attributes.*
- [BasicCodedEntry](#) (const char \*a\_CV, const char \*a\_CSD, const char \*a\_CSV, const char \*a\_CM)  
*constructor which defines attributes.*
- bool [IsEmpty](#) (const bool checkOptionalAttributes=false) const  
*Check if each attributes of the basic coded entry is defined.*

## Public Attributes

- std::string [CM](#)  
*Coding Scheme [Version](#) attribute.*
- std::string [CSD](#)  
*Code [Value](#) attribute.*
- std::string [CSV](#)  
*Coding Scheme Designator attribute.*
- std::string [CV](#)

### 10.33.1 Detailed Description

This structure defines a basic coded entry with all of its attributes.

See also

PS 3.3 section 8.8.

### 10.33.2 Constructor & Destructor Documentation

#### 10.33.2.1 BasicCodedEntry() [1/3]

```
gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry ( ) [inline]
```

Constructor.

#### 10.33.2.2 BasicCodedEntry() [2/3]

```
gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry (
    const char * a_CV,
    const char * a_CSD,
    const char * a_CM ) [inline]
```

constructor which defines type 1 attributes.

### 10.33.2.3 BasicCodedEntry() [3/3]

```
gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry (
    const char * a_CV,
    const char * a_CSD,
    const char * a_CSV,
    const char * a_CM ) [inline]
```

constructor which defines attributes.

## 10.33.3 Member Function Documentation

### 10.33.3.1 IsEmpty()

```
bool gdcm::SegmentHelper::BasicCodedEntry::IsEmpty (
    const bool checkOptionalAttributes = false ) const
```

Check if each attributes of the basic coded entry is defined.

Parameters

<i>checkOptionalAttributes</i>	Check also type 1C attributes.
--------------------------------	--------------------------------

## 10.33.4 Member Data Documentation

### 10.33.4.1 CM

```
std::string gdcm::SegmentHelper::BasicCodedEntry::CM
```

Coding Scheme [Version](#) attribute.

### 10.33.4.2 CSD

```
std::string gdcm::SegmentHelper::BasicCodedEntry::CSD
```

Code [Value](#) attribute.

### 10.33.4.3 CSV

```
std::string gdcm::SegmentHelper::BasicCodedEntry::CSV
```

Coding Scheme Designator attribute.

#### 10.33.4.4 CV

```
std::string gdcm::SegmentHelper::BasicCodedEntry::CV
```

The documentation for this struct was generated from the following file:

- [gdcmSegmentHelper.h](#)

### 10.34 gdcm::BasicOffsetTable Class Reference

Class to represent a [BasicOffsetTable](#).

```
#include <gdcmBasicOffsetTable.h>
```

Inheritance diagram for gdcm::BasicOffsetTable:



Collaboration diagram for gdcm::BasicOffsetTable:



### Public Member Functions

- [BasicOffsetTable](#) ()
- template<typename TSwap >  
std::istream & [Read](#) (std::istream &is)

### Public Member Functions inherited from [gdcm::Fragment](#)

- [Fragment](#) ()
- [VL ComputeLength](#) () const
- [VL GetLength](#) () const
- template<typename TSwap >  
std::istream & [Read](#) (std::istream &is)
- template<typename TSwap >  
std::istream & [ReadBacktrack](#) (std::istream &is)
- template<typename TSwap >  
std::istream & [ReadPreValue](#) (std::istream &is)
- template<typename TSwap >  
std::istream & [ReadValue](#) (std::istream &is)
- template<typename TSwap >  
std::ostream & [Write](#) (std::ostream &os) const

## Public Member Functions inherited from `gdcm::DataElement`

- `DataElement` (const `DataElement` &\_val)
- `DataElement` (const `Tag` &t=`Tag`(0), const `VL` &vl=0, const `VR` &vr=`VR::INVALID`)
- void `Clear` ()
  - Clear Data `Element` (make `Value` empty and invalidate `Tag` & `VR`)*
- void `Empty` ()
  - Make Data `Element` empty (no `Value`)*
- const `ByteValue` \* `GetByteValue` () const
- template<typename TDE >  
`VL` `GetLength` () const
- `SequenceOfFragments` \* `GetSequenceOfFragments` ()
- const `SequenceOfFragments` \* `GetSequenceOfFragments` () const
- `Tag` & `GetTag` ()
- const `Tag` & `GetTag` () const
  - Get `Tag`.*
- `Value` & `GetValue` ()
- `Value` const & `GetValue` () const
  - Set/Get `Value` (bytes array, SQ of items, SQ of fragments):*
- `SmartPointer`< `SequenceOfItems` > `GetValueAsSQ` () const
- `VL` & `GetVL` ()
- const `VL` & `GetVL` () const
  - Get `VL`.*
- `VR` const & `GetVR` () const
- bool `IsEmpty` () const
  - Check if Data `Element` is empty.*
- bool `IsUndefinedLength` () const
  - return if `Value` Length if of undefined length*
- bool `operator`< (const `DataElement` &de) const
- `DataElement` & `operator`= (const `DataElement` &)=default
- bool `operator`== (const `DataElement` &de) const
- template<typename TDE , typename TSwap >  
`std::istream` & `Read` (`std::istream` &is)
- template<typename TDE , typename TSwap >  
`std::istream` & `ReadOrSkip` (`std::istream` &is, `std::set`< `Tag` > const &skiptags)
- template<typename TDE , typename TSwap >  
`std::istream` & `ReadPreValue` (`std::istream` &is, `std::set`< `Tag` > const &skiptags)
- template<typename TDE , typename TSwap >  
`std::istream` & `ReadValue` (`std::istream` &is, `std::set`< `Tag` > const &skiptags)
- template<typename TDE , typename TSwap >  
`std::istream` & `ReadValueWithLength` (`std::istream` &is, `VL` &length, `std::set`< `Tag` > const &skiptags)
- template<typename TDE , typename TSwap >  
`std::istream` & `ReadWithLength` (`std::istream` &is, `VL` &length)
- void `SetByteValue` (const char \*array, `VL` length)
- void `SetTag` (const `Tag` &t)
- void `SetValue` (`Value` const &vl)
- void `SetVL` (const `VL` &vl)
- void `SetVLToUndefined` ()
- void `SetVR` (`VR` const &vr)
- template<typename TDE , typename TSwap >  
`const std::ostream` & `Write` (`std::ostream` &os) const



## Friends

- `std::ostream & operator<< (std::ostream &os, const BasicOffsetTable &val)`

## Additional Inherited Members

## Protected Types inherited from [gdcm::DataElement](#)

- `typedef SmartPointer< Value > ValuePtr`

## Protected Member Functions inherited from [gdcm::DataElement](#)

- `void SetValueFieldLength (VL vl, bool readvalues)`

## Protected Attributes inherited from [gdcm::DataElement](#)

- [Tag](#) [TagField](#)
- [ValuePtr](#) [ValueField](#)
- [VL](#) [ValueLengthField](#)
- [VR](#) [VRField](#)

### 10.34.1 Detailed Description

Class to represent a [BasicOffsetTable](#).

### 10.34.2 Constructor & Destructor Documentation

#### 10.34.2.1 [BasicOffsetTable\(\)](#)

```
gdcm::BasicOffsetTable::BasicOffsetTable ( ) [inline]
```

### 10.34.3 Member Function Documentation

#### 10.34.3.1 [Read\(\)](#)

```
template<typename TSwap >
std::istream & gdcm::BasicOffsetTable::Read (
    std::istream & is ) [inline]
```

References [gdcmAssertAlwaysMacro](#), and [gdcm::ParseException::SetLastElement\(\)](#).

### 10.34.4 Friends And Related Symbol Documentation

#### 10.34.4.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & os,  
    const BasicOffsetTable & val ) [friend]
```

The documentation for this class was generated from the following file:

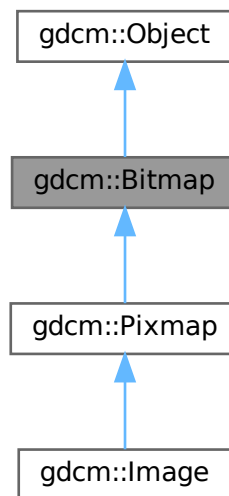
- [gdcmBasicOffsetTable.h](#)

## 10.35 gdcm::Bitmap Class Reference

[Bitmap](#) class.

```
#include <gdcmBitmap.h>
```

Inheritance diagram for gdcm::Bitmap:



Collaboration diagram for gdcm::Bitmap:



## Public Member Functions

- [Bitmap](#) ()
- [~Bitmap](#) () override
- virtual bool [AreOverlaysInPixelData](#) () const
- void [Clear](#) ()
- bool [GetBuffer](#) (char \*buffer) const  
*Access the raw data.*
- unsigned long [GetBufferLength](#) () const
- unsigned int [GetColumns](#) () const
- [DataElement](#) & [GetDataElement](#) ()
- const [DataElement](#) & [GetDataElement](#) () const
- unsigned int [GetDimension](#) (unsigned int idx) const
- const unsigned int \* [GetDimensions](#) () const  
*Return the dimension of the pixel data, first dimension (x), then 2nd (y), then 3rd (z)...*
- [LookupTable](#) & [GetLUT](#) ()
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const  
*INTERNAL do not use.*
- unsigned int [GetNumberOfDimensions](#) () const  
*Return the number of dimension of the pixel data bytes; for example 2 for a 2D matrices of values.*
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const  
*return the photometric interpretation*
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const  
*Get/Set PixelFormat.*
- unsigned int [GetPlanarConfiguration](#) () const  
*return the planar configuration*
- unsigned int [GetRows](#) () const
- const [TransferSyntax](#) & [GetTransferSyntax](#) () const
- bool [IsEmpty](#) () const
- bool [IsLossy](#) () const  
*Return whether or not the image was compressed using a lossy compressor or not.*

- bool [IsTransferSyntaxCompatible](#) ([TransferSyntax](#) const &ts) const
- void [Print](#) (std::ostream &) const override
- void [SetColumns](#) (unsigned int col)
- void [SetDataElement](#) ([DataElement](#) const &de)
- void [SetDimension](#) (unsigned int idx, unsigned int dim)
- void [SetDimensions](#) (const unsigned int dims[3])
- void [SetLossyFlag](#) (bool f)  
*Specifically set that the image was compressed using a lossy compression mechanism.*
- void [SetLUT](#) ([LookupTable](#) const &lut)  
*Set/Get LUT.*
- void [SetNeedByteSwap](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)
- void [SetRows](#) (unsigned int rows)
- void [SetTransferSyntax](#) ([TransferSyntax](#) const &ts)  
*Transfer syntax.*
- virtual bool [UnusedBitsPresentInPixelData](#) () const

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

## Protected Types

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

## Protected Member Functions

- bool [ComputeLossyFlag](#) ()
- bool [GetBuffer2](#) (std::ostream &os) const
- bool [TryJPEG2000Codec](#) (char \*buffer, bool &lossyflag) const
- bool [TryJPEG2000Codec2](#) (std::ostream &os) const
- bool [TryJPEGCodec](#) (char \*buffer, bool &lossyflag) const
- bool [TryJPEGCodec2](#) (std::ostream &os) const
- bool [TryJPEGLSCodec](#) (char \*buffer, bool &lossyflag) const
- bool [TryKAKADUCodec](#) (char \*buffer, bool &lossyflag) const
- bool [TryPVRGCodec](#) (char \*buffer, bool &lossyflag) const
- bool [TryRAWCodec](#) (char \*buffer, bool &lossyflag) const
- bool [TryRLECodec](#) (char \*buffer, bool &lossyflag) const

## Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

## Protected Attributes

- `std::vector< unsigned int >` [Dimensions](#)
- bool [LossyFlag](#)
- [LUTPtr](#) [LUT](#)
- bool [NeedByteSwap](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) [PF](#)
- [PhotometricInterpretation](#) [PI](#)
- [DataElement](#) [PixelData](#)
- unsigned int [PlanarConfiguration](#)
- [TransferSyntax](#) [TS](#)

## Friends

- class [ImageChangeTransferSyntax](#)
- class [PixmapReader](#)

## 10.35.1 Detailed Description

[Bitmap](#) class.

A bitmap based image. Used as parent for both [IconImage](#) and the main Pixel Data [Image](#) It does not contains any World Space information (IPP, IOP)

### Examples

[ExtractIconFromFile.cxx](#).

## 10.35.2 Member Typedef Documentation

### 10.35.2.1 LUTPtr

```
typedef SmartPointer<LookupTable> gdcm::Bitmap::LUTPtr [protected]
```

## 10.35.3 Constructor & Destructor Documentation

### 10.35.3.1 [Bitmap](#)()

```
gdcm::Bitmap::Bitmap ( )
```

### 10.35.3.2 ~Bitmap()

```
gdcm::Bitmap::~~Bitmap ( ) [override]
```

## 10.35.4 Member Function Documentation

### 10.35.4.1 AreOverlaysInPixelData()

```
virtual bool gdcm::Bitmap::AreOverlaysInPixelData ( ) const [inline], [virtual]
```

Reimplemented in [gdcm::Pixmap](#).

### 10.35.4.2 Clear()

```
void gdcm::Bitmap::Clear ( )
```

### 10.35.4.3 ComputeLossyFlag()

```
bool gdcm::Bitmap::ComputeLossyFlag ( ) [protected]
```

### 10.35.4.4 GetBuffer()

```
bool gdcm::Bitmap::GetBuffer (
    char * buffer ) const
```

Access the raw data.

#### Examples

[BasicImageAnonymizer.cs](#), [ConvertToQImage.cxx](#), [DecompressImage.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [GetArray.cs](#), [ReadMultiTimesException.cxx](#), and [threadgdcm.cxx](#).

### 10.35.4.5 GetBuffer2()

```
bool gdcm::Bitmap::GetBuffer2 (
    std::ostream & os ) const [protected]
```

#### 10.35.4.6 GetBufferLength()

```
unsigned long gdcm::Bitmap::GetBufferLength ( ) const
```

Return the length of the image after decompression WARNING for palette color: It will NOT take into account the Palette Color thus you need to multiply this length by 3 if computing the size of equivalent RGB image

##### Examples

[BasicImageAnonymizer.cs](#), [ConvertToQImage.cxx](#), [DecompressImage.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [GetArray.cs](#), [PatchFile.cxx](#), [ReadMultiTimesException.cxx](#), [RescaleImage.cs](#), and [threadgdcm.cxx](#).

#### 10.35.4.7 GetColumns()

```
unsigned int gdcm::Bitmap::GetColumns ( ) const [inline]
```

#### 10.35.4.8 GetDataElement() [1/2]

```
DataElement & gdcm::Bitmap::GetDataElement ( ) [inline]
```

#### 10.35.4.9 GetDataElement() [2/2]

```
const DataElement & gdcm::Bitmap::GetDataElement ( ) const [inline]
```

##### Examples

[ExtractIconFromFile.cxx](#).

#### 10.35.4.10 GetDimension()

```
unsigned int gdcm::Bitmap::GetDimension (
    unsigned int idx ) const
```

##### Examples

[BasicImageAnonymizer.cs](#), [DecompressImage.cs](#), and [GetArray.cs](#).

#### 10.35.4.11 GetDimensions()

```
const unsigned int * gdcm::Bitmap::GetDimensions ( ) const
```

Return the dimension of the pixel data, first dimension (x), then 2nd (y), then 3rd (z)...

##### Examples

[ConvertToQImage.cxx](#), [ExtractIconFromFile.cxx](#), [FixJAIBugJPEGLS.cxx](#), [HelloVizWorld.cxx](#), and [threadgdcm.cxx](#).

#### 10.35.4.12 GetLUT() [1/2]

```
LookupTable & gdcm::Bitmap::GetLUT ( ) [inline]
```

#### 10.35.4.13 GetLUT() [2/2]

```
const LookupTable & gdcm::Bitmap::GetLUT ( ) const [inline]
```

##### Examples

[ExtractIconFromFile.cxx](#), [ExtractImageRegionWithLUT.cs](#), and [PrintLUT.cxx](#).

#### 10.35.4.14 GetNeedByteSwap()

```
bool gdcm::Bitmap::GetNeedByteSwap ( ) const [inline]
```

INTERNAL do not use.

#### 10.35.4.15 GetNumberOfDimensions()

```
unsigned int gdcm::Bitmap::GetNumberOfDimensions ( ) const
```

Return the number of dimension of the pixel data bytes; for example 2 for a 2D matrices of values.

##### Examples

[DecompressImage.cs](#), [GetArray.cs](#), [HelloVizWorld.cxx](#), and [threadgdcm.cxx](#).



#### 10.35.4.16 GetPhotometricInterpretation()

```
const PhotometricInterpretation & gdcm::Bitmap::GetPhotometricInterpretation ( ) const
```

return the photometric interpretation

##### Examples

[ConvertToQImage.cxx](#), [DecompressImage.cs](#), [ExtractIconFromFile.cxx](#), and [HelloVizWorld.cxx](#).

#### 10.35.4.17 GetPixelFormat() [1/2]

```
PixelFormat & gdcm::Bitmap::GetPixelFormat ( ) [inline]
```

#### 10.35.4.18 GetPixelFormat() [2/2]

```
const PixelFormat & gdcm::Bitmap::GetPixelFormat ( ) const [inline]
```

Get/Set [PixelFormat](#).

##### Examples

[ConvertToQImage.cxx](#), [DecompressImage.cs](#), [ExtractIconFromFile.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetArray.cs](#), [GetJPEGSamplePrecision.cxx](#), [RescaleImage.cs](#), [TemplateEmptyImage.cxx](#), and [threadgdcm.cxx](#).

#### 10.35.4.19 GetPlanarConfiguration()

```
unsigned int gdcm::Bitmap::GetPlanarConfiguration ( ) const
```

return the planar configuration

#### 10.35.4.20 GetRows()

```
unsigned int gdcm::Bitmap::GetRows ( ) const [inline]
```

#### 10.35.4.21 GetTransferSyntax()

```
const TransferSyntax & gdcm::Bitmap::GetTransferSyntax ( ) const [inline]
```

##### Examples

[ExtractIconFromFile.cxx](#).

#### 10.35.4.22 IsEmpty()

```
bool gdcm::Bitmap::IsEmpty ( ) const [inline]
```

#### 10.35.4.23 IsLossy()

```
bool gdcm::Bitmap::IsLossy ( ) const
```

Return whether or not the image was compressed using a lossy compressor or not.

#### 10.35.4.24 IsTransferSyntaxCompatible()

```
bool gdcm::Bitmap::IsTransferSyntaxCompatible (
    TransferSyntax const & ts ) const
```

#### 10.35.4.25 Print()

```
void gdcm::Bitmap::Print (
    std::ostream & ) const [override], [virtual]
```

Reimplemented from [gdcm::Object](#).

Reimplemented in [gdcm::Pixmap](#), and [gdcm::Image](#).

#### Examples

[ExtractIconFromFile.cxx](#).

#### 10.35.4.26 SetColumns()

```
void gdcm::Bitmap::SetColumns (
    unsigned int col ) [inline]
```

#### 10.35.4.27 SetDataElement()

```
void gdcm::Bitmap::SetDataElement (
    DataElement const & de ) [inline]
```

#### Examples

[BasicImageAnonymizer.cs](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImage.cs](#), [DecompressImageMultiframe.c](#), [DecompressJPEGFile.cs](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [MpegVideoInfo.cs](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

#### 10.35.4.28 SetDimension()

```
void gdcm::Bitmap::SetDimension (
    unsigned int idx,
    unsigned int dim )
```

##### Examples

[DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [MpegVideoInfo.cs](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

#### 10.35.4.29 SetDimensions()

```
void gdcm::Bitmap::SetDimensions (
    const unsigned int dims[3] )
```

##### Examples

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), and [DecompressImage.cs](#).

#### 10.35.4.30 SetLossyFlag()

```
void gdcm::Bitmap::SetLossyFlag (
    bool f ) [inline]
```

Specifically set that the image was compressed using a lossy compression mechanism.

#### 10.35.4.31 SetLUT()

```
void gdcm::Bitmap::SetLUT (
    LookupTable const & lut ) [inline]
```

Set/Get LUT.

#### 10.35.4.32 SetNeedByteSwap()

```
void gdcm::Bitmap::SetNeedByteSwap (
    bool b ) [inline]
```

#### 10.35.4.33 SetNumberOfDimensions()

```
void gdcM::Bitmap::SetNumberOfDimensions (
    unsigned int dim )
```

##### Examples

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImage.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [MpegVideoInfo.cs](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

#### 10.35.4.34 SetPhotometricInterpretation()

```
void gdcM::Bitmap::SetPhotometricInterpretation (
    PhotometricInterpretation const & pi )
```

##### Examples

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImage.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [MpegVideoInfo.cs](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

#### 10.35.4.35 SetPixelFormat()

```
void gdcM::Bitmap::SetPixelFormat (
    PixelFormat const & pf ) [inline]
```

##### Examples

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImage.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [MpegVideoInfo.cs](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

References [gdcM::PixelFormat::Validate\(\)](#).

#### 10.35.4.36 SetPlanarConfiguration()

```
void gdcM::Bitmap::SetPlanarConfiguration (
    unsigned int pc )
```

##### Warning

you need to call `SetPixelFormat` first (before `SetPlanarConfiguration`) for consistency checking

#### 10.35.4.37 SetRows()

```
void gdcm::Bitmap::SetRows (
    unsigned int rows ) [inline]
```

#### 10.35.4.38 SetTransferSyntax()

```
void gdcm::Bitmap::SetTransferSyntax (
    TransferSyntax const & ts ) [inline]
```

Transfer syntax.

#### Examples

[BasicImageAnonymizer.cs](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [MergeTwoFiles.cxx](#), and [MpegVideoInfo.cs](#).

#### 10.35.4.39 TryJPEG2000Codec()

```
bool gdcm::Bitmap::TryJPEG2000Codec (
    char * buffer,
    bool & lossyflag ) const [protected]
```

#### 10.35.4.40 TryJPEG2000Codec2()

```
bool gdcm::Bitmap::TryJPEG2000Codec2 (
    std::ostream & os ) const [protected]
```

#### 10.35.4.41 TryJPEGCodec()

```
bool gdcm::Bitmap::TryJPEGCodec (
    char * buffer,
    bool & lossyflag ) const [protected]
```

#### 10.35.4.42 TryJPEGCodec2()

```
bool gdcm::Bitmap::TryJPEGCodec2 (
    std::ostream & os ) const [protected]
```

#### 10.35.4.43 TryJPEGLSCodec()

```
bool gdcm::Bitmap::TryJPEGLSCodec (
    char * buffer,
    bool & lossyflag ) const [protected]
```

#### 10.35.4.44 TryKAKADUCodec()

```
bool gdcmm::Bitmap::TryKAKADUCodec (
    char * buffer,
    bool & lossyflag ) const    [protected]
```

#### 10.35.4.45 TryPVRGCodec()

```
bool gdcmm::Bitmap::TryPVRGCodec (
    char * buffer,
    bool & lossyflag ) const    [protected]
```

#### 10.35.4.46 TryRAWCodec()

```
bool gdcmm::Bitmap::TryRAWCodec (
    char * buffer,
    bool & lossyflag ) const    [protected]
```

#### 10.35.4.47 TryRLECodec()

```
bool gdcmm::Bitmap::TryRLECodec (
    char * buffer,
    bool & lossyflag ) const    [protected]
```

#### 10.35.4.48 UnusedBitsPresentInPixelData()

```
virtual bool gdcmm::Bitmap::UnusedBitsPresentInPixelData ( ) const    [inline], [virtual]
```

Reimplemented in [gdcmm::Pixmap](#).

### 10.35.5 Friends And Related Symbol Documentation

#### 10.35.5.1 ImageChangeTransferSyntax

```
friend class ImageChangeTransferSyntax    [friend]
```

#### 10.35.5.2 PixmapReader

```
friend class PixmapReader    [friend]
```

## 10.35.6 Member Data Documentation

### 10.35.6.1 Dimensions

`std::vector<unsigned int> gdcm::Bitmap::Dimensions` [protected]

### 10.35.6.2 LossyFlag

`bool gdcm::Bitmap::LossyFlag` [protected]

### 10.35.6.3 LUT

`LUTPtr gdcm::Bitmap::LUT` [protected]

### 10.35.6.4 NeedByteSwap

`bool gdcm::Bitmap::NeedByteSwap` [protected]

### 10.35.6.5 NumberOfDimensions

`unsigned int gdcm::Bitmap::NumberOfDimensions` [protected]

### 10.35.6.6 PF

`PixelFormat gdcm::Bitmap::PF` [protected]

### 10.35.6.7 PI

`PhotometricInterpretation gdcm::Bitmap::PI` [protected]

### 10.35.6.8 PixelData

`DataElement gdcm::Bitmap::PixelData` [protected]

### 10.35.6.9 PlanarConfiguration

`unsigned int gdcm::Bitmap::PlanarConfiguration` [protected]

### 10.35.6.10 TS

`TransferSyntax` `gdcm::Bitmap::TS` [protected]

The documentation for this class was generated from the following file:

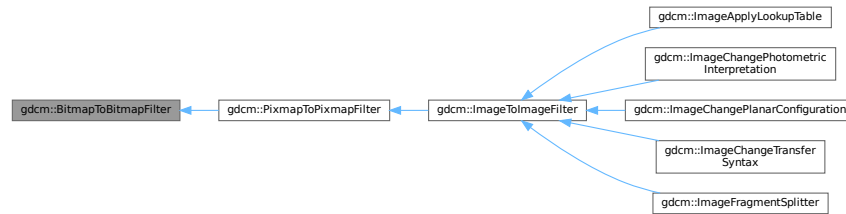
- [gdcmBitmap.h](#)

## 10.36 gdcm::BitmapToBitmapFilter Class Reference

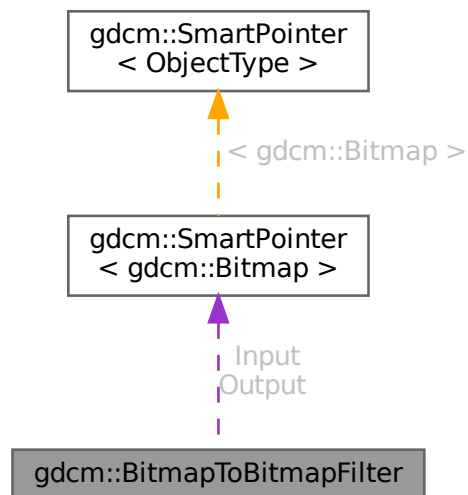
[BitmapToBitmapFilter](#) class.

```
#include <gdcmBitmapToBitmapFilter.h>
```

Inheritance diagram for `gdcm::BitmapToBitmapFilter`:



Collaboration diagram for `gdcm::BitmapToBitmapFilter`:





## Public Member Functions

- [BitmapToBitmapFilter](#) ()
- [~BitmapToBitmapFilter](#) ()=default
- const [Bitmap](#) & [GetOutput](#) () const  
*Get Output image.*
- const [Bitmap](#) & [GetOutputAsBitmap](#) () const
- void [SetInput](#) (const [Bitmap](#) &image)  
*Set input image.*

## Protected Attributes

- [SmartPointer](#)< [Bitmap](#) > [Input](#)
- [SmartPointer](#)< [Bitmap](#) > [Output](#)

### 10.36.1 Detailed Description

[BitmapToBitmapFilter](#) class.

Super class for all filter taking an image and producing an output image

### 10.36.2 Constructor & Destructor Documentation

#### 10.36.2.1 [BitmapToBitmapFilter](#)()

```
gdcm::BitmapToBitmapFilter::BitmapToBitmapFilter ( )
```

#### 10.36.2.2 [~BitmapToBitmapFilter](#)()

```
gdcm::BitmapToBitmapFilter::~~BitmapToBitmapFilter ( ) [default]
```

### 10.36.3 Member Function Documentation

#### 10.36.3.1 [GetOutput](#)()

```
const Bitmap & gdcm::BitmapToBitmapFilter::GetOutput ( ) const [inline]
```

Get Output image.

#### 10.36.3.2 [GetOutputAsBitmap](#)()

```
const Bitmap & gdcm::BitmapToBitmapFilter::GetOutputAsBitmap ( ) const
```

### 10.36.3.3 SetInput()

```
void gdcm::BitmapToBitmapFilter::SetInput (
    const Bitmap & image )
```

Set input image.

#### Examples

[BasicImageAnonymizer.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), and [StandardizeFiles.cs](#).

## 10.36.4 Member Data Documentation

### 10.36.4.1 Input

```
SmartPointer<Bitmap> gdcm::BitmapToBitmapFilter::Input [protected]
```

### 10.36.4.2 Output

```
SmartPointer<Bitmap> gdcm::BitmapToBitmapFilter::Output [protected]
```

The documentation for this class was generated from the following file:

- [gdcmBitmapToBitmapFilter.h](#)

## 10.37 gdcm::BoxRegion Class Reference

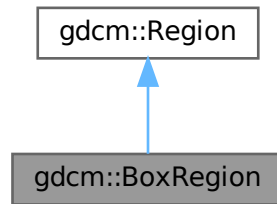
Class for manipulation box region.

```
#include <gdcmBoxRegion.h>
```

Inheritance diagram for gdcm::BoxRegion:



Collaboration diagram for gdcm::BoxRegion:



### Public Member Functions

- [BoxRegion](#) ()
- [BoxRegion](#) (const [BoxRegion](#) &)  
*copy/cstor and al.*
- [~BoxRegion](#) () override
- [size\\_t Area](#) () const override  
*compute the area*
- [Region \\* Clone](#) () const override
- [BoxRegion ComputeBoundingBox](#) () override  
*Return the Axis-Aligned minimum bounding box for all regions.*
- [bool Empty](#) () const override  
*return whether this domain is empty:*
- [unsigned int GetXMax](#) () const
- [unsigned int GetXMin](#) () const  
*Get domain.*
- [unsigned int GetYMax](#) () const
- [unsigned int GetYMin](#) () const
- [unsigned int GetZMax](#) () const
- [unsigned int GetZMin](#) () const
- [bool IsValid](#) () const override  
*return whether this is valid domain*
- [void operator=](#) (const [BoxRegion](#) &)
- [void Print](#) (std::ostream &os=std::cout) const override  
*Print.*
- [void SetDomain](#) (unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax)  
*Set domain.*

### Public Member Functions inherited from [gdcm::Region](#)

- [Region](#) ()
- virtual [~Region](#) ()

## Static Public Member Functions

- static [BoxRegion BoundingBox](#) ([BoxRegion](#) const &b1, [BoxRegion](#) const &b2)

*Helper class to compute the bounding box of two [BoxRegion](#).*

### 10.37.1 Detailed Description

Class for manipulation box region.

This is a very simple implementation of the [Region](#) class. It only support 3D box type region. It assumes the 3D Box does not have a tilt Origin is as (0,0,0)

#### Examples

[ExtractImageRegion.cs](#), and [ExtractImageRegionWithLUT.cs](#).

### 10.37.2 Constructor & Destructor Documentation

#### 10.37.2.1 [BoxRegion\(\)](#) [1/2]

```
gdcM::BoxRegion::BoxRegion ( )
```

#### 10.37.2.2 [~BoxRegion\(\)](#)

```
gdcM::BoxRegion::~~BoxRegion ( ) [override]
```

#### 10.37.2.3 [BoxRegion\(\)](#) [2/2]

```
gdcM::BoxRegion::BoxRegion (
    const BoxRegion & )
```

copy/cstor and al.

### 10.37.3 Member Function Documentation

#### 10.37.3.1 [Area\(\)](#)

```
size_t gdcM::BoxRegion::Area ( ) const [override], [virtual]
```

compute the area

Implements [gdcM::Region](#).

### 10.37.3.2 BoundingBox()

```
static BoxRegion gdcm::BoxRegion::BoundingBox (
    BoxRegion const & b1,
    BoxRegion const & b2 ) [static]
```

Helper class to compute the bounding box of two [BoxRegion](#).

### 10.37.3.3 Clone()

```
Region * gdcm::BoxRegion::Clone ( ) const [override], [virtual]
```

Implements [gdcm::Region](#).

### 10.37.3.4 ComputeBoundingBox()

```
BoxRegion gdcm::BoxRegion::ComputeBoundingBox ( ) [override], [virtual]
```

Return the Axis-Aligned minimum bounding box for all regions.

Implements [gdcm::Region](#).

### 10.37.3.5 Empty()

```
bool gdcm::BoxRegion::Empty ( ) const [override], [virtual]
```

return whether this domain is empty:

Implements [gdcm::Region](#).

### 10.37.3.6 GetXMax()

```
unsigned int gdcm::BoxRegion::GetXMax ( ) const
```

### 10.37.3.7 GetXMin()

```
unsigned int gdcm::BoxRegion::GetXMin ( ) const
```

Get domain.

### 10.37.3.8 GetYMax()

```
unsigned int gdcm::BoxRegion::GetYMax ( ) const
```

### 10.37.3.9 GetYMin()

```
unsigned int gdcm::BoxRegion::GetYMin ( ) const
```

### 10.37.3.10 GetZMax()

```
unsigned int gdcm::BoxRegion::GetZMax ( ) const
```

### 10.37.3.11 GetZMin()

```
unsigned int gdcm::BoxRegion::GetZMin ( ) const
```

### 10.37.3.12 IsValid()

```
bool gdcm::BoxRegion::IsValid ( ) const [override], [virtual]
```

return whether this is valid domain

Implements [gdcm::Region](#).

### 10.37.3.13 operator=()

```
void gdcm::BoxRegion::operator= (
    const BoxRegion & )
```

### 10.37.3.14 Print()

```
void gdcm::BoxRegion::Print (
    std::ostream & os = std::cout ) const [override], [virtual]
```

Print.

Reimplemented from [gdcm::Region](#).

### 10.37.3.15 SetDomain()

```
void gdcm::BoxRegion::SetDomain (
    unsigned int xmin,
    unsigned int xmax,
    unsigned int ymin,
    unsigned int ymax,
    unsigned int zmin,
    unsigned int zmax )
```

Set domain.

#### Examples

[ExtractImageRegion.cs](#), and [ExtractImageRegionWithLUT.cs](#).

The documentation for this class was generated from the following file:

- [gdcmBoxRegion.h](#)

## 10.38 gdcm::ByteBuffer Class Reference

[ByteBuffer](#).

```
#include <gdcmByteBuffer.h>
```

### Public Member Functions

- [ByteBuffer](#) ()
- char \* [Get](#) (int len)
- const char \* [GetStart](#) () const
- void [ShiftEnd](#) (int len)
- void [UpdatePosition](#) ()

### 10.38.1 Detailed Description

[ByteBuffer](#).

Detailed description here

#### Note

looks like a std::streambuf or std::filebuf class with the get and peek pointer

## 10.38.2 Constructor & Destructor Documentation

### 10.38.2.1 ByteBuffer()

```
gdcm::ByteBuffer::ByteBuffer ( ) [inline]
```

## 10.38.3 Member Function Documentation

### 10.38.3.1 Get()

```
char * gdcm::ByteBuffer::Get (
    int len ) [inline]
```

### 10.38.3.2 GetStart()

```
const char * gdcm::ByteBuffer::GetStart ( ) const [inline]
```

### 10.38.3.3 ShiftEnd()

```
void gdcm::ByteBuffer::ShiftEnd (
    int len ) [inline]
```

### 10.38.3.4 UpdatePosition()

```
void gdcm::ByteBuffer::UpdatePosition ( ) [inline]
```

The documentation for this class was generated from the following file:

- [gdcmByteBuffer.h](#)

## 10.39 gdcm::ByteSwap< T > Class Template Reference

[ByteSwap](#).

```
#include <gdcmByteSwap.h>
```

### Static Public Member Functions

- static void [Swap](#) (T &p)
- static void [SwapFromSwapCodeIntoSystem](#) (T &p, [SwapCode](#) const &sc)
- static void [SwapRange](#) (T \*p, unsigned int num)
- static void [SwapRangeFromSwapCodeIntoSystem](#) (T \*p, [SwapCode](#) const &sc, std::streamoff num)
- static bool [SystemIsBigEndian](#) ()
- static bool [SystemIsLittleEndian](#) ()



## 10.39.1 Detailed Description

```
template<class T>
class gdcm::ByteSwap< T >
```

[ByteSwap](#).

Perform machine dependent byte swapping (Little Endian, Big Endian, Bad Little Endian, Bad Big Endian). TODO: bswap\_32 / bswap\_64 ...

### Examples

[TestByteSwap.cxx](#).

## 10.39.2 Member Function Documentation

### 10.39.2.1 Swap()

```
template<class T >
static void gdcm::ByteSwap< T >::Swap (
    T & p ) [static]
```

### 10.39.2.2 SwapFromSwapCodeIntoSystem()

```
template<class T >
static void gdcm::ByteSwap< T >::SwapFromSwapCodeIntoSystem (
    T & p,
    SwapCode const & sc ) [static]
```

### Examples

[TestByteSwap.cxx](#).

### 10.39.2.3 SwapRange()

```
template<class T >
static void gdcm::ByteSwap< T >::SwapRange (
    T * p,
    unsigned int num ) [static]
```

### 10.39.2.4 SwapRangeFromSwapCodeIntoSystem()

```
template<class T >
static void gdcM::ByteSwap< T >::SwapRangeFromSwapCodeIntoSystem (
    T * p,
    SwapCode const & sc,
    std::streamoff num ) [static]
```

#### Examples

[TestByteSwap.cxx](#).

### 10.39.2.5 SystemIsBigEndian()

```
template<class T >
static bool gdcM::ByteSwap< T >::SystemIsBigEndian ( ) [static]
```

Query the machine Endian-ness.

### 10.39.2.6 SystemIsLittleEndian()

```
template<class T >
static bool gdcM::ByteSwap< T >::SystemIsLittleEndian ( ) [static]
```

The documentation for this class was generated from the following file:

- [gdcMByteSwap.h](#)

## 10.40 gdcM::ByteSwapFilter Class Reference

[ByteSwapFilter](#).

```
#include <gdcMByteSwapFilter.h>
```

#### Public Member Functions

- [ByteSwapFilter](#) (const [ByteSwapFilter](#) &)=delete
- [ByteSwapFilter](#) ([DataSet](#) &ds)
- [~ByteSwapFilter](#) ()=default
- bool [ByteSwap](#) ()
- [ByteSwapFilter](#) & [operator=](#) (const [ByteSwapFilter](#) &)=delete
- void [SetByteSwapTag](#) (bool b)

## 10.40.1 Detailed Description

[ByteSwapFilter](#).

In place byte-swapping of a dataset FIXME: FL status ??

## 10.40.2 Constructor & Destructor Documentation

### 10.40.2.1 ByteSwapFilter() [1/2]

```
gdcm::ByteSwapFilter::ByteSwapFilter (  
    DataSet & ds ) [inline]
```

### 10.40.2.2 ~ByteSwapFilter()

```
gdcm::ByteSwapFilter::~~ByteSwapFilter ( ) [default]
```

### 10.40.2.3 ByteSwapFilter() [2/2]

```
gdcm::ByteSwapFilter::ByteSwapFilter (  
    const ByteSwapFilter & ) [delete]
```

## 10.40.3 Member Function Documentation

### 10.40.3.1 ByteSwap()

```
bool gdcm::ByteSwapFilter::ByteSwap ( )
```

Referenced by [gdcm::Item::Read\(\)](#).

### 10.40.3.2 operator=()

```
ByteSwapFilter & gdcm::ByteSwapFilter::operator= (  
    const ByteSwapFilter & ) [delete]
```

### 10.40.3.3 SetByteSwapTag()

```
void gdcm::ByteSwapFilter::SetByteSwapTag (  
    bool b ) [inline]
```

Referenced by [gdcm::Item::Read\(\)](#).

The documentation for this class was generated from the following file:

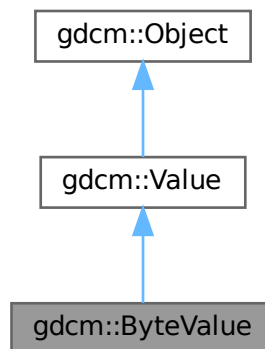
- [gdcmByteSwapFilter.h](#)

## 10.41 gdcm::ByteValue Class Reference

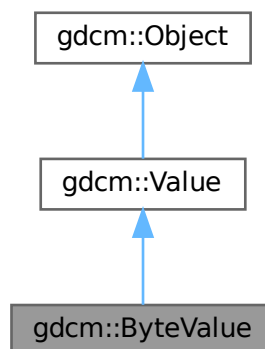
Class to represent binary value (array of bytes)

```
#include <gdcmByteValue.h>
```

Inheritance diagram for gdcm::ByteValue:



Collaboration diagram for gdcm::ByteValue:



**Public Member Functions**

- [ByteValue](#) (const char \*array=nullptr, [VL](#) const &vl=0)
- [ByteValue](#) (std::vector< char > &v)
- [~ByteValue](#) () override
- void [Append](#) ([ByteValue](#) const &bv)
- void [Clear](#) () override
- [VL ComputeLength](#) () const
- void [Fill](#) (char c)
- bool [GetBuffer](#) (char \*buffer, unsigned long length) const
- [VL GetLength](#) () const override
- const char \* [GetPointer](#) () const
- void \* [GetVoidPointer](#) ()
- const void \* [GetVoidPointer](#) () const
- bool [IsEmpty](#) () const
- bool [IsPrintable](#) ([VL](#) length) const  
*Checks whether a 'ByteValue' is printable or not (in order to avoid corrupting the terminal of invocation when printing) / don't think this function is working since it does not handle UNICODE or character set...*
- [operator const std::vector< char > &](#) () const
- [ByteValue](#) & [operator=](#) (const [ByteValue](#) &val)
- bool [operator==](#) (const [ByteValue](#) &val) const
- bool [operator==](#) (const [Value](#) &val) const override
- void [PrintASCII](#) (std::ostream &os, [VL](#) maxlength) const
- void [PrintASCIIXML](#) (std::ostream &os) const
- void [PrintGroupLength](#) (std::ostream &os)
- void [PrintHex](#) (std::ostream &os, [VL](#) maxlength) const
- void [PrintHexXML](#) (std::ostream &os) const
- void [PrintPNXML](#) (std::ostream &os) const
- template<typename TSwap >  
std::istream & [Read](#) (std::istream &is)
- template<typename TSwap , typename TType >  
std::istream & [Read](#) (std::istream &is, bool readvalues=true)
- void [SetLength](#) ([VL](#) vl) override
- template<typename TSwap , typename TType >  
std::ostream const & [Write](#) (std::ostream &os) const
- template<typename TSwap >  
std::ostream const & [Write](#) (std::ostream &os) const
- bool [WriteBuffer](#) (std::ostream &os) const

**Public Member Functions inherited from [gdcm::Value](#)**

- [Value](#) ()=default
- [~Value](#) () override=default

**Public Member Functions inherited from [gdcm::Object](#)**

- [Object](#) ()
- [Object](#) (const [Object](#) &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

## Protected Member Functions

- void [Print](#) (std::ostream &os) const override
- void [SetLengthOnly](#) ([VL](#) vl) override

## Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

### 10.41.1 Detailed Description

Class to represent binary value (array of bytes)

#### Examples

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetSubSequenceData.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

### 10.41.2 Constructor & Destructor Documentation

#### 10.41.2.1 ByteValue() [1/2]

```
gdcm::ByteValue::ByteValue (
    const char * array = nullptr,
    VL const & vl = 0 ) [inline]
```

References [gdcmDebugMacro](#).

#### 10.41.2.2 ByteValue() [2/2]

```
gdcm::ByteValue::ByteValue (
    std::vector< char > & v ) [inline]
```

#### Warning

casting to uint32\_t

#### 10.41.2.3 ~ByteValue()

```
gdcm::ByteValue::~~ByteValue ( ) [inline], [override]
```

## 10.41.3 Member Function Documentation

### 10.41.3.1 Append()

```
void gdcmm::ByteValue::Append (
    ByteValue const & bv )
```

### 10.41.3.2 Clear()

```
void gdcmm::ByteValue::Clear ( ) [inline], [override], [virtual]
```

Implements [gdcmm::Value](#).

### 10.41.3.3 ComputeLength()

```
VL gdcmm::ByteValue::ComputeLength ( ) const [inline]
```

Referenced by [gdcmm::Fragment::Write\(\)](#).

### 10.41.3.4 Fill()

```
void gdcmm::ByteValue::Fill (
    char c ) [inline]
```

#### Examples

[DuplicatePCDE.cxx](#).

### 10.41.3.5 GetBuffer()

```
bool gdcmm::ByteValue::GetBuffer (
    char * buffer,
    unsigned long length ) const
```

#### Examples

[ExtractEncapsulatedFile.cs](#), and [FixJAIBugJPEGLS.cxx](#).

### 10.41.3.6 GetLength()

```
VL gdcm::ByteValue::GetLength ( ) const [inline], [override], [virtual]
```

Implements [gdcm::Value](#).

#### Examples

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetSubSequenceData.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

Referenced by [gdcm::SequenceOfFragments::ReadValue\(\)](#), [gdcm::Element< TVR, TVM >::Set\(\)](#), [gdcm::Element< TVR, VM::VM1\\_n >::Set\(\)](#), [gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >::SetByteValue\(\)](#), [gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap\(\)](#), [gdcm::Element< TVR, TVM >::SetNoSwap\(\)](#), [gdcm::Element< TVR, VM::VM1\\_n >::SetNoSwap\(\)](#), and [gdcm::Fragment::Write\(\)](#).

### 10.41.3.7 GetPointer()

```
const char * gdcm::ByteValue::GetPointer ( ) const [inline]
```

#### Examples

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [GetSubSequenceData.cxx](#), [MrProtocol.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

Referenced by [gdcm::SequenceOfFragments::ReadValue\(\)](#), [gdcm::Element< TVR, TVM >::Set\(\)](#), [gdcm::Element< TVR, VM::VM1\\_n >::Set\(\)](#), [gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >::SetByteValue\(\)](#), [gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap\(\)](#), [gdcm::Element< TVR, TVM >::SetNoSwap\(\)](#), and [gdcm::Element< TVR, VM::VM1\\_n >::SetNoSwap\(\)](#).

### 10.41.3.8 GetVoidPointer() [1/2]

```
void * gdcm::ByteValue::GetVoidPointer ( ) [inline]
```

### 10.41.3.9 GetVoidPointer() [2/2]

```
const void * gdcm::ByteValue::GetVoidPointer ( ) const [inline]
```

#### Examples

[FixBrokenJ2K.cxx](#).

Referenced by [gdcm::Element< TVR, VM::VM1\\_n >::Set\(\)](#).



#### 10.41.3.10 IsEmpty()

```
bool gdcmm::ByteValue::IsEmpty ( ) const [inline]
```

#### 10.41.3.11 IsPrintable()

```
bool gdcmm::ByteValue::IsPrintable (
    VL length ) const [inline]
```

Checks whether a 'ByteValue' is printable or not (in order to avoid corrupting the terminal of invocation when printing) ! don't think this function is working since it does not handle UNICODE or character set...

#### 10.41.3.12 operator const std::vector< char > &()

```
gdcmm::ByteValue::operator const std::vector< char > & ( ) const [inline]
```

#### 10.41.3.13 operator=()

```
ByteValue & gdcmm::ByteValue::operator= (
    const ByteValue & val ) [inline]
```

#### 10.41.3.14 operator==( ) [1/2]

```
bool gdcmm::ByteValue::operator== (
    const ByteValue & val ) const [inline]
```

#### 10.41.3.15 operator==( ) [2/2]

```
bool gdcmm::ByteValue::operator== (
    const Value & val ) const [inline], [override], [virtual]
```

Implements [gdcmm::Value](#).

#### 10.41.3.16 Print()

```
void gdcmm::ByteValue::Print (
    std::ostream & os ) const [inline], [override], [protected], [virtual]
```

Reimplemented from [gdcmm::Object](#).

#### 10.41.3.17 PrintASCII()

```
void gdcm::ByteValue::PrintASCII (
    std::ostream & os,
    VL maxlength ) const
```

#### 10.41.3.18 PrintASCIIXML()

```
void gdcm::ByteValue::PrintASCIIXML (
    std::ostream & os ) const
```

#### 10.41.3.19 PrintGroupLength()

```
void gdcm::ByteValue::PrintGroupLength (
    std::ostream & os ) [inline]
```

#### 10.41.3.20 PrintHex()

```
void gdcm::ByteValue::PrintHex (
    std::ostream & os,
    VL maxlength ) const
```

#### 10.41.3.21 PrintHexXML()

```
void gdcm::ByteValue::PrintHexXML (
    std::ostream & os ) const
```

#### 10.41.3.22 PrintPNXML()

```
void gdcm::ByteValue::PrintPNXML (
    std::ostream & os ) const
```

To Print Values in Native DICOM format

#### 10.41.3.23 Read() [1/2]

```
template<typename TSwap >
std::istream & gdcm::ByteValue::Read (
    std::istream & is ) [inline]
```

#### 10.41.3.24 Read() [2/2]

```
template<typename TSwap , typename TType >
std::istream & gdcm::ByteValue::Read (
    std::istream & is,
    bool readvalues = true ) [inline]
```

#### 10.41.3.25 SetLength()

```
void gdcm::ByteValue::SetLength (
    VL vl ) [override], [virtual]
```

Implements [gdcm::Value](#).

#### 10.41.3.26 SetLengthOnly()

```
void gdcm::ByteValue::SetLengthOnly (
    VL vl ) [inline], [override], [protected], [virtual]
```

Reimplemented from [gdcm::Value](#).

#### 10.41.3.27 Write() [1/2]

```
template<typename TSwap , typename TType >
std::ostream const & gdcm::ByteValue::Write (
    std::ostream & os ) const [inline]
```

Referenced by [gdcm::Fragment::Write\(\)](#).

#### 10.41.3.28 Write() [2/2]

```
template<typename TSwap >
std::ostream const & gdcm::ByteValue::Write (
    std::ostream & os ) const [inline]
```

#### 10.41.3.29 WriteBuffer()

```
bool gdcm::ByteValue::WriteBuffer (
    std::ostream & os ) const [inline]
```

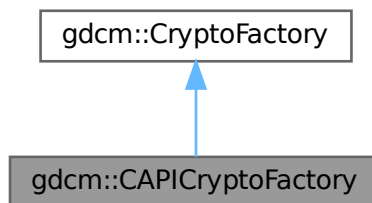
The documentation for this class was generated from the following file:

- [gdcmByteValue.h](#)

## 10.42 gdcm::CAPICryptoFactory Class Reference

```
#include <gdcmCAPICryptoFactory.h>
```

Inheritance diagram for gdcm::CAPICryptoFactory:



Collaboration diagram for gdcm::CAPICryptoFactory:



### Public Member Functions

- [CAPICryptoFactory](#) ([CryptoLib](#) id)
- [CryptographicMessageSyntax](#) \* [CreateCMSProvider](#) ()

### Additional Inherited Members

### Public Types inherited from [gdcm::CryptoFactory](#)

- enum [CryptoLib](#) {  
    [DEFAULT](#) = 0 ,  
    [OPENSSL](#) = 1 ,  
    [CAPI](#) = 2 ,  
    [OPENSSLP7](#) = 3 }

## Static Public Member Functions inherited from [gdcM::CryptoFactory](#)

- static [CryptoFactory](#) \* [GetFactoryInstance](#) ([CryptoLib](#) id=DEFAULT)

## Protected Member Functions inherited from [gdcM::CryptoFactory](#)

- [CryptoFactory](#) ()=default
- [CryptoFactory](#) ([CryptoLib](#) id)
- [~CryptoFactory](#) ()=default

## 10.42.1 Constructor & Destructor Documentation

### 10.42.1.1 CAPICryptoFactory()

```
gdcM::CAPICryptoFactory::CAPICryptoFactory (
    CryptoLib id )
```

## 10.42.2 Member Function Documentation

### 10.42.2.1 CreateCMSProvider()

```
CryptographicMessageSyntax * gdcM::CAPICryptoFactory::CreateCMSProvider ( ) [virtual]
```

Implements [gdcM::CryptoFactory](#).

The documentation for this class was generated from the following file:

- [gdcMCAPICryptoFactory.h](#)

## 10.43 gdcM::CAPICryptographicMessageSyntax Class Reference

```
#include <gdcMCAPICryptographicMessageSyntax.h>
```

Inheritance diagram for gdcM::CAPICryptographicMessageSyntax:



Collaboration diagram for `gdcm::CAPICryptographicMessageSyntax`:



### Public Member Functions

- [CAPICryptographicMessageSyntax](#) ()
- [~CAPICryptographicMessageSyntax](#) ()
- bool [Decrypt](#) (char \*output, size\_t &outlen, const char \*array, size\_t len) const  
*decrypt content from a CMS envelopedData structure*
- bool [Encrypt](#) (char \*output, size\_t &outlen, const char \*array, size\_t len) const  
*create a CMS envelopedData structure*
- [CipherTypes](#) [GetCipherType](#) () const
- bool [GetInitialized](#) () const
- bool [ParseCertificateFile](#) (const char \*filename)
- bool [ParseKeyFile](#) (const char \*filename)
- void [SetCipherType](#) ([CipherTypes](#) type)
- bool [SetPassword](#) (const char \*pass, size\_t passLen)

### Public Member Functions inherited from [gdcm::CryptographicMessageSyntax](#)

- [CryptographicMessageSyntax](#) ()=default
- [CryptographicMessageSyntax](#) (const [CryptographicMessageSyntax](#) &)=delete
- virtual [~CryptographicMessageSyntax](#) ()=default
- void [operator=](#) (const [CryptographicMessageSyntax](#) &)=delete

### Additional Inherited Members

### Public Types inherited from [gdcm::CryptographicMessageSyntax](#)

- enum [CipherTypes](#) {  
[DES3\\_CIPHER](#) ,  
[AES128\\_CIPHER](#) ,  
[AES192\\_CIPHER](#) ,  
[AES256\\_CIPHER](#) }

## 10.43.1 Constructor & Destructor Documentation

### 10.43.1.1 CAPICryptographicMessageSyntax()

```
gdcmm::CAPICryptographicMessageSyntax::CAPICryptographicMessageSyntax ( )
```

### 10.43.1.2 ~CAPICryptographicMessageSyntax()

```
gdcmm::CAPICryptographicMessageSyntax::~~CAPICryptographicMessageSyntax ( )
```

## 10.43.2 Member Function Documentation

### 10.43.2.1 Decrypt()

```
bool gdcmm::CAPICryptographicMessageSyntax::Decrypt (
    char * output,
    size_t & outlen,
    const char * array,
    size_t len ) const [virtual]
```

decrypt content from a CMS envelopedData structure

Implements [gdcmm::CryptographicMessageSyntax](#).

### 10.43.2.2 Encrypt()

```
bool gdcmm::CAPICryptographicMessageSyntax::Encrypt (
    char * output,
    size_t & outlen,
    const char * array,
    size_t len ) const [virtual]
```

create a CMS envelopedData structure

Implements [gdcmm::CryptographicMessageSyntax](#).

### 10.43.2.3 GetCipherType()

```
CipherTypes gdcmm::CAPICryptographicMessageSyntax::GetCipherType ( ) const [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

### 10.43.2.4 GetInitialized()

```
bool gdcmm::CAPICryptographicMessageSyntax::GetInitialized ( ) const [inline]
```

#### 10.43.2.5 ParseCertificateFile()

```
bool gdcM::CAPICryptographicMessageSyntax::ParseCertificateFile (
    const char * filename ) [virtual]
```

Implements [gdcM::CryptographicMessageSyntax](#).

#### 10.43.2.6 ParseKeyFile()

```
bool gdcM::CAPICryptographicMessageSyntax::ParseKeyFile (
    const char * filename ) [virtual]
```

Implements [gdcM::CryptographicMessageSyntax](#).

#### 10.43.2.7 SetCipherType()

```
void gdcM::CAPICryptographicMessageSyntax::SetCipherType (
    CipherTypes type ) [virtual]
```

Implements [gdcM::CryptographicMessageSyntax](#).

#### 10.43.2.8 SetPassword()

```
bool gdcM::CAPICryptographicMessageSyntax::SetPassword (
    const char * pass,
    size_t passLen ) [virtual]
```

Implements [gdcM::CryptographicMessageSyntax](#).

The documentation for this class was generated from the following file:

- [gdcMCAPICryptographicMessageSyntax.h](#)

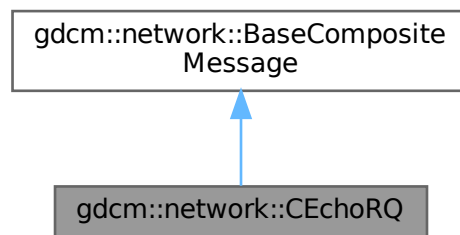


## 10.44 gdcm::network::CEchoRQ Class Reference

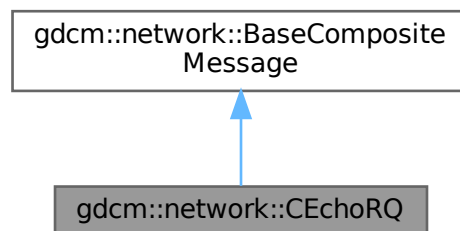
[CEchoRQ](#).

```
#include <gdcmCEchoMessages.h>
```

Inheritance diagram for gdcm::network::CEchoRQ:



Collaboration diagram for gdcm::network::CEchoRQ:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseRootQuery](#) \*inRootQuery) override

### Public Member Functions inherited from [gdcm::network::BaseCompositeMessage](#)

- virtual `~BaseCompositeMessage` ()=default

## Public Attributes

- [UIComp AffectedSOPClassUID](#)
- `uint16_t` [MessageID](#)

### 10.44.1 Detailed Description

[CEchoRQ](#).

this file defines the messages for the cecho action

### 10.44.2 Member Function Documentation

#### 10.44.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcmm::network::CEchoRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery ) [override], [virtual]
```

Implements [gdcmm::network::BaseCompositeMessage](#).

### 10.44.3 Member Data Documentation

#### 10.44.3.1 AffectedSOPClassUID

[UIComp](#) `gdcmm::network::CEchoRQ::AffectedSOPClassUID`

#### 10.44.3.2 MessageID

`uint16_t` `gdcmm::network::CEchoRQ::MessageID`

The documentation for this class was generated from the following files:

- [gdcmmCEchoMessages.h](#)
- [gdcmmDIMSE.h](#)

## 10.45 gdcm::network::CEchoRSP Class Reference

[CEchoRSP](#) this file defines the messages for the cecho action.

```
#include <gdcmCEchoMessages.h>
```

Inheritance diagram for gdcm::network::CEchoRSP:



Collaboration diagram for gdcm::network::CEchoRSP:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet` (const [DataSet](#) \*inDataSet)

### Public Member Functions inherited from [gdcm::network::BaseCompositeMessage](#)

- virtual `~BaseCompositeMessage ()=default`
- virtual `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseRootQuery](#) \*inRootQuery)=0

### 10.45.1 Detailed Description

[CEchoRSP](#) this file defines the messages for the cecho action.

### 10.45.2 Member Function Documentation

#### 10.45.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::CEchoRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

The documentation for this class was generated from the following file:

- [gdcmCEchoMessages.h](#)

## 10.46 gdcm::network::CFind Class Reference

```
#include <gdcmDIMSE.h>
```

### 10.46.1 Detailed Description

PS 3.4 - 2009 [Table B.2-1 C-STORE STATUS](#)

The documentation for this class was generated from the following file:

- [gdcmDIMSE.h](#)

## 10.47 gdcm::network::CFindCancelRQ Class Reference

[CFindCancelRQ](#) this file defines the messages for the cfind action.

```
#include <gdcmCFindMessages.h>
```

Inheritance diagram for gdcm::network::CFindCancelRQ:



Collaboration diagram for gdcm::network::CFindCancelRQ:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)`

### Public Member Functions inherited from [gdcm::network::BaseCompositeMessage](#)

- `virtual ~BaseCompositeMessage ()=default`
- `virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)=0`

## 10.47.1 Detailed Description

[CFindCancelRQ](#) this file defines the messages for the cfind action.

## 10.47.2 Member Function Documentation

### 10.47.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::CFindCancelRQ::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

The documentation for this class was generated from the following file:

- [gdcmCFindMessages.h](#)

## 10.48 gdcm::network::CFindRQ Class Reference

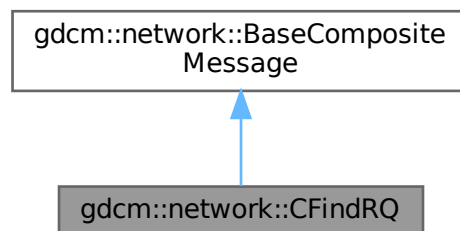
[CFindRQ](#).

```
#include <gdcmCFindMessages.h>
```

Inheritance diagram for `gdcm::network::CFindRQ`:



Collaboration diagram for `gdcm::network::CFindRQ`:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseRootQuery](#) \*inRootQuery) override

### Public Member Functions inherited from [gdcm::network::BaseCompositeMessage](#)

- virtual `~BaseCompositeMessage` ()=default

## 10.48.1 Detailed Description

[CFindRQ](#).

this file defines the messages for the cfind action

## 10.48.2 Member Function Documentation

### 10.48.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::CFindRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery ) [override], [virtual]
```

Implements [gdcm::network::BaseCompositeMessage](#).

The documentation for this class was generated from the following file:

- [gdcmCFindMessages.h](#)

## 10.49 gdcm::network::CFindRSP Class Reference

[CFindRSP](#) this file defines the messages for the cfind action.

```
#include <gdcmCFindMessages.h>
```

Inheritance diagram for gdcm::network::CFindRSP:



Collaboration diagram for `gdcm::network::CFindRSP`:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)`

### Public Member Functions inherited from [gdcm::network::BaseCompositeMessage](#)

- `virtual ~BaseCompositeMessage ()=default`
- `virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)=0`

## 10.49.1 Detailed Description

[CFindRSP](#) this file defines the messages for the cfind action.

## 10.49.2 Member Function Documentation

### 10.49.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::CFindRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

The documentation for this class was generated from the following file:

- [gdcmCFindMessages.h](#)



## 10.50 gdcm::Cleaner Class Reference

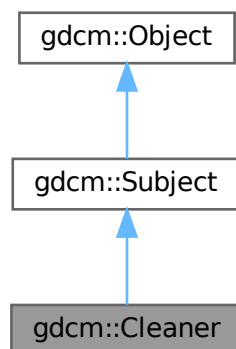
[Cleaner.](#)

```
#include <gdcmCleaner.h>
```

Inheritance diagram for gdcm::Cleaner:



Collaboration diagram for gdcm::Cleaner:



## Public Member Functions

- [Cleaner](#) ()
- [~Cleaner](#) () override
- bool [Clean](#) ()
  - main loop*
- bool [Empty](#) (DPath const &dpath)
- bool [Empty](#) (PrivateTag const &pt)
- bool [Empty](#) (Tag const &t)
- bool [Empty](#) (VR const &vr)
- void [EmptyWhenScrubFails](#) (bool empty)
  - Should I empty instead of scrub upon failure.*
- [File](#) & [GetFile](#) ()
- bool [Preserve](#) (DPath const &dpath)
- bool [Remove](#) (DPath const &dpath)
- bool [Remove](#) (PrivateTag const &pt)
- bool [Remove](#) (Tag const &t)
- bool [Remove](#) (VR const &vr)
- void [RemoveAllGroupLength](#) (bool remove)
  - Should I remove all group length (deprecated). Default: true.*
- void [RemoveAllIllegal](#) (bool remove)
  - Should I remove all illegal attribute. Default: true.*
- void [RemoveAllMissingPrivateCreator](#) (bool remove)
- bool [RemoveMissingPrivateCreator](#) (Tag const &t)
- bool [Scrub](#) (DPath const &dpath)
- bool [Scrub](#) (PrivateTag const &pt)
- bool [Scrub](#) (Tag const &t)
  - Clean digital tash (typically SIEMENS CSA header):*
- bool [Scrub](#) (VR const &vr)
- void [SetFile](#) (const [File](#) &f)
  - Set/Get File.*

## Public Member Functions inherited from [gdcm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*) const
- [Command](#) \* [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

## Static Public Member Functions

- static [SmartPointer](#)< [Cleaner](#) > [New](#) ()  
*for wrapped language: instantiate a reference counted object*

## Additional Inherited Members

## Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

### 10.50.1 Detailed Description

[Cleaner](#).

This class implement the Subject/Observer pattern trigger the following event:

- [AnonymizeEvent](#)
- [IterationEvent](#)
- [StartEvent](#)
- [EndEvent](#)

#### Examples

[Cleaner.cs](#).

### 10.50.2 Constructor & Destructor Documentation

#### 10.50.2.1 Cleaner()

```
gdcm::Cleaner::Cleaner ( )
```

### 10.50.2.2 ~Cleaner()

```
gdcM::Cleaner::~~Cleaner ( ) [override]
```

## 10.50.3 Member Function Documentation

### 10.50.3.1 Clean()

```
bool gdcM::Cleaner::Clean ( )
```

main loop

#### Examples

[Cleaner.cs](#).

### 10.50.3.2 Empty() [1/4]

```
bool gdcM::Cleaner::Empty (
    DPath const & dpath )
```

### 10.50.3.3 Empty() [2/4]

```
bool gdcM::Cleaner::Empty (
    PrivateTag const & pt )
```

### 10.50.3.4 Empty() [3/4]

```
bool gdcM::Cleaner::Empty (
    Tag const & t )
```

#### Examples

[Cleaner.cs](#).

### 10.50.3.5 Empty() [4/4]

```
bool gdcM::Cleaner::Empty (
    VR const & vr )
```

### 10.50.3.6 EmptyWhenScrubFails()

```
void gdcm::Cleaner::EmptyWhenScrubFails (
    bool empty )
```

Should I empty instead of scrub upon failure.

### 10.50.3.7 GetFile()

```
File & gdcm::Cleaner::GetFile ( ) [inline]
```

#### Examples

[Cleaner.cs.](#)

### 10.50.3.8 New()

```
static SmartPointer< Cleaner > gdcm::Cleaner::New ( ) [inline], [static]
```

for wrapped language: instantiate a reference counted object

#### Examples

[Cleaner.cs.](#)

### 10.50.3.9 Preserve()

```
bool gdcm::Cleaner::Preserve (
    DPath const & dpath )
```

#### Examples

[Cleaner.cs.](#)

### 10.50.3.10 Remove() [1/4]

```
bool gdcm::Cleaner::Remove (
    DPath const & dpath )
```

### 10.50.3.11 Remove() [2/4]

```
bool gdcm::Cleaner::Remove (
    PrivateTag const & pt )
```

**10.50.3.12 Remove() [3/4]**

```
bool gdcM::Cleaner::Remove (
    Tag const & t )
```

**Examples**

[Cleaner.cs](#).

**10.50.3.13 Remove() [4/4]**

```
bool gdcM::Cleaner::Remove (
    VR const & vr )
```

**10.50.3.14 RemoveAllGroupLength()**

```
void gdcM::Cleaner::RemoveAllGroupLength (
    bool remove )
```

Should I remove all group length (deprecated). Default: true.

**10.50.3.15 RemoveAllIllegal()**

```
void gdcM::Cleaner::RemoveAllIllegal (
    bool remove )
```

Should I remove all illegal attribute. Default: true.

**10.50.3.16 RemoveAllMissingPrivateCreator()**

```
void gdcM::Cleaner::RemoveAllMissingPrivateCreator (
    bool remove )
```

Should I remove all private tag for which no private creator is found. Default: true

**10.50.3.17 RemoveMissingPrivateCreator()**

```
bool gdcM::Cleaner::RemoveMissingPrivateCreator (
    Tag const & t )
```

Specify a private tag (odd number) without a private creator (root level only for now):

**10.50.3.18 Scrub()** [1/4]

```
bool gdcm::Cleaner::Scrub (
    DPath const & dpath )
```

**10.50.3.19 Scrub()** [2/4]

```
bool gdcm::Cleaner::Scrub (
    PrivateTag const & pt )
```

**10.50.3.20 Scrub()** [3/4]

```
bool gdcm::Cleaner::Scrub (
    Tag const & t )
```

Clean digital tash (typically SIEMENS CSA header):

**Examples**

[Cleaner.cs](#).

**10.50.3.21 Scrub()** [4/4]

```
bool gdcm::Cleaner::Scrub (
    VR const & vr )
```

**10.50.3.22 SetFile()**

```
void gdcm::Cleaner::SetFile (
    const File & f ) [inline]
```

Set/Get [File](#).

**Examples**

[Cleaner.cs](#).

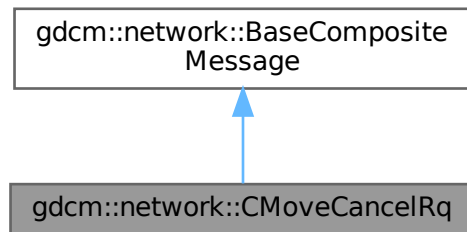
The documentation for this class was generated from the following file:

- [gdcmCleaner.h](#)

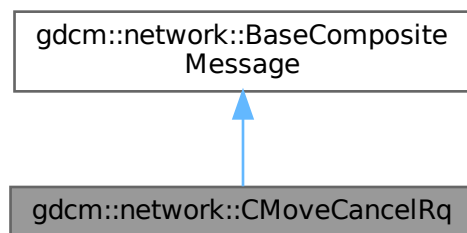
## 10.51 gdcm::network::CMoveCancelRq Class Reference

```
#include <gdcmCMoveMessages.h>
```

Inheritance diagram for `gdcm::network::CMoveCancelRq`:



Collaboration diagram for `gdcm::network::CMoveCancelRq`:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet` (const [DataSet](#) \*inDataSet)

### Public Member Functions inherited from [gdcm::network::BaseCompositeMessage](#)

- virtual `~BaseCompositeMessage` ()=default
- virtual `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseRootQuery](#) \*inRootQuery)=0



## 10.51.1 Member Function Documentation

### 10.51.1.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::CMoveCancelRq::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

The documentation for this class was generated from the following file:

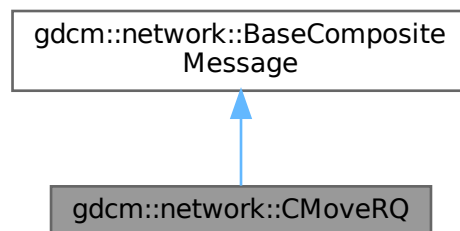
- [gdcmCMoveMessages.h](#)

## 10.52 gdcm::network::CMoveRQ Class Reference

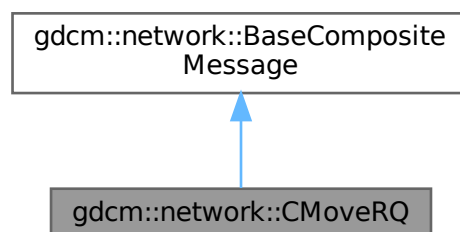
[CMoveRQ](#).

```
#include <gdcmCMoveMessages.h>
```

Inheritance diagram for gdcm::network::CMoveRQ:



Collaboration diagram for gdcm::network::CMoveRQ:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseRootQuery](#) \*inRootQuery) override

### Public Member Functions inherited from [gdcm::network::BaseCompositeMessage](#)

- virtual `~BaseCompositeMessage` ()=default

## 10.52.1 Detailed Description

[CMoveRQ](#).

this file defines the messages for the cmove action

## 10.52.2 Member Function Documentation

### 10.52.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::CMoveRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery ) [override], [virtual]
```

Implements [gdcm::network::BaseCompositeMessage](#).

The documentation for this class was generated from the following file:

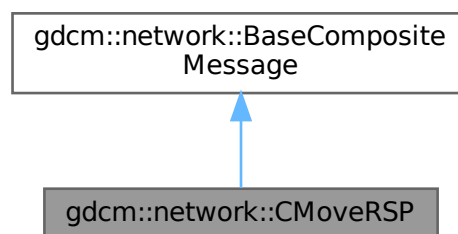
- [gdcmCMoveMessages.h](#)

## 10.53 [gdcm::network::CMoveRSP](#) Class Reference

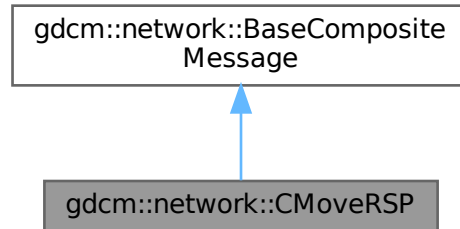
[CMoveRSP](#) this file defines the messages for the cmove action.

```
#include <gdcmCMoveMessages.h>
```

Inheritance diagram for [gdcm::network::CMoveRSP](#):



Collaboration diagram for gdcm::network::CMoveRSP:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)`

### Public Member Functions inherited from [gdcm::network::BaseCompositeMessage](#)

- `virtual ~BaseCompositeMessage ()=default`
- `virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)=0`

## 10.53.1 Detailed Description

[CMoveRSP](#) this file defines the messages for the cmove action.

## 10.53.2 Member Function Documentation

### 10.53.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::CMoveRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

The documentation for this class was generated from the following file:

- [gdcmCMoveMessages.h](#)

## 10.54 gdcm::Codec Class Reference

[Codec](#) class.

```
#include <gdcmCodec.h>
```

Inheritance diagram for `gdcm::Codec`:



Collaboration diagram for `gdcm::Codec`:



### Additional Inherited Members

#### Public Member Functions inherited from [gdcm::Coder](#)

- virtual `~Coder()`=default
- virtual bool `CanCode(TransferSyntax const &) const` =0  
*Return whether this coder support this transfer syntax (can code it)*
- virtual bool `Code(DataElement const &in_, DataElement &out_)`  
*Code.*

## Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default
- virtual bool [CanDecode](#) ([TransferSyntax](#) const &) const =0  
*Return whether this decoder support this transfer syntax (can decode it)*
- virtual bool [Decode](#) ([DataElement](#) const &, [DataElement](#) &)  
*Decode.*

## Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char \*bv, unsigned long len, std::ostream &os)

## Protected Member Functions inherited from [gdcm::Decoder](#)

- virtual bool [DecodeByStreams](#) (std::istream &, std::ostream &)

### 10.54.1 Detailed Description

[Codec](#) class.

The documentation for this class was generated from the following file:

- [gdcmCodec.h](#)

## 10.55 gdcm::Coder Class Reference

[Coder](#).

```
#include <gdcmCoder.h>
```

Inheritance diagram for [gdcm::Coder](#):



## Public Member Functions

- virtual [~Coder](#) ()=default
- virtual bool [CanCode](#) ([TransferSyntax](#) const &) const =0  
*Return whether this coder support this transfer syntax (can code it)*
- virtual bool [Code](#) ([DataElement](#) const &in\_, [DataElement](#) &out\_)  
*Code.*

## Protected Member Functions

- virtual bool [InternalCode](#) (const char \*bv, unsigned long len, std::ostream &os)

## 10.55.1 Detailed Description

[Coder](#).

## 10.55.2 Constructor & Destructor Documentation

### 10.55.2.1 ~Coder()

```
virtual gdcm::Coder::~Coder ( ) [virtual], [default]
```

## 10.55.3 Member Function Documentation

### 10.55.3.1 CanCode()

```
virtual bool gdcm::Coder::CanCode (
    TransferSyntax const & ) const [pure virtual]
```

Return whether this coder support this transfer syntax (can code it)

Implemented in [gdcm::AudioCodec](#), [gdcm::ImageCodec](#), [gdcm::PDFCodec](#), [gdcm::JPEG2000Codec](#), [gdcm::JPEGCodec](#), [gdcm::JPEGLSCodec](#), [gdcm::KAKADUCodec](#), [gdcm::PGXCodec](#), [gdcm::PNMCodec](#), [gdcm::PVRGCodec](#), [gdcm::RAWCodec](#), and [gdcm::RLECodec](#).

### 10.55.3.2 Code()

```
virtual bool gdcm::Coder::Code (
    DataElement const & in_,
    DataElement & out_ ) [inline], [virtual]
```

Code.

Reimplemented in [gdcm::JPEG2000Codec](#), [gdcm::JPEGCodec](#), [gdcm::JPEGLSCodec](#), [gdcm::KAKADUCodec](#), [gdcm::PVRGCodec](#), [gdcm::RAWCodec](#), and [gdcm::RLECodec](#).

### 10.55.3.3 InternalCode()

```
virtual bool gdcm::Coder::InternalCode (
    const char * bv,
    unsigned long len,
    std::ostream & os ) [inline], [protected], [virtual]
```

Reimplemented in [gdcm::JPEG12Codec](#), [gdcm::JPEG16Codec](#), and [gdcm::JPEG8Codec](#).

The documentation for this class was generated from the following file:

- [gdcmCoder.h](#)

## 10.56 gdcm::CodeString Class Reference

[CodeString](#).

```
#include <gdcmCodeString.h>
```

### Public Types

- typedef [InternalClass::const\\_iterator](#) const\_iterator
- typedef [InternalClass::const\\_reference](#) const\_reference
- typedef [InternalClass::const\\_reverse\\_iterator](#) const\_reverse\_iterator
- typedef [InternalClass::difference\\_type](#) difference\_type
- typedef [InternalClass::iterator](#) iterator
- typedef [InternalClass::pointer](#) pointer
- typedef [InternalClass::reference](#) reference
- typedef [InternalClass::reverse\\_iterator](#) reverse\_iterator
- typedef [InternalClass::size\\_type](#) size\_type
- typedef [InternalClass::value\\_type](#) value\_type

### Public Member Functions

- [CodeString](#) ()  
*CodeString constructors.*
- [CodeString](#) (const [InternalClass](#) &s, [size\\_type](#) pos=0, [size\\_type](#) n=[InternalClass::npos](#))
- [CodeString](#) (const [value\\_type](#) \*s)
- [CodeString](#) (const [value\\_type](#) \*s, [size\\_type](#) n)
- [std::string](#) [GetAsString](#) () const  
*Return the full code string as std::string.*
- bool [IsValid](#) () const  
*Check if CodeString obj is correct..*
- [size\\_type](#) [Size](#) () const  
*Return the size of the string.*

## Protected Member Functions

- `std::string TrimInternal () const`

## Friends

- `bool operator!= (const CodeString &ref, const CodeString &cs)`
- `std::ostream & operator<< (std::ostream &os, const CodeString &str)`
- `bool operator== (const CodeString &ref, const CodeString &cs)`

## 10.56.1 Detailed Description

[CodeString](#).

This is an implementation of DICOM [VR](#): CS The ctor will properly Trim so that operator== is correct.

### Note

the ctor of [CodeString](#) will Trim the string on the fly so as to remove the extra leading and ending spaces. However it will not perform validation on the fly ([CodeString](#) obj can contains invalid char such as lower cases). This design was chosen to be a little tolerant to broken DICOM implementation, and thus allow user to compare lower case CS from there input file without the need to first rewrite them to get rid of invalid character (validation is a different operation from searching, querying).

### Warning

when writing out DICOM file it is highly recommended to perform the [IsValid\(\)](#) call, at least to check that the length of the string match the definition in the standard.

## 10.56.2 Member Typedef Documentation

### 10.56.2.1 const\_iterator

```
typedef InternalClass::const\_iterator gdcm::CodeString::const\_iterator
```

### 10.56.2.2 const\_reference

```
typedef InternalClass::const\_reference gdcm::CodeString::const\_reference
```

### 10.56.2.3 const\_reverse\_iterator

```
typedef InternalClass::const\_reverse\_iterator gdcm::CodeString::const\_reverse\_iterator
```



#### 10.56.2.4 difference\_type

```
typedef InternalClass::difference\_type gdcm::CodeString::difference_type
```

#### 10.56.2.5 iterator

```
typedef InternalClass::iterator gdcm::CodeString::iterator
```

#### 10.56.2.6 pointer

```
typedef InternalClass::pointer gdcm::CodeString::pointer
```

#### 10.56.2.7 reference

```
typedef InternalClass::reference gdcm::CodeString::reference
```

#### 10.56.2.8 reverse\_iterator

```
typedef InternalClass::reverse\_iterator gdcm::CodeString::reverse_iterator
```

#### 10.56.2.9 size\_type

```
typedef InternalClass::size\_type gdcm::CodeString::size_type
```

#### 10.56.2.10 value\_type

```
typedef InternalClass::value\_type gdcm::CodeString::value_type
```

### 10.56.3 Constructor & Destructor Documentation

#### 10.56.3.1 CodeString() [1/4]

```
gdcm::CodeString::CodeString ( ) [inline]
```

[CodeString](#) constructors.

#### 10.56.3.2 CodeString() [2/4]

```
gdcm::CodeString::CodeString (
    const value\_type * s ) [inline]
```

**10.56.3.3 CodeString()** [3/4]

```
gdcmm::CodeString::CodeString (
    const value_type * s,
    size_type n ) [inline]
```

**10.56.3.4 CodeString()** [4/4]

```
gdcmm::CodeString::CodeString (
    const InternalClass & s,
    size_type pos = 0,
    size_type n = InternalClass::npos ) [inline]
```

**10.56.4 Member Function Documentation****10.56.4.1 GetAsString()**

```
std::string gdcmm::CodeString::GetAsString ( ) const [inline]
```

Return the full code string as std::string.

**10.56.4.2 IsValid()**

```
bool gdcmm::CodeString::IsValid ( ) const
```

Check if [CodeString](#) obj is correct..

**10.56.4.3 Size()**

```
size_type gdcmm::CodeString::Size ( ) const [inline]
```

Return the size of the string.

**10.56.4.4 TrimInternal()**

```
std::string gdcmm::CodeString::TrimInternal ( ) const [inline], [protected]
```

**10.56.5 Friends And Related Symbol Documentation****10.56.5.1 operator"!="**

```
bool operator!= (
    const CodeString & ref,
    const CodeString & cs ) [friend]
```

### 10.56.5.2 operator<<

```
std::ostream & operator<< (  
    std::ostream & os,  
    const CodeString & str ) [friend]
```

### 10.56.5.3 operator==

```
bool operator== (  
    const CodeString & ref,  
    const CodeString & cs ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmCodeString.h](#)

## 10.57 gdcm::Command Class Reference

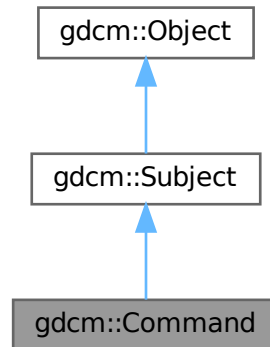
[Command](#) superclass for callback/observer methods.

```
#include <gdcmCommand.h>
```

Inheritance diagram for gdcm::Command:



Collaboration diagram for `gdcm::Command`:



### Public Member Functions

- `Command` (const `Command` &)=delete
- virtual void `Execute` (const `Subject` \*caller, const `Event` &event)=0
- virtual void `Execute` (`Subject` \*caller, const `Event` &event)=0  
*Abstract method that defines the action to be taken by the command.*
- void `operator=` (const `Command` &)=delete

### Public Member Functions inherited from `gdcm::Subject`

- `Subject` ()
- `~Subject` () override
- unsigned long `AddObserver` (const `Event` &event, `Command` \*)
- unsigned long `AddObserver` (const `Event` &event, `Command` \*) const
- `Command` \* `GetCommand` (unsigned long tag)
- bool `HasObserver` (const `Event` &event) const
- void `InvokeEvent` (const `Event` &)
- void `InvokeEvent` (const `Event` &) const
- void `RemoveAllObservers` ()
- void `RemoveObserver` (unsigned long tag)

### Public Member Functions inherited from `gdcm::Object`

- `Object` ()
- `Object` (const `Object` &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual `~Object` ()
- void `operator=` (const `Object` &)
- virtual void `Print` (std::ostream &) const

**Protected Member Functions**

- [Command](#) ()
- [~Command](#) () override

**Protected Member Functions inherited from [gdcmm::Object](#)**

- void [Register](#) ()
- void [UnRegister](#) ()

**10.57.1 Detailed Description**

[Command](#) superclass for callback/observer methods.

See also

[Subject](#)

**10.57.2 Constructor & Destructor Documentation****10.57.2.1 Command() [1/2]**

```
gdcmm::Command::Command (
    const Command & ) [delete]
```

**10.57.2.2 Command() [2/2]**

```
gdcmm::Command::Command ( ) [protected]
```

**10.57.2.3 ~Command()**

```
gdcmm::Command::~~Command ( ) [override], [protected]
```

**10.57.3 Member Function Documentation****10.57.3.1 Execute() [1/2]**

```
virtual void gdcmm::Command::Execute (
    const Subject * caller,
    const Event & event ) [pure virtual]
```

Abstract method that defines the action to be taken by the command. This variant is expected to be used when requests comes from a const [Object](#)

Implemented in [gdcmm::SimpleMemberCommand< T >](#), and [gdcmm::MemberCommand< T >](#).

### 10.57.3.2 Execute() [2/2]

```
virtual void gdcM::Command::Execute (
    Subject * caller,
    const Event & event ) [pure virtual]
```

Abstract method that defines the action to be taken by the command.

Implemented in [gdcM::SimpleMemberCommand< T >](#), and [gdcM::MemberCommand< T >](#).

### 10.57.3.3 operator=()

```
void gdcM::Command::operator= (
    const Command & ) [delete]
```

The documentation for this class was generated from the following file:

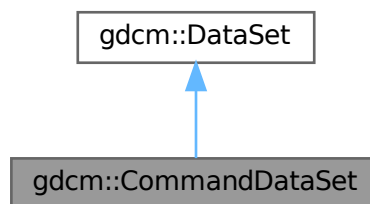
- [gdcMCommand.h](#)

## 10.58 gdcM::CommandDataSet Class Reference

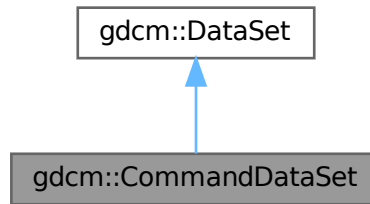
Class to represent a [Command DataSet](#).

```
#include <gdcMCommandDataSet.h>
```

Inheritance diagram for gdcM::CommandDataSet:



Collaboration diagram for gdcm::CommandDataSet:



### Public Member Functions

- [CommandDataSet](#) ()=default
- [~CommandDataSet](#) ()=default
- void [Insert](#) (const [DataElement](#) &de)
- std::istream & [Read](#) (std::istream &is)  
*Read.*
- void [Replace](#) (const [DataElement](#) &de)
- std::ostream & [Write](#) (std::ostream &os) const  
*Write.*

### Public Member Functions inherited from [gdcm::DataSet](#)

- [Iterator Begin](#) ()
- [ConstIterator Begin](#) () const
- void [Clear](#) ()
- template<typename TDE >  
unsigned int [ComputeGroupLength](#) ([Tag](#) const &tag) const
- [Iterator End](#) ()
- [ConstIterator End](#) () const
- bool [FindDataElement](#) (const [PrivateTag](#) &t) const  
*Look up if private tag 't' is present in the dataset:*
- bool [FindDataElement](#) (const [Tag](#) &t) const
- const [DataElement](#) & [FindNextDataElement](#) (const [Tag](#) &t) const
- const [DataElement](#) & [GetDataElement](#) (const [PrivateTag](#) &t) const  
*Return the dataelement.*
- const [DataElement](#) & [GetDataElement](#) (const [Tag](#) &t) const
- [DataElementSet](#) & [GetDES](#) ()
- const [DataElementSet](#) & [GetDES](#) () const
- template<typename TDE >  
[VL GetLength](#) () const
- [MediaStorage GetMediaStorage](#) () const

- `std::string GetPrivateCreator (const Tag &t) const`
- `PrivateTag GetPrivateTag (const Tag &t) const`  
*Return the private tag of the private tag 't', private creator will be set to empty if not found.*
- `void Insert (const DataElement &de)`
- `bool IsEmpty () const`  
*Returns if the dataset is empty.*
- `const DataElement & operator() (uint16_t group, uint16_t element) const`
- `DataSet & operator= (DataSet const &)=default`
- `const DataElement & operator[] (const Tag &t) const`
- `void Print (std::ostream &os, std::string const &indent="") const`
- `template<typename TDE , typename TSwap >  
std::istream & Read (std::istream &is)`
- `template<typename TDE , typename TSwap >  
std::istream & ReadNested (std::istream &is)`
- `template<typename TDE , typename TSwap >  
std::istream & ReadSelectedPrivateTags (std::istream &is, const std::set< PrivateTag > &tags, bool readvalues=true)`
- `template<typename TDE , typename TSwap >  
std::istream & ReadSelectedPrivateTagsWithLength (std::istream &is, const std::set< PrivateTag > &tags, VL &length, bool readvalues=true)`
- `template<typename TDE , typename TSwap >  
std::istream & ReadSelectedTags (std::istream &is, const std::set< Tag > &tags, bool readvalues=true)`
- `template<typename TDE , typename TSwap >  
std::istream & ReadSelectedTagsWithLength (std::istream &is, const std::set< Tag > &tags, VL &length, bool readvalues=true)`
- `template<typename TDE , typename TSwap >  
std::istream & ReadUpToTag (std::istream &is, const Tag &t, std::set< Tag > const &skiptags)`
- `template<typename TDE , typename TSwap >  
std::istream & ReadUpToTagWithLength (std::istream &is, const Tag &t, std::set< Tag > const &skiptags, VL &length)`
- `template<typename TDE , typename TSwap >  
std::istream & ReadWithLength (std::istream &is, VL &length)`
- `SizeType Remove (const Tag &tag)`  
*Completely remove a dataelement from the dataset.*
- `void Replace (const DataElement &de)`  
*Replace a dataelement with another one.*
- `void ReplaceEmpty (const DataElement &de)`  
*Only replace a DICOM attribute when it is missing or empty.*
- `SizeType Size () const`
- `template<typename TDE , typename TSwap >  
std::ostream const & Write (std::ostream &os) const`

## Friends

- `std::ostream & operator<< (std::ostream &_os, const CommandDataSet &_val)`



## Additional Inherited Members

### Public Types inherited from [gdcm::DataSet](#)

- typedef DataSet::const\_iterator [ConstIterator](#)
- typedef std::set< [DataElement](#) > [DataElementSet](#)
- typedef DataSet::iterator [Iterator](#)
- typedef DataSet::size\_type [SizeType](#)

### Protected Member Functions inherited from [gdcm::DataSet](#)

- [Tag ComputeDataElement](#) (const [PrivateTag](#) &t) const
- const [DataElement](#) & [GetDEEnd](#) () const
- void [InsertDataElement](#) (const [DataElement](#) &de)

## 10.58.1 Detailed Description

Class to represent a [Command DataSet](#).

See also

[DataSet](#)

## 10.58.2 Constructor & Destructor Documentation

### 10.58.2.1 CommandDataSet()

```
gdcm::CommandDataSet::CommandDataSet ( ) [default]
```

### 10.58.2.2 ~CommandDataSet()

```
gdcm::CommandDataSet::~~CommandDataSet ( ) [default]
```

References [gdcm::operator<<\(\)](#).

## 10.58.3 Member Function Documentation

### 10.58.3.1 Insert()

```
void gdcm::CommandDataSet::Insert (
    const DataElement & de ) [inline]
```

References [gdcmErrorMacro](#), [gdcm::Tag::GetGroup\(\)](#), and [gdcm::DataElement::GetTag\(\)](#).

### 10.58.3.2 Read()

```
std::istream & gdcM::CommandDataSet::Read (
    std::istream & is )
```

Read.

### 10.58.3.3 Replace()

```
void gdcM::CommandDataSet::Replace (
    const DataElement & de ) [inline]
```

References [gdcM::DataElement::GetTag\(\)](#).

### 10.58.3.4 Write()

```
std::ostream & gdcM::CommandDataSet::Write (
    std::ostream & os ) const
```

Write.

## 10.58.4 Friends And Related Symbol Documentation

### 10.58.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const CommandDataSet & _val ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcMCommandDataSet.h](#)

## 10.59 gdcM::network::CompositeMessageFactory Class Reference

[CompositeMessageFactory](#).

```
#include <gdcMCompositeMessageFactory.h>
```

## Static Public Member Functions

- static std::vector< [PresentationDataValue](#) > [ConstructCEchoRQ](#) (const [ULConnection](#) &inConnection)
- static std::vector< [PresentationDataValue](#) > [ConstructCFindRQ](#) (const [ULConnection](#) &inConnection, const [BaseRootQuery](#) \*inRootQuery)
- static std::vector< [PresentationDataValue](#) > [ConstructCMoveRQ](#) (const [ULConnection](#) &inConnection, const [BaseRootQuery](#) \*inRootQuery)
- static std::vector< [PresentationDataValue](#) > [ConstructCStoreRQ](#) (const [ULConnection](#) &inConnection, const [File](#) &file, bool writeDataSet=true)
- static std::vector< [PresentationDataValue](#) > [ConstructCStoreRSP](#) (const [DataSet](#) \*inDataSet, const [BasePDU](#) \*inPC)

## 10.59.1 Detailed Description

[CompositeMessageFactory](#).

This class constructs PDataPDUs, but that have been specifically constructed for the composite DICOM services (C-Echo, C-Find, C-Get, C-Move, and C-Store). It will also handle parsing the incoming data to determine which of the CompositePDUs the incoming data is, and so therefore allowing the scu to determine what to do with incoming data (if acting as a storescp server, for instance).

## 10.59.2 Member Function Documentation

### 10.59.2.1 ConstructCEchoRQ()

```
static std::vector< PresentationDataValue > gdcn::network::CompositeMessageFactory::ConstructCEchoRQ (
    const ULConnection & inConnection ) [static]
```

### 10.59.2.2 ConstructCFindRQ()

```
static std::vector< PresentationDataValue > gdcn::network::CompositeMessageFactory::ConstructCFindRQ (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery ) [static]
```

### 10.59.2.3 ConstructCMoveRQ()

```
static std::vector< PresentationDataValue > gdcn::network::CompositeMessageFactory::ConstructCMoveRQ (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery ) [static]
```

### 10.59.2.4 ConstructCStoreRQ()

```
static std::vector< PresentationDataValue > gdcM::network::CompositeMessageFactory::ConstructCStoreRQ (
    const ULConnection & inConnection,
    const File & file,
    bool writeDataSet = true ) [static]
```

### 10.59.2.5 ConstructCStoreRSP()

```
static std::vector< PresentationDataValue > gdcM::network::CompositeMessageFactory::ConstructCStoreRSP (
    const DataSet * inDataSet,
    const BasePDU * inPC ) [static]
```

The documentation for this class was generated from the following file:

- [gdcMCompositeMessageFactory.h](#)

## 10.60 gdcM::CompositeNetworkFunctions Class Reference

Composite Network Functions.

```
#include <gdcMCompositeNetworkFunctions.h>
```

### Public Types

- typedef std::vector< [KeyValuePairType](#) > [KeyValuePairArrayType](#)
- typedef std::pair< [Tag](#), std::string > [KeyValuePairType](#)

### Static Public Member Functions

- static bool [CEcho](#) (const char \*remote, uint16\_t portno, const char \*aetitle=nullptr, const char \*call=nullptr)
- static bool [CFind](#) (const char \*remote, uint16\_t portno, const [BaseRootQuery](#) \*query, std::vector< [DataSet](#) > &retDataSets, const char \*aetitle=nullptr, const char \*call=nullptr)
- static bool [CMove](#) (const char \*remote, uint16\_t portno, const [BaseRootQuery](#) \*query, uint16\_t portscp, const char \*aetitle=nullptr, const char \*call=nullptr, const char \*outputdir=nullptr)
- static [BaseRootQuery](#) \* [ConstructQuery](#) ([ERootType](#) inRootType, [EQueryLevel](#) inQueryLevel, const [DataSet](#) &queryds, [EQueryType](#) queryType=eFind)
- static [BaseRootQuery](#) \* [ConstructQuery](#) ([ERootType](#) inRootType, [EQueryLevel](#) inQueryLevel, const [KeyValuePairArrayType](#) &keys, [EQueryType](#) queryType=eFind)
- static bool [CStore](#) (const char \*remote, uint16\_t portno, const [Directory::FileNamesType](#) &filenames, const char \*aetitle=nullptr, const char \*call=nullptr)

## 10.60.1 Detailed Description

Composite Network Functions.

These functions provide a generic API to the DICOM functions implemented in GDCM. Advanced users can use this code as a template for building their own versions of these functions (for instance, to provide progress bars or some other way of handling returned query information), but for most users, these functions should be sufficient to interface with a PACS to a local machine. Note that these functions are not contained within a static class or some other class-style interface, because multiple connections can be instantiated in the same program. The DICOM standard is much more function oriented rather than class oriented in this instance, so the design of this API reflects that functional approach. These functions implements the following SCU operations:

- C-ECHO SCU
- C-FIND SCU
- C-STORE SCU
- C-MOVE SCU (+internal C-STORE SCP)

### Examples

[SendFileSCU.cs](#).

## 10.60.2 Member Typedef Documentation

### 10.60.2.1 KeyValuePairArrayType

```
typedef std::vector< KeyValuePairType > gdcm::CompositeNetworkFunctions::KeyValuePairArrayType
```

### 10.60.2.2 KeyValuePairType

```
typedef std::pair<Tag, std::string> gdcm::CompositeNetworkFunctions::KeyValuePairType
```

## 10.60.3 Member Function Documentation

### 10.60.3.1 CEcho()

```
static bool gdcm::CompositeNetworkFunctions::CEcho (  
    const char * remote,  
    uint16_t portno,  
    const char * aetitle = nullptr,  
    const char * call = nullptr ) [static]
```

The most basic network function. Use this function to ensure that the remote server is responding on the given IP and port number as expected.

**Parameters**

<i>aetitle</i>	when not set will default to 'GDCMSCU'
<i>call</i>	when not set will default to 'ANY-SCP'

**Warning**

This is an error to set remote to NULL or portno to 0

**Returns**

true if it worked.

**Examples**

[SendFileSCU.cs](#).

**10.60.3.2 CFind()**

```
static bool gdc::CompositeNetworkFunctions::CFind (
    const char * remote,
    uint16_t portno,
    const BaseRootQuery * query,
    std::vector< DataSet > & retDataSets,
    const char * aetitle = nullptr,
    const char * call = nullptr ) [static]
```

This function will use the provided query to determine what files a remote server contains that match the query strings. The return is a vector of datasets that contain tags as reported by the server. If the dataset is empty, then it is possible that an error condition was encountered; in which case, the user should monitor the error and warning streams.

**Parameters**

<i>aetitle</i>	when not set will default to 'GDCMSCU'
<i>call</i>	when not set will default to 'ANY-SCP'

**Warning**

This is an error to set remote to NULL or portno to 0

**Returns**

true if it worked.

### 10.60.3.3 CMove()

```
static bool gdcm::CompositeNetworkFunctions::CMove (
    const char * remote,
    uint16_t portno,
    const BaseRootQuery * query,
    uint16_t portscp,
    const char * aetitle = nullptr,
    const char * call = nullptr,
    const char * outputdir = nullptr ) [static]
```

This function will use the provided query to get files from a remote server. NOTE that this functionality is essentially equivalent to C-GET in the DICOM standard; however, C-GET has been deprecated, so this function allows for the user to ask a remote server for files matching a query and return them to the local machine. Files will be written to the given output directory. If the operation succeeds, the function returns true. This function is a prime candidate for being overwritten by expert users; if the datasets should remain in memory, for instance, that behavior can be changed by creating a user-level version of this function.

#### Parameters

<i>aetitle</i>	when not set will default to 'GDCMSCU'
<i>call</i>	when not set will default to 'ANY-SCP' This is an error to set remote to NULL or portno to 0 when
<i>outputdir</i>	is not set default to current dir ('.')

#### Returns

true if it worked.

### 10.60.3.4 ConstructQuery() [1/2]

```
static BaseRootQuery * gdcm::CompositeNetworkFunctions::ConstructQuery (
    ERootType inRootType,
    EQueryLevel inQueryLevel,
    const DataSet & queryds,
    EQueryType queryType = eFind ) [static]
```

This function will take a list of strings and tags and fill in a query that can be used for either CFind or CMove (depending on the input boolean

#### Parameters

<i>inMove</i> ).	Note that the caller is responsible for deleting the constructed query. This function is used to build both a move and a find query (true for inMove if it's move, false if it's find)
------------------	--

### 10.60.3.5 ConstructQuery() [2/2]

```
static BaseRootQuery * gdcm::CompositeNetworkFunctions::ConstructQuery (
```

```

ERootType inRootType,
EQueryLevel inQueryLevel,
const KeyValuePairArrayType & keys,
EQueryType queryType = eFind ) [static]

```

## Deprecated

### 10.60.3.6 CStore()

```

static bool gdcm::CompositeNetworkFunctions::CStore (
    const char * remote,
    uint16_t portno,
    const Directory::FileNamesType & filenames,
    const char * aetitle = nullptr,
    const char * call = nullptr ) [static]

```

This function will place the provided files into the remote server. The function returns true if it worked for all files.

#### Warning

the server side can refuse an association on a given file

#### Parameters

<i>aetitle</i>	when not set will default to 'GDCMSCU'
<i>call</i>	when not set will default to 'ANY-SCP'

#### Warning

This is an error to set remote to NULL or portno to 0

#### Returns

true if it worked for all files

#### Examples

[SendFileSCU.cs.](#)

The documentation for this class was generated from the following file:

- [gdcmCompositeNetworkFunctions.h](#)



## 10.61 gdcm::ConstCharWrapper Class Reference

Do not use me.

```
#include <gdcmConstCharWrapper.h>
```

### Public Member Functions

- [ConstCharWrapper](#) (const char \*i=0)
- [operator const char \\* \(\)](#) const

### 10.61.1 Detailed Description

Do not use me.

### 10.61.2 Constructor & Destructor Documentation

#### 10.61.2.1 ConstCharWrapper()

```
gdcm::ConstCharWrapper::ConstCharWrapper (  
    const char * i = 0 ) [inline]
```

### 10.61.3 Member Function Documentation

#### 10.61.3.1 operator const char \*()

```
gdcm::ConstCharWrapper::operator const char * ( ) const [inline]
```

The documentation for this class was generated from the following file:

- [gdcmConstCharWrapper.h](#)

## 10.62 gdcm::CP246ExplicitDataElement Class Reference

Class to read/write a [DataElement](#) as CP246Explicit Data [Element](#).

```
#include <gdcmCP246ExplicitDataElement.h>
```

Inheritance diagram for gdcm::CP246ExplicitDataElement:



Collaboration diagram for gdcm::CP246ExplicitDataElement:



### Public Member Functions

- [VL GetLength](#) () const
- template<typename TSwap >  
std::istream & [Read](#) (std::istream &is)

- template<typename TSwap >  
std::istream & [ReadPreValue](#) (std::istream &is)
- template<typename TSwap >  
std::istream & [ReadValue](#) (std::istream &is, bool readvalues=true)
- template<typename TSwap >  
std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)

## Public Member Functions inherited from [gdcm::DataElement](#)

- [DataElement](#) (const [DataElement](#) &\_val)
- [DataElement](#) (const [Tag](#) &t=[Tag](#)(0), const [VL](#) &vl=0, const [VR](#) &vr=[VR::INVALID](#))
- void [Clear](#) ()  
*Clear Data [Element](#) (make [Value](#) empty and invalidate [Tag](#) & [VR](#))*
- void [Empty](#) ()  
*Make Data [Element](#) empty (no [Value](#))*
- const [ByteValue](#) \* [GetByteValue](#) () const
- template<typename TDE >  
[VL](#) [GetLength](#) () const
- [SequenceOfFragments](#) \* [GetSequenceOfFragments](#) ()
- const [SequenceOfFragments](#) \* [GetSequenceOfFragments](#) () const
- [Tag](#) & [GetTag](#) ()
- const [Tag](#) & [GetTag](#) () const  
*Get [Tag](#).*
- [Value](#) & [GetValue](#) ()
- [Value](#) const & [GetValue](#) () const  
*Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):*
- [SmartPointer](#)< [SequenceOfItems](#) > [GetValueAsSQ](#) () const
- [VL](#) & [GetVL](#) ()
- const [VL](#) & [GetVL](#) () const  
*Get [VL](#).*
- [VR](#) const & [GetVR](#) () const
- bool [IsEmpty](#) () const  
*Check if Data [Element](#) is empty.*
- bool [IsUndefinedLength](#) () const  
*return if [Value](#) Length if of undefined length*
- bool [operator<](#) (const [DataElement](#) &de) const
- [DataElement](#) & [operator=](#) (const [DataElement](#) &)=default
- bool [operator==](#) (const [DataElement](#) &de) const
- template<typename TDE , typename TSwap >  
std::istream & [Read](#) (std::istream &is)
- template<typename TDE , typename TSwap >  
std::istream & [ReadOrSkip](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE , typename TSwap >  
std::istream & [ReadPreValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE , typename TSwap >  
std::istream & [ReadValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE , typename TSwap >  
std::istream & [ReadValueWithLength](#) (std::istream &is, [VL](#) &length, std::set< [Tag](#) > const &skiptags)

- `template<typename TDE , typename TSwap >`  
`std::istream & ReadWithLength (std::istream &is, VL &length)`
- `void SetByteValue (const char *array, VL length)`
- `void SetTag (const Tag &t)`
- `void SetValue (Value const &vl)`
- `void SetVL (const VL &vl)`
- `void SetVLToUndefined ()`
- `void SetVR (VR const &vr)`
- `template<typename TDE , typename TSwap >`  
`const std::ostream & Write (std::ostream &os) const`

### Additional Inherited Members

### Protected Types inherited from [gdcm::DataElement](#)

- `typedef SmartPointer< Value > ValuePtr`

### Protected Member Functions inherited from [gdcm::DataElement](#)

- `void SetValueFieldLength (VL vl, bool readvalues)`

### Protected Attributes inherited from [gdcm::DataElement](#)

- `Tag TagField`
- `ValuePtr ValueField`
- `VL ValueLengthField`
- `VR VRField`

## 10.62.1 Detailed Description

Class to read/write a [DataElement](#) as CP246Explicit Data [Element](#).

### Note

Some system are producing SQ, declare them as UN, but encode the SQ as 'Explicit' instead of Implicit

## 10.62.2 Member Function Documentation

### 10.62.2.1 GetLength()

```
VL gdcm::CP246ExplicitDataElement::GetLength ( ) const
```

### 10.62.2.2 Read()

```
template<typename TSwap >
std::istream & gdcm::CP246ExplicitDataElement::Read (
    std::istream & is )
```

### 10.62.2.3 ReadPreValue()

```
template<typename TSwap >
std::istream & gdcm::CP246ExplicitDataElement::ReadPreValue (
    std::istream & is )
```

### 10.62.2.4 ReadValue()

```
template<typename TSwap >
std::istream & gdcm::CP246ExplicitDataElement::ReadValue (
    std::istream & is,
    bool readvalues = true )
```

### 10.62.2.5 ReadWithLength()

```
template<typename TSwap >
std::istream & gdcm::CP246ExplicitDataElement::ReadWithLength (
    std::istream & is,
    VL & length )
```

The documentation for this class was generated from the following file:

- [gdcmCP246ExplicitDataElement.h](#)

## 10.63 gdcm::CryptoFactory Class Reference

Class to do handle the crypto factory.

```
#include <gdcmCryptoFactory.h>
```

Inheritance diagram for gdcm::CryptoFactory:



## Public Types

- enum [CryptoLib](#) {  
    [DEFAULT](#) = 0 ,  
    [OPENSSL](#) = 1 ,  
    [CAPI](#) = 2 ,  
    [OPENSSL7](#) = 3 }

## Public Member Functions

- virtual [CryptographicMessageSyntax](#) \* [CreateCMSProvider](#) ()=0

## Static Public Member Functions

- static [CryptoFactory](#) \* [GetFactoryInstance](#) ([CryptoLib](#) id=[DEFAULT](#))

## Protected Member Functions

- [CryptoFactory](#) ()=default
- [CryptoFactory](#) ([CryptoLib](#) id)
- [~CryptoFactory](#) ()=default

## 10.63.1 Detailed Description

Class to do handle the crypto factory.

GDCM needs to access in a platform independent way the user specified crypto engine. It can be:

- CAPI (windows only)
- OPENSSL (portable)
- OPENSSL7 (portable) By default the factory will try: CAPI if on windows OPENSSL if possible OPENSSL7 when older OpenSSL is used.

## Examples

[BasicAnonymizer.cs](#), and [ClinicalTrialIdentificationWorkflow.cs](#).

## 10.63.2 Member Enumeration Documentation

### 10.63.2.1 CryptoLib

enum [gdcm::CryptoFactory::CryptoLib](#)

## Enumerator

DEFAULT	
OPENSSL	
CAPI	
OPENSSL7	

### 10.63.3 Constructor & Destructor Documentation

#### 10.63.3.1 CryptoFactory() [1/2]

```
gdcm::CryptoFactory::CryptoFactory (
    CryptoLib id ) [inline], [protected]
```

#### 10.63.3.2 CryptoFactory() [2/2]

```
gdcm::CryptoFactory::CryptoFactory ( ) [protected], [default]
```

#### 10.63.3.3 ~CryptoFactory()

```
gdcm::CryptoFactory::~CryptoFactory ( ) [protected], [default]
```

### 10.63.4 Member Function Documentation

#### 10.63.4.1 CreateCMSProvider()

```
virtual CryptographicMessageSyntax * gdcm::CryptoFactory::CreateCMSProvider ( ) [pure virtual]
```

Implemented in [gdcm::CAPICryptoFactory](#), [gdcm::OpenSSLCryptoFactory](#), and [gdcm::OpenSSL7CryptoFactory](#).

## Examples

[BasicAnonymizer.cs](#), and [ClinicalTrialIdentificationWorkflow.cs](#).

#### 10.63.4.2 GetFactoryInstance()

```
static CryptoFactory * gdcm::CryptoFactory::GetFactoryInstance (
    CryptoLib id = DEFAULT ) [static]
```

## Examples

[BasicAnonymizer.cs](#), and [ClinicalTrialIdentificationWorkflow.cs](#).

The documentation for this class was generated from the following file:

- [gdcmCryptoFactory.h](#)

## 10.64 gdcm::CryptographicMessageSyntax Class Reference

```
#include <gdcmCryptographicMessageSyntax.h>
```

Inheritance diagram for gdcm::CryptographicMessageSyntax:



### Public Types

- enum [CipherTypes](#) {  
[DES3\\_CIPHER](#) ,  
[AES128\\_CIPHER](#) ,  
[AES192\\_CIPHER](#) ,  
[AES256\\_CIPHER](#) }

### Public Member Functions

- [CryptographicMessageSyntax](#) ()=default
- [CryptographicMessageSyntax](#) (const [CryptographicMessageSyntax](#) &)=delete
- virtual [~CryptographicMessageSyntax](#) ()=default
- virtual bool [Decrypt](#) (char \*output, size\_t &outlen, const char \*array, size\_t len) const =0  
*decrypt content from a CMS envelopedData structure*
- virtual bool [Encrypt](#) (char \*output, size\_t &outlen, const char \*array, size\_t len) const =0  
*create a CMS envelopedData structure*
- virtual [CipherTypes](#) [GetCipherType](#) () const =0
- void [operator=](#) (const [CryptographicMessageSyntax](#) &)=delete
- virtual bool [ParseCertificateFile](#) (const char \*filename)=0
- virtual bool [ParseKeyFile](#) (const char \*filename)=0
- virtual void [SetCipherType](#) ([CipherTypes](#) type)=0
- virtual bool [SetPassword](#) (const char \*pass, size\_t passLen)=0

### 10.64.1 Detailed Description

#### Examples

[BasicAnonymizer.cs](#), and [ClinicalTrialIdentificationWorkflow.cs](#).



## 10.64.2 Member Enumeration Documentation

### 10.64.2.1 CipherTypes

```
enum gdcmm::CryptographicMessageSyntax::CipherTypes
```

Enumerator

DES3_CIPHER	
AES128_CIPHER	
AES192_CIPHER	
AES256_CIPHER	

## 10.64.3 Constructor & Destructor Documentation

### 10.64.3.1 CryptographicMessageSyntax() [1/2]

```
gdcmm::CryptographicMessageSyntax::CryptographicMessageSyntax ( ) [default]
```

### 10.64.3.2 ~CryptographicMessageSyntax()

```
virtual gdcmm::CryptographicMessageSyntax::~~CryptographicMessageSyntax ( ) [virtual], [default]
```

### 10.64.3.3 CryptographicMessageSyntax() [2/2]

```
gdcmm::CryptographicMessageSyntax::CryptographicMessageSyntax (
    const CryptographicMessageSyntax & ) [delete]
```

## 10.64.4 Member Function Documentation

### 10.64.4.1 Decrypt()

```
virtual bool gdcmm::CryptographicMessageSyntax::Decrypt (
    char * output,
    size_t & outlen,
    const char * array,
    size_t len ) const [pure virtual]
```

decrypt content from a CMS envelopedData structure

Implemented in [gdcmm::CAPICryptographicMessageSyntax](#), [gdcmm::OpenSSLCryptographicMessageSyntax](#), and [gdcmm::OpenSSL7CryptographicMessageSyntax](#).

#### 10.64.4.2 Encrypt()

```
virtual bool gdcM::CryptographicMessageSyntax::Encrypt (
    char * output,
    size_t & outlen,
    const char * array,
    size_t len ) const [pure virtual]
```

create a CMS envelopedData structure

Implemented in [gdcM::CAPICryptographicMessageSyntax](#), [gdcM::OpenSSLCryptographicMessageSyntax](#), and [gdcM::OpenSSL7CryptographicMessageSyntax](#).

#### 10.64.4.3 GetCipherType()

```
virtual CipherTypes gdcM::CryptographicMessageSyntax::GetCipherType ( ) const [pure virtual]
```

Implemented in [gdcM::CAPICryptographicMessageSyntax](#), [gdcM::OpenSSLCryptographicMessageSyntax](#), and [gdcM::OpenSSL7CryptographicMessageSyntax](#).

#### 10.64.4.4 operator=()

```
void gdcM::CryptographicMessageSyntax::operator= (
    const CryptographicMessageSyntax & ) [delete]
```

#### 10.64.4.5 ParseCertificateFile()

```
virtual bool gdcM::CryptographicMessageSyntax::ParseCertificateFile (
    const char * filename ) [pure virtual]
```

Implemented in [gdcM::CAPICryptographicMessageSyntax](#), [gdcM::OpenSSLCryptographicMessageSyntax](#), and [gdcM::OpenSSL7CryptographicMessageSyntax](#).

#### Examples

[BasicAnonymizer.cs](#), and [ClinicalTrialIdentificationWorkflow.cs](#).

#### 10.64.4.6 ParseKeyFile()

```
virtual bool gdcM::CryptographicMessageSyntax::ParseKeyFile (
    const char * filename ) [pure virtual]
```

Implemented in [gdcM::CAPICryptographicMessageSyntax](#), [gdcM::OpenSSLCryptographicMessageSyntax](#), and [gdcM::OpenSSL7CryptographicMessageSyntax](#).

#### 10.64.4.7 SetCipherType()

```
virtual void gdcm::CryptographicMessageSyntax::SetCipherType (
    CipherTypes type ) [pure virtual]
```

Implemented in [gdcm::CAPICryptographicMessageSyntax](#), [gdcm::OpenSSLCryptographicMessageSyntax](#), and [gdcm::OpenSSL7CryptographicMessageSyntax](#).

#### 10.64.4.8 SetPassword()

```
virtual bool gdcm::CryptographicMessageSyntax::SetPassword (
    const char * pass,
    size_t passLen ) [pure virtual]
```

Implemented in [gdcm::OpenSSL7CryptographicMessageSyntax](#), [gdcm::CAPICryptographicMessageSyntax](#), and [gdcm::OpenSSLCryptographicMessageSyntax](#).

The documentation for this class was generated from the following file:

- [gdcmCryptographicMessageSyntax.h](#)

## 10.65 gdcm::CSAElement Class Reference

Class to represent a CSA [Element](#).

```
#include <gdcmCSAElement.h>
```

Collaboration diagram for gdcm::CSAElement:



## Public Member Functions

- [CSAElement](#) (const [CSAElement](#) &\_val)
- [CSAElement](#) (unsigned int kf=0)
- const [ByteValue](#) \* [GetByteValue](#) () const
- unsigned int [GetKey](#) () const  
*Set/Get Key.*
- const char \* [GetName](#) () const  
*Set/Get Name.*
- unsigned int [GetNoOfItems](#) () const  
*Set/Get NoOfItems.*
- unsigned int [GetSyngoDT](#) () const  
*Set/Get SyngoDT.*
- [Value](#) & [GetValue](#) ()
- [Value](#) const & [GetValue](#) () const  
*Set/Get Value (bytes array, SQ of items, SQ of fragments):*
- const [VM](#) & [GetVM](#) () const  
*Set/Get VM.*
- [VR](#) const & [GetVR](#) () const  
*Set/Get VR.*
- bool [IsEmpty](#) () const  
*Check if CSA Element is empty.*
- bool [operator<](#) (const [CSAElement](#) &de) const
- [CSAElement](#) & [operator=](#) (const [CSAElement](#) &de)=default
- bool [operator==](#) (const [CSAElement](#) &de) const
- void [SetByteValue](#) (const char \*array, [VL](#) length)  
*Set.*
- void [SetKey](#) (unsigned int key)
- void [SetName](#) (const char \*name)
- void [SetNoOfItems](#) (unsigned int items)
- void [SetSyngoDT](#) (unsigned int syngodt)
- void [SetValue](#) ([Value](#) const &vl)
- void [SetVM](#) (const [VM](#) &vm)
- void [SetVR](#) ([VR](#) const &vr)

## Protected Types

- typedef [SmartPointer](#)< [Value](#) > [DataPtr](#)

## Protected Attributes

- [DataPtr](#) [DataField](#)
- unsigned int [KeyField](#)
- std::string [NameField](#)
- unsigned int [NoOfItemsField](#)
- unsigned int [SyngoDTField](#)
- [VM](#) [ValueMultiplicityField](#)
- [VR](#) [VRField](#)

## Friends

- `std::ostream & operator<< (std::ostream &os, const CSAElement &val)`

## 10.65.1 Detailed Description

Class to represent a CSA [Element](#).

See also

[CSAHeader](#)

Examples

[DumpCSA.cs](#), [DumpSiemensBase64.cxx](#), [MrProtocol.cxx](#), and [csa2img.cxx](#).

## 10.65.2 Member Typedef Documentation

### 10.65.2.1 DataPtr

```
typedef SmartPointer<Value> gdcm::CSAElement::DataPtr [protected]
```

## 10.65.3 Constructor & Destructor Documentation

### 10.65.3.1 CSAElement() [1/2]

```
gdcm::CSAElement::CSAElement (
    unsigned int kf = 0 ) [inline]
```

### 10.65.3.2 CSAElement() [2/2]

```
gdcm::CSAElement::CSAElement (
    const CSAElement & _val ) [inline]
```

## 10.65.4 Member Function Documentation

### 10.65.4.1 GetByteValue()

```
const ByteValue * gdcm::CSAElement::GetByteValue ( ) const [inline]
```

Return the [Value](#) of [CSAElement](#) as a [ByteValue](#) (if possible)

Warning

: You need to check for NULL return value

Examples

[DumpSiemensBase64.cxx](#), and [MrProtocol.cxx](#).

#### 10.65.4.2 GetKey()

```
unsigned int gdcm::CSAElement::GetKey ( ) const [inline]
```

Set/Get Key.

Referenced by [operator<\(\)](#).

#### 10.65.4.3 GetName()

```
const char * gdcm::CSAElement::GetName ( ) const [inline]
```

Set/Get Name.

#### 10.65.4.4 GetNoOfItems()

```
unsigned int gdcm::CSAElement::GetNoOfItems ( ) const [inline]
```

Set/Get NoOfItems.

#### 10.65.4.5 GetSyngoDT()

```
unsigned int gdcm::CSAElement::GetSyngoDT ( ) const [inline]
```

Set/Get SyngoDT.

#### 10.65.4.6 GetValue() [1/2]

```
Value & gdcm::CSAElement::GetValue ( ) [inline]
```

#### 10.65.4.7 GetValue() [2/2]

```
Value const & gdcm::CSAElement::GetValue ( ) const [inline]
```

Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):

#### Examples

[csa2img.cxx](#).

#### 10.65.4.8 GetVM()

```
const VM & gdcm::CSAElement::GetVM ( ) const [inline]
```

Set/Get VM.

#### 10.65.4.9 GetVR()

```
VR const & gdcm::CSAElement::GetVR ( ) const [inline]
```

Set/Get VR.

#### 10.65.4.10 IsEmpty()

```
bool gdcm::CSAElement::IsEmpty ( ) const [inline]
```

Check if CSA Element is empty.

#### Examples

[csa2img.cxx](#).

#### 10.65.4.11 operator<()

```
bool gdcm::CSAElement::operator< (
    const CSAElement & de ) const [inline]
```

References [GetKey\(\)](#).

#### 10.65.4.12 operator=()

```
CSAElement & gdcm::CSAElement::operator= (
    const CSAElement & de ) [default]
```

#### 10.65.4.13 operator==( )

```
bool gdcm::CSAElement::operator== (
    const CSAElement & de ) const [inline]
```

References [KeyField](#), [NameField](#), [SyngoDTField](#), [ValueMultiplicityField](#), and [VRField](#).

**10.65.4.14 SetByteValue()**

```
void gdcm::CSAElement::SetByteValue (
    const char * array,
    VL length ) [inline]
```

Set.

**10.65.4.15 SetKey()**

```
void gdcm::CSAElement::SetKey (
    unsigned int key ) [inline]
```

**10.65.4.16 SetName()**

```
void gdcm::CSAElement::SetName (
    const char * name ) [inline]
```

**10.65.4.17 SetNoOfItems()**

```
void gdcm::CSAElement::SetNoOfItems (
    unsigned int items ) [inline]
```

**10.65.4.18 SetSyngoDT()**

```
void gdcm::CSAElement::SetSyngoDT (
    unsigned int syngodt ) [inline]
```

**10.65.4.19 SetValue()**

```
void gdcm::CSAElement::SetValue (
    Value const & vl ) [inline]
```

**10.65.4.20 SetVM()**

```
void gdcm::CSAElement::SetVM (
    const VM & vm ) [inline]
```

**10.65.4.21 SetVR()**

```
void gdcm::CSAElement::SetVR (
    VR const & vr ) [inline]
```



## 10.65.5 Friends And Related Symbol Documentation

### 10.65.5.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & os,  
    const CSAElement & val ) [friend]
```

## 10.65.6 Member Data Documentation

### 10.65.6.1 DataField

```
DataPtr gdcM::CSAElement::DataField [protected]
```

### 10.65.6.2 KeyField

```
unsigned int gdcM::CSAElement::KeyField [protected]
```

Referenced by [operator==\(\(\)\)](#).

### 10.65.6.3 NameField

```
std::string gdcM::CSAElement::NameField [protected]
```

Referenced by [operator==\(\(\)\)](#).

### 10.65.6.4 NoOfItemsField

```
unsigned int gdcM::CSAElement::NoOfItemsField [protected]
```

### 10.65.6.5 SyngoDTField

```
unsigned int gdcM::CSAElement::SyngoDTField [protected]
```

Referenced by [operator==\(\(\)\)](#).

### 10.65.6.6 ValueMultiplicityField

```
VM gdcM::CSAElement::ValueMultiplicityField [protected]
```

Referenced by [operator==\(\(\)\)](#).

### 10.65.6.7 VRField

`VR gdcmm::CSAElement::VRField [protected]`

Referenced by `operator==()`.

The documentation for this class was generated from the following file:

- `gdcmmCSAElement.h`

## 10.66 gdcmm::CSAHeader Class Reference

Class for `CSAHeader`.

```
#include <gdcmmCSAHeader.h>
```

### Public Types

- enum `CSAHeaderType` {  
`UNKNOWN = 0` ,  
`SV10` ,  
`NOMAGIC` ,  
`DATASET_FORMAT` ,  
`INTERFILE` ,  
`ZEROED_OUT` }

*Diverse format of `CSAHeader` as found 'in the wild'.*

### Public Member Functions

- `CSAHeader ()`
- `~CSAHeader ()=default`
- `bool FindCSAElementByName (const char *name)`
- `const CSAElement & GetCSAElementByName (const char *name)`
- `const DataSet & GetDataSet () const`  
*Return the `DataSet` output (use only if `Format == DATASET_FORMAT`)*
- `CSAHeaderType GetFormat () const`
- `const char * GetInterfile () const`  
*Return the string output (use only if `Format == Interfile`)*
- `bool GetMrProtocol (const DataSet &ds, MrProtocol &mrProtocol)`  
*Retrieve the ASCII portion stored within the MrProtocol/MrPhoenixProtocol:*
- `bool LoadFromDataElement (DataElement const &de)`  
*Decode the `CSAHeader` from element 'de'.*
- `void Print (std::ostream &os) const`  
*Print the `CSAHeader` (use only if `Format == SV10` or `NOMAGIC`)*

### Static Public Member Functions

- static const [PrivateTag](#) & [GetCSADataInfo](#) ()
- static const [PrivateTag](#) & [GetCSAImageHeaderInfoTag](#) ()
- static const [PrivateTag](#) & [GetCSASeriesHeaderInfoTag](#) ()

### Protected Member Functions

- const [CSAElement](#) & [GetCSAEEnd](#) () const

### Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [CSAHeader](#) &d)

## 10.66.1 Detailed Description

Class for [CSAHeader](#).

SIEMENS store private information in tag (0x0029,0x10,"SIEMENS CSA HEADER") this class is meant for user wishing to access values stored within this private attribute. There are basically two main 'format' for this attribute : SV10/↔ NOMAGIC and DATASET\_FORMAT SV10 and NOMAGIC are from a user prospective identical, see CSAHeader.xml for possible name / value stored in this format. DATASET\_FORMAT is in fact simply just another DICOM dataset (implicit) with -currently unknown- value. This can be only be printed for now.

#### Warning

Everything you do with this code is at your own risk, since decoding process was not written from specification documents.

the API of this class might change.

**Todo** MrEvaProtocol in 29,1020 contains ^M that would be nice to get rid of on UNIX system...

#### See also

[PDBHeader](#)

External references: 5.1.3.2.4.1 MEDCOM History Information and 5.1.4.3 CSA Non-Image [Module](#) in [http://tamsinfo.toshiba.com/docrequest/pdf/E.Soft\\_v2.0.pdf](http://tamsinfo.toshiba.com/docrequest/pdf/E.Soft_v2.0.pdf)

#### Examples

[DumpCSA.cs](#), [DumpSiemensBase64.cxx](#), [MrProtocol.cxx](#), and [csa2img.cxx](#).

## 10.66.2 Member Enumeration Documentation

### 10.66.2.1 CSAHeaderType

```
enum gdcm::CSAHeader::CSAHeaderType
```

Diverse format of [CSAHeader](#) as found 'in the wild'.

## Enumerator

UNKNOWN	
SV10	
NOMAGIC	
DATASET_FORMAT	
INTERFILE	
ZEROED_OUT	

### 10.66.3 Constructor & Destructor Documentation

#### 10.66.3.1 CSAHeader()

```
gdcm::CSAHeader::CSAHeader ( ) [inline]
```

#### 10.66.3.2 ~CSAHeader()

```
gdcm::CSAHeader::~~CSAHeader ( ) [default]
```

### 10.66.4 Member Function Documentation

#### 10.66.4.1 FindCSAElementByName()

```
bool gdcm::CSAHeader::FindCSAElementByName (
    const char * name )
```

Return true if the CSA element matching 'name' is found or not

#### Warning

Case Sensitive

#### Examples

[DumpCSA.cs](#), [DumpSiemensBase64.cxx](#), [MrProtocol.cxx](#), and [csa2img.cxx](#).

#### 10.66.4.2 GetCSADataInfo()

```
static const PrivateTag & gdcm::CSAHeader::GetCSADataInfo ( ) [static]
```

Return the private tag used by SIEMENS to store the CSA Data Info This is: [PrivateTag](#)(0x0029,0x10,"SIEMENS CSA NON-IMAGE");

### 10.66.4.3 GetCSAEEnd()

```
const CSAElement & gdcm::CSAHeader::GetCSAEEnd ( ) const [protected]
```

### 10.66.4.4 GetCSAElementByName()

```
const CSAElement & gdcm::CSAHeader::GetCSAElementByName (
    const char * name )
```

Return the [CSAElement](#) corresponding to name 'name'

#### Warning

Case Sensitive

#### Examples

[DumpCSA.cs](#), [DumpSiemensBase64.cxx](#), [MrProtocol.cxx](#), and [csa2img.cxx](#).

### 10.66.4.5 GetCSAImageHeaderInfoTag()

```
static const PrivateTag & gdcm::CSAHeader::GetCSAImageHeaderInfoTag ( ) [static]
```

Return the private tag used by SIEMENS to store the CSA [Image](#) Header This is: [PrivateTag](#)(0x0029,0x10,"SIEMENS CSA HEADER");

#### Examples

[DumpCSA.cs](#), [DumpSiemensBase64.cxx](#), [PublicDict.cxx](#), and [csa2img.cxx](#).

### 10.66.4.6 GetCSASeriesHeaderInfoTag()

```
static const PrivateTag & gdcm::CSAHeader::GetCSASeriesHeaderInfoTag ( ) [static]
```

Return the private tag used by SIEMENS to store the CSA [Series](#) Header This is: [PrivateTag](#)(0x0029,0x20,"SIEMENS CSA HEADER");

#### Examples

[MrProtocol.cxx](#).

### 10.66.4.7 GetDataSet()

```
const DataSet & gdcm::CSAHeader::GetDataSet ( ) const [inline]
```

Return the [DataSet](#) output (use only if Format == DATASET\_FORMAT )

#### 10.66.4.8 GetFormat()

```
CSAHeaderType gdcm::CSAHeader::GetFormat ( ) const
```

return the format of the [CSAHeader](#) SV10 and NOMAGIC are equivalent.

#### 10.66.4.9 GetInterfile()

```
const char * gdcm::CSAHeader::GetInterfile ( ) const [inline]
```

Return the string output (use only if Format == Interfile)

#### 10.66.4.10 GetMrProtocol()

```
bool gdcm::CSAHeader::GetMrProtocol (
    const DataSet & ds,
    MrProtocol & mrProtocol )
```

Retrieve the ASCII portion stored within the MrProtocol/MrPhoenixProtocol:

##### Examples

[MrProtocol.cxx](#).

#### 10.66.4.11 LoadFromDataElement()

```
bool gdcm::CSAHeader::LoadFromDataElement (
    DataElement const & de )
```

Decode the [CSAHeader](#) from element 'de'.

##### Examples

[DumpCSA.cs](#), [DumpSiemensBase64.cxx](#), [MrProtocol.cxx](#), and [csa2img.cxx](#).

#### 10.66.4.12 Print()

```
void gdcm::CSAHeader::Print (
    std::ostream & os ) const
```

Print the [CSAHeader](#) (use only if Format == SV10 or NOMAGIC)

##### Examples

[csa2img.cxx](#).

## 10.66.5 Friends And Related Symbol Documentation

### 10.66.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const CSAHeader & d ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmCSAHeader.h](#)

## 10.67 gdcm::CSAHeaderDict Class Reference

Class to represent a map of [CSAHeaderDictEntry](#).

```
#include <gdcmCSAHeaderDict.h>
```

### Public Types

- typedef MapCSAHeaderDictEntry::const\_iterator [ConstIterator](#)
- typedef MapCSAHeaderDictEntry::iterator [Iterator](#)
- typedef std::set< [CSAHeaderDictEntry](#) > [MapCSAHeaderDictEntry](#)

### Public Member Functions

- [CSAHeaderDict](#) ()
- [CSAHeaderDict](#) (const [CSAHeaderDict](#) &\_val)=delete
- void [AddCSAHeaderDictEntry](#) (const [CSAHeaderDictEntry](#) &de)
- [ConstIterator](#) [Begin](#) () const
- [ConstIterator](#) [End](#) () const
- const [CSAHeaderDictEntry](#) & [GetCSAHeaderDictEntry](#) (const char \*name) const
- bool [IsEmpty](#) () const
- [CSAHeaderDict](#) & [operator=](#) (const [CSAHeaderDict](#) &\_val)=delete

### Protected Member Functions

- void [LoadDefault](#) ()

### Friends

- class [Dicts](#)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [CSAHeaderDict](#) &\_val)

### 10.67.1 Detailed Description

Class to represent a map of [CSAHeaderDictEntry](#).

#### Examples

[MrProtocol.cxx](#).

### 10.67.2 Member Typedef Documentation

#### 10.67.2.1 ConstIterator

```
typedef MapCSAHeaderDictEntry::const_iterator gdcM::CSAHeaderDict::ConstIterator
```

#### 10.67.2.2 Iterator

```
typedef MapCSAHeaderDictEntry::iterator gdcM::CSAHeaderDict::Iterator
```

#### 10.67.2.3 MapCSAHeaderDictEntry

```
typedef std::set<CSAHeaderDictEntry> gdcM::CSAHeaderDict::MapCSAHeaderDictEntry
```

### 10.67.3 Constructor & Destructor Documentation

#### 10.67.3.1 CSAHeaderDict() [1/2]

```
gdcM::CSAHeaderDict::CSAHeaderDict ( ) [inline]
```

#### 10.67.3.2 CSAHeaderDict() [2/2]

```
gdcM::CSAHeaderDict::CSAHeaderDict (
    const CSAHeaderDict & _val ) [delete]
```

References [gdcM::operator<<\(\)](#).

### 10.67.4 Member Function Documentation

#### 10.67.4.1 AddCSAHeaderDictEntry()

```
void gdcM::CSAHeaderDict::AddCSAHeaderDictEntry (
    const CSAHeaderDictEntry & de ) [inline]
```



#### 10.67.4.2 Begin()

```
ConstIterator gdcm::CSAHeaderDict::Begin ( ) const [inline]
```

#### 10.67.4.3 End()

```
ConstIterator gdcm::CSAHeaderDict::End ( ) const [inline]
```

#### 10.67.4.4 GetCSAHeaderDictEntry()

```
const CSAHeaderDictEntry & gdcm::CSAHeaderDict::GetCSAHeaderDictEntry (
    const char * name ) const [inline]
```

#### Examples

[MrProtocol.cxx](#).

#### 10.67.4.5 IsEmpty()

```
bool gdcm::CSAHeaderDict::IsEmpty ( ) const [inline]
```

#### 10.67.4.6 LoadDefault()

```
void gdcm::CSAHeaderDict::LoadDefault ( ) [protected]
```

#### 10.67.4.7 operator=()

```
CSAHeaderDict & gdcm::CSAHeaderDict::operator= (
    const CSAHeaderDict & _val ) [delete]
```

### 10.67.5 Friends And Related Symbol Documentation

#### 10.67.5.1 Dicts

```
friend class Dicts [friend]
```

### 10.67.5.2 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const CSAHeaderDict & _val ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmCSAHeaderDict.h](#)

## 10.68 gdcm::CSAHeaderDictEntry Class Reference

Class to represent an Entry in the [Dict](#).

```
#include <gdcmCSAHeaderDictEntry.h>
```

### Public Member Functions

- [CSAHeaderDictEntry](#) (const char \*name="", [VR](#) const &vr=[VR::INVALID](#), [VM](#) const &vm=[VM::VM0](#), const char \*desc="")
- const char \* [GetDescription](#) () const  
*Set/Get Description.*
- const char \* [GetName](#) () const  
*Set/Get Name.*
- const [VM](#) & [GetVM](#) () const  
*Set/Get VM.*
- const [VR](#) & [GetVR](#) () const  
*Set/Get VR.*
- bool [operator<](#) (const [CSAHeaderDictEntry](#) &entry) const
- void [SetDescription](#) (const char \*desc)
- void [SetName](#) (const char \*name)
- void [SetVM](#) ([VM](#) const &vm)
- void [SetVR](#) (const [VR](#) &vr)

### Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [CSAHeaderDictEntry](#) &\_val)

## 10.68.1 Detailed Description

Class to represent an Entry in the [Dict](#).

Does not really exist within the DICOM definition, just a way to minimize storage and have a mapping from [gdcm::Tag](#) to the needed information

### Note

bla TODO FIXME: Need a PublicCSAHeaderDictEntry...indeed [CSAHeaderDictEntry](#) has a notion of retired which does not exist in PrivateCSAHeaderDictEntry...

### See also

[gdcm::Dict](#)

### Examples

[MrProtocol.cxx](#).

## 10.68.2 Constructor & Destructor Documentation

### 10.68.2.1 CSAHeaderDictEntry()

```
gdcm::CSAHeaderDictEntry::CSAHeaderDictEntry (
    const char * name = "",
    VR const & vr = VR::INVALID,
    VM const & vm = VM::VMO,
    const char * desc = "" ) [inline]
```

## 10.68.3 Member Function Documentation

### 10.68.3.1 GetDescription()

```
const char * gdcm::CSAHeaderDictEntry::GetDescription ( ) const [inline]
```

Set/Get Description.

### 10.68.3.2 GetName()

```
const char * gdcm::CSAHeaderDictEntry::GetName ( ) const [inline]
```

Set/Get Name.

Referenced by [operator<\(\)](#).

### 10.68.3.3 GetVM()

```
const VM & gdcm::CSAHeaderDictEntry::GetVM ( ) const [inline]
```

Set/Get [VM](#).

### 10.68.3.4 GetVR()

```
const VR & gdcm::CSAHeaderDictEntry::GetVR ( ) const [inline]
```

Set/Get [VR](#).

### 10.68.3.5 operator<()

```
bool gdcm::CSAHeaderDictEntry::operator< (
    const CSAHeaderDictEntry & entry ) const [inline]
```

References [GetName\(\)](#).

### 10.68.3.6 SetDescription()

```
void gdcm::CSAHeaderDictEntry::SetDescription (
    const char * desc ) [inline]
```

### 10.68.3.7 SetName()

```
void gdcm::CSAHeaderDictEntry::SetName (
    const char * name ) [inline]
```

### 10.68.3.8 SetVM()

```
void gdcm::CSAHeaderDictEntry::SetVM (
    VM const & vm ) [inline]
```

### 10.68.3.9 SetVR()

```
void gdcm::CSAHeaderDictEntry::SetVR (
    const VR & vr ) [inline]
```

## 10.68.4 Friends And Related Symbol Documentation

### 10.68.4.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & _os,  
    const CSAHeaderDictEntry & _val ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmCSAHeaderDictEntry.h](#)

## 10.69 gdcm::CSAHeaderDictException Class Reference

```
#include <gdcmCSAHeaderDict.h>
```

Inheritance diagram for gdcm::CSAHeaderDictException:



Collaboration diagram for gdcm::CSAHeaderDictException:



The documentation for this class was generated from the following file:

- [gdcmCSAHeaderDict.h](#)

## 10.70 gdcm::network::CStoreRQ Class Reference

[CStoreRQ](#).

```
#include <gdcmCStoreMessages.h>
```

Inheritance diagram for gdcm::network::CStoreRQ:



Collaboration diagram for gdcm::network::CStoreRQ:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [File](#) &file, bool writeDataSet=true)

### Public Member Functions inherited from [gdcm::network::BaseCompositeMessage](#)

- virtual `~BaseCompositeMessage` ()=default

## 10.70.1 Detailed Description

[CStoreRQ](#).

this file defines the messages for the cecho action

## 10.70.2 Member Function Documentation

### 10.70.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::CStoreRQ::ConstructPDV (
    const ULConnection & inConnection,
    const File & file,
    bool writeDataSet = true )
```

The documentation for this class was generated from the following file:

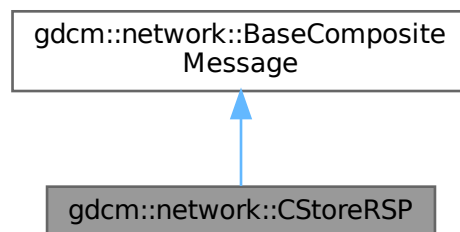
- [gdcmCStoreMessages.h](#)

## 10.71 gdcm::network::CStoreRSP Class Reference

[CStoreRSP](#) this file defines the messages for the cecho action.

```
#include <gdcmCStoreMessages.h>
```

Inheritance diagram for gdcm::network::CStoreRSP:



Collaboration diagram for `gdcm::network::CStoreRSP`:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [DataSet](#) \*inDataSet, const [BasePDU](#) \*inPC)

### Public Member Functions inherited from [gdcm::network::BaseCompositeMessage](#)

- virtual `~BaseCompositeMessage` ()=default

## 10.71.1 Detailed Description

[CStoreRSP](#) this file defines the messages for the cecho action.

## 10.71.2 Member Function Documentation

### 10.71.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::CStoreRSP::ConstructPDV (
    const DataSet * inDataSet,
    const BasePDU * inPC )
```

The documentation for this class was generated from the following file:

- [gdcmCStoreMessages.h](#)



## 10.72 gdcm::Curve Class Reference

[Curve](#) class to handle element 50xx,3000 [Curve](#) Data.

```
#include <gdcmCurve.h>
```

Inheritance diagram for gdcm::Curve:



Collaboration diagram for gdcm::Curve:



### Public Member Functions

- [Curve](#) ()
- [Curve](#) ([Curve](#) const &ov)
- [~Curve](#) () override
- void [Decode](#) (std::istream &is, std::ostream &os)
- void [GetAsPoints](#) (float \*array) const
- std::vector< unsigned short > const & [GetCurveDataDescriptor](#) () const
- unsigned short [GetDataValueRepresentation](#) () const
- unsigned short [GetDimensions](#) () const
- unsigned short [GetGroup](#) () const

- unsigned short [GetNumberOfPoints](#) () const
- const char \* [GetTypeOfData](#) () const
- const char \* [GetTypeOfDataDescription](#) () const
- bool [IsEmpty](#) () const
- void [Print](#) (std::ostream &) const override
- void [SetCoordinateStartValue](#) (unsigned short v)
- void [SetCoordinateStepValue](#) (unsigned short v)
- void [SetCurve](#) (const char \*array, unsigned int length)
- void [SetCurveDataDescriptor](#) (const uint16\_t \*values, size\_t num)
- void [SetCurveDescription](#) (const char \*curvedescription)
- void [SetDataValueRepresentation](#) (unsigned short datavaluerepresentation)
- void [SetDimensions](#) (unsigned short dimensions)
- void [SetGroup](#) (unsigned short group)
- void [SetNumberOfPoints](#) (unsigned short numberofpoints)
- void [SetTypeOfData](#) (const char \*typeofdata)
- void [Update](#) (const [DataElement](#) &de)

### Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

### Static Public Member Functions

- static unsigned int [GetNumberOfCurves](#) ([DataSet](#) const &ds)

### Additional Inherited Members

### Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

## 10.72.1 Detailed Description

[Curve](#) class to handle element 50xx,3000 [Curve](#) Data.

WARNING: This is deprecated and lastly defined in PS 3.3 - 2004

Examples:

- GE\_DLX-8-MONO2-Multiframe-Jpeg\_Lossless.dcm
- GE\_DLX-8-MONO2-Multiframe.dcm
- gdcmSampleData/Philips\_Medical\_Images/integris\_HV\_5000/xa\_integris.dcm
- TOSHIBA-CurveData[1-3].dcm

## 10.72.2 Constructor & Destructor Documentation

### 10.72.2.1 Curve() [1/2]

```
gdcm::Curve::Curve ( )
```

### 10.72.2.2 ~Curve()

```
gdcm::Curve::~~Curve ( ) [override]
```

### 10.72.2.3 Curve() [2/2]

```
gdcm::Curve::Curve (
    Curve const & ov )
```

## 10.72.3 Member Function Documentation

### 10.72.3.1 Decode()

```
void gdcm::Curve::Decode (
    std::istream & is,
    std::ostream & os )
```

### 10.72.3.2 GetAsPoints()

```
void gdcm::Curve::GetAsPoints (
    float * array ) const
```

### 10.72.3.3 GetCurveDataDescriptor()

```
std::vector< unsigned short > const & gdcm::Curve::GetCurveDataDescriptor ( ) const
```

### 10.72.3.4 GetDataValueRepresentation()

```
unsigned short gdcm::Curve::GetDataValueRepresentation ( ) const
```

### 10.72.3.5 GetDimensions()

```
unsigned short gdcm::Curve::GetDimensions ( ) const
```

#### 10.72.3.6 GetGroup()

```
unsigned short gdcm::Curve::GetGroup ( ) const
```

#### 10.72.3.7 GetNumberOfCurves()

```
static unsigned int gdcm::Curve::GetNumberOfCurves (
    DataSet const & ds ) [static]
```

#### 10.72.3.8 GetNumberOfPoints()

```
unsigned short gdcm::Curve::GetNumberOfPoints ( ) const
```

#### 10.72.3.9 GetTypeInfoData()

```
const char * gdcm::Curve::GetTypeInfoData ( ) const
```

#### 10.72.3.10 GetTypeInfoDataDescription()

```
const char * gdcm::Curve::GetTypeInfoDataDescription ( ) const
```

#### 10.72.3.11 IsEmpty()

```
bool gdcm::Curve::IsEmpty ( ) const
```

#### 10.72.3.12 Print()

```
void gdcm::Curve::Print (
    std::ostream & ) const [override], [virtual]
```

Reimplemented from [gdcm::Object](#).

#### 10.72.3.13 SetCoordinateStartValue()

```
void gdcm::Curve::SetCoordinateStartValue (
    unsigned short v )
```

#### 10.72.3.14 SetCoordinateStepValue()

```
void gdcm::Curve::SetCoordinateStepValue (
    unsigned short v )
```

**10.72.3.15 SetCurve()**

```
void gdcm::Curve::SetCurve (
    const char * array,
    unsigned int length )
```

**10.72.3.16 SetCurveDataDescriptor()**

```
void gdcm::Curve::SetCurveDataDescriptor (
    const uint16_t * values,
    size_t num )
```

**10.72.3.17 SetCurveDescription()**

```
void gdcm::Curve::SetCurveDescription (
    const char * curvedescription )
```

**10.72.3.18 SetDataValueRepresentation()**

```
void gdcm::Curve::SetDataValueRepresentation (
    unsigned short datavaluerepresentation )
```

**10.72.3.19 SetDimensions()**

```
void gdcm::Curve::SetDimensions (
    unsigned short dimensions )
```

**10.72.3.20 SetGroup()**

```
void gdcm::Curve::SetGroup (
    unsigned short group )
```

**10.72.3.21 SetNumberOfPoints()**

```
void gdcm::Curve::SetNumberOfPoints (
    unsigned short numberofpoints )
```

**10.72.3.22 SetTypeOfData()**

```
void gdcm::Curve::SetTypeOfData (
    const char * typeofdata )
```

### 10.72.3.23 Update()

```
void gdcM::Curve::Update (
    const DataElement & de )
```

The documentation for this class was generated from the following file:

- [gdcMCurve.h](#)

## 10.73 gdcM::DataElement Class Reference

Class to represent a Data [Element](#) either Implicit or Explicit.

```
#include <gdcMDataElement.h>
```

Inheritance diagram for gdcM::DataElement:



Collaboration diagram for gdcmm::DataElement:



## Public Member Functions

- `DataElement` (const `DataElement` &\_val)
- `DataElement` (const `Tag` &t=`Tag`(0), const `VL` &vl=0, const `VR` &vr=`VR::INVALID`)
- void `Clear` ()
  - Clear Data Element (make Value empty and invalidate Tag & VR)*
- void `Empty` ()
  - Make Data Element empty (no Value)*
- const `ByteValue` \* `GetByteValue` () const
- template<typename TDE >
  - `VL` `GetLength` () const
- `SequenceOfFragments` \* `GetSequenceOfFragments` ()
- const `SequenceOfFragments` \* `GetSequenceOfFragments` () const
- `Tag` & `GetTag` ()
- const `Tag` & `GetTag` () const
  - Get Tag.*
- `Value` & `GetValue` ()
- `Value` const & `GetValue` () const
  - Set/Get Value (bytes array, SQ of items, SQ of fragments):*
- `SmartPointer`< `SequenceOfItems` > `GetValueAsSQ` () const
- `VL` & `GetVL` ()
- const `VL` & `GetVL` () const
  - Get VL.*
- `VR` const & `GetVR` () const
- bool `IsEmpty` () const
  - Check if Data Element is empty.*
- bool `IsUndefinedLength` () const
  - return if Value Length if of undefined length*
- bool `operator<` (const `DataElement` &de) const
- `DataElement` & `operator=` (const `DataElement` &)=default

- bool `operator==` (const `DataElement` &de) const
- template<typename TDE , typename TSwap >  
std::istream & `Read` (std::istream &is)
- template<typename TDE , typename TSwap >  
std::istream & `ReadOrSkip` (std::istream &is, std::set< `Tag` > const &skiptags)
- template<typename TDE , typename TSwap >  
std::istream & `ReadPreValue` (std::istream &is, std::set< `Tag` > const &skiptags)
- template<typename TDE , typename TSwap >  
std::istream & `ReadValue` (std::istream &is, std::set< `Tag` > const &skiptags)
- template<typename TDE , typename TSwap >  
std::istream & `ReadValueWithLength` (std::istream &is, `VL` &length, std::set< `Tag` > const &skiptags)
- template<typename TDE , typename TSwap >  
std::istream & `ReadWithLength` (std::istream &is, `VL` &length)
- void `SetByteValue` (const char \*array, `VL` length)
- void `SetTag` (const `Tag` &t)
- void `SetValue` (`Value` const &vl)
- void `SetVL` (const `VL` &vl)
- void `SetVLToUndefined` ()
- void `SetVR` (`VR` const &vr)
- template<typename TDE , typename TSwap >  
const std::ostream & `Write` (std::ostream &os) const

### Protected Types

- typedef `SmartPointer`< `Value` > `ValuePtr`

### Protected Member Functions

- void `SetValueFieldLength` (`VL` vl, bool readvalues)

### Protected Attributes

- `Tag` TagField
- `ValuePtr` ValueField
- `VL` ValueLengthField
- `VR` VRField

### Friends

- std::ostream & `operator<<` (std::ostream &\_os, const `DataElement` &\_val)



### 10.73.1 Detailed Description

Class to represent a Data [Element](#) either Implicit or Explicit.

DATA ELEMENT: A unit of information as defined by a single entry in the data dictionary. An encoded Information [Object](#) Definition (IOD) [Attribute](#) that is composed of, at a minimum, three fields: a Data [Element Tag](#), a [Value](#) Length, and a [Value](#) Field. For some specific Transfer Syntaxes, a Data [Element](#) also contains a [VR](#) Field where the [Value](#) Representation of that Data [Element](#) is specified explicitly.

Design:

- A [DataElement](#) in GDCM always store [VL](#) ([Value](#) Length) on a 32 bits integer even when [VL](#) is 16 bits
- A [DataElement](#) always store the [VR](#) even for Implicit TS, in which case [VR](#) is defaulted to [VR::INVALID](#)
- For [Item](#) start/end (See 0xfffe tags), [Value](#) is NULL

See also

[ExplicitDataElement](#) [ImplicitDataElement](#)

Examples

[BasicImageAnonymizer.cs](#), [ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImage.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenAIIVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [MpegVideoInfo.cs](#), [NewSequence.cs](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [SimplePrint.cs](#), [StreamImageReaderTest.cxx](#), [csa2img.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

### 10.73.2 Member Typedef Documentation

#### 10.73.2.1 ValuePtr

```
typedef SmartPointer<Value> gdcm::DataElement::ValuePtr [protected]
```

### 10.73.3 Constructor & Destructor Documentation

#### 10.73.3.1 DataElement() [1/2]

```
gdcm::DataElement::DataElement (
    const Tag & t = Tag(0),
    const VL & vl = 0,
    const VR & vr = VR::INVALID ) [inline]
```

### 10.73.3.2 DataElement() [2/2]

```
gdcmm::DataElement::DataElement (
    const DataElement & _val ) [inline]
```

## 10.73.4 Member Function Documentation

### 10.73.4.1 Clear()

```
void gdcmm::DataElement::Clear ( ) [inline]
```

Clear Data [Element](#) (make [Value](#) empty and invalidate [Tag](#) & [VR](#))

### 10.73.4.2 Empty()

```
void gdcmm::DataElement::Empty ( ) [inline]
```

Make Data [Element](#) empty (no [Value](#))

### 10.73.4.3 GetByteValue()

```
const ByteValue * gdcmm::DataElement::GetByteValue ( ) const [inline]
```

Return the [Value](#) of [DataElement](#) as a [ByteValue](#) (if possible)

#### Warning

: You need to check for NULL return value

#### Examples

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAI BugJPEGLS.cxx](#), [GetSubSequenceData.cxx](#), [PatchFile.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

Referenced by [gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement\(\)](#), [gdcmm::Attribute< Group, Element, TVR, VM::VM1\\_n >::SetFromDataElement\(\)](#), [gdcmm::Element< TVR, TVM >::SetFromDataElement\(\)](#), and [gdcmm::Element< TVR, VM::VM1\\_n >::SetFromDataElement\(\)](#).

### 10.73.4.4 GetLength()

```
template<typename TDE >
VL gdcmm::DataElement::GetLength ( ) const [inline]
```

**10.73.4.5 GetSequenceOfFragments() [1/2]**

```
SequenceOfFragments * gdcm::DataElement::GetSequenceOfFragments ( )
```

**10.73.4.6 GetSequenceOfFragments() [2/2]**

```
const SequenceOfFragments * gdcm::DataElement::GetSequenceOfFragments ( ) const
```

Return the [Value](#) of [DataElement](#) as a Sequence Of Fragments (if possible)

**Warning**

: You need to check for NULL return value

**Examples**

[DecompressImage.cs](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), and [GetJPEGSamplePrecision.cxx](#).

**10.73.4.7 GetTag() [1/2]**

```
Tag & gdcm::DataElement::GetTag ( ) [inline]
```

**10.73.4.8 GetTag() [2/2]**

```
const Tag & gdcm::DataElement::GetTag ( ) const [inline]
```

Get [Tag](#).

**Examples**

[DumpGEMSMovieGroup.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [SimplePrint.cs](#), and [pmsct\\_rgb1.cxx](#).

Referenced by [gdcm::DataSet::Insert\(\)](#), [gdcm::FileMetaInformation::Insert\(\)](#), [gdcm::CommandDataSet::Insert\(\)](#), [operator<\(\)](#), [gdcm::SequenceOfItems::Read\(\)](#), [gdcm::SequenceOfFragments::ReadValue\(\)](#), [gdcm::FileMetaInformation::Replace\(\)](#), [gdcm::CommandDataSet::Replace\(\)](#), [gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement\(\)](#), and [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >::SetFromDataElement\(\)](#).

**10.73.4.9 GetValue() [1/2]**

```
Value & gdcm::DataElement::GetValue ( ) [inline]
```

References [gdcmAssertAlwaysMacro](#).

**10.73.4.10 GetValue() [2/2]**

```
Value const & gdcm::DataElement::GetValue ( ) const [inline]
```

Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):

**Examples**

[ReadAndDumpDICOMDIR.cxx](#).

References [gdcmAssertAlwaysMacro](#).

Referenced by [gdcm::DataSet::InsertDataElement\(\)](#), [gdcm::Element< TVR, TVM >::SetFromDataElement\(\)](#), and [gdcm::Element< TVR, VM::VM1\\_n >::SetFromDataElement\(\)](#).

**10.73.4.11 GetValueAsSQ()**

```
SmartPointer< SequenceOfItems > gdcm::DataElement::GetValueAsSQ ( ) const
```

Interpret the [Value](#) stored in the [DataElement](#). This is more robust (but also more expensive) to call this function rather than the simplest form: [GetSequenceOfItems\(\)](#) It also return NULL when the [Value](#) is NOT of type [SequenceOfItems](#)

**Warning**

in case [GetSequenceOfItems\(\)](#) succeed the function return this value, otherwise it creates a new [SequenceOfItems](#), you should handle that in your case, for instance: `SmartPointer<SequenceOfItems> sqi = de.GetValueAsSQ();`

**Examples**

[ChangeSequenceUltrasound.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [ExtractEncryptedContent.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [SimplePrint.cs](#), [gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

**10.73.4.12 GetVL() [1/2]**

```
VL & gdcm::DataElement::GetVL ( ) [inline]
```

**10.73.4.13 GetVL() [2/2]**

```
const VL & gdcm::DataElement::GetVL ( ) const [inline]
```

Get [VL](#).

**Examples**

[SimplePrint.cs](#).

Referenced by [gdcm::DataSet::InsertDataElement\(\)](#), [gdcm::SequenceOfItems::Read\(\)](#), and [gdcm::SequenceOfFragments::ReadValue\(\)](#).

#### 10.73.4.14 GetVR()

```
VR const & gdcm::DataElement::GetVR ( ) const [inline]
```

Get [VR](#) do not set [VR::SQ](#) on bytevalue data element

##### Examples

[DuplicatePCDE.cxx](#), and [GenFakeIdentifyFile.cxx](#).

Referenced by [gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >::GetAsDataElement\(\)](#), [gdcm::Element< TVR, TVM >::GetAsDataElement\(\)](#), [gdcm::Element< TVR, VM::VM1\\_n >::GetAsDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >::SetFromDataElement\(\)](#), [gdcm::Element< TVR, TVM >::SetFromDataElement\(\)](#), and [gdcm::Element< TVR, VM::VM1\\_n >::SetFromDataElement\(\)](#).

#### 10.73.4.15 IsEmpty()

```
bool gdcm::DataElement::IsEmpty ( ) const [inline]
```

Check if Data [Element](#) is empty.

##### Examples

[DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [ELSCINT1WaveToText.cxx](#), [FixJAIBugJPEGLS.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

Referenced by [gdcm::DataSet::InsertDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement\(\)](#), and [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >::SetFromDataElement\(\)](#).

#### 10.73.4.16 IsUndefinedLength()

```
bool gdcm::DataElement::IsUndefinedLength ( ) const [inline]
```

return if [Value](#) Length if of undefined length

#### 10.73.4.17 operator<()

```
bool gdcm::DataElement::operator< (
    const DataElement & de ) const [inline]
```

References [GetTag\(\)](#).

#### 10.73.4.18 operator=()

```
DataElement & gdcM::DataElement::operator= (
    const DataElement & ) [default]
```

#### 10.73.4.19 operator==(

```
bool gdcM::DataElement::operator== (
    const DataElement & de ) const [inline]
```

References [TagField](#), [ValueField](#), [ValueLengthField](#), and [VRField](#).

#### 10.73.4.20 Read()

```
template<typename TDE , typename TSwap >
std::istream & gdcM::DataElement::Read (
    std::istream & is ) [inline]
```

#### Examples

[DumpSiemensBase64.cxx](#).

#### 10.73.4.21 ReadOrSkip()

```
template<typename TDE , typename TSwap >
std::istream & gdcM::DataElement::ReadOrSkip (
    std::istream & is,
    std::set< Tag > const & skiptags ) [inline]
```

#### 10.73.4.22 ReadPreValue()

```
template<typename TDE , typename TSwap >
std::istream & gdcM::DataElement::ReadPreValue (
    std::istream & is,
    std::set< Tag > const & skiptags ) [inline]
```

#### 10.73.4.23 ReadValue()

```
template<typename TDE , typename TSwap >
std::istream & gdcM::DataElement::ReadValue (
    std::istream & is,
    std::set< Tag > const & skiptags ) [inline]
```

#### 10.73.4.24 ReadValueWithLength()

```
template<typename TDE , typename TSwap >
std::istream & gdcm::DataElement::ReadValueWithLength (
    std::istream & is,
    VL & length,
    std::set< Tag > const & skiptags ) [inline]
```

#### 10.73.4.25 ReadWithLength()

```
template<typename TDE , typename TSwap >
std::istream & gdcm::DataElement::ReadWithLength (
    std::istream & is,
    VL & length ) [inline]
```

#### 10.73.4.26 SetByteValue()

```
void gdcm::DataElement::SetByteValue (
    const char * array,
    VL length ) [inline]
```

Set the byte value

#### Warning

user need to read DICOM standard for an understanding of:

- even padding
- \0 vs space padding By default even padding is achieved using \0 regardless of the of [VR](#)

#### Examples

[BasicImageAnonymizer.cs](#), [ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImage.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [FixBrokenJ2K.cxx](#), [FixJABugJPEGLS.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSubSequenceData.cxx](#), [MpegVideoInfo.cs](#), [NewSequence.cs](#), [StreamImageReaderTest.cxx](#), [iU22tomultisc.cxx](#), and [rle2img.cxx](#).

Referenced by [gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >::GetAsDataElement\(\)](#), [gdcm::Element< TVR, TVM >::GetAsDataElement\(\)](#), and [gdcm::Element< TVR, VM::VM1\\_n >::GetAsDataElement\(\)](#).

#### 10.73.4.27 SetTag()

```
void gdcM::DataElement::SetTag (
    const Tag & t ) [inline]
```

Set [Tag](#) Use with cautious (need to match Part 6)

##### Examples

[Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GenFakeIdentifyFile.cxx](#), and [GetSubSequenceData.cxx](#).

#### 10.73.4.28 SetValue()

```
void gdcM::DataElement::SetValue (
    Value const & vl ) [inline]
```

##### Warning

you need to set the ValueLengthField explicitly

##### Examples

[DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [DuplicatePCDE.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [MpegVideoInfo.cs](#), and [NewSequence.cs](#).

References [gdcM::Value::GetLength\(\)](#).

#### 10.73.4.29 SetValueFieldLength()

```
void gdcM::DataElement::SetValueFieldLength (
    VL vl,
    bool readvalues ) [protected]
```

#### 10.73.4.30 SetVL()

```
void gdcM::DataElement::SetVL (
    const VL & vl ) [inline]
```

Set [VL](#) Use with cautious (need to match Part 6), advanced user only

##### See also

[SetByteValue](#)



### 10.73.4.31 SetVLToUndefined()

```
void gdcm::DataElement::SetVLToUndefined ( )
```

#### Examples

[Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), and [NewSequence.cs](#).

### 10.73.4.32 SetVR()

```
void gdcm::DataElement::SetVR (
    VR const & vr ) [inline]
```

Set [VR](#) Use with cautious (need to match Part 6), advanced user only

#### Precondition

vr is a [VR::VRALL](#) (not a dual one such as OB\_OW)

#### Examples

[Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSubSequenceData.cxx](#), [NewSequence.cs](#), [StreamImageReaderTest.cxx](#), [iU22tomultisc.cxx](#), and [rle2img.cxx](#).

References [gdcm::VR::IsVRFile\(\)](#).

Referenced by [gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >::GetAsDataElement\(\)](#), [gdcm::Element< TVR, TVM >::GetAsDataElement\(\)](#), and [gdcm::Element< TVR, VM::VM1\\_n >::GetAsDataElement\(\)](#).

### 10.73.4.33 Write()

```
template<typename TDE , typename TSwap >
const std::ostream & gdcm::DataElement::Write (
    std::ostream & os ) const [inline]
```

## 10.73.5 Friends And Related Symbol Documentation

### 10.73.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const DataElement & _val ) [friend]
```

## 10.73.6 Member Data Documentation

### 10.73.6.1 TagField

`Tag` `gdcm::DataElement::TagField` [protected]

Referenced by [operator==\(\)](#).

### 10.73.6.2 ValueField

`ValuePtr` `gdcm::DataElement::ValueField` [protected]

Referenced by [operator==\(\)](#).

### 10.73.6.3 ValueLengthField

`VL` `gdcm::DataElement::ValueLengthField` [protected]

Referenced by [operator==\(\)](#).

### 10.73.6.4 VRField

`VR` `gdcm::DataElement::VRField` [protected]

Referenced by [operator==\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmDataElement.h](#)

## 10.74 gdcm::DataElementException Class Reference

```
#include <gdcmDataSet.h>
```

Inheritance diagram for gdcm::DataElementException:



Collaboration diagram for gdcm::DataElementException:



The documentation for this class was generated from the following file:

- [gdcmDataSet.h](#)

## 10.75 gdcm::DataSet Class Reference

[DataSet](#).

```
#include <gdcmDataSet.h>
```

Inheritance diagram for `gdcM::DataEvent`:



Collaboration diagram for `gdcM::DataEvent`:



### Public Types

- typedef [DataEvent](#) Self
- typedef [AnyEvent](#) Superclass

### Public Member Functions

- [DataEvent](#) (const char \*bytes=nullptr, size\_t len=0)

- [DataEvent](#) (const [Self](#) &s)
- [~DataEvent](#) () override=default
- bool [CheckEvent](#) (const [::gdcm::Event](#) \*e) const override
- const char \* [GetData](#) () const
- size\_t [GetDataLength](#) () const
- const char \* [GetEventName](#) () const override
- [::gdcm::Event](#) \* [MakeObject](#) () const override
- void [operator=](#) (const [Self](#) &)=delete
- void [SetData](#) (const char \*bytes, size\_t len)

## Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()
- [Event](#) (const [Event](#) &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) \*) const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

## 10.75.1 Detailed Description

[DataEvent](#).

## 10.75.2 Member Typedef Documentation

### 10.75.2.1 Self

```
typedef DataEvent gdcm::DataEvent::Self
```

### 10.75.2.2 Superclass

```
typedef AnyEvent gdcm::DataEvent::Superclass
```

## 10.75.3 Constructor & Destructor Documentation

### 10.75.3.1 DataEvent() [1/2]

```
gdcm::DataEvent::DataEvent (
    const char * bytes = nullptr,
    size_t len = 0 ) [inline]
```

### 10.75.3.2 ~DataEvent()

```
gdcmm::DataEvent::~~DataEvent ( ) [override], [default]
```

### 10.75.3.3 DataEvent() [2/2]

```
gdcmm::DataEvent::DataEvent (
    const Self & s ) [inline]
```

## 10.75.4 Member Function Documentation

### 10.75.4.1 CheckEvent()

```
bool gdcmm::DataEvent::CheckEvent (
    const ::gdcmm::Event * e ) const [inline], [override]
```

### 10.75.4.2 GetData()

```
const char * gdcmm::DataEvent::GetData ( ) const [inline]
```

### 10.75.4.3 GetDataLength()

```
size_t gdcmm::DataEvent::GetDataLength ( ) const [inline]
```

### 10.75.4.4 GetEventName()

```
const char * gdcmm::DataEvent::GetEventName ( ) const [inline], [override], [virtual]
```

Return the StringName associated with the event.

Implements [gdcmm::Event](#).

### 10.75.4.5 MakeObject()

```
::gdcmm::Event * gdcmm::DataEvent::MakeObject ( ) const [inline], [override], [virtual]
```

Create an [Event](#) of this type This method work as a Factory for creating events of each particular type.

Implements [gdcmm::Event](#).

#### 10.75.4.6 operator=()

```
void gdcm::DataEvent::operator= (
    const Self & ) [delete]
```

#### 10.75.4.7 SetData()

```
void gdcm::DataEvent::SetData (
    const char * bytes,
    size_t len ) [inline]
```

The documentation for this class was generated from the following file:

- [gdcmDataEvent.h](#)

## 10.76 gdcm::DataSet Class Reference

Class to represent a Data Set (which contains Data Elements)

```
#include <gdcmDataSet.h>
```

Inheritance diagram for gdcm::DataSet:



### Public Types

- typedef DataElementSet::const\_iterator [ConstIterator](#)
- typedef std::set< [DataElement](#) > [DataElementSet](#)
- typedef DataElementSet::iterator [Iterator](#)
- typedef DataElementSet::size\_type [SizeType](#)

## Public Member Functions

- [Iterator Begin](#) ()
- [ConstIterator Begin](#) () const
- void [Clear](#) ()
- template<typename TDE >  
unsigned int [ComputeGroupLength](#) ([Tag](#) const &tag) const
- [Iterator End](#) ()
- [ConstIterator End](#) () const
- bool [FindDataElement](#) (const [PrivateTag](#) &t) const  
*Look up if private tag 't' is present in the dataset:*
- bool [FindDataElement](#) (const [Tag](#) &t) const
- const [DataElement](#) & [FindNextDataElement](#) (const [Tag](#) &t) const
- const [DataElement](#) & [GetDataElement](#) (const [PrivateTag](#) &t) const  
*Return the dataelement.*
- const [DataElement](#) & [GetDataElement](#) (const [Tag](#) &t) const
- [DataElementSet](#) & [GetDES](#) ()
- const [DataElementSet](#) & [GetDES](#) () const
- template<typename TDE >  
[VL GetLength](#) () const
- [MediaStorage GetMediaStorage](#) () const
- std::string [GetPrivateCreator](#) (const [Tag](#) &t) const
- [PrivateTag GetPrivateTag](#) (const [Tag](#) &t) const  
*Return the private tag of the private tag 't', private creator will be set to empty if not found.*
- void [Insert](#) (const [DataElement](#) &de)
- bool [IsEmpty](#) () const  
*Returns if the dataset is empty.*
- const [DataElement](#) & [operator\(\)](#) (uint16\_t group, uint16\_t element) const
- [DataSet](#) & [operator=](#) ([DataSet](#) const &)=default
- const [DataElement](#) & [operator\[\]](#) (const [Tag](#) &t) const
- void [Print](#) (std::ostream &os, std::string const &indent="") const
- template<typename TDE , typename TSwap >  
std::istream & [Read](#) (std::istream &is)
- template<typename TDE , typename TSwap >  
std::istream & [ReadNested](#) (std::istream &is)
- template<typename TDE , typename TSwap >  
std::istream & [ReadSelectedPrivateTags](#) (std::istream &is, const std::set< [PrivateTag](#) > &tags, bool readvalues=true)
- template<typename TDE , typename TSwap >  
std::istream & [ReadSelectedPrivateTagsWithLength](#) (std::istream &is, const std::set< [PrivateTag](#) > &tags, [VL](#) &length, bool readvalues=true)
- template<typename TDE , typename TSwap >  
std::istream & [ReadSelectedTags](#) (std::istream &is, const std::set< [Tag](#) > &tags, bool readvalues=true)
- template<typename TDE , typename TSwap >  
std::istream & [ReadSelectedTagsWithLength](#) (std::istream &is, const std::set< [Tag](#) > &tags, [VL](#) &length, bool readvalues=true)
- template<typename TDE , typename TSwap >  
std::istream & [ReadUpToTag](#) (std::istream &is, const [Tag](#) &t, std::set< [Tag](#) > const &skiptags)
- template<typename TDE , typename TSwap >  
std::istream & [ReadUpToTagWithLength](#) (std::istream &is, const [Tag](#) &t, std::set< [Tag](#) > const &skiptags, [VL](#) &length)



- `template<typename TDE , typename TSwap >`  
`std::istream & ReadWithLength (std::istream &is, VL &length)`
- `SizeType Remove (const Tag &tag)`  
*Completely remove a dataelement from the dataset.*
- `void Replace (const DataElement &de)`  
*Replace a dataelement with another one.*
- `void ReplaceEmpty (const DataElement &de)`  
*Only replace a DICOM attribute when it is missing or empty.*
- `SizeType Size () const`
- `template<typename TDE , typename TSwap >`  
`std::ostream const & Write (std::ostream &os) const`

### Protected Member Functions

- `Tag ComputeDataElement (const PrivateTag &t) const`
- `const DataElement & GetDEEnd () const`
- `void InsertDataElement (const DataElement &de)`

### Friends

- class `CSAHeader`
- `std::ostream & operator<< (std::ostream &_os, const DataSet &)`

## 10.76.1 Detailed Description

Class to represent a Data Set (which contains Data Elements)

A Data Set represents an instance of a real world Information [Object](#)

### Note

DATA SET: Exchanged information consisting of a structured set of [Attribute](#) values directly or indirectly related to Information Objects. The value of each [Attribute](#) in a Data Set is expressed as a Data [Element](#). A collection of Data Elements ordered by increasing Data [Element Tag](#) number that is an encoding of the values of Attributes of a real world object.

Implementation note. If one do: `DataSet ds; ds.SetLength(0); ds.Read(is);` setting length to 0 actually means try to read is as if it was a root [DataSet](#). Other value are undefined (nested dataset with undefined length) or defined length (different from 0) means nested dataset with defined length.

**Warning**

a [DataSet](#) does not have a Transfer Syntax type, only a [File](#) does.

**Examples**

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CompressLossyJPEG.cs](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Write.cxx](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [FixOrientation.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [SimplePrint.cs](#), [SortImage.cxx](#), [SortImage2.cs](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), [VolumeSorter.cxx](#), [csa2img.cxx](#), [gdcmrtonplan.cxx](#), [gdcmrtpplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

**10.76.2 Member Typedef Documentation****10.76.2.1 ConstIterator**

```
typedef DataSet::const_iterator gdcm::DataSet::ConstIterator
```

**10.76.2.2 DataSet**

```
typedef std::set<DataElement> gdcm::DataSet::DataSet
```

**10.76.2.3 Iterator**

```
typedef DataSet::iterator gdcm::DataSet::Iterator
```

**10.76.2.4 SizeType**

```
typedef DataSet::size_type gdcm::DataSet::SizeType
```

**10.76.3 Member Function Documentation****10.76.3.1 Begin() [1/2]**

```
Iterator gdcm::DataSet::Begin ( ) [inline]
```

### 10.76.3.2 Begin() [2/2]

```
ConstIterator gdcm::DataSet::Begin ( ) const [inline]
```

#### Examples

[DiffFile.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpVisusChange.cxx](#), and [DuplicatePCDE.cxx](#).

### 10.76.3.3 Clear()

```
void gdcm::DataSet::Clear ( ) [inline]
```

Referenced by [gdcm::Item::Read\(\)](#).

### 10.76.3.4 ComputeDataElement()

```
Tag gdcm::DataSet::ComputeDataElement (
    const PrivateTag & t ) const [protected]
```

References [gdcm::operator<<\(\)](#).

### 10.76.3.5 ComputeGroupLength()

```
template<typename TDE >
unsigned int gdcm::DataSet::ComputeGroupLength (
    Tag const & tag ) const [inline]
```

References [gdcm::Tag::GetElement\(\)](#), and [gdcm::Tag::GetGroup\(\)](#).

### 10.76.3.6 End() [1/2]

```
Iterator gdcm::DataSet::End ( ) [inline]
```

### 10.76.3.7 End() [2/2]

```
ConstIterator gdcm::DataSet::End ( ) const [inline]
```

#### Examples

[DiffFile.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpVisusChange.cxx](#), and [DuplicatePCDE.cxx](#).

**10.76.3.8 FindDataElement()** [1/2]

```
bool gdcM::DataSet::FindDataElement (
    const PrivateTag & t ) const
```

Look up if private tag 't' is present in the dataset:

**Examples**

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [MrProtocol.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadGEMSSDO.cxx](#), [csa2img.cxx](#), [gdcMrtionplan.cxx](#), [gdcMrtplan.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

Referenced by [gdcM::Attribute< Group, Element, TVR, TVM >::SetFromDataSet\(\)](#), [gdcM::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet\(\)](#), and [gdcM::Attribute< Group, Element, TVR, VM::VM1\\_n >::SetFromDataSet\(\)](#).

**10.76.3.9 FindDataElement()** [2/2]

```
bool gdcM::DataSet::FindDataElement (
    const Tag & t ) const [inline]
```

**10.76.3.10 FindNextDataElement()**

```
const DataElement & gdcM::DataSet::FindNextDataElement (
    const Tag & t ) const [inline]
```

**Examples**

[DuplicatePCDE.cxx](#).

**10.76.3.11 GetDataElement()** [1/2]

```
const DataElement & gdcM::DataSet::GetDataElement (
    const PrivateTag & t ) const
```

Return the dataelement.

**10.76.3.12 GetDataElement() [2/2]**

```
const DataElement & gdcm::DataSet::GetDataElement (
    const Tag & t ) const [inline]
```

Return the [DataElement](#) with [Tag](#) 't'

**Warning**

: This only search at the 'root level' of the [DataSet](#)

**Examples**

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [DecompressImage.cs](#), [DeriveSeries.cxx](#), [DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [csa2img.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

Referenced by [gdcm::Attribute< Group, Element, TVR, TVM >::Set\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Set\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >::Set\(\)](#), [gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet\(\)](#), and [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >::SetFromDataSet\(\)](#).

**10.76.3.13 GetDEEnd()**

```
const DataElement & gdcm::DataSet::GetDEEnd ( ) const [protected]
```

**10.76.3.14 GetDES() [1/2]**

```
DataElementSet & gdcm::DataSet::GetDES ( ) [inline]
```

**10.76.3.15 GetDES() [2/2]**

```
const DataElementSet & gdcm::DataSet::GetDES ( ) const [inline]
```

**Examples**

[ReadAndDumpDICOMDIR.cxx](#).

**10.76.3.16 GetLength()**

```
template<typename TDE >
VL gdcm::DataSet::GetLength ( ) const [inline]
```

References [gdcm::VL::GetLength\(\)](#).

### 10.76.3.17 GetMediaStorage()

```
MediaStorage gdcm::DataSet::GetMediaStorage ( ) const
```

### 10.76.3.18 GetPrivateCreator()

```
std::string gdcm::DataSet::GetPrivateCreator (
    const Tag & t ) const
```

Return the private creator of the private tag 't': or an empty string when not found

#### Examples

[DuplicatePCDE.cxx](#).

### 10.76.3.19 GetPrivateTag()

```
PrivateTag gdcm::DataSet::GetPrivateTag (
    const Tag & t ) const
```

Return the private tag of the private tag 't', private creator will be set to empty if not found.

### 10.76.3.20 Insert()

```
void gdcm::DataSet::Insert (
    const DataElement & de ) [inline]
```

Insert a [DataElement](#) in the [DataSet](#).

#### Warning

: [Tag](#) need to be  $\geq 0x8$  to be considered valid data element

#### Examples

[CreateJPIPDataSet.cxx](#), [DumpSiemensBase64.cxx](#), [DuplicatePCDE.cxx](#), [Extracting\\_All\\_Resolution.cxx](#),  
[Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [GenAllIVR.cxx](#),  
[GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [NewSequence.cs](#), [StreamImageReaderTest.cxx](#),  
and [TemplateEmptyImage.cxx](#).

References [gdcmErrorMacro](#), [gdcm::Tag::GetGroup\(\)](#), and [gdcm::DataElement::GetTag\(\)](#).

**10.76.3.21 InsertDataElement()**

```
void gdcm::DataSet::InsertDataElement (
    const DataElement & de ) [inline], [protected]
```

References [gdcmWarningMacro](#), [gdcm::Value::GetLength\(\)](#), [gdcm::DataElement::GetValue\(\)](#), [gdcm::DataElement::GetVL\(\)](#), and [gdcm::DataElement::IsEmpty\(\)](#).

**10.76.3.22 IsEmpty()**

```
bool gdcm::DataSet::IsEmpty ( ) const [inline]
```

Returns if the dataset is empty.

Referenced by [gdcm::Item::Read\(\)](#).

**10.76.3.23 operator()()**

```
const DataElement & gdcm::DataSet::operator() (
    uint16_t group,
    uint16_t element ) const [inline]
```

**10.76.3.24 operator=()**

```
DataSet & gdcm::DataSet::operator= (
    DataSet const & ) [default]
```

**10.76.3.25 operator[]()**

```
const DataElement & gdcm::DataSet::operator[] (
    const Tag & t ) const [inline]
```

**10.76.3.26 Print()**

```
void gdcm::DataSet::Print (
    std::ostream & os,
    std::string const & indent = "" ) const [inline]
```

**10.76.3.27 Read()**

```
template<typename TDE , typename TSwap >
std::istream & gdcm::DataSet::Read (
    std::istream & is )
```

**Examples**

[DumpToshibaDTI.cxx](#), and [DumpToshibaDTI2.cxx](#).

### 10.76.3.28 ReadNested()

```
template<typename TDE , typename TSwap >
std::istream & gdcmm::DataSet::ReadNested (
    std::istream & is )
```

### 10.76.3.29 ReadSelectedPrivateTags()

```
template<typename TDE , typename TSwap >
std::istream & gdcmm::DataSet::ReadSelectedPrivateTags (
    std::istream & is,
    const std::set< PrivateTag > & tags,
    bool readvalues = true )
```

### 10.76.3.30 ReadSelectedPrivateTagsWithLength()

```
template<typename TDE , typename TSwap >
std::istream & gdcmm::DataSet::ReadSelectedPrivateTagsWithLength (
    std::istream & is,
    const std::set< PrivateTag > & tags,
    VL & length,
    bool readvalues = true )
```

### 10.76.3.31 ReadSelectedTags()

```
template<typename TDE , typename TSwap >
std::istream & gdcmm::DataSet::ReadSelectedTags (
    std::istream & is,
    const std::set< Tag > & tags,
    bool readvalues = true )
```

### 10.76.3.32 ReadSelectedTagsWithLength()

```
template<typename TDE , typename TSwap >
std::istream & gdcmm::DataSet::ReadSelectedTagsWithLength (
    std::istream & is,
    const std::set< Tag > & tags,
    VL & length,
    bool readvalues = true )
```

### 10.76.3.33 ReadUpToTag()

```
template<typename TDE , typename TSwap >
std::istream & gdcmm::DataSet::ReadUpToTag (
    std::istream & is,
    const Tag & t,
    std::set< Tag > const & skiptags )
```



### 10.76.3.34 ReadUpToTagWithLength()

```
template<typename TDE , typename TSwap >
std::istream & gdcm::DataSet::ReadUpToTagWithLength (
    std::istream & is,
    const Tag & t,
    std::set< Tag > const & skiptags,
    VL & length )
```

### 10.76.3.35 ReadWithLength()

```
template<typename TDE , typename TSwap >
std::istream & gdcm::DataSet::ReadWithLength (
    std::istream & is,
    VL & length )
```

### 10.76.3.36 Remove()

```
SizeType gdcm::DataSet::Remove (
    const Tag & tag ) [inline]
```

Completely remove a dataelement from the dataset.

#### Examples

[ClinicalTrialIdentificationWorkflow.cs](#), [GenFakeIdentifyFile.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [ReformatFile.cs](#), [StandardizeFiles.cs](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

### 10.76.3.37 Replace()

```
void gdcm::DataSet::Replace (
    const DataElement & de ) [inline]
```

Replace a dataelement with another one.

#### Examples

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CreateFakeRTDOSE.cxx](#), [DeriveSeries.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenFakeIdentifyFile.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [PatchFile.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

References [gdcmAssertAlwaysMacro](#).

### 10.76.3.38 ReplaceEmpty()

```
void gdcM::DataSet::ReplaceEmpty (
    const DataElement & de ) [inline]
```

Only replace a DICOM attribute when it is missing or empty.

#### Examples

[rle2img.cxx](#).

References [gdcMAssertAlwaysMacro](#).

### 10.76.3.39 Size()

```
SizeType gdcM::DataSet::Size ( ) const [inline]
```

#### Examples

[DumpGEMSMovieGroup.cxx](#).

Referenced by [gdcM::SequenceOfItems::Read\(\)](#).

### 10.76.3.40 Write()

```
template<typename TDE , typename TSwap >
std::ostream const & gdcM::DataSet::Write (
    std::ostream & os ) const
```

## 10.76.4 Friends And Related Symbol Documentation

### 10.76.4.1 CSAHeader

```
friend class CSAHeader [friend]
```

### 10.76.4.2 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const DataSet & val ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcMDataSet.h](#)

## 10.77 gdcm::DataSetEvent Class Reference

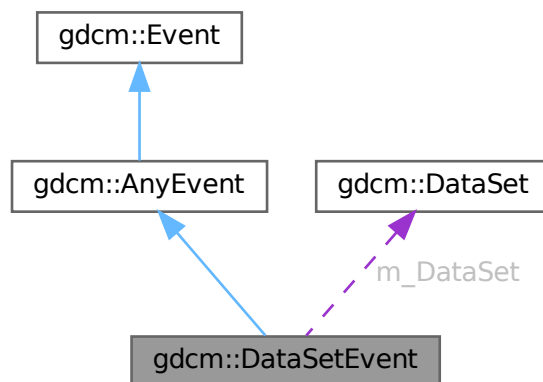
[DataSetEvent](#).

```
#include <gdcmDataSetEvent.h>
```

Inheritance diagram for gdcm::DataSetEvent:



Collaboration diagram for gdcm::DataSetEvent:



### Public Types

- typedef [DataSetEvent](#) Self
- typedef [AnyEvent](#) Superclass

## Public Member Functions

- [DataSetEvent](#) (const [Self](#) &s)
- [DataSetEvent](#) ([DataSet](#) const \*ds=nullptr)
- [~DataSetEvent](#) () override=default
- bool [CheckEvent](#) (const [::gdcmm::Event](#) \*e) const override
- [DataSet](#) const & [GetDataSet](#) () const
- const char \* [GetEventName](#) () const override
- [::gdcmm::Event](#) \* [MakeObject](#) () const override
- void [operator=](#) (const [Self](#) &)=delete

## Public Member Functions inherited from [gdcmm::Event](#)

- [Event](#) ()
- [Event](#) (const [Event](#) &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) \*) const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

## Public Attributes

- const [DataSet](#) \* [m\\_DataSet](#)

## 10.77.1 Detailed Description

[DataSetEvent](#).

Special type of event triggered during the [DataSet](#) store/move process

## 10.77.2 Member Typedef Documentation

### 10.77.2.1 Self

```
typedef DataSetEvent gdcmm::DataSetEvent::Self
```

### 10.77.2.2 Superclass

```
typedef AnyEvent gdcmm::DataSetEvent::Superclass
```

## 10.77.3 Constructor & Destructor Documentation

### 10.77.3.1 DataSetEvent() [1/2]

```
gdcm::DataSetEvent::DataSetEvent (
    DataSet const * ds = nullptr ) [inline]
```

### 10.77.3.2 ~DataSetEvent()

```
gdcm::DataSetEvent::~~DataSetEvent ( ) [override], [default]
```

### 10.77.3.3 DataSetEvent() [2/2]

```
gdcm::DataSetEvent::DataSetEvent (
    const Self & s ) [inline]
```

## 10.77.4 Member Function Documentation

### 10.77.4.1 CheckEvent()

```
bool gdcm::DataSetEvent::CheckEvent (
    const ::gdcm::Event * e ) const [inline], [override]
```

### 10.77.4.2 GetDataSet()

```
DataSet const & gdcm::DataSetEvent::GetDataSet ( ) const [inline]
```

References [m\\_DataSet](#).

### 10.77.4.3 GetEventName()

```
const char * gdcm::DataSetEvent::GetEventName ( ) const [inline], [override], [virtual]
```

Return the StringName associated with the event.

Implements [gdcm::Event](#).

### 10.77.4.4 MakeObject()

```
::gdcm::Event * gdcm::DataSetEvent::MakeObject ( ) const [inline], [override], [virtual]
```

Create an [Event](#) of this type This method work as a Factory for creating events of each particular type.

Implements [gdcm::Event](#).

#### 10.77.4.5 operator=()

```
void gdcM::DataSetEvent::operator= (
    const Self & ) [delete]
```

### 10.77.5 Member Data Documentation

#### 10.77.5.1 m\_DataSet

```
const DataSet* gdcM::DataSetEvent::m_DataSet
```

Referenced by [GetDataSet\(\)](#).

The documentation for this class was generated from the following file:

- [gdcMDataSetEvent.h](#)

## 10.78 gdcM::DataSetHelper Class Reference

[DataSetHelper](#) (internal class, not intended for user level)

```
#include <gdcMDataSetHelper.h>
```

### Static Public Member Functions

- static [VR ComputeVR](#) ([File](#) const &file, [DataSet](#) const &ds, const [Tag](#) &tag)

#### 10.78.1 Detailed Description

[DataSetHelper](#) (internal class, not intended for user level)

### Examples

[SimplePrint.cs](#).

## 10.78.2 Member Function Documentation

### 10.78.2.1 ComputeVR()

```
static VR gdcmm::DataSetHelper::ComputeVR (
    File const & file,
    DataSet const & ds,
    const Tag & tag ) [static]
```

ds -> current dataset, which is not the same as the root dataset return `VR::INVALID` in case of error

#### Examples

[SimplePrint.cs](#).

The documentation for this class was generated from the following file:

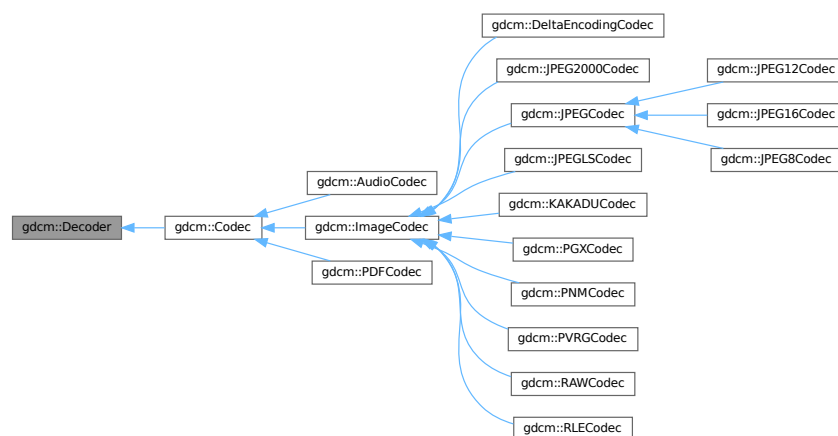
- [gdcmmDataSetHelper.h](#)

## 10.79 gdcmm::Decoder Class Reference

[Decoder](#).

```
#include <gdcmmDecoder.h>
```

Inheritance diagram for gdcmm::Decoder:



## Public Member Functions

- virtual [~Decoder](#) ()=default
- virtual bool [CanDecode](#) ([TransferSyntax](#) const &) const =0  
*Return whether this decoder support this transfer syntax (can decode it)*
- virtual bool [Decode](#) ([DataElement](#) const &, [DataElement](#) &)  
*Decode.*

## Protected Member Functions

- virtual bool [DecodeByStreams](#) (std::istream &, std::ostream &)

## 10.79.1 Detailed Description

[Decoder](#).

## 10.79.2 Constructor & Destructor Documentation

### 10.79.2.1 ~Decoder()

```
virtual gdcm::Decoder::~~Decoder ( ) [virtual], [default]
```

## 10.79.3 Member Function Documentation

### 10.79.3.1 CanDecode()

```
virtual bool gdcm::Decoder::CanDecode (
    TransferSyntax const & ) const [pure virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Implemented in [gdcm::AudioCodec](#), [gdcm::ImageCodec](#), [gdcm::PDFCodec](#), [gdcm::JPEG2000Codec](#), [gdcm::JPEGCodec](#), [gdcm::JPEGLSCodec](#), [gdcm::KAKADUCodec](#), [gdcm::PGXCodec](#), [gdcm::PNMCodec](#), [gdcm::PVRGCodec](#), [gdcm::RAWCodec](#), and [gdcm::RLECodec](#).

### 10.79.3.2 Decode()

```
virtual bool gdcm::Decoder::Decode (
    DataElement const & ,
    DataElement & ) [inline], [virtual]
```

Decode.

Reimplemented in [gdcm::DeltaEncodingCodec](#), [gdcm::AudioCodec](#), [gdcm::JPEG2000Codec](#), [gdcm::JPEGCodec](#), [gdcm::JPEGLSCodec](#), [gdcm::KAKADUCodec](#), [gdcm::PDFCodec](#), [gdcm::PVRGCodec](#), [gdcm::RAWCodec](#), [gdcm::RLECodec](#), and [gdcm::ImageCodec](#).



### 10.79.3.3 DecodeByStreams()

```
virtual bool gdcm::Decoder::DecodeByStreams (
    std::istream & ,
    std::ostream & ) [inline], [protected], [virtual]
```

Reimplemented in [gdcm::JPEG12Codec](#), [gdcm::JPEG16Codec](#), [gdcm::JPEG2000Codec](#), [gdcm::JPEG8Codec](#), [gdcm::JPEGCodec](#), [gdcm::RAWCodec](#), [gdcm::RLECodec](#), and [gdcm::ImageCodec](#).

The documentation for this class was generated from the following file:

- [gdcmDecoder.h](#)

## 10.80 gdcm::DefinedTerms Class Reference

Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data [Element](#) with Defined Terms that does not contain a [Value](#) equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation [Type](#) ID (4008,0210) is an example of a Data [Element](#) having Defined Terms. It is defined to have a [Value](#) that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data [Element](#) has Defined Terms other Interpretation [Type](#) IDs may be defined by the implementor.

```
#include <gdcmDefinedTerms.h>
```

### Public Member Functions

- [DefinedTerms](#) ()=default

### 10.80.1 Detailed Description

Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data [Element](#) with Defined Terms that does not contain a [Value](#) equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation [Type](#) ID (4008,0210) is an example of a Data [Element](#) having Defined Terms. It is defined to have a [Value](#) that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data [Element](#) has Defined Terms other Interpretation [Type](#) IDs may be defined by the implementor.

### 10.80.2 Constructor & Destructor Documentation

#### 10.80.2.1 DefinedTerms()

```
gdcm::DefinedTerms::DefinedTerms ( ) [default]
```

The documentation for this class was generated from the following file:

- [gdcmDefinedTerms.h](#)

## 10.81 gdcm::Defs Class Reference

FIXME I do not like the name 'Defs'.

```
#include <gdcmDefs.h>
```

### Public Member Functions

- [Defs](#) ()
- [Defs](#) (const [Defs](#) &val)=delete
- [~Defs](#) ()
- const [IOD](#) & [GetIODFromFile](#) (const [File](#) &file) const
- [IODs](#) & [GetIODs](#) ()
- const [IODs](#) & [GetIODs](#) () const
- [Macros](#) & [GetMacros](#) ()
- const [Macros](#) & [GetMacros](#) () const
- [Modules](#) & [GetModules](#) ()
- const [Modules](#) & [GetModules](#) () const
- [Type](#) [GetTypeFromTag](#) (const [File](#) &file, const [Tag](#) &tag) const
- bool [IsEmpty](#) () const
- [Defs](#) & [operator=](#) (const [Defs](#) &val)=delete
- bool [Verify](#) (const [DataSet](#) &ds) const
- bool [Verify](#) (const [File](#) &file) const

### Static Public Member Functions

- static const char \* [GetIODNameFromMediaStorage](#) ([MediaStorage](#) const &ms)

### Protected Member Functions

- void [LoadDefaults](#) ()
- void [LoadFromFile](#) (const char \*filename)

### Friends

- class [Global](#)

### 10.81.1 Detailed Description

FIXME I do not like the name 'Defs'.

#### Note

bla

#### Examples

[GenerateStandardSOPClasses.cxx](#), and [TraverseModules.cxx](#).

## 10.81.2 Constructor & Destructor Documentation

### 10.81.2.1 Defs() [1/2]

```
gdcm::Defs::Defs ( )
```

### 10.81.2.2 ~Defs()

```
gdcm::Defs::~~Defs ( )
```

### 10.81.2.3 Defs() [2/2]

```
gdcm::Defs::Defs (
    const Defs & val ) [delete]
```

## 10.81.3 Member Function Documentation

### 10.81.3.1 GetIODFromFile()

```
const IOD & gdcm::Defs::GetIODFromFile (
    const File & file ) const
```

### 10.81.3.2 GetIODNameFromMediaStorage()

```
static const char * gdcm::Defs::GetIODNameFromMediaStorage (
    MediaStorage const & ms ) [static]
```

#### Examples

[GenerateStandardSOPClasses.cxx](#).

### 10.81.3.3 GetIODs() [1/2]

```
IODs & gdcm::Defs::GetIODs ( ) [inline]
```

### 10.81.3.4 GetIODs() [2/2]

```
const IODs & gdcm::Defs::GetIODs ( ) const [inline]
```

#### Examples

[TraverseModules.cxx](#).

### 10.81.3.5 GetMacros() [1/2]

```
Macros & gdcm::Defs::GetMacros ( ) [inline]
```

### 10.81.3.6 GetMacros() [2/2]

```
const Macros & gdcm::Defs::GetMacros ( ) const [inline]
```

Users should not directly use [Macro](#). [Macro](#) are simply a way for DICOM WG to re-use Tables. [Macros](#) are conveniently wrapped within [Modules](#). See [gdcm::Module](#) API directly

#### Examples

[TraverseModules.cxx](#).

### 10.81.3.7 GetModules() [1/2]

```
Modules & gdcm::Defs::GetModules ( ) [inline]
```

### 10.81.3.8 GetModules() [2/2]

```
const Modules & gdcm::Defs::GetModules ( ) const [inline]
```

#### Examples

[TraverseModules.cxx](#).

### 10.81.3.9 GetTypeFromTag()

```
Type gdcm::Defs::GetTypeFromTag (
    const File & file,
    const Tag & tag ) const
```

### 10.81.3.10 IsEmpty()

```
bool gdcm::Defs::IsEmpty ( ) const [inline]
```

### 10.81.3.11 LoadDefaults()

```
void gdcm::Defs::LoadDefaults ( ) [protected]
```

### 10.81.3.12 LoadFromFile()

```
void gdcmm::Defs::LoadFromFile (
    const char * filename ) [protected]
```

### 10.81.3.13 operator=()

```
Defs & gdcmm::Defs::operator= (
    const Defs & val ) [delete]
```

### 10.81.3.14 Verify() [1/2]

```
bool gdcmm::Defs::Verify (
    const DataSet & ds ) const
```

### 10.81.3.15 Verify() [2/2]

```
bool gdcmm::Defs::Verify (
    const File & file ) const
```

## 10.81.4 Friends And Related Symbol Documentation

### 10.81.4.1 Global

```
friend class Global [friend]
```

The documentation for this class was generated from the following file:

- [gdcmmDefs.h](#)

## 10.82 gdcm::DeltaEncodingCodec Class Reference

[DeltaEncodingCodec](#) compression used by some private vendor.

```
#include <gdcmDeltaEncodingCodec.h>
```

Inheritance diagram for `gdcm::DeltaEncodingCodec`:



Collaboration diagram for `gdcm::DeltaEncodingCodec`:



### Public Member Functions

- [DeltaEncodingCodec](#) ()
- [~DeltaEncodingCodec](#) ()
- bool [CanDecode](#) ([TransferSyntax](#) const &ts)
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os)

*Decode.*

## Public Member Functions inherited from [gdcm::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &) const override  
*Return whether this coder support this transfer syntax (can code it)*
- bool [CanDecode](#) ([TransferSyntax](#) const &) const override  
*Return whether this decoder support this transfer syntax (can decode it)*
- bool [CleanupUnusedBits](#) (char \*data, size\_t datalen)
- virtual [ImageCodec](#) \* [Clone](#) () const =0
- bool [Decode](#) ([DataElement](#) const &is\_, [DataElement](#) &os) override  
*Decode.*
- const unsigned int \* [GetDimensions](#) () const
- virtual bool [GetHeaderInfo](#) (std::istream &is\_, [TransferSyntax](#) &ts)
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- virtual void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)

## Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default
- virtual bool [Code](#) ([DataElement](#) const &in\_, [DataElement](#) &out\_)  
*Code.*

## Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

## Protected Member Functions

- bool [Decode](#) (std::istream &is, std::ostream &os)

## Protected Member Functions inherited from [gdcm::ImageCodec](#)

- virtual bool [AppendFrameEncode](#) (std::ostream &out, const char \*data, size\_t datalen)
- virtual bool [AppendRowEncode](#) (std::ostream &out, const char \*data, size\_t datalen)
- bool [DecodeByStreams](#) (std::istream &is\_, std::ostream &os) override
- bool [DoByteSwap](#) (std::istream &is\_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is\_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is\_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is\_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is\_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is\_, std::ostream &os)
- virtual bool [IsFrameEncoder](#) ()
- virtual bool [IsRowEncoder](#) ()
- virtual bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)
- virtual bool [StartEncode](#) (std::ostream &os)
- virtual bool [StopEncode](#) (std::ostream &os)

## Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char \*bv, unsigned long len, std::ostream &os)

## Additional Inherited Members

## Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

## Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) [LUT](#)
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) [PF](#)
- [PhotometricInterpretation](#) [PI](#)
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

## 10.82.1 Detailed Description

[DeltaEncodingCodec](#) compression used by some private vendor.



## 10.82.2 Constructor & Destructor Documentation

### 10.82.2.1 DeltaEncodingCodec()

```
gdcm::DeltaEncodingCodec::DeltaEncodingCodec ( )
```

### 10.82.2.2 ~DeltaEncodingCodec()

```
gdcm::DeltaEncodingCodec::~~DeltaEncodingCodec ( )
```

## 10.82.3 Member Function Documentation

### 10.82.3.1 CanDecode()

```
bool gdcm::DeltaEncodingCodec::CanDecode (
    TransferSyntax const & ts )
```

### 10.82.3.2 Decode() [1/2]

```
bool gdcm::DeltaEncodingCodec::Decode (
    DataElement const & ,
    DataElement & ) [virtual]
```

Decode.

Reimplemented from [gdcm::Decoder](#).

### 10.82.3.3 Decode() [2/2]

```
bool gdcm::DeltaEncodingCodec::Decode (
    std::istream & is,
    std::ostream & os ) [protected]
```

The documentation for this class was generated from the following file:

- [gdcmDeltaEncodingCodec.h](#)

## 10.83 gdcm::DICOMDIR Class Reference

[DICOMDIR](#) class.

```
#include <gdcmDICOMDIR.h>
```

## Public Member Functions

- [DICOMDIR](#) ()=default
- [DICOMDIR](#) ([FileSet](#) fs)

### 10.83.1 Detailed Description

[DICOMDIR](#) class.

Structured for handling [DICOMDIR](#)

### 10.83.2 Constructor & Destructor Documentation

#### 10.83.2.1 [DICOMDIR](#)() [1/2]

```
gdcm::DICOMDIR::DICOMDIR ( ) [default]
```

#### 10.83.2.2 [DICOMDIR](#)() [2/2]

```
gdcm::DICOMDIR::DICOMDIR (
    FileSet fs ) [inline]
```

The documentation for this class was generated from the following file:

- [gdcmDICOMDIR.h](#)

## 10.84 [gdcm::DICOMDIRGenerator](#) Class Reference

[DICOMDIRGenerator](#) class.

```
#include <gdcmDICOMDIRGenerator.h>
```

## Public Types

- typedef [Directory::FileNamesType](#) FileNamesType
- typedef [Directory::FilenameType](#) FilenameType

**Public Member Functions**

- [DICOMDIRGenerator](#) ()
- [~DICOMDIRGenerator](#) ()
- bool [Generate](#) ()  
*Main function to generate the [DICOMDIR](#).*
- [File](#) & [GetFile](#) ()
- void [SetDescriptor](#) (const char \*d)
- void [SetFile](#) (const [File](#) &f)  
*Set/Get file. The [DICOMDIR](#) file will be valid once a call to Generate has been done.*
- void [SetFilenames](#) ([FilenamesType](#) const &fns)  
*Set the list of filenames from which the [DICOMDIR](#) should be generated from.*
- void [SetRootDirectory](#) ([FilenameType](#) const &root)  
*Set the root directory from which the filenames should be considered.*

**Protected Member Functions**

- bool [AddImageDirectoryRecord](#) ()
- bool [AddPatientDirectoryRecord](#) ()
- bool [AddSeriesDirectoryRecord](#) ()
- bool [AddStudyDirectoryRecord](#) ()
- [Scanner](#) & [GetScanner](#) ()

**10.84.1 Detailed Description**

[DICOMDIRGenerator](#) class.

This is a STD-GEN-CD [DICOMDIR](#) generator. ref: PS 3.11-2008 Annex D (Normative) - General Purpose CD-R and DVD Interchange Profiles

**Note**

PS 3.11 - 2008 / D.3.2 Physical Medium And Medium Format The STD-GEN-CD and STD-GEN-SEC-CD application profiles require the 120 mm CD-R physical medium with the ISO/IEC 9660 Media Format, as defined in PS3.12. See also PS 3.12 - 2008 / Annex F 120mm CD-R Medium (Normative) and PS 3.10 - 2008 / 8 DICOM [File](#) Service / 8.1 FILE-SET

**Warning**

: PS 3.11 - 2008 / D.3.1 SOP Classes and Transfer Syntaxes Composite [Image](#) & Stand-alone Storage are required to be stored as Explicit [VR](#) Little Endian Uncompressed (1.2.840.10008.1.2.1). When a DICOM file is found using another Transfer Syntax the generator will simply stops.

- Input files should be Explicit [VR](#) Little Endian
- filenames should be valid [VR::CS](#) value (16 bytes, upper case ...)

**Bug** : There is a current limitation of not handling Referenced SOP Class UID / Referenced SOP Instance UID simply because the [Scanner](#) does not allow us See PS 3.11 / [Table](#) D.3-2 STD-GEN Additional [DICOMDIR](#) Keys

**Examples**

[GenerateDICOMDIR.cs](#).

## 10.84.2 Member Typedef Documentation

### 10.84.2.1 FilenamesType

```
typedef Directory::FilenamesType gdcm::DICOMDIRGenerator::FilenamesType
```

### 10.84.2.2 FilenameType

```
typedef Directory::FilenameType gdcm::DICOMDIRGenerator::FilenameType
```

## 10.84.3 Constructor & Destructor Documentation

### 10.84.3.1 DICOMDIRGenerator()

```
gdcm::DICOMDIRGenerator::DICOMDIRGenerator ( )
```

### 10.84.3.2 ~DICOMDIRGenerator()

```
gdcm::DICOMDIRGenerator::~~DICOMDIRGenerator ( )
```

## 10.84.4 Member Function Documentation

### 10.84.4.1 AddImageDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddImageDirectoryRecord ( ) [protected]
```

### 10.84.4.2 AddPatientDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddPatientDirectoryRecord ( ) [protected]
```

### 10.84.4.3 AddSeriesDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddSeriesDirectoryRecord ( ) [protected]
```

### 10.84.4.4 AddStudyDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddStudyDirectoryRecord ( ) [protected]
```

#### 10.84.4.5 Generate()

```
bool gdcm::DICOMDIRGenerator::Generate ( )
```

Main function to generate the [DICOMDIR](#).

##### Examples

[GenerateDICOMDIR.cs](#).

#### 10.84.4.6 GetFile()

```
File & gdcm::DICOMDIRGenerator::GetFile ( )
```

##### Examples

[GenerateDICOMDIR.cs](#).

#### 10.84.4.7 GetScanner()

```
Scanner & gdcm::DICOMDIRGenerator::GetScanner ( ) [protected]
```

#### 10.84.4.8 SetDescriptor()

```
void gdcm::DICOMDIRGenerator::SetDescriptor (
    const char * d )
```

Set the [File](#) Set ID.

##### Warning

this need to be a valid [VR::CS](#) value

##### Examples

[GenerateDICOMDIR.cs](#).

#### 10.84.4.9 SetFile()

```
void gdcm::DICOMDIRGenerator::SetFile (
    const File & f )
```

Set/Get file. The [DICOMDIR](#) file will be valid once a call to Generate has been done.

#### 10.84.4.10 SetFileNames()

```
void gdcmm::DICOMDIRGenerator::SetFileNames (
    FilenamesType const & fns )
```

Set the list of filenames from which the [DICOMDIR](#) should be generated from.

##### Examples

[GenerateDICOMDIR.cs.](#)

#### 10.84.4.11 SetRootDirectory()

```
void gdcmm::DICOMDIRGenerator::SetRootDirectory (
    FilenameType const & root )
```

Set the root directory from which the filenames should be considered.

The documentation for this class was generated from the following file:

- [gdcmmDICOMDIRGenerator.h](#)

## 10.85 gdcmm::Dict Class Reference

Class to represent a map of [DictEntry](#).

```
#include <gdcmmDict.h>
```

### Public Types

- typedef MapDictEntry::const\_iterator [ConstIterator](#)
- typedef MapDictEntry::iterator [Iterator](#)
- typedef std::map< [Tag](#), [DictEntry](#) > [MapDictEntry](#)

### Public Member Functions

- [Dict](#) ()
  - [Dict](#) (const [Dict](#) &\_val)=delete
  - void [AddDictEntry](#) (const [Tag](#) &tag, const [DictEntry](#) &de)
  - [ConstIterator](#) [Begin](#) () const
  - [ConstIterator](#) [End](#) () const
  - const [DictEntry](#) & [GetDictEntry](#) (const [Tag](#) &tag) const
  - const [DictEntry](#) & [GetDictEntryByKeyword](#) (const char \*keyword, [Tag](#) &tag) const
  - const [DictEntry](#) & [GetDictEntryByName](#) (const char \*name, [Tag](#) &tag) const
  - const char \* [GetKeywordFromTag](#) ([Tag](#) const &tag) const
- Function to return the Keyword from a [Tag](#).*
- bool [IsEmpty](#) () const
  - [Dict](#) & [operator=](#) (const [Dict](#) &\_val)=delete

## Protected Member Functions

- void [LoadDefault](#) ()

## Friends

- class [Dicts](#)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [Dict](#) &\_val)

## 10.85.1 Detailed Description

Class to represent a map of [DictEntry](#).

### Note

bla TODO FIXME: For [Element](#) == 0x0 need to return Name = Group Length ValueRepresentation = UL Value↵ Multiplicity = 1

### Examples

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [PublicDict.cxx](#), and [ReadAndPrintAttributes.cxx](#).

## 10.85.2 Member Typedef Documentation

### 10.85.2.1 ConstIterator

```
typedef MapDictEntry::const_iterator gdcm::Dict::ConstIterator
```

### 10.85.2.2 Iterator

```
typedef MapDictEntry::iterator gdcm::Dict::Iterator
```

### 10.85.2.3 MapDictEntry

```
typedef std::map<Tag, DictEntry> gdcm::Dict::MapDictEntry
```

## 10.85.3 Constructor & Destructor Documentation

### 10.85.3.1 Dict() [1/2]

```
gdcm::Dict::Dict ( ) [inline]
```

### 10.85.3.2 Dict() [2/2]

```
gdcmm::Dict::Dict (
    const Dict & _val ) [delete]
```

References [gdcmm::operator<<\(\)](#).

## 10.85.4 Member Function Documentation

### 10.85.4.1 AddDictEntry()

```
void gdcmm::Dict::AddDictEntry (
    const Tag & tag,
    const DictEntry & de ) [inline]
```

### 10.85.4.2 Begin()

```
ConstIterator gdcmm::Dict::Begin ( ) const [inline]
```

#### Examples

[GenAllVR.cxx](#), and [GenFakeIdentifyFile.cxx](#).

### 10.85.4.3 End()

```
ConstIterator gdcmm::Dict::End ( ) const [inline]
```

#### Examples

[GenAllVR.cxx](#), and [GenFakeIdentifyFile.cxx](#).

### 10.85.4.4 GetDictEntry()

```
const DictEntry & gdcmm::Dict::GetDictEntry (
    const Tag & tag ) const [inline]
```

#### Examples

[GenFakeIdentifyFile.cxx](#), and [PublicDict.cxx](#).



#### 10.85.4.5 GetDictEntryByKeyword()

```
const DictEntry & gdcm::Dict::GetDictEntryByKeyword (
    const char * keyword,
    Tag & tag ) const [inline]
```

Lookup [DictEntry](#) by keyword. Even if DICOM standard defines keyword as being unique. The lookup table is built on [Tag](#). Therefore looking up a [DictEntry](#) by Keyword is more inefficient than looking up by [Tag](#).

#### 10.85.4.6 GetDictEntryByName()

```
const DictEntry & gdcm::Dict::GetDictEntryByName (
    const char * name,
    Tag & tag ) const [inline]
```

Inefficient way of looking up tag by name. Technically DICOM does not guarantee uniqueness (and [Curve](#) / [Overlay](#) are there to prove it). But most of the time name is in fact uniq and can be uniquely link to a tag

#### Examples

[ReadAndPrintAttributes.cxx](#).

#### 10.85.4.7 GetKeywordFromTag()

```
const char * gdcm::Dict::GetKeywordFromTag (
    Tag const & tag ) const [inline]
```

Function to return the Keyword from a [Tag](#).

#### 10.85.4.8 IsEmpty()

```
bool gdcm::Dict::IsEmpty ( ) const [inline]
```

#### 10.85.4.9 LoadDefault()

```
void gdcm::Dict::LoadDefault ( ) [protected]
```

#### 10.85.4.10 operator=()

```
Dict & gdcm::Dict::operator= (
    const Dict & _val ) [delete]
```

## 10.85.5 Friends And Related Symbol Documentation

### 10.85.5.1 Dicts

```
friend class Dicts [friend]
```

### 10.85.5.2 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const Dict & _val ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmDict.h](#)

## 10.86 gdcm::DictConverter Class Reference

Class to convert a .dic file into something else:

```
#include <gdcmDictConverter.h>
```

### Public Types

- enum [OutputTypes](#) {  
[DICT\\_DEFAULT](#) = 0 ,  
[DICT\\_DEBUG](#) ,  
[DICT\\_XML](#) }

### Public Member Functions

- [DictConverter](#) ()
- [~DictConverter](#) ()
- void [Convert](#) ()
- const std::string & [GetDictName](#) () const
- const std::string & [GetInputFilename](#) () const
- const std::string & [GetOutputFilename](#) () const
- int [GetOutputType](#) () const
- void [SetDictName](#) (const char \*name)
- void [SetInputFileName](#) (const char \*filename)
- void [SetOutputFileName](#) (const char \*filename)
- void [SetOutputType](#) (int type)

### Static Public Member Functions

- static bool [Readuint16](#) (const char \*raw, uint16\_t &ov)
- static bool [ReadVM](#) (const char \*raw, [VM::VMType](#) &type)
- static bool [ReadVR](#) (const char \*raw, [VR::VRType](#) &type)

### Protected Member Functions

- void [AddGroupLength](#) ()
- bool [ConvertToCXX](#) (const char \*raw, std::string &cxx)
- bool [ConvertToXML](#) (const char \*raw, std::string &cxx)
- void [WriteFooter](#) ()
- void [WriteHeader](#) ()

## 10.86.1 Detailed Description

Class to convert a .dic file into something else:

- CXX code : embed dict into shared lib (DICT\_DEFAULT)
- Debug mode (DICT\_DEBUG)
- XML dict (DICT\_XML)

Note

## 10.86.2 Member Enumeration Documentation

### 10.86.2.1 OutputTypes

```
enum gdcmm::DictConverter::OutputTypes
```

Enumerator

DICT_DEFAULT	
DICT_DEBUG	
DICT_XML	

## 10.86.3 Constructor & Destructor Documentation

### 10.86.3.1 DictConverter()

```
gdcmm::DictConverter::DictConverter ( )
```

### 10.86.3.2 ~DictConverter()

```
gdcM::DictConverter::~~DictConverter ( )
```

## 10.86.4 Member Function Documentation

### 10.86.4.1 AddGroupLength()

```
void gdcM::DictConverter::AddGroupLength ( ) [protected]
```

### 10.86.4.2 Convert()

```
void gdcM::DictConverter::Convert ( )
```

### 10.86.4.3 ConvertToCXX()

```
bool gdcM::DictConverter::ConvertToCXX (
    const char * raw,
    std::string & cxx ) [protected]
```

### 10.86.4.4 ConvertToXML()

```
bool gdcM::DictConverter::ConvertToXML (
    const char * raw,
    std::string & cxx ) [protected]
```

### 10.86.4.5 GetDictName()

```
const std::string & gdcM::DictConverter::GetDictName ( ) const
```

### 10.86.4.6 GetInputFilename()

```
const std::string & gdcM::DictConverter::GetInputFilename ( ) const
```

### 10.86.4.7 GetOutputFilename()

```
const std::string & gdcM::DictConverter::GetOutputFilename ( ) const
```

#### 10.86.4.8 GetOutputType()

```
int gdcm::DictConverter::GetOutputType ( ) const [inline]
```

#### 10.86.4.9 Readuint16()

```
static bool gdcm::DictConverter::Readuint16 (
    const char * raw,
    uint16_t & ov ) [static]
```

#### 10.86.4.10 ReadVM()

```
static bool gdcm::DictConverter::ReadVM (
    const char * raw,
    VM::VMType & type ) [static]
```

#### 10.86.4.11 ReadVR()

```
static bool gdcm::DictConverter::ReadVR (
    const char * raw,
    VR::VRType & type ) [static]
```

#### 10.86.4.12 SetDictName()

```
void gdcm::DictConverter::SetDictName (
    const char * name )
```

#### 10.86.4.13 SetInputFileName()

```
void gdcm::DictConverter::SetInputFileName (
    const char * filename )
```

#### 10.86.4.14 SetOutputFileName()

```
void gdcm::DictConverter::SetOutputFileName (
    const char * filename )
```

#### 10.86.4.15 SetOutputType()

```
void gdcm::DictConverter::SetOutputType (
    int type ) [inline]
```

#### 10.86.4.16 WriteFooter()

```
void gdcM::DictConverter::WriteFooter ( ) [protected]
```

#### 10.86.4.17 WriteHeader()

```
void gdcM::DictConverter::WriteHeader ( ) [protected]
```

The documentation for this class was generated from the following file:

- [gdcMDictConverter.h](#)

## 10.87 gdcM::DictEntry Class Reference

Class to represent an Entry in the [Dict](#).

```
#include <gdcMDictEntry.h>
```

### Public Member Functions

- [DictEntry](#) (const char \*name="", const char \*keyword="", [VR](#) const &vr=[VR::INVALID](#), [VM](#) const &vm=[VM::VM0](#), bool ret=false)
- const char \* [GetKeyword](#) () const  
*same as GetName but without spaces...*
- const char \* [GetName](#) () const  
*Set/Get Name.*
- bool [GetRetired](#) () const  
*Set/Get Retired flag.*
- const [VM](#) & [GetVM](#) () const  
*Set/Get VM.*
- const [VR](#) & [GetVR](#) () const  
*Set/Get VR.*
- bool [IsUnique](#) () const
- void [SetElementXX](#) (bool v)  
*Set whether element is shared in multiple elements (Source [Image](#) IDs typically)*
- void [SetGroupXX](#) (bool v)  
*Set whether element is shared in multiple groups (Curve/Overlay typically)*
- void [SetKeyword](#) (const char \*keyword)
- void [SetName](#) (const char \*name)
- void [SetRetired](#) (bool retired)
- void [SetVM](#) ([VM](#) const &vm)
- void [SetVR](#) (const [VR](#) &vr)

## Friends

- class [Dict](#)
- `std::ostream & operator<< (std::ostream &_os, const DictEntry &_val)`

## 10.87.1 Detailed Description

Class to represent an Entry in the [Dict](#).

Does not really exist within the DICOM definition, just a way to minimize storage and have a mapping from [gdcmm::Tag](#) to the needed information

### Note

bla TODO FIXME: Need a PublicDictEntry...indeed [DictEntry](#) has a notion of retired which does not exist in PrivateDictEntry...

### See also

[gdcmm::Dict](#)

### Examples

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [PublicDict.cxx](#), and [TraverseModules.cxx](#).

## 10.87.2 Constructor & Destructor Documentation

### 10.87.2.1 DictEntry()

```
gdcmm::DictEntry::DictEntry (
    const char * name = "",
    const char * keyword = "",
    VR const & vr = VR::INVALID,
    VM const & vm = VM::VMO,
    bool ret = false ) [inline]
```

## 10.87.3 Member Function Documentation

### 10.87.3.1 GetKeyword()

```
const char * gdcmm::DictEntry::GetKeyword ( ) const [inline]
```

same as GetName but without spaces...

### 10.87.3.2 GetName()

```
const char * gdcM::DictEntry::GetName ( ) const [inline]
```

Set/Get Name.

Referenced by [gdcM::PrivateDict::PrintXML\(\)](#).

### 10.87.3.3 GetRetired()

```
bool gdcM::DictEntry::GetRetired ( ) const [inline]
```

Set/Get Retired flag.

Examples

[GenAllVR.cxx](#).

### 10.87.3.4 GetVM()

```
const VM & gdcM::DictEntry::GetVM ( ) const [inline]
```

Set/Get VM.

Referenced by [gdcM::PrivateDict::AddDictEntry\(\)](#), and [gdcM::PrivateDict::PrintXML\(\)](#).

### 10.87.3.5 GetVR()

```
const VR & gdcM::DictEntry::GetVR ( ) const [inline]
```

Set/Get VR.

Examples

[GenAllVR.cxx](#), and [GenFakeIdentifyFile.cxx](#).

Referenced by [gdcM::PrivateDict::AddDictEntry\(\)](#), and [gdcM::PrivateDict::PrintXML\(\)](#).

### 10.87.3.6 IsUnique()

```
bool gdcM::DictEntry::IsUnique ( ) const [inline]
```

Return whether the name of the [DataElement](#) can be considered to be unique. As of 2008 all elements name were unique (except the explicitly 'XX' ones)



### 10.87.3.7 SetElementXX()

```
void gdcm::DictEntry::SetElementXX (
    bool v ) [inline]
```

Set whether element is shared in multiple elements (Source [Image](#) IDs typically)

### 10.87.3.8 SetGroupXX()

```
void gdcm::DictEntry::SetGroupXX (
    bool v ) [inline]
```

Set whether element is shared in multiple groups (Curve/Overlay typically)

### 10.87.3.9 SetKeyword()

```
void gdcm::DictEntry::SetKeyword (
    const char * keyword ) [inline]
```

### 10.87.3.10 SetName()

```
void gdcm::DictEntry::SetName (
    const char * name ) [inline]
```

### 10.87.3.11 SetRetired()

```
void gdcm::DictEntry::SetRetired (
    bool retired ) [inline]
```

### 10.87.3.12 SetVM()

```
void gdcm::DictEntry::SetVM (
    VM const & vm ) [inline]
```

Referenced by [gdcm::PrivateDict::AddDictEntry\(\)](#).

### 10.87.3.13 SetVR()

```
void gdcm::DictEntry::SetVR (
    const VR & vr ) [inline]
```

Referenced by [gdcm::PrivateDict::AddDictEntry\(\)](#).

## 10.87.4 Friends And Related Symbol Documentation

### 10.87.4.1 Dict

```
friend class Dict [friend]
```

### 10.87.4.2 operator<<

```
std::ostream & operator<< (  
    std::ostream & _os,  
    const DictEntry & _val ) [friend]
```

The documentation for this class was generated from the following file:

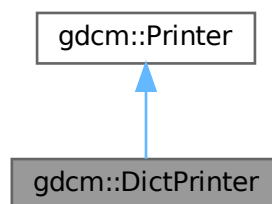
- [gdcmDictEntry.h](#)

## 10.88 gdcm::DictPrinter Class Reference

[DictPrinter](#) class.

```
#include <gdcmDictPrinter.h>
```

Inheritance diagram for `gdcm::DictPrinter`:



Collaboration diagram for gdcmm::DictPrinter:



### Public Member Functions

- [DictPrinter](#) ()
- [~DictPrinter](#) ()=default
- void [Print](#) (std::ostream &os)

### Public Member Functions inherited from [gdcmm::Printer](#)

- [Printer](#) ()
- [~Printer](#) ()=default
- [PrintStyles GetPrintStyle](#) () const  
*Get PrintStyle value.*
- void [Print](#) (std::ostream &os)  
*Print.*
- void [PrintDataSet](#) (const [DataSet](#) &ds, std::ostream &os, const std::string &s="")  
*Print an individual dataset.*
- void [SetColor](#) (bool c)  
*Set color mode or not.*
- void [SetFile](#) ([File](#) const &f)  
*Set file.*
- void [SetStyle](#) ([PrintStyles](#) ps)  
*Set PrintStyle value.*

### Protected Member Functions

- void [PrintDataElement2](#) (std::ostream &os, const [DataSet](#) &ds, const [DataElement](#) &ide)
- void [PrintDataSet2](#) (std::ostream &os, const [DataSet](#) &ds)

### Protected Member Functions inherited from [gdcm::Printer](#)

- [VR PrintDataElement](#) (std::ostream &os, const [Dicts](#) &dicts, const [DataSet](#) &ds, const [DataElement](#) &de, std::ostream &out, std::string const &indent)
- void [PrintSQ](#) (const [SequenceOfItems](#) \*sqi, std::ostream &os, std::string const &indent)

### Additional Inherited Members

### Public Types inherited from [gdcm::Printer](#)

- enum [PrintStyles](#) {  
[VERBOSE\\_STYLE](#) = 0 ,  
[CONDENSED\\_STYLE](#) ,  
[XML](#) ,  
[CXX](#) }

### Protected Attributes inherited from [gdcm::Printer](#)

- const [File](#) \* [F](#)
- [VL MaxPrintLength](#)
- [PrintStyles](#) [PrintStyle](#)

## 10.88.1 Detailed Description

[DictPrinter](#) class.

## 10.88.2 Constructor & Destructor Documentation

### 10.88.2.1 DictPrinter()

```
gdcm::DictPrinter::DictPrinter ( )
```

### 10.88.2.2 ~DictPrinter()

```
gdcm::DictPrinter::~DictPrinter ( ) [default]
```

## 10.88.3 Member Function Documentation

### 10.88.3.1 Print()

```
void gdcm::DictPrinter::Print (
    std::ostream & os )
```

### 10.88.3.2 PrintDataElement2()

```
void gdcm::DictPrinter::PrintDataElement2 (
    std::ostream & os,
    const DataSet & ds,
    const DataElement & ide ) [protected]
```

### 10.88.3.3 PrintDataSet2()

```
void gdcm::DictPrinter::PrintDataSet2 (
    std::ostream & os,
    const DataSet & ds ) [protected]
```

The documentation for this class was generated from the following file:

- [gdcmDictPrinter.h](#)

## 10.89 gdcm::Dicts Class Reference

Class to manipulate the sum of knowledge (all the dict user load)

```
#include <gdcmDicts.h>
```

### Public Member Functions

- [Dicts](#) ()
- [Dicts](#) (const [Dicts](#) &\_val)=delete
- [~Dicts](#) ()
- const [CSAHeaderDict](#) & [GetCSAHeaderDict](#) () const
- const [DictEntry](#) & [GetDictEntry](#) (const [PrivateTag](#) &tag) const
- const [DictEntry](#) & [GetDictEntry](#) (const [Tag](#) &tag, const char \*owner=nullptr) const
- *THREAD SAFE.*
- [PrivateDict](#) & [GetPrivateDict](#) ()
- const [PrivateDict](#) & [GetPrivateDict](#) () const
- const [Dict](#) & [GetPublicDict](#) () const
- bool [IsEmpty](#) () const
- [Dicts](#) & [operator=](#) (const [Dicts](#) &\_val)=delete

### Protected Types

- enum [ConstructorType](#) {  
    [PHILIPS](#) ,  
    [GEMS](#) ,  
    [SIEMENS](#) }

### Protected Member Functions

- void [LoadDefaults](#) ()

### Static Protected Member Functions

- static const char \* [GetConstructorString](#) ([ConstructorType](#) type)

### Friends

- class [Global](#)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [Dicts](#) &d)

## 10.89.1 Detailed Description

Class to manipulate the sum of knowledge (all the dict user load)

#### Note

bla

#### Examples

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [PublicDict.cxx](#), [ReadAndPrintAttributes.cxx](#), and [TraverseModules.cxx](#).

## 10.89.2 Member Enumeration Documentation

### 10.89.2.1 ConstructorType

```
enum gdcmm::Dicts::ConstructorType [protected]
```

#### Enumerator

PHILIPS	
GEMS	
SIEMENS	

## 10.89.3 Constructor & Destructor Documentation

### 10.89.3.1 Dicts() [1/2]

```
gdcm::Dicts::Dicts ( )
```

### 10.89.3.2 ~Dicts()

```
gdcm::Dicts::~~Dicts ( )
```

### 10.89.3.3 Dicts() [2/2]

```
gdcm::Dicts::Dicts (
    const Dicts & _val ) [delete]
```

## 10.89.4 Member Function Documentation

### 10.89.4.1 GetConstructorString()

```
static const char * gdcm::Dicts::GetConstructorString (
    ConstructorType type ) [static], [protected]
```

### 10.89.4.2 GetCSAHeaderDict()

```
const CSAHeaderDict & gdcm::Dicts::GetCSAHeaderDict ( ) const
```

#### Examples

[MrProtocol.cxx](#).

### 10.89.4.3 GetDictEntry() [1/2]

```
const DictEntry & gdcm::Dicts::GetDictEntry (
    const PrivateTag & tag ) const
```

#### 10.89.4.4 GetDictEntry() [2/2]

```
const DictEntry & gdc::Dicts::GetDictEntry (
    const Tag & tag,
    const char * owner = nullptr ) const
```

THREAD SAFE.

works for both public and private dicts: owner is null for public dict

##### Warning

owner need to be set to appropriate owner for call to work. see

##### Examples

[PublicDict.cxx](#), and [TraverseModules.cxx](#).

#### 10.89.4.5 GetPrivateDict() [1/2]

```
PrivateDict & gdc::Dicts::GetPrivateDict ( )
```

#### 10.89.4.6 GetPrivateDict() [2/2]

```
const PrivateDict & gdc::Dicts::GetPrivateDict ( ) const
```

#### 10.89.4.7 GetPublicDict()

```
const Dict & gdc::Dicts::GetPublicDict ( ) const
```

##### Examples

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [PublicDict.cxx](#), and [ReadAndPrintAttributes.cxx](#).

#### 10.89.4.8 IsEmpty()

```
bool gdc::Dicts::IsEmpty ( ) const [inline]
```

#### 10.89.4.9 LoadDefaults()

```
void gdc::Dicts::LoadDefaults ( ) [protected]
```



#### 10.89.4.10 operator=()

```
Dicts & gdcm::Dicts::operator= (
    const Dicts & _val ) [delete]
```

### 10.89.5 Friends And Related Symbol Documentation

#### 10.89.5.1 Global

```
friend class Global [friend]
```

#### 10.89.5.2 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const Dicts & d ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmDicts.h](#)

## 10.90 gdcm::network::DIMSE Class Reference

[DIMSE.](#)

```
#include <gdcmDIMSE.h>
```

### Public Types

- enum [CommandTypes](#) {
  - [C\\_STORE\\_RQ](#) = 0x0001 ,
  - [C\\_STORE\\_RSP](#) = 0x8001 ,
  - [C\\_GET\\_RQ](#) = 0x0010 ,
  - [C\\_GET\\_RSP](#) = 0x8010 ,
  - [C\\_FIND\\_RQ](#) = 0x0020 ,
  - [C\\_FIND\\_RSP](#) = 0x8020 ,
  - [C\\_MOVE\\_RQ](#) = 0x0021 ,
  - [C\\_MOVE\\_RSP](#) = 0x8021 ,
  - [C\\_ECHO\\_RQ](#) = 0x0030 ,
  - [C\\_ECHO\\_RSP](#) = 0x8030 ,
  - [N\\_EVENT\\_REPORT\\_RQ](#) = 0x0100 ,
  - [N\\_EVENT\\_REPORT\\_RSP](#) = 0x8100 ,
  - [N\\_GET\\_RQ](#) = 0x0110 ,
  - [N\\_GET\\_RSP](#) = 0x8110 ,
  - [N\\_SET\\_RQ](#) = 0x0120 ,
  - [N\\_SET\\_RSP](#) = 0x8120 ,
  - [N\\_ACTION\\_RQ](#) = 0x0130 ,
  - [N\\_ACTION\\_RSP](#) = 0x8130 ,
  - [N\\_CREATE\\_RQ](#) = 0x0140 ,
  - [N\\_CREATE\\_RSP](#) = 0x8140 ,
  - [N\\_DELETE\\_RQ](#) = 0x0150 ,
  - [N\\_DELETE\\_RSP](#) = 0x8150 ,
  - [C\\_CANCEL\\_RQ](#) = 0x0FFF }

### 10.90.1 Detailed Description

[DIMSE.](#)

PS 3.7 - 2009 Annex E [Command](#) Dictionary (Normative) E.1 REGISTRY OF DICOM COMMAND ELEMENTS [Table](#)  
E.1-1 COMMAND FIELDS (PART 1)

### 10.90.2 Member Enumeration Documentation

#### 10.90.2.1 CommandTypes

```
enum gdcm::network::DIMSE::CommandTypes
```

Enumerator

C_STORE_RQ	
C_STORE_RSP	
C_GET_RQ	
C_GET_RSP	
C_FIND_RQ	
C_FIND_RSP	
C_MOVE_RQ	
C_MOVE_RSP	
C_ECHO_RQ	
C_ECHO_RSP	
N_EVENT_REPORT_RQ	
N_EVENT_REPORT_RSP	
N_GET_RQ	
N_GET_RSP	
N_SET_RQ	
N_SET_RSP	
N_ACTION_RQ	
N_ACTION_RSP	
N_CREATE_RQ	
N_CREATE_RSP	
N_DELETE_RQ	
N_DELETE_RSP	
C_CANCEL_RQ	

The documentation for this class was generated from the following file:

- [gdcmDIMSE.h](#)

## 10.91 [gdcm::DirectionCosines](#) Class Reference

class to handle [DirectionCosines](#)

```
#include <gdcmDirectionCosines.h>
```

## Public Member Functions

- [DirectionCosines](#) ()
- [DirectionCosines](#) (const double dircos[6])
- [~DirectionCosines](#) ()
- double [ComputeDistAlongNormal](#) (const double ipp[3]) const  
*Compute the distance along the normal.*
- void [Cross](#) (double z[3]) const  
*Compute Cross product.*
- double [CrossDot](#) ([DirectionCosines](#) const &dc) const  
*Compute the Dot product of the two cross vector of both [DirectionCosines](#) object.*
- double [Dot](#) () const  
*Compute Dot.*
- bool [IsValid](#) () const  
*Return whether or not this is a valid direction cosines.*
- void [Normalize](#) ()  
*Normalize in-place.*
- [operator const double \\*](#) () const  
*Make the class behave like a const double \*.*
- void [Print](#) (std::ostream &) const  
*Print.*
- bool [SetFromString](#) (const char \*str)

## Static Public Member Functions

- static double [Dot](#) (const double x[3], const double y[3])  
*Compute Dot.*
- static double [Norm](#) (const double v[3])  
*Return norm of the vector.*
- static void [Normalize](#) (double v[3])  
*Normalize in-place.*

### 10.91.1 Detailed Description

class to handle [DirectionCosines](#)

#### Examples

[DiscriminateVolume.cxx](#).

### 10.91.2 Constructor & Destructor Documentation

#### 10.91.2.1 [DirectionCosines\(\)](#) [1/2]

```
gdcm::DirectionCosines::DirectionCosines ( )
```

### 10.91.2.2 DirectionCosines() [2/2]

```
gdc::DirectionCosines::DirectionCosines (
    const double dircos[6] )
```

### 10.91.2.3 ~DirectionCosines()

```
gdc::DirectionCosines::~~DirectionCosines ( )
```

## 10.91.3 Member Function Documentation

### 10.91.3.1 ComputeDistAlongNormal()

```
double gdc::DirectionCosines::ComputeDistAlongNormal (
    const double ipp[3] ) const
```

Compute the distance along the normal.

### 10.91.3.2 Cross()

```
void gdc::DirectionCosines::Cross (
    double z[3] ) const
```

Compute Cross product.

### 10.91.3.3 CrossDot()

```
double gdc::DirectionCosines::CrossDot (
    DirectionCosines const & dc ) const
```

Compute the Dot product of the two cross vector of both [DirectionCosines](#) object.

#### Examples

[DiscriminateVolume.cxx](#).

### 10.91.3.4 Dot() [1/2]

```
double gdc::DirectionCosines::Dot ( ) const
```

Compute Dot.

### 10.91.3.5 Dot() [2/2]

```
static double gdcM::DirectionCosines::Dot (
    const double x[3],
    const double y[3] ) [static]
```

Compute Dot.

### 10.91.3.6 IsValid()

```
bool gdcM::DirectionCosines::IsValid ( ) const
```

Return whether or not this is a valid direction cosines.

### 10.91.3.7 Norm()

```
static double gdcM::DirectionCosines::Norm (
    const double v[3] ) [static]
```

Return norm of the vector.

### 10.91.3.8 Normalize() [1/2]

```
void gdcM::DirectionCosines::Normalize ( )
```

Normalize in-place.

### 10.91.3.9 Normalize() [2/2]

```
static void gdcM::DirectionCosines::Normalize (
    double v[3] ) [static]
```

Normalize in-place.

### 10.91.3.10 operator const double \*()

```
gdcM::DirectionCosines::operator const double * ( ) const [inline]
```

Make the class behave like a const double \*.

### 10.91.3.11 Print()

```
void gdcM::DirectionCosines::Print (
    std::ostream & ) const
```

Print.

### 10.91.3.12 SetFromString()

```
bool gdcm::DirectionCosines::SetFromString (
    const char * str )
```

Initialize from string str. It requires 6 floating point separated by a backslash character.

#### Examples

[DiscriminateVolume.cxx](#).

The documentation for this class was generated from the following file:

- [gdcmDirectionCosines.h](#)

## 10.92 gdcm::Directory Class Reference

Class for manipulation directories.

```
#include <gdcmDirectory.h>
```

### Public Types

- typedef std::vector< [FilenameType](#) > [FileNamesType](#)
- typedef std::string [FilenameType](#)

### Public Member Functions

- [Directory](#) ()=default
- [~Directory](#) ()=default
- [FileNamesType](#) const & [GetDirectories](#) () const  
*Return the Directories traversed.*
- [FileNamesType](#) const & [GetFileNames](#) () const  
*Set/Get the file names within the directory.*
- [FilenameType](#) const & [GetToplevel](#) () const  
*Get the name of the toplevel directory.*
- unsigned int [Load](#) ([FilenameType](#) const &name, bool recursive=false)
- void [Print](#) (std::ostream &os=std::cout) const  
*Print.*

### Protected Member Functions

- unsigned int [Explore](#) ([FilenameType](#) const &name, bool recursive)  
*Return number of file found when 'recursive'ly exploring directory name*

## Friends

- `std::ostream & operator<< (std::ostream &_os, const Directory &d)`

## 10.92.1 Detailed Description

Class for manipulation directories.

### Note

This implementation provide a cross platform implementation for manipulating directories: basically traversing directories and harvesting files

will not take into account unix type hidden file recursive option will not look into UNIX type hidden directory (those starting with a '.')

Since python or C# provide there own equivalent implementation, in which case [gdcm::Directory](#) does not make much sense.

### Examples

[ClinicalTrialIdentificationWorkflow.cs](#), [DecompressImageMultiframe.cs](#), [DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [DumpVisusChange.cxx](#), [GenerateDICOMDIR.cs](#), [GenerateRTSTRUCT.cxx](#), [ReadUTF8QtDir.cxx](#), [ScanDirectory.cs](#), [SortImage.cxx](#), [StandardizeFiles.cs](#), [VolumeSorter.cxx](#), [gdcmorthoplanes.cxx](#), [reslicesphere.cxx](#), and [threadgdcm.cxx](#).

## 10.92.2 Member Typedef Documentation

### 10.92.2.1 FilenamesType

```
typedef std::vector<FilenameType> gdcm::Directory::FilenamesType
```

### 10.92.2.2 FilenameType

```
typedef std::string gdcm::Directory::FilenameType
```

## 10.92.3 Constructor & Destructor Documentation

### 10.92.3.1 Directory()

```
gdcm::Directory::Directory ( ) [default]
```

### 10.92.3.2 ~Directory()

```
gdcm::Directory::~~Directory ( ) [default]
```

## 10.92.4 Member Function Documentation

### 10.92.4.1 Explore()

```
unsigned int gdcm::Directory::Explore (
    FilenameType const & name,
    bool recursive ) [protected]
```

Return number of file found when 'recursive'ly exploring directory name

### 10.92.4.2 GetDirectories()

```
FilenameType const & gdcm::Directory::GetDirectories ( ) const [inline]
```

Return the Directories traversed.

### 10.92.4.3 GetFileNames()

```
FilenameType const & gdcm::Directory::GetFileNames ( ) const [inline]
```

Set/Get the file names within the directory.

#### Examples

[ClinicalTrialIdentificationWorkflow.cs](#), [DecompressImageMultiframe.cs](#), [DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [DumpVisusChange.cxx](#), [GenerateDICOMDIR.cs](#), [GenerateRTSTRUCT.cxx](#), [ReadUTF8QtDir.cxx](#), [ScanDirectory.cs](#), [SortImage.cxx](#), [StandardizeFiles.cs](#), [VolumeSorter.cxx](#), [gdcmorthoplanes.cxx](#), [reslicesphere.cxx](#), and [threadgdcm.cxx](#).

### 10.92.4.4 GetToplevel()

```
FilenameType const & gdcm::Directory::GetToplevel ( ) const [inline]
```

Get the name of the toplevel directory.

### 10.92.4.5 Load()

```
unsigned int gdcm::Directory::Load (
    FilenameType const & name,
    bool recursive = false )
```

construct a list of filenames and subdirectory beneath directory: name

#### Warning

: hidden file and hidden directory are not loaded.

#### Examples

[ClinicalTrialIdentificationWorkflow.cs](#), [DecompressImageMultiframe.cs](#), [DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [DumpVisusChange.cxx](#), [GenerateDICOMDIR.cs](#), [GenerateRTSTRUCT.cxx](#), [ReadUTF8QtDir.cxx](#), [ScanDirectory.cs](#), [SortImage.cxx](#), [StandardizeFiles.cs](#), [VolumeSorter.cxx](#), [gdcmorthoplanes.cxx](#), [reslicesphere.cxx](#), and [threadgdcm.cxx](#).



#### 10.92.4.6 Print()

```
void gdcm::Directory::Print (
    std::ostream & os = std::cout ) const
```

Print.

#### Examples

[SortImage.cxx](#).

### 10.92.5 Friends And Related Symbol Documentation

#### 10.92.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const Directory & d ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmDirectory.h](#)

## 10.93 gdcm::DirectoryHelper Class Reference

[DirectoryHelper](#).

```
#include <gdcmDirectoryHelper.h>
```

#### Static Public Member Functions

- static [Directory::FilenameType GetCTImageSeriesUIDs](#) (const std::string &inDirectory)
- static [Directory::FilenameType GetFileNamesFromSeriesUIDs](#) (const std::string &inDirectory, const std::string &inSeriesUID)
- static std::string [GetFrameOfReference](#) (const std::vector< [DataSet](#) > &inDS)
- static [Directory::FilenameType GetMRImageSeriesUIDs](#) (const std::string &inDirectory)
- static [Directory::FilenameType GetRTStructSeriesUIDs](#) (const std::string &inDirectory)
- static [Directory::FilenameType GetSeriesUIDsBySOPClassUID](#) (const std::string &inDirectory, const std::string &inSOPClassUID)
- static std::string [GetSOPClassUID](#) (const std::vector< [DataSet](#) > &inDS)
- static std::string [GetStringValueFromTag](#) (const [Tag](#) &t, const [DataSet](#) &ds)
- static std::vector< [DataSet](#) > [LoadImageFromFiles](#) (const std::string &inDirectory, const std::string &inSeriesUID)
- static std::string [RetrieveSOPInstanceUIDFromIndex](#) (int inIndex, const std::vector< [DataSet](#) > &inDS)
- static std::string [RetrieveSOPInstanceUIDFromZPosition](#) (double inZPos, const std::vector< [DataSet](#) > &inDS)

### 10.93.1 Detailed Description

[DirectoryHelper](#).

this class is designed to help mitigate some of the commonly performed operations on directories. namely: 1) the ability to determine the number of series in a directory by what type of series is present 2) the ability to find all ct series in a directory 3) the ability to find all mr series in a directory 4) to load a set of DataSets from a series that's already been sorted by the IPP sorter 5) For rtstruct stuff, you need to know the sopinstanceuid of each z plane, so there's a retrieval function for that 6) then a few other functions for rtstruct writeouts

### 10.93.2 Member Function Documentation

#### 10.93.2.1 GetCTImageSeriesUIDs()

```
static Directory::FileNamesType gdcm::DirectoryHelper::GetCTImageSeriesUIDs (
    const std::string & inDirectory ) [static]
```

#### 10.93.2.2 GetFileNamesFromSeriesUIDs()

```
static Directory::FileNamesType gdcm::DirectoryHelper::GetFileNamesFromSeriesUIDs (
    const std::string & inDirectory,
    const std::string & inSeriesUID ) [static]
```

#### 10.93.2.3 GetFrameOfReference()

```
static std::string gdcm::DirectoryHelper::GetFrameOfReference (
    const std::vector< DataSet > & inDS ) [static]
```

#### 10.93.2.4 GetMRImageSeriesUIDs()

```
static Directory::FileNamesType gdcm::DirectoryHelper::GetMRImageSeriesUIDs (
    const std::string & inDirectory ) [static]
```

#### 10.93.2.5 GetRTStructSeriesUIDs()

```
static Directory::FileNamesType gdcm::DirectoryHelper::GetRTStructSeriesUIDs (
    const std::string & inDirectory ) [static]
```

#### 10.93.2.6 GetSeriesUIDsBySOPClassUID()

```
static Directory::FileNamesType gdcm::DirectoryHelper::GetSeriesUIDsBySOPClassUID (
    const std::string & inDirectory,
    const std::string & inSOPClassUID ) [static]
```

### 10.93.2.7 GetSOPClassUID()

```
static std::string gdcm::DirectoryHelper::GetSOPClassUID (
    const std::vector< DataSet > & inDS ) [static]
```

### 10.93.2.8 GetStringValueFromTag()

```
static std::string gdcm::DirectoryHelper::GetStringValueFromTag (
    const Tag & t,
    const DataSet & ds ) [static]
```

### 10.93.2.9 LoadImageFromFiles()

```
static std::vector< DataSet > gdcm::DirectoryHelper::LoadImageFromFiles (
    const std::string & inDirectory,
    const std::string & inSeriesUID ) [static]
```

### 10.93.2.10 RetrieveSOPInstanceUIDFromIndex()

```
static std::string gdcm::DirectoryHelper::RetrieveSOPInstanceUIDFromIndex (
    int inIndex,
    const std::vector< DataSet > & inDS ) [static]
```

### 10.93.2.11 RetrieveSOPInstanceUIDFromZPosition()

```
static std::string gdcm::DirectoryHelper::RetrieveSOPInstanceUIDFromZPosition (
    double inZPos,
    const std::vector< DataSet > & inDS ) [static]
```

The documentation for this class was generated from the following file:

- [gdcmDirectoryHelper.h](#)

## 10.94 gdcm::DPath Class Reference

class to handle a DICOM path While supp 118 did introduced a notion of XPath for XML Native model this convention is too XML-centric. Instead prefer DCMTK style notation <https://groups.google.com/g/comp.protocols.dicom/c/IyIH0IOBMPA>

```
#include <gdcmDPath.h>
```

## Public Member Functions

- [DPath](#) ()
- [~DPath](#) ()
- bool [ConstructFromString](#) (const char \*path)
- bool [Match](#) ([DPath](#) const &other) const  
*Return whether or not 'other' match the template [DPath](#).*
- bool [operator<](#) (const [DPath](#) &rhs) const
- void [Print](#) (std::ostream &) const

## Static Public Member Functions

- static bool [IsValid](#) (const char \*path)  
*Return if path is valid or not.*

## Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [DPath](#) &\_val)

## 10.94.1 Detailed Description

class to handle a DICOM path While supp 118 did introduced a notion of XPath for XML Native model this convention is too XML-centric. Instead prefer DCMTK style notation [https://groups.google.com/g/comp.↵protocols.dicom/c/IyIH0IOBMPA](https://groups.google.com/g/comp.protocols.dicom/c/IyIH0IOBMPA)

### Examples

[Cleaner.cs](#).

## 10.94.2 Constructor & Destructor Documentation

### 10.94.2.1 DPath()

```
gdcm::DPath::DPath ( )
```

### 10.94.2.2 ~DPath()

```
gdcm::DPath::~~DPath ( )
```

## 10.94.3 Member Function Documentation

### 10.94.3.1 ConstructFromString()

```
bool gdcmm::DPath::ConstructFromString (
    const char * path )
```

#### Examples

[Cleaner.cs](#).

### 10.94.3.2 IsValid()

```
static bool gdcmm::DPath::IsValid (
    const char * path ) [static]
```

Return if path is valid or not.

### 10.94.3.3 Match()

```
bool gdcmm::DPath::Match (
    DPath const & other ) const
```

Return whether or not 'other' match the template [DPath](#).

### 10.94.3.4 operator<()

```
bool gdcmm::DPath::operator< (
    const DPath & rhs ) const
```

### 10.94.3.5 Print()

```
void gdcmm::DPath::Print (
    std::ostream & ) const
```

## 10.94.4 Friends And Related Symbol Documentation

### 10.94.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const DPath & _val ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmmDPath.h](#)

## 10.95 gdcm::DummyValueGenerator Class Reference

Class for generating dummy value.

```
#include <gdcmDummyValueGenerator.h>
```

### Static Public Member Functions

- static const char \* [Generate](#) (const char \*input)

### 10.95.1 Detailed Description

Class for generating dummy value.

See also

[Anonymizer](#)

### 10.95.2 Member Function Documentation

#### 10.95.2.1 Generate()

```
static const char * gdcm::DummyValueGenerator::Generate (  
    const char * input ) [static]
```

Generate a dummy value from an input value. This is guarantee to always return the same output value when input is identical. Return an array of bytes that can be used for anonymization purpose, return NULL on error NOT THREAD SAFE

The documentation for this class was generated from the following file:

- [gdcmDummyValueGenerator.h](#)

## 10.96 gdcm::Dumper Class Reference

[Codec](#) class.

```
#include <gdcmDumper.h>
```

Inheritance diagram for gdcm::Dumper:



Collaboration diagram for gdcm::Dumper:



### Public Member Functions

- [Dumper](#) ()
- [~Dumper](#) ()=default

## Public Member Functions inherited from [gdcm::Printer](#)

- [Printer](#) ()
- [~Printer](#) ()=default
- [PrintStyles GetPrintStyle](#) () const  
*Get PrintStyle value.*
- void [Print](#) (std::ostream &os)  
*Print.*
- void [PrintDataSet](#) (const [DataSet](#) &ds, std::ostream &os, const std::string &s="")  
*Print an individual dataset.*
- void [SetColor](#) (bool c)  
*Set color mode or not.*
- void [SetFile](#) ([File](#) const &f)  
*Set file.*
- void [SetStyle](#) ([PrintStyles](#) ps)  
*Set PrintStyle value.*

## Additional Inherited Members

## Public Types inherited from [gdcm::Printer](#)

- enum [PrintStyles](#) {  
    [VERBOSE\\_STYLE](#) = 0 ,  
    [CONDENSED\\_STYLE](#) ,  
    [XML](#) ,  
    [CXX](#) }

## Protected Member Functions inherited from [gdcm::Printer](#)

- [VR PrintDataElement](#) (std::ostream &os, const [Dicts](#) &dicts, const [DataSet](#) &ds, const [DataElement](#) &de, std::ostream &out, std::string const &indent)
- void [PrintSQ](#) (const [SequenceOfItems](#) \*sqi, std::ostream &os, std::string const &indent)

## Protected Attributes inherited from [gdcm::Printer](#)

- const [File](#) \* F
- [VL MaxPrintLength](#)
- [PrintStyles PrintStyle](#)

## 10.96.1 Detailed Description

[Codec](#) class.

### Note

Use it to simply dump value read from the file. No interpretation is done. But it is real fast ! Almost no overhead



## 10.96.2 Constructor & Destructor Documentation

### 10.96.2.1 Dumper()

```
gdcmm::Dumper::Dumper ( ) [inline]
```

### 10.96.2.2 ~Dumper()

```
gdcmm::Dumper::~~Dumper ( ) [default]
```

The documentation for this class was generated from the following file:

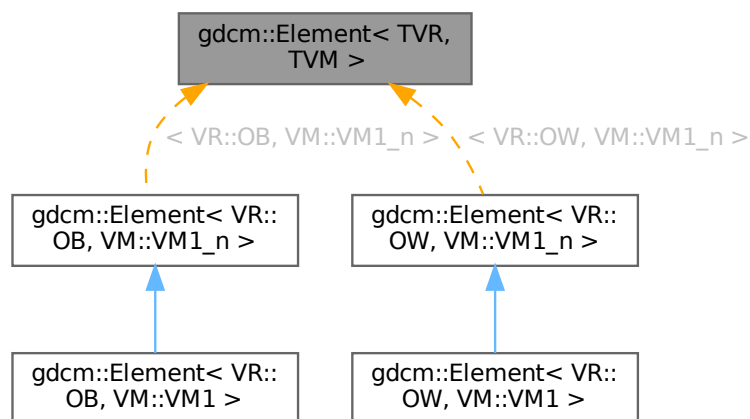
- [gdcmmDumper.h](#)

## 10.97 gdcmm::Element< TVR, TVM > Class Template Reference

[Element](#) class.

```
#include <gdcmmElement.h>
```

Inheritance diagram for gdcmm::Element< TVR, TVM >:



Collaboration diagram for `gdcm::Element< TVR, TVM >`:



## Public Types

- typedef `VRToType< TVR >::Type` `Type`

## Public Member Functions

- `DataElement GetAsDataElement ()` const
- unsigned long `GetLength ()` const
- `VRToType< TVR >::Type & GetValue` (unsigned int idx=0)
- const `VRToType< TVR >::Type & GetValue` (unsigned int idx=0) const
- const `VRToType< TVR >::Type * GetValues ()` const
- `VRToType< TVR >::Type operator[]` (unsigned int idx) const
- void `Print` (std::ostream &\_os) const
- void `Read` (std::istream &\_is)
- void `Set` (Value const &v)
- void `SetFromDataElement` (DataElement const &de)
- void `SetValue` (typename `VRToType< TVR >::Type` v, unsigned int idx=0)
- void `Write` (std::ostream &\_os) const

## Static Public Member Functions

- static `VM GetVM ()`
- static `VR GetVR ()`

## Public Attributes

- [VRToType](#)< TVR >::Type Internal [[VMToLength](#)< TVM >::Length]

## Protected Member Functions

- void [SetNoSwap](#) ([Value](#) const &v)

## 10.97.1 Detailed Description

```
template<long long TVR, int TVM>
class gdcm::Element< TVR, TVM >
```

[Element](#) class.

### Note

TODO

### Examples

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GetSubSequenceData.cxx](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

## 10.97.2 Member Typedef Documentation

### 10.97.2.1 Type

```
template<long long TVR, int TVM>
typedef VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Type
```

## 10.97.3 Member Function Documentation

### 10.97.3.1 GetAsDataElement()

```
template<long long TVR, int TVM>
DataElement gdcm::Element< TVR, TVM >::GetAsDataElement ( ) const [inline]
```

### Examples

[Extracting\\_All\\_Resolution.cxx](#), and [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#).

References [gdcm::DataElement::GetVR\(\)](#), [gdcm::DataElement::SetByteValue\(\)](#), and [gdcm::DataElement::SetVR\(\)](#).

### 10.97.3.2 GetLength()

```
template<long long TVR, int TVM>
unsigned long gdcm::Element< TVR, TVM >::GetLength ( ) const [inline]
```

#### Examples

[DumpGEMSMovieGroup.cxx](#).

### 10.97.3.3 GetValue() [1/2]

```
template<long long TVR, int TVM>
VRToType< TVR >::Type & gdcm::Element< TVR, TVM >::GetValue (
    unsigned int idx = 0 ) [inline]
```

### 10.97.3.4 GetValue() [2/2]

```
template<long long TVR, int TVM>
const VRToType< TVR >::Type & gdcm::Element< TVR, TVM >::GetValue (
    unsigned int idx = 0 ) const [inline]
```

#### Examples

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [GetSubSequenceData.cxx](#),  
and [csa2img.cxx](#).

### 10.97.3.5 GetValues()

```
template<long long TVR, int TVM>
const VRToType< TVR >::Type * gdcm::Element< TVR, TVM >::GetValues ( ) const [inline]
```

### 10.97.3.6 GetVM()

```
template<long long TVR, int TVM>
static VM gdcm::Element< TVR, TVM >::GetVM ( ) [inline], [static]
```

### 10.97.3.7 GetVR()

```
template<long long TVR, int TVM>
static VR gdcm::Element< TVR, TVM >::GetVR ( ) [inline], [static]
```

### 10.97.3.8 operator[]()

```
template<long long TVR, int TVM>
VRToType< TVR >::Type gdcm::Element< TVR, TVM >::operator[] (
    unsigned int idx ) const [inline]
```

### 10.97.3.9 Print()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::Print (
    std::ostream & _os ) const [inline]
```

#### Examples

[DumpGEMSMovieGroup.cxx](#).

### 10.97.3.10 Read()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::Read (
    std::istream & _is ) [inline]
```

### 10.97.3.11 Set()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::Set (
    Value const & v ) [inline]
```

#### Examples

[csa2img.cxx](#).

References [gdcm::ByteValue::GetLength\(\)](#), and [gdcm::ByteValue::GetPointer\(\)](#).

### 10.97.3.12 SetFromDataElement()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::SetFromDataElement (
    DataElement< TVR, TVM > const & de ) [inline]
```

#### Examples

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [GetSubSequenceData.cxx](#), and [iU22tomultisc.cxx](#).

References [gdcm::DataElement::GetByteValue\(\)](#), [gdcm::DataElement::GetValue\(\)](#), and [gdcm::DataElement::GetVR\(\)](#).

### 10.97.3.13 SetNoSwap()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::SetNoSwap (
    Value const & v ) [inline], [protected]
```

References [gdcm::ByteValue::GetLength\(\)](#), and [gdcm::ByteValue::GetPointer\(\)](#).

### 10.97.3.14 SetValue()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::SetValue (
    typename VRToType< TVR >::Type v,
    unsigned int idx = 0 ) [inline]
```

#### Examples

[Extracting\\_All\\_Resolution.cxx](#), and [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#).

### 10.97.3.15 Write()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::Write (
    std::ostream & _os ) const [inline]
```

## 10.97.4 Member Data Documentation

### 10.97.4.1 Internal

```
template<long long TVR, int TVM>
VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

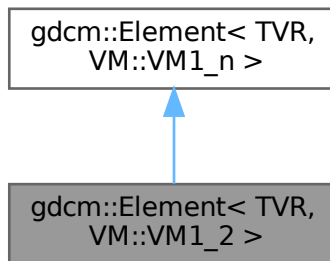
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

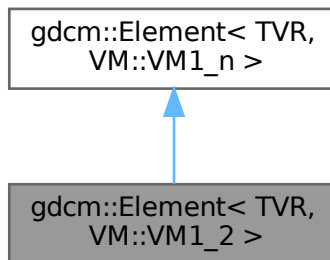
## 10.98 gdcm::Element< TVR, VM::VM1\_2 > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, VM::VM1\_2 >:



Collaboration diagram for gdcm::Element< TVR, VM::VM1\_2 >:



### Public Types

- typedef `Element< TVR, VM::VM1_n >` `Parent`

### Public Types inherited from `gdcm::Element< TVR, VM::VM1_n >`

- typedef `VRToType< TVR >::Type` `Type`

## Public Member Functions

- void [SetLength](#) (int len)

## Public Member Functions inherited from [gdcm::Element< TVR, VM::VM1\\_n >](#)

- [Element](#) ()
- [Element](#) (const [Element](#) &\_val)
- [~Element](#) ()
- [DataElement](#) [GetAsDataElement](#) () const
- unsigned long [GetLength](#) () const
- [VRToType< TVR >::Type](#) & [GetValue](#) (unsigned int idx=0)
- const [VRToType< TVR >::Type](#) & [GetValue](#) (unsigned int idx=0) const
- [Element](#) & [operator=](#) (const [Element](#) &\_val)
- [VRToType< TVR >::Type](#) [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &\_os) const
- void [Read](#) (std::istream &\_is)
- void [Set](#) ([Value](#) const &v)
- void [SetArray](#) (const [Type](#) \*array, unsigned long len, bool save=false)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetLength](#) (unsigned long len)
- void [SetValue](#) (typename [VRToType< TVR >::Type](#) v, unsigned int idx=0)
- void [Write](#) (std::ostream &\_os) const
- void [WriteASCII](#) (std::ostream &os) const

## Additional Inherited Members

## Static Public Member Functions inherited from [gdcm::Element< TVR, VM::VM1\\_n >](#)

- static [VM](#) [GetVM](#) ()
- static [VR](#) [GetVR](#) ()

## Protected Member Functions inherited from [gdcm::Element< TVR, VM::VM1\\_n >](#)

- void [SetNoSwap](#) ([Value](#) const &v)

## 10.98.1 Member Typedef Documentation

### 10.98.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM1\_n> gdcm::Element< TVR, VM::VM1\_2 >::Parent
```



## 10.98.2 Member Function Documentation

### 10.98.2.1 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_2 >::SetLength (
    int len ) [inline]
```

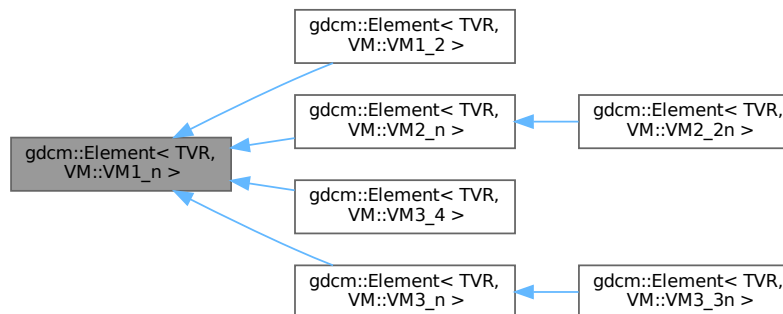
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

## 10.99 gdcm::Element< TVR, VM::VM1\_n > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, VM::VM1\_n >:



### Public Types

- typedef `VRToType< TVR >::Type` `Type`

### Public Member Functions

- `Element ()`
- `Element (const Element &_val)`
- `~Element ()`
- `DataElement GetAsDataElement () const`
- `unsigned long GetLength () const`
- `VRToType< TVR >::Type & GetValue (unsigned int idx=0)`
- `const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const`

- [Element](#) & [operator=](#) (const [Element](#) &\_val)
- [VRToType](#)< TVR >::[Type](#) [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &\_os) const
- void [Read](#) (std::istream &\_is)
- void [Set](#) ([Value](#) const &v)
- void [SetArray](#) (const [Type](#) \*array, unsigned long len, bool save=false)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetLength](#) (unsigned long len)
- void [SetValue](#) (typename [VRToType](#)< TVR >::[Type](#) v, unsigned int idx=0)
- void [Write](#) (std::ostream &\_os) const
- void [WriteASCII](#) (std::ostream &os) const

### Static Public Member Functions

- static [VM](#) [GetVM](#) ()
- static [VR](#) [GetVR](#) ()

### Protected Member Functions

- void [SetNoSwap](#) ([Value](#) const &v)

## 10.99.1 Member Typedef Documentation

### 10.99.1.1 Type

```
template<long long TVR>
typedef VRToType<TVR>::Type gdcmm::Element< TVR, VM::VM1\_n >::Type
```

## 10.99.2 Constructor & Destructor Documentation

### 10.99.2.1 [Element\(\)](#) [1/2]

```
template<long long TVR>
gdcmm::Element< TVR, VM::VM1\_n >::Element ( ) [inline], [explicit]
```

### 10.99.2.2 [~Element\(\)](#)

```
template<long long TVR>
gdcmm::Element< TVR, VM::VM1\_n >::~Element ( ) [inline]
```

### 10.99.2.3 Element() [2/2]

```
template<long long TVR>
gdcm::Element< TVR, VM::VM1_n >::Element (
    const Element< TVR, VM::VM1_n > & _val ) [inline]
```

## 10.99.3 Member Function Documentation

### 10.99.3.1 GetAsDataElement()

```
template<long long TVR>
DataElement gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement ( ) const [inline]
```

References [gdcm::DataElement::GetVR\(\)](#), [gdcm::DataElement::SetByteValue\(\)](#), and [gdcm::DataElement::SetVR\(\)](#).

### 10.99.3.2 GetLength()

```
template<long long TVR>
unsigned long gdcm::Element< TVR, VM::VM1_n >::GetLength ( ) const [inline]
```

### 10.99.3.3 GetValue() [1/2]

```
template<long long TVR>
VRToType< TVR >::Type & gdcm::Element< TVR, VM::VM1_n >::GetValue (
    unsigned int idx = 0 ) [inline]
```

### 10.99.3.4 GetValue() [2/2]

```
template<long long TVR>
const VRToType< TVR >::Type & gdcm::Element< TVR, VM::VM1_n >::GetValue (
    unsigned int idx = 0 ) const [inline]
```

### 10.99.3.5 GetVM()

```
template<long long TVR>
static VM gdcm::Element< TVR, VM::VM1_n >::GetVM ( ) [inline], [static]
```

### 10.99.3.6 GetVR()

```
template<long long TVR>
static VR gdcm::Element< TVR, VM::VM1_n >::GetVR ( ) [inline], [static]
```

**10.99.3.7 operator=()**

```
template<long long TVR>
Element & gdcm::Element< TVR, VM::VM1_n >::operator= (
    const Element< TVR, VM::VM1_n > & _val ) [inline]
```

**10.99.3.8 operator[]()**

```
template<long long TVR>
VRToType< TVR >::Type gdcm::Element< TVR, VM::VM1_n >::operator[] (
    unsigned int idx ) const [inline]
```

**10.99.3.9 Print()**

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::Print (
    std::ostream & _os ) const [inline]
```

**10.99.3.10 Read()**

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::Read (
    std::istream & _is ) [inline]
```

**10.99.3.11 Set()**

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::Set (
    Value const & v ) [inline]
```

References [gdcm::ByteValue::GetLength\(\)](#), [gdcm::ByteValue::GetPointer\(\)](#), and [gdcm::ByteValue::GetVoidPointer\(\)](#).

**10.99.3.12 SetArray()**

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetArray (
    const Type * array,
    unsigned long len,
    bool save = false ) [inline]
```

### 10.99.3.13 SetFromDataElement()

```
template<long long TVR>
void gdcmm::Element< TVR, VM::VM1_n >::SetFromDataElement (
    DataElement< TVR, VM::VM1_n > const & de ) [inline]
```

References [gdcmm::DataElement::GetByteValue\(\)](#), [gdcmm::DataElement::GetValue\(\)](#), and [gdcmm::DataElement::GetVR\(\)](#).

### 10.99.3.14 SetLength()

```
template<long long TVR>
void gdcmm::Element< TVR, VM::VM1_n >::SetLength (
    unsigned long len ) [inline]
```

### 10.99.3.15 SetNoSwap()

```
template<long long TVR>
void gdcmm::Element< TVR, VM::VM1_n >::SetNoSwap (
    Value const & v ) [inline], [protected]
```

References [gdcmm::ByteValue::GetLength\(\)](#), and [gdcmm::ByteValue::GetPointer\(\)](#).

### 10.99.3.16 SetValue()

```
template<long long TVR>
void gdcmm::Element< TVR, VM::VM1_n >::SetValue (
    typename VRToType< TVR >::Type v,
    unsigned int idx = 0 ) [inline]
```

### 10.99.3.17 Write()

```
template<long long TVR>
void gdcmm::Element< TVR, VM::VM1_n >::Write (
    std::ostream & _os ) const [inline]
```

### 10.99.3.18 WriteASCII()

```
template<long long TVR>
void gdcmm::Element< TVR, VM::VM1_n >::WriteASCII (
    std::ostream & os ) const [inline]
```

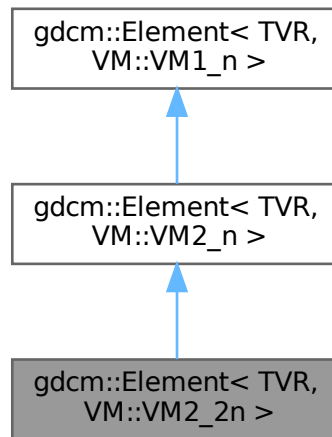
The documentation for this class was generated from the following file:

- [gdcmmElement.h](#)

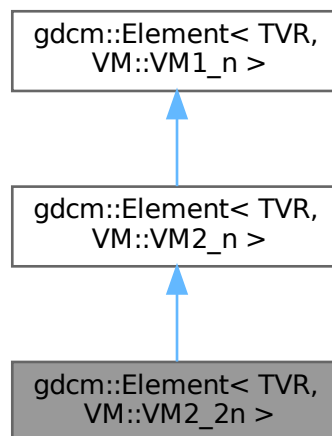
## 10.100 gdcmm::Element< TVR, VM::VM2\_2n > Class Template Reference

```
#include <gdcmmElement.h>
```

Inheritance diagram for gdcmm::Element< TVR, VM::VM2\_2n >:



Collaboration diagram for gdcmm::Element< TVR, VM::VM2\_2n >:



**Public Types**

- typedef [Element](#)< TVR, [VM::VM2\\_n](#) > [Parent](#)

**Public Types inherited from [gdcm::Element](#)< TVR, [VM::VM2\\_n](#) >**

- typedef [Element](#)< TVR, [VM::VM1\\_n](#) > [Parent](#)

**Public Types inherited from [gdcm::Element](#)< TVR, [VM::VM1\\_n](#) >**

- typedef [VRToType](#)< TVR >::Type [Type](#)

**Public Member Functions**

- void [SetLength](#) (int len)

**Public Member Functions inherited from [gdcm::Element](#)< TVR, [VM::VM2\\_n](#) >**

- void [SetLength](#) (int len)

**Public Member Functions inherited from [gdcm::Element](#)< TVR, [VM::VM1\\_n](#) >**

- [Element](#) ()
- [Element](#) (const [Element](#) &\_val)
- [~Element](#) ()
- [DataElement](#) [GetAsDataElement](#) () const
- unsigned long [GetLength](#) () const
- [VRToType](#)< TVR >::Type & [GetValue](#) (unsigned int idx=0)
- const [VRToType](#)< TVR >::Type & [GetValue](#) (unsigned int idx=0) const
- [Element](#) & [operator=](#) (const [Element](#) &\_val)
- [VRToType](#)< TVR >::Type [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &\_os) const
- void [Read](#) (std::istream &\_is)
- void [Set](#) ([Value](#) const &v)
- void [SetArray](#) (const [Type](#) \*array, unsigned long len, bool save=false)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetLength](#) (unsigned long len)
- void [SetValue](#) (typename [VRToType](#)< TVR >::Type v, unsigned int idx=0)
- void [Write](#) (std::ostream &\_os) const
- void [WriteASCII](#) (std::ostream &os) const

**Additional Inherited Members****Static Public Member Functions inherited from [gdcm::Element](#)< TVR, [VM::VM1\\_n](#) >**

- static [VM](#) [GetVM](#) ()
- static [VR](#) [GetVR](#) ()

## Protected Member Functions inherited from [gdcm::Element< TVR, VM::VM1\\_n >](#)

- void [SetNoSwap](#) ([Value](#) const &*v*)

### 10.100.1 Member Typedef Documentation

#### 10.100.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM2\_n> gdcm::Element< TVR, VM::VM2\_2n >::Parent
```

### 10.100.2 Member Function Documentation

#### 10.100.2.1 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM2\_2n >::SetLength (
    int len ) [inline]
```

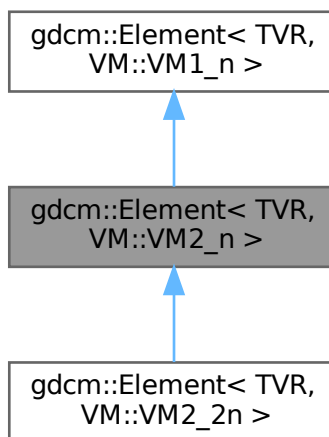
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

## 10.101 [gdcm::Element< TVR, VM::VM2\\_n >](#) Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for [gdcm::Element< TVR, VM::VM2\\_n >](#):





Collaboration diagram for gdcm::Element< TVR, VM::VM2\_n >:



### Public Types

- typedef `Element< TVR, VM::VM1_n >` `Parent`

### Public Types inherited from `gdcm::Element< TVR, VM::VM1_n >`

- typedef `VRToType< TVR >::Type` `Type`

### Public Member Functions

- void `SetLength` (int len)

### Public Member Functions inherited from `gdcm::Element< TVR, VM::VM1_n >`

- `Element` ()
- `Element` (const `Element` &\_val)
- `~Element` ()
- `DataElement GetAsDataElement` () const
- unsigned long `GetLength` () const
- `VRToType< TVR >::Type` & `GetValue` (unsigned int idx=0)
- const `VRToType< TVR >::Type` & `GetValue` (unsigned int idx=0) const
- `Element` & `operator=` (const `Element` &\_val)
- `VRToType< TVR >::Type` `operator[]` (unsigned int idx) const
- void `Print` (std::ostream &\_os) const
- void `Read` (std::istream &\_is)
- void `Set` (`Value` const &v)
- void `SetArray` (const `Type` \*array, unsigned long len, bool save=false)
- void `SetFromDataElement` (`DataElement` const &de)
- void `SetLength` (unsigned long len)
- void `SetValue` (typename `VRToType< TVR >::Type` v, unsigned int idx=0)
- void `Write` (std::ostream &\_os) const
- void `WriteASCII` (std::ostream &os) const

## Additional Inherited Members

### Static Public Member Functions inherited from [gdcm::Element< TVR, VM::VM1\\_n >](#)

- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

### Protected Member Functions inherited from [gdcm::Element< TVR, VM::VM1\\_n >](#)

- void [SetNoSwap](#) ([Value](#) const &v)

## 10.101.1 Member Typedef Documentation

### 10.101.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM1\_n> gdcm::Element< TVR, VM::VM2\_n >::Parent
```

## 10.101.2 Member Function Documentation

### 10.101.2.1 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM2\_n >::SetLength (
    int len ) [inline]
```

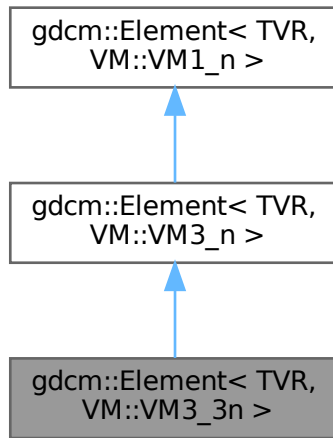
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

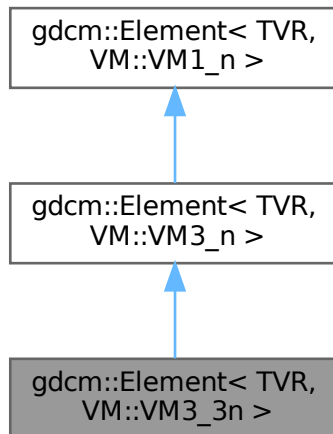
## 10.102 [gdcm::Element< TVR, VM::VM3\\_3n >](#) Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcmm::Element< TVR, VM::VM3\_3n >:



Collaboration diagram for gdcmm::Element< TVR, VM::VM3\_3n >:



## Public Types

- typedef [Element](#)< TVR, [VM::VM3\\_n](#) > [Parent](#)

### Public Types inherited from [gdcm::Element< TVR, VM::VM3\\_n >](#)

- typedef [Element< TVR, VM::VM1\\_n >](#) [Parent](#)

### Public Types inherited from [gdcm::Element< TVR, VM::VM1\\_n >](#)

- typedef [VRToType< TVR >::Type](#) [Type](#)

### Public Member Functions

- void [SetLength](#) (int len)

### Public Member Functions inherited from [gdcm::Element< TVR, VM::VM3\\_n >](#)

- void [SetLength](#) (int len)

### Public Member Functions inherited from [gdcm::Element< TVR, VM::VM1\\_n >](#)

- [Element](#) ()
- [Element](#) (const [Element](#) &\_val)
- [~Element](#) ()
- [DataElement](#) [GetAsDataElement](#) () const
- unsigned long [GetLength](#) () const
- [VRToType< TVR >::Type](#) & [GetValue](#) (unsigned int idx=0)
- const [VRToType< TVR >::Type](#) & [GetValue](#) (unsigned int idx=0) const
- [Element](#) & [operator=](#) (const [Element](#) &\_val)
- [VRToType< TVR >::Type](#) [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &\_os) const
- void [Read](#) (std::istream &\_is)
- void [Set](#) ([Value](#) const &v)
- void [SetArray](#) (const [Type](#) \*array, unsigned long len, bool save=false)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetLength](#) (unsigned long len)
- void [SetValue](#) (typename [VRToType< TVR >::Type](#) v, unsigned int idx=0)
- void [Write](#) (std::ostream &\_os) const
- void [WriteASCII](#) (std::ostream &os) const

### Additional Inherited Members

### Static Public Member Functions inherited from [gdcm::Element< TVR, VM::VM1\\_n >](#)

- static [VM](#) [GetVM](#) ()
- static [VR](#) [GetVR](#) ()

## Protected Member Functions inherited from [gdcm::Element< TVR, VM::VM1\\_n >](#)

- void [SetNoSwap](#) (Value const &v)

### 10.102.1 Member Typedef Documentation

#### 10.102.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM3_n> gdcm::Element< TVR, VM::VM3_3n >::Parent
```

### 10.102.2 Member Function Documentation

#### 10.102.2.1 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM3_3n >::SetLength (
    int len ) [inline]
```

The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

## 10.103 gdcm::Element< TVR, VM::VM3\_4 > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for [gdcm::Element< TVR, VM::VM3\\_4 >](#):



Collaboration diagram for `gdcm::Element< TVR, VM::VM3_4 >`:



### Public Types

- typedef `Element< TVR, VM::VM1_n >` `Parent`

### Public Types inherited from `gdcm::Element< TVR, VM::VM1_n >`

- typedef `VRToType< TVR >::Type` `Type`

### Public Member Functions

- void `SetLength` (int len)

### Public Member Functions inherited from `gdcm::Element< TVR, VM::VM1_n >`

- `Element` ()
- `Element` (const `Element` &\_val)
- `~Element` ()
- `DataElement GetAsDataElement` () const
- unsigned long `GetLength` () const
- `VRToType< TVR >::Type` & `GetValue` (unsigned int idx=0)
- const `VRToType< TVR >::Type` & `GetValue` (unsigned int idx=0) const
- `Element` & `operator=` (const `Element` &\_val)
- `VRToType< TVR >::Type` `operator[]` (unsigned int idx) const
- void `Print` (std::ostream &\_os) const
- void `Read` (std::istream &\_is)
- void `Set` (`Value` const &v)
- void `SetArray` (const `Type` \*array, unsigned long len, bool save=false)
- void `SetFromDataElement` (`DataElement` const &de)
- void `SetLength` (unsigned long len)
- void `SetValue` (typename `VRToType< TVR >::Type` v, unsigned int idx=0)
- void `Write` (std::ostream &\_os) const
- void `WriteASCII` (std::ostream &os) const

### Additional Inherited Members

### Static Public Member Functions inherited from `gdcm::Element< TVR, VM::VM1_n >`

- static `VM GetVM ()`
- static `VR GetVR ()`

### Protected Member Functions inherited from `gdcm::Element< TVR, VM::VM1_n >`

- void `SetNoSwap (Value const &v)`

## 10.103.1 Member Typedef Documentation

### 10.103.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM3_4 >::Parent
```

## 10.103.2 Member Function Documentation

### 10.103.2.1 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM3_4 >::SetLength (
    int len ) [inline]
```

The documentation for this class was generated from the following file:

- `gdcmElement.h`

## 10.104 `gdcm::Element< TVR, VM::VM3_n >` Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for `gdcM::Element< TVR, VM::VM3_n >`:



Collaboration diagram for `gdcM::Element< TVR, VM::VM3_n >`:



## Public Types

- typedef `Element< TVR, VM::VM1_n >` `Parent`

## Public Types inherited from `gdcM::Element< TVR, VM::VM1_n >`

- typedef `VRToType< TVR >::Type` `Type`



**Public Member Functions**

- void [SetLength](#) (int len)

**Public Member Functions inherited from [gdcm::Element< TVR, VM::VM1\\_n >](#)**

- [Element](#) ()
- [Element](#) (const [Element](#) &\_val)
- [~Element](#) ()
- [DataElement](#) [GetAsDataElement](#) () const
- unsigned long [GetLength](#) () const
- [VRToType< TVR >::Type](#) & [GetValue](#) (unsigned int idx=0)
- const [VRToType< TVR >::Type](#) & [GetValue](#) (unsigned int idx=0) const
- [Element](#) & [operator=](#) (const [Element](#) &\_val)
- [VRToType< TVR >::Type](#) [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &\_os) const
- void [Read](#) (std::istream &\_is)
- void [Set](#) ([Value](#) const &v)
- void [SetArray](#) (const [Type](#) \*array, unsigned long len, bool save=false)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetLength](#) (unsigned long len)
- void [SetValue](#) (typename [VRToType< TVR >::Type](#) v, unsigned int idx=0)
- void [Write](#) (std::ostream &\_os) const
- void [WriteASCII](#) (std::ostream &os) const

**Additional Inherited Members****Static Public Member Functions inherited from [gdcm::Element< TVR, VM::VM1\\_n >](#)**

- static [VM](#) [GetVM](#) ()
- static [VR](#) [GetVR](#) ()

**Protected Member Functions inherited from [gdcm::Element< TVR, VM::VM1\\_n >](#)**

- void [SetNoSwap](#) ([Value](#) const &v)

**10.104.1 Member Typedef Documentation****10.104.1.1 Parent**

```
template<long long TVR>
typedef Element<TVR, VM::VM1\_n> gdcm::Element< TVR, VM::VM3\_n >::Parent
```

## 10.104.2 Member Function Documentation

### 10.104.2.1 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM3_n >::SetLength (
    int len ) [inline]
```

The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

## 10.105 gdcm::Element< VR::AS, VM::VM5 > Class Reference

```
#include <gdcmElement.h>
```

### Public Member Functions

- unsigned long [GetLength](#) () const
- void [Print](#) (std::ostream &\_os) const

### Public Attributes

- char [Internal](#) [VMToLength< VM::VM5 >::Length \*sizeof(VRToType< VR::AS >::Type)]

## 10.105.1 Member Function Documentation

### 10.105.1.1 GetLength()

```
unsigned long gdcm::Element< VR::AS, VM::VM5 >::GetLength ( ) const [inline]
```

### 10.105.1.2 Print()

```
void gdcm::Element< VR::AS, VM::VM5 >::Print (
    std::ostream &_os ) const [inline]
```

## 10.105.2 Member Data Documentation

### 10.105.2.1 Internal

```
char gdcm::Element< VR::AS, VM::VM5 >::Internal[VMToLength< VM::VM5 >::Length *sizeof(VRToType< VR::AS >::Type)]
```

The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

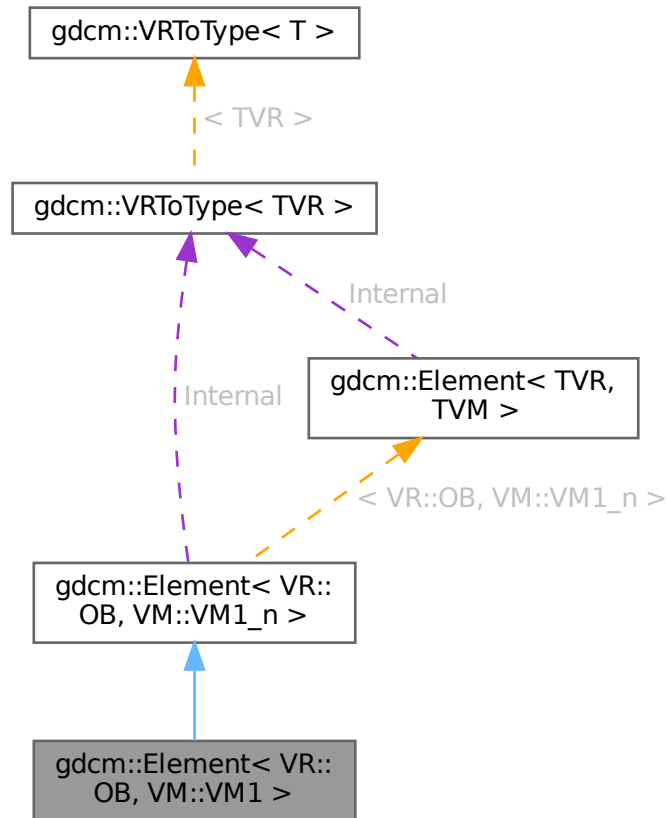
## 10.106 gdcm::Element< VR::OB, VM::VM1 > Class Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< VR::OB, VM::VM1 >:



Collaboration diagram for `gdcm::Element< VR::OB, VM::VM1 >`:



#### Additional Inherited Members

#### Public Types inherited from `gdcm::Element< VR::OB, VM::VM1_n >`

- typedef `VRToType< TVR >::Type` `Type`

#### Public Member Functions inherited from `gdcm::Element< VR::OB, VM::VM1_n >`

- `DataElement` `GetAsDataElement` () const
- unsigned long `GetLength` () const
- `VRToType< TVR >::Type` & `GetValue` (unsigned int idx=0)
- const `VRToType< TVR >::Type` & `GetValue` (unsigned int idx=0) const
- const `VRToType< TVR >::Type` \* `GetValues` () const
- `VRToType< TVR >::Type` `operator[]` (unsigned int idx) const

- void [Print](#) (std::ostream &\_os) const
- void [Read](#) (std::istream &\_is)
- void [Set](#) ([Value](#) const &v)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetValue](#) (typename [VRToType](#)< TVR >::Type v, unsigned int idx=0)
- void [Write](#) (std::ostream &\_os) const

#### Static Public Member Functions inherited from [gdcm::Element< VR::OB, VM::VM1\\_n >](#)

- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

#### Public Attributes inherited from [gdcm::Element< VR::OB, VM::VM1\\_n >](#)

- [VRToType](#)< TVR >::Type [Internal](#) [[VMToLength](#)< TVM >::Length]

#### Protected Member Functions inherited from [gdcm::Element< VR::OB, VM::VM1\\_n >](#)

- void [SetNoSwap](#) ([Value](#) const &v)

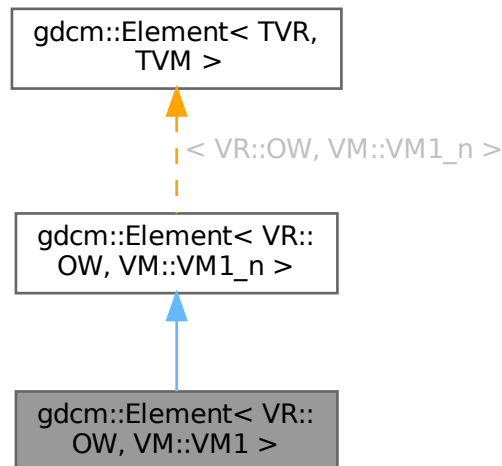
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

## 10.107 gdcm::Element< VR::OW, VM::VM1 > Class Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< VR::OW, VM::VM1 >:



Collaboration diagram for `gdcm::Element< VR::OW, VM::VM1 >`:



#### Additional Inherited Members

#### Public Types inherited from `gdcm::Element< VR::OW, VM::VM1_n >`

- typedef `VRToType< TVR >::Type` `Type`

#### Public Member Functions inherited from `gdcm::Element< VR::OW, VM::VM1_n >`

- `DataElement` `GetAsDataElement` () const
- unsigned long `GetLength` () const
- `VRToType< TVR >::Type` & `GetValue` (unsigned int idx=0)
- const `VRToType< TVR >::Type` & `GetValue` (unsigned int idx=0) const
- const `VRToType< TVR >::Type` \* `GetValues` () const
- `VRToType< TVR >::Type` `operator[]` (unsigned int idx) const

- void [Print](#) (std::ostream &\_os) const
- void [Read](#) (std::istream &\_is)
- void [Set](#) ([Value](#) const &v)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetValue](#) (typename [VRToType](#)< TVR >::Type v, unsigned int idx=0)
- void [Write](#) (std::ostream &\_os) const

### Static Public Member Functions inherited from [gdcm::Element](#)< [VR::OW](#), [VM::VM1\\_n](#) >

- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

### Public Attributes inherited from [gdcm::Element](#)< [VR::OW](#), [VM::VM1\\_n](#) >

- [VRToType](#)< TVR >::Type [Internal](#) [[VMToLength](#)< TVM >::Length]

### Protected Member Functions inherited from [gdcm::Element](#)< [VR::OW](#), [VM::VM1\\_n](#) >

- void [SetNoSwap](#) ([Value](#) const &v)

The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

## 10.108 gdcm::ElementDisableCombinations< TVR, TVM > Class Template Reference

A class which is used to produce compile errors for an invalid combination of template parameters.

```
#include <gdcmElement.h>
```

### 10.108.1 Detailed Description

```
template<long long TVR, int TVM>
class gdcm::ElementDisableCombinations< TVR, TVM >
```

A class which is used to produce compile errors for an invalid combination of template parameters.

Invalid combinations have specialized declarations with no definition.

The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

## 10.109 `gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n >` Class Reference

```
#include <gdcmElement.h>
```

The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

## 10.110 `gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n >` Class Reference

```
#include <gdcmElement.h>
```

The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

## 10.111 `gdcm::EmptyMaskGenerator` Class Reference

[EmptyMaskGenerator](#) Main class to generate a Empty Mask [Series](#) from an input [Series](#). This class takes an input folder and generates a series of DICOM files in the specified output directory. This class handles multiples DICOM [Series](#) within the same input directory.

```
#include <gdcmEmptyMaskGenerator.h>
```

### Public Types

- enum [SOPClassUIDMode](#) {  
    [UseOriginalSOPClassUID](#) = 0 ,  
    [UseGrayscaleSecondaryImageStorage](#) }

### Public Member Functions

- [EmptyMaskGenerator](#) ()
- [~EmptyMaskGenerator](#) ()
- bool [Execute](#) ()  
    *Main loop.*
- void [SetInputDirectory](#) (const char \*dirname)  
    *Specify input directory.*
- void [SetOutputDirectory](#) (const char \*dirname)  
    *Specify output directory.*
- void [SetSOPClassUIDMode](#) ([SOPClassUIDMode](#) mode)



### 10.111.1 Detailed Description

[EmptyMaskGenerator](#) Main class to generate a Empty Mask [Series](#) from an input [Series](#). This class takes an input folder and generates a series of DICOM files in the specified output directory. This class handles multiples DICOM [Series](#) within the same input directory.

The class allow two mode of operations:

- UseOriginalSOPClassUID
- UseGrayscaleSecondaryImageStorage

UseOriginalSOPClassUID is the mode where original attributes are copied from the original DICOM instance.

UseGrayscaleSecondaryImageStorage is the mode where attributes are generated so as to create a Multiframe↔ GrayscaleByteSecondaryCaptureImageStorage (MultiframeGrayscaleWordSecondaryCaptureImageStorage) instance.

In both mode:

- the [Study](#) references (StudyInstanceUID and StudyID) are preserved.
- the PatientID reference is preserved.
- the [Image Type](#) attribute will be setup so that the fourth element is set to 'MASK'.
- a new [Series](#) Instance UID is generated. It is thus required to run the process over all files using the same input [Series](#) Instance UID so that a proper mapping from the old [Series](#) UID is done to the new one. Since a new [Series](#) Instance UID is generated, there is no sense to preserve the original Frame of Reference UID, although it would have made sense here.

#### Examples

[EmptyMask.cxx](#).

### 10.111.2 Member Enumeration Documentation

#### 10.111.2.1 SOPClassUIDMode

```
enum gdcm::EmptyMaskGenerator::SOPClassUIDMode
```

##### Enumerator

UseOriginalSOPClassUID	
UseGrayscaleSecondaryImageStorage	

### 10.111.3 Constructor & Destructor Documentation

#### 10.111.3.1 EmptyMaskGenerator()

```
gdcmm::EmptyMaskGenerator::EmptyMaskGenerator ( )
```

#### 10.111.3.2 ~EmptyMaskGenerator()

```
gdcmm::EmptyMaskGenerator::~~EmptyMaskGenerator ( )
```

### 10.111.4 Member Function Documentation

#### 10.111.4.1 Execute()

```
bool gdcmm::EmptyMaskGenerator::Execute ( )
```

Main loop.

##### Examples

[EmptyMask.cxx](#).

#### 10.111.4.2 SetInputDirectory()

```
void gdcmm::EmptyMaskGenerator::SetInputDirectory (
    const char * dirname )
```

Specify input directory.

##### Examples

[EmptyMask.cxx](#).

#### 10.111.4.3 SetOutputDirectory()

```
void gdcmm::EmptyMaskGenerator::SetOutputDirectory (
    const char * dirname )
```

Specify output directory.

##### Examples

[EmptyMask.cxx](#).

#### 10.111.4.4 SetSOPClassUIDMode()

```
void gdcm::EmptyMaskGenerator::SetSOPClassUIDMode (
    SOPClassUIDMode mode )
```

Select generation of SOP Class UID method: Default is UseOriginalSOPClassUID

#### Examples

[EmptyMask.cxx](#).

The documentation for this class was generated from the following file:

- [gdcmEmptyMaskGenerator.h](#)

## 10.112 gdcm::EncapsulatedDocument Class Reference

[EncapsulatedDocument](#).

```
#include <gdcmEncapsulatedDocument.h>
```

#### Public Member Functions

- [EncapsulatedDocument](#) ()=default

### 10.112.1 Detailed Description

[EncapsulatedDocument](#).

### 10.112.2 Constructor & Destructor Documentation

#### 10.112.2.1 EncapsulatedDocument()

```
gdcm::EncapsulatedDocument::EncapsulatedDocument ( ) [default]
```

The documentation for this class was generated from the following file:

- [gdcmEncapsulatedDocument.h](#)

## 10.113 gdcm::EncodingImplementation< T > Class Template Reference

[EncodingImplementation](#).

### 10.113.1 Detailed Description

```
template<long long T>
class gdcm::EncodingImplementation< T >
```

[EncodingImplementation.](#)

Note

TODO

The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

## 10.114 gdcm::EncodingImplementation< VR::VRASCII > Class Reference

```
#include <gdcmElement.h>
```

### Public Member Functions

- template<> void [Write](#) (const double \*data, unsigned long length, std::ostream &\_os)

### Static Public Member Functions

- template<typename T >  
static void [Read](#) (T \*data, unsigned long length, std::istream &\_is)
- template<typename T >  
static void [ReadComputeLength](#) (T \*data, unsigned int &length, std::istream &\_is)
- template<typename T >  
static void [ReadNoSwap](#) (T \*data, unsigned long length, std::istream &\_is)
- template<typename T >  
static void [Write](#) (const T \*data, unsigned long length, std::ostream &\_os)

### 10.114.1 Member Function Documentation

#### 10.114.1.1 Read()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRASCII >::Read (
    T * data,
    unsigned long length,
    std::istream & _is ) [inline], [static]
```

### 10.114.1.2 ReadComputeLength()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRASCII >::ReadComputeLength (
    T * data,
    unsigned int & length,
    std::istream & _is ) [inline], [static]
```

References [gdcm::backslash\(\)](#).

### 10.114.1.3 ReadNoSwap()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRASCII >::ReadNoSwap (
    T * data,
    unsigned long length,
    std::istream & _is ) [inline], [static]
```

### 10.114.1.4 Write() [1/2]

```
template<>
void gdcm::EncodingImplementation< VR::VRASCII >::Write (
    const double * data,
    unsigned long length,
    std::ostream & _os ) [inline]
```

References [gdcm::x16printf\(\)](#).

### 10.114.1.5 Write() [2/2]

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRASCII >::Write (
    const T * data,
    unsigned long length,
    std::ostream & _os ) [inline], [static]
```

The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

## 10.115 gdcm::EncodingImplementation< VR::VRBINARY > Class Reference

```
#include <gdcmElement.h>
```

## Static Public Member Functions

- `template<typename T >`  
`static void Read (T *data, unsigned long length, std::istream &_is)`
- `template<typename T >`  
`static void ReadComputeLength (T *data, unsigned int &length, std::istream &_is)`
- `template<typename T >`  
`static void ReadNoSwap (T *data, unsigned long length, std::istream &_is)`
- `template<typename T >`  
`static void Write (const T *data, unsigned long length, std::ostream &_os)`

## 10.115.1 Member Function Documentation

### 10.115.1.1 [Read\(\)](#)

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRBINARY >::Read (
    T * data,
    unsigned long length,
    std::istream & _is ) [inline], [static]
```

### 10.115.1.2 [ReadComputeLength\(\)](#)

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadComputeLength (
    T * data,
    unsigned int & length,
    std::istream & _is ) [inline], [static]
```

### 10.115.1.3 [ReadNoSwap\(\)](#)

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadNoSwap (
    T * data,
    unsigned long length,
    std::istream & _is ) [inline], [static]
```

### 10.115.1.4 [Write\(\)](#)

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRBINARY >::Write (
    const T * data,
    unsigned long length,
    std::ostream & _os ) [inline], [static]
```

The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

## 10.116 gdcm::EndEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::EndEvent:



Collaboration diagram for gdcm::EndEvent:



### Additional Inherited Members

### Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()

- [Event](#) (const [Event](#) &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) \*) const =0
- virtual const char \* [GetEventName](#) () const =0
- virtual [Event](#) \* [MakeObject](#) () const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

The documentation for this class was generated from the following file:

- [gdcmEvent.h](#)

## 10.117 gdcm::EnumeratedValues Class Reference

**Element.** A Data [Element](#) with Enumerated Values that does not have a [Value](#) equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:

```
#include <gdcmEnumeratedValues.h>
```

### Public Member Functions

- [EnumeratedValues](#) ()=default

### 10.117.1 Detailed Description

**Element.** A Data [Element](#) with Enumerated Values that does not have a [Value](#) equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:

1. [Patient](#) Sex (0010, 0040) is an example of a Data [Element](#) having Enumerated Values. It is defined to have a [Value](#) that is either "M", "F", or "O" (see PS 3.3). No other [Value](#) shall be given to this Data [Element](#).
2. Future modifications of this standard may add to the set of allowed values for Data Elements with Enumerated Values. Such additions by themselves may or may not require a change in SOP Class [UIDs](#), depending on the semantics of the Data [Element](#).

## 10.117.2 Constructor & Destructor Documentation

### 10.117.2.1 EnumeratedValues()

```
gdcm::EnumeratedValues::EnumeratedValues ( ) [default]
```

The documentation for this class was generated from the following file:

- [gdcmEnumeratedValues.h](#)



## 10.118 gdcm::EquipmentManufacturer Class Reference

```
#include <gdcmEquipmentManufacturer.h>
```

### Public Types

- enum [Type](#) {  
[UNKNOWN](#) = 0 ,  
[FUJI](#) ,  
[GEMS](#) ,  
[HITACHI](#) ,  
[KODAK](#) ,  
[MARCONI](#) ,  
[PMS](#) ,  
[SIEMENS](#) ,  
[TOSHIBA](#) ,  
[AGFA](#) ,  
[SAMSUNG](#) ,  
[UIH](#) }

### Static Public Member Functions

- static [Type](#) [Compute](#) ([DataSet](#) const &ds)
- static const char \* [TypeToString](#) ([Type](#) type)

### 10.118.1 Detailed Description

The intent is for private tags handling. This class is not meant to handle all possible vendors in the world, simply those well known where we intend to read private tags afterwards (typically SIEMENS+CSA, GEMS+PDB ...)

### 10.118.2 Member Enumeration Documentation

#### 10.118.2.1 Type

```
enum gdcm::EquipmentManufacturer::Type
```

#### Enumerator

UNKNOWN	
FUJI	
GEMS	
HITACHI	
KODAK	
MARCONI	
PMS	
SIEMENS	
TOSHIBA	
AGFA	
SAMSUNG	
UIH	

### 10.118.3 Member Function Documentation

#### 10.118.3.1 Compute()

```
static Type gdcM::EquipmentManufacturer::Compute (  
    DataSet const & ds ) [static]
```

#### 10.118.3.2 TypeToString()

```
static const char * gdcM::EquipmentManufacturer::TypeToString (  
    Type type ) [static]
```

The documentation for this class was generated from the following file:

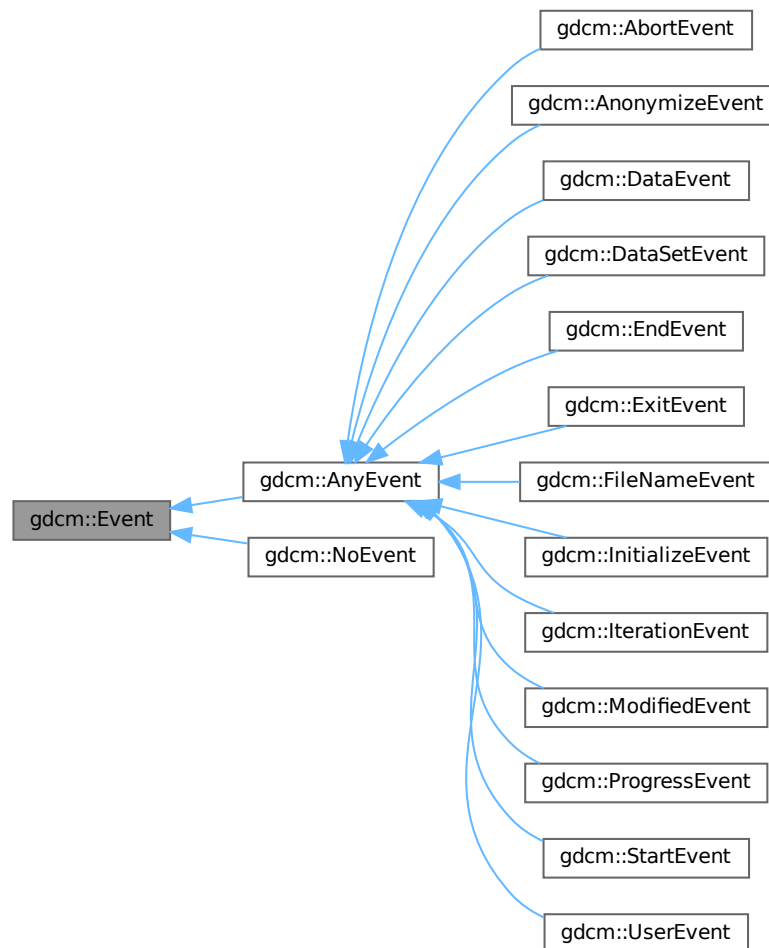
- [gdcMEquipmentManufacturer.h](#)

## 10.119 gdcM::Event Class Reference

superclass for callback/observer methods

```
#include <gdcMEvent.h>
```

Inheritance diagram for gdcmm::Event:



### Public Member Functions

- [Event](#) ()
- [Event](#) (const [Event](#) &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) \*) const =0
- virtual const char \* [GetEventName](#) () const =0
- virtual [Event](#) \* [MakeObject](#) () const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

### 10.119.1 Detailed Description

superclass for callback/observer methods

See also

[Command Subject](#)

Examples

[BasicAnonymizer.cs](#), [Cleaner.cs](#), [ClinicalTrialIdentificationWorkflow.cs](#), [ScanDirectory.cs](#), and [SimpleScanner.cxx](#).

### 10.119.2 Constructor & Destructor Documentation

#### 10.119.2.1 Event() [1/2]

```
gdcmm::Event::Event ( )
```

#### 10.119.2.2 ~Event()

```
virtual gdcmm::Event::~~Event ( ) [virtual]
```

#### 10.119.2.3 Event() [2/2]

```
gdcmm::Event::Event (
    const Event & )
```

### 10.119.3 Member Function Documentation

#### 10.119.3.1 CheckEvent()

```
virtual bool gdcmm::Event::CheckEvent (
    const Event * ) const [pure virtual]
```

Check if given event matches or derives from this event.

#### 10.119.3.2 GetEventName()

```
virtual const char * gdcmm::Event::GetEventName ( ) const [pure virtual]
```

Return the StringName associated with the event.

Implemented in [gdcmm::DataEvent](#), [gdcmm::FileNameEvent](#), [gdcmm::ProgressEvent](#), [gdcmm::DataSetEvent](#), and [gdcmm::AnonymizeEvent](#).

Examples

[BasicAnonymizer.cs](#), [Cleaner.cs](#), [ClinicalTrialIdentificationWorkflow.cs](#), and [ScanDirectory.cs](#).

### 10.119.3.3 MakeObject()

```
virtual Event * gdcm::Event::MakeObject ( ) const [pure virtual]
```

Create an [Event](#) of this type This method work as a Factory for creating events of each particular type.

Implemented in [gdcm::DataEvent](#), [gdcm::FileNameEvent](#), [gdcm::ProgressEvent](#), [gdcm::DataSetEvent](#), and [gdcm::AnonymizeEvent](#).

### 10.119.3.4 operator=()

```
void gdcm::Event::operator= (
    const Event & ) [delete]
```

### 10.119.3.5 Print()

```
virtual void gdcm::Event::Print (
    std::ostream & os ) const [virtual]
```

Print [Event](#) information. This method can be overridden by specific [Event](#) subtypes. The default is to print out the type of the event.

Referenced by [gdcm::operator<<\(\)](#).

The documentation for this class was generated from the following file:

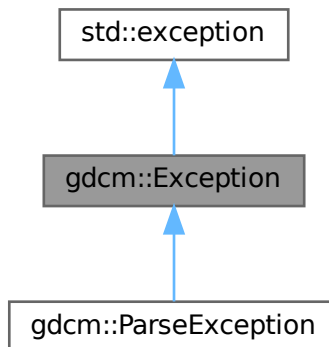
- [gdcmEvent.h](#)

## 10.120 gdcm::Exception Class Reference

[Exception](#).

```
#include <gdcmException.h>
```

Inheritance diagram for gdcm::Exception:



Collaboration diagram for `gdcm::Exception`:



### Public Member Functions

- [Exception](#) (const char \*desc="None", const char \*file=\_\_FILE\_\_, unsigned int lineNumber=\_\_LINE\_\_, const char \*func="")
- [~Exception](#) () override throw ()
- const char \* [GetDescription](#) () const  
*Return the Description.*
- const char \* [what](#) () const override throw ()  
*what implementation*

### 10.120.1 Detailed Description

[Exception](#).

Standard exception handling object.

#### Note

Its copy-constructor and assignment operator are generated by the compiler.

#### Examples

[ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), [ExtractOneFrame.cs](#), [FileChangeTS.cs](#), and [FileChangeTSLossy.cs](#).

### 10.120.2 Constructor & Destructor Documentation

#### 10.120.2.1 Exception()

```

gdcm::Exception::Exception (
    const char * desc = "None",
    const char * file = __FILE__,
    unsigned int lineNumber = __LINE__,
    const char * func = "" ) [inline], [explicit]
  
```

Explicit constructor, initializing the description and the text returned by [what\(\)](#).

**Note**

The last parameter is ignored for the time being. It may be used to specify the function where the exception was thrown.

**10.120.2.2 ~Exception()**

```
gdcm::Exception::~Exception ( ) throw ( )    [inline], [override]
```

**10.120.3 Member Function Documentation****10.120.3.1 GetDescription()**

```
const char * gdcm::Exception::GetDescription ( ) const    [inline]
```

Return the Description.

Referenced by [gdcm::SequenceOfItems::Read\(\)](#).

**10.120.3.2 what()**

```
const char * gdcm::Exception::what ( ) const throw ( )    [inline], [override]
```

what implementation

Referenced by [gdcm::SequenceOfFragments::ReadValue\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmException.h](#)

## 10.121 gdcm::ExitEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::ExitEvent:



Collaboration diagram for gdcm::ExitEvent:



### Additional Inherited Members

### Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()



- [Event](#) (const [Event](#) &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) \*) const =0
- virtual const char \* [GetEventName](#) () const =0
- virtual [Event](#) \* [MakeObject](#) () const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

The documentation for this class was generated from the following file:

- [gdcmEvent.h](#)

## 10.122 gdcm::ExplicitDataElement Class Reference

Class to read/write a [DataElement](#) as Explicit Data [Element](#).

```
#include <gdcmExplicitDataElement.h>
```

Inheritance diagram for gdcm::ExplicitDataElement:



Collaboration diagram for `gdcm::ExplicitDataElement`:



## Public Member Functions

- [VL GetLength](#) () const
- template<typename TSwap >  
std::istream & [Read](#) (std::istream &is)
- template<typename TSwap >  
std::istream & [ReadPreValue](#) (std::istream &is)
- template<typename TSwap >  
std::istream & [ReadValue](#) (std::istream &is, bool readvalues=true)
- template<typename TSwap >  
std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)
- template<typename TSwap >  
const std::ostream & [Write](#) (std::ostream &os) const

## Public Member Functions inherited from [gdcm::DataElement](#)

- [DataElement](#) (const [DataElement](#) &\_val)
- [DataElement](#) (const [Tag](#) &t=[Tag](#)(0), const [VL](#) &vl=0, const [VR](#) &vr=[VR::INVALID](#))
- void [Clear](#) ()  
*Clear Data [Element](#) (make [Value](#) empty and invalidate [Tag](#) & [VR](#))*
- void [Empty](#) ()  
*Make Data [Element](#) empty (no [Value](#))*
- const [ByteValue](#) \* [GetByteValue](#) () const
- template<typename TDE >  
[VL GetLength](#) () const
- [SequenceOfFragments](#) \* [GetSequenceOfFragments](#) ()
- const [SequenceOfFragments](#) \* [GetSequenceOfFragments](#) () const

- [Tag](#) & [GetTag](#) ()
- const [Tag](#) & [GetTag](#) () const  
*Get Tag.*
- [Value](#) & [GetValue](#) ()
- [Value](#) const & [GetValue](#) () const  
*Set/Get Value (bytes array, SQ of items, SQ of fragments):*
- [SmartPointer](#)< [SequenceOfItems](#) > [GetValueAsSQ](#) () const
- [VL](#) & [GetVL](#) ()
- const [VL](#) & [GetVL](#) () const  
*Get VL.*
- [VR](#) const & [GetVR](#) () const
- bool [IsEmpty](#) () const  
*Check if Data Element is empty.*
- bool [IsUndefinedLength](#) () const  
*return if Value Length if of undefined length*
- bool [operator](#)< (const [DataElement](#) &de) const
- [DataElement](#) & [operator](#)= (const [DataElement](#) &)=default
- bool [operator](#)== (const [DataElement](#) &de) const
- template<typename TDE , typename TSwap >  
std::istream & [Read](#) (std::istream &is)
- template<typename TDE , typename TSwap >  
std::istream & [ReadOrSkip](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE , typename TSwap >  
std::istream & [ReadPreValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE , typename TSwap >  
std::istream & [ReadValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE , typename TSwap >  
std::istream & [ReadValueWithLength](#) (std::istream &is, [VL](#) &length, std::set< [Tag](#) > const &skiptags)
- template<typename TDE , typename TSwap >  
std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)
- void [SetByteValue](#) (const char \*array, [VL](#) length)
- void [SetTag](#) (const [Tag](#) &t)
- void [SetValue](#) ([Value](#) const &vl)
- void [SetVL](#) (const [VL](#) &vl)
- void [SetVLToUndefined](#) ()
- void [SetVR](#) ([VR](#) const &vr)
- template<typename TDE , typename TSwap >  
const std::ostream & [Write](#) (std::ostream &os) const

### Additional Inherited Members

### Protected Types inherited from [gdcm::DataElement](#)

- typedef [SmartPointer](#)< [Value](#) > [ValuePtr](#)

### Protected Member Functions inherited from [gdcm::DataElement](#)

- void [SetValueFieldLength](#) ([VL](#) vl, bool readvalues)

## Protected Attributes inherited from [gdcm::DataElement](#)

- [Tag TagField](#)
- [ValuePtr ValueField](#)
- [VL ValueLengthField](#)
- [VR VRField](#)

### 10.122.1 Detailed Description

Class to read/write a [DataElement](#) as Explicit Data [Element](#).

#### Note

bla

#### Examples

[DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), and [ReadAndDumpDICOMDIR2.cxx](#).

### 10.122.2 Member Function Documentation

#### 10.122.2.1 GetLength()

```
VL gdcm::ExplicitDataElement::GetLength ( ) const
```

#### 10.122.2.2 Read()

```
template<typename TSwap >  
std::istream & gdcm::ExplicitDataElement::Read (  
    std::istream & is )
```

#### 10.122.2.3 ReadPreValue()

```
template<typename TSwap >  
std::istream & gdcm::ExplicitDataElement::ReadPreValue (  
    std::istream & is )
```

#### 10.122.2.4 ReadValue()

```
template<typename TSwap >  
std::istream & gdcm::ExplicitDataElement::ReadValue (  
    std::istream & is,  
    bool readvalues = true )
```

### 10.122.2.5 ReadWithLength()

```
template<typename TSwap >
std::istream & gdcm::ExplicitDataElement::ReadWithLength (
    std::istream & is,
    VL & length )
```

### 10.122.2.6 Write()

```
template<typename TSwap >
const std::ostream & gdcm::ExplicitDataElement::Write (
    std::ostream & os ) const
```

The documentation for this class was generated from the following file:

- [gdcmExplicitDataElement.h](#)

## 10.123 gdcm::ExplicitImplicitDataElement Class Reference

Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#).

```
#include <gdcmExplicitImplicitDataElement.h>
```

Inheritance diagram for gdcm::ExplicitImplicitDataElement:



Collaboration diagram for `gdcm::ExplicitImplicitDataElement`:



## Public Member Functions

- [VL GetLength](#) () const
- template<typename TSwap >  
std::istream & [Read](#) (std::istream &is)
- template<typename TSwap >  
std::istream & [ReadPreValue](#) (std::istream &is)
- template<typename TSwap >  
std::istream & [ReadValue](#) (std::istream &is, bool readvalues=true)
- template<typename TSwap >  
std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)

## Public Member Functions inherited from [gdcm::DataElement](#)

- [DataElement](#) (const [DataElement](#) &\_val)
- [DataElement](#) (const [Tag](#) &t=[Tag](#)(0), const [VL](#) &vl=0, const [VR](#) &vr=[VR::INVALID](#))
- void [Clear](#) ()  
*Clear Data [Element](#) (make [Value](#) empty and invalidate [Tag](#) & [VR](#))*
- void [Empty](#) ()  
*Make Data [Element](#) empty (no [Value](#))*
- const [ByteValue](#) \* [GetByteValue](#) () const
- template<typename TDE >  
[VL GetLength](#) () const
- [SequenceOfFragments](#) \* [GetSequenceOfFragments](#) ()
- const [SequenceOfFragments](#) \* [GetSequenceOfFragments](#) () const
- [Tag](#) & [GetTag](#) ()

- const [Tag](#) & [GetTag](#) () const  
*Get [Tag](#).*
- [Value](#) & [GetValue](#) ()
- [Value](#) const & [GetValue](#) () const  
*Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):*
- [SmartPointer](#)< [SequenceOfItems](#) > [GetValueAsSQ](#) () const
- [VL](#) & [GetVL](#) ()
- const [VL](#) & [GetVL](#) () const  
*Get [VL](#).*
- [VR](#) const & [GetVR](#) () const
- bool [IsEmpty](#) () const  
*Check if Data [Element](#) is empty.*
- bool [IsUndefinedLength](#) () const  
*return if [Value](#) Length if of undefined length*
- bool [operator](#)< (const [DataElement](#) &de) const
- [DataElement](#) & [operator](#)= (const [DataElement](#) &)=default
- bool [operator](#)== (const [DataElement](#) &de) const
- template<typename TDE , typename TSwap >  
std::istream & [Read](#) (std::istream &is)
- template<typename TDE , typename TSwap >  
std::istream & [ReadOrSkip](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE , typename TSwap >  
std::istream & [ReadPreValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE , typename TSwap >  
std::istream & [ReadValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE , typename TSwap >  
std::istream & [ReadValueWithLength](#) (std::istream &is, [VL](#) &length, std::set< [Tag](#) > const &skiptags)
- template<typename TDE , typename TSwap >  
std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)
- void [SetByteValue](#) (const char \*array, [VL](#) length)
- void [SetTag](#) (const [Tag](#) &t)
- void [SetValue](#) ([Value](#) const &vl)
- void [SetVL](#) (const [VL](#) &vl)
- void [SetVLToUndefined](#) ()
- void [SetVR](#) ([VR](#) const &vr)
- template<typename TDE , typename TSwap >  
const std::ostream & [Write](#) (std::ostream &os) const

### Additional Inherited Members

### Protected Types inherited from [gdcm::DataElement](#)

- typedef [SmartPointer](#)< [Value](#) > [ValuePtr](#)

### Protected Member Functions inherited from [gdcm::DataElement](#)

- void [SetValueFieldLength](#) ([VL](#) vl, bool readvalues)

## Protected Attributes inherited from [gdcm::DataElement](#)

- [Tag](#) [TagField](#)
- [ValuePtr](#) [ValueField](#)
- [VL](#) [ValueLengthField](#)
- [VR](#) [VRField](#)

### 10.123.1 Detailed Description

Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#).

#### Note

This only happen for some Philips images Should I derive from [ExplicitDataElement](#) instead ? This is the class that is the closest the GDCM1.x parser. At each element we try first to read it as explicit, if this fails, then we try again as an implicit element.

### 10.123.2 Member Function Documentation

#### 10.123.2.1 GetLength()

```
VL gdcm::ExplicitImplicitDataElement::GetLength ( ) const
```

#### 10.123.2.2 Read()

```
template<typename TSwap >  
std::istream & gdcm::ExplicitImplicitDataElement::Read (  
    std::istream & is )
```

#### 10.123.2.3 ReadPreValue()

```
template<typename TSwap >  
std::istream & gdcm::ExplicitImplicitDataElement::ReadPreValue (  
    std::istream & is )
```

#### 10.123.2.4 ReadValue()

```
template<typename TSwap >  
std::istream & gdcm::ExplicitImplicitDataElement::ReadValue (  
    std::istream & is,  
    bool readvalues = true )
```



### 10.123.2.5 ReadWithLength()

```
template<typename TSwap >
std::istream & gdcm::ExplicitImplicitDataElement::ReadWithLength (
    std::istream & is,
    VL & length ) [inline]
```

The documentation for this class was generated from the following file:

- [gdcmExplicitImplicitDataElement.h](#)

## 10.124 gdcm::Fiducials Class Reference

[Fiducials.](#)

```
#include <gdcmFiducials.h>
```

### Public Member Functions

- [Fiducials](#) ()=default

### 10.124.1 Detailed Description

[Fiducials.](#)

## 10.124.2 Constructor & Destructor Documentation

### 10.124.2.1 Fiducials()

```
gdcm::Fiducials::Fiducials ( ) [default]
```

The documentation for this class was generated from the following file:

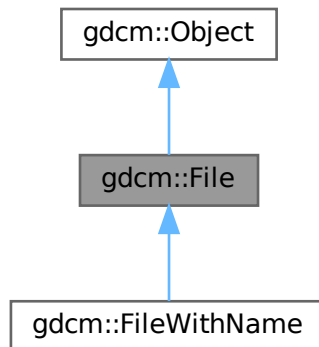
- [gdcmFiducials.h](#)

## 10.125 gdcm::File Class Reference

a DICOM [File](#)

```
#include <gdcmFile.h>
```

Inheritance diagram for gdcm::File:



Collaboration diagram for gdcm::File:



### Public Member Functions

- [File](#) ()
- [~File](#) () override
- [DataSet](#) & [GetDataSet](#) ()

*Get Data Set.*

- const [DataSet](#) & [GetDataSet](#) () const  
*Get Data Set.*
- [FileMetaInformation](#) & [GetHeader](#) ()  
*Get File Meta Information.*
- const [FileMetaInformation](#) & [GetHeader](#) () const  
*Get File Meta Information.*
- std::istream & [Read](#) (std::istream &is)  
*Read.*
- void [SetDataSet](#) (const [DataSet](#) &ds)  
*Set Data Set.*
- void [SetHeader](#) (const [FileMetaInformation](#) &fmi)  
*Set File Meta Information.*
- std::ostream const & [Write](#) (std::ostream &os) const  
*Write.*

### Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

### Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [File](#) &val)

### Additional Inherited Members

### Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

## 10.125.1 Detailed Description

a DICOM [File](#)

See PS 3.10 [File](#): A [File](#) is an ordered string of zero or more bytes, where the first byte is at the beginning of the file and the last byte at the end of the [File](#). Files are identified by a unique [File](#) ID and may be written, read and/or deleted.

See also

[Reader Writer](#)

Examples

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CompressLossyJPEG.cs](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpCSA.cs](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), [ExtractOneFrame.cs](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [FixBrokenJ2K.cxx](#), [FixOrientation.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [MpegVideoInfo.cs](#), [NewSequence.cs](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadGEMSSDO.cxx](#), [SimplePrint.cs](#), [SimplePrintPatientName.cs](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), and [iU22tomultisc.cxx](#).

## 10.125.2 Constructor & Destructor Documentation

### 10.125.2.1 File()

```
gdcm::File::File ( )
```

### 10.125.2.2 ~File()

```
gdcm::File::~~File ( ) [override]
```

References [gdcm::operator<<\(\)](#).

## 10.125.3 Member Function Documentation

### 10.125.3.1 GetDataSet() [1/2]

```
DataSet & gdcm::File::GetDataSet ( ) [inline]
```

Get Data Set.

**10.125.3.2 GetDataSet()** [2/2]

```
const DataSet & gdcm::File::GetDataSet ( ) const [inline]
```

Get Data Set.

**Examples**

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CompressLossyJPEG.cs](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [DecompressImage.cs](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [SimplePrint.cs](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), [csa2img.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

**10.125.3.3 GetHeader()** [1/2]

```
FileMetaInformation & gdcm::File::GetHeader ( ) [inline]
```

Get [File](#) Meta Information.

**10.125.3.4 GetHeader()** [2/2]

```
const FileMetaInformation & gdcm::File::GetHeader ( ) const [inline]
```

Get [File](#) Meta Information.

**Examples**

[CreateJPIPDataSet.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [EncapsulateFileInRawData.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GetJPEGSamplePrecision.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [MergeTwoFiles.cxx](#), [MpegVideoInfo.cs](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReformatFile.cs](#), [StandardizeFiles.cs](#), [StreamImageReaderTest.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

**10.125.3.5 Read()**

```
std::istream & gdcm::File::Read (
    std::istream & is )
```

Read.

### 10.125.3.6 SetDataSet()

```
void gdcM::File::SetDataSet (
    const DataSet & ds ) [inline]
```

Set Data Set.

### 10.125.3.7 SetHeader()

```
void gdcM::File::SetHeader (
    const FileMetaInformation & fmi ) [inline]
```

Set [File](#) Meta Information.

### 10.125.3.8 Write()

```
std::ostream const & gdcM::File::Write (
    std::ostream & os ) const
```

Write.

## 10.125.4 Friends And Related Symbol Documentation

### 10.125.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & os,
    const File & val ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcMFile.h](#)

## 10.126 gdcm::FileAnonymizer Class Reference

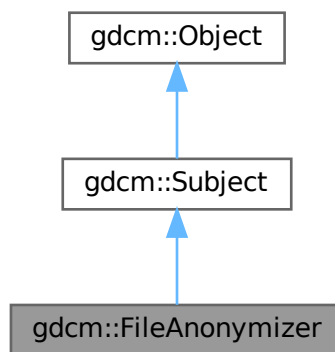
[FileAnonymizer](#).

```
#include <gdcmFileAnonymizer.h>
```

Inheritance diagram for gdcm::FileAnonymizer:



Collaboration diagram for gdcm::FileAnonymizer:



## Public Member Functions

- [FileAnonymizer](#) ()
- [~FileAnonymizer](#) () override
- void [Empty](#) ([Tag](#) const &t)
- void [Remove](#) ([Tag](#) const &t)  
*remove a tag (even a SQ can be removed)*
- void [Replace](#) ([Tag](#) const &t, const char \*value\_data, [VL](#) const &vl)
- void [Replace](#) ([Tag](#) const &t, const char \*value\_str)
- void [SetInputFileName](#) (const char \*filename\_native)  
*Set input filename.*
- void [SetOutputFileName](#) (const char \*filename\_native)  
*Set output filename.*
- bool [Write](#) ()  
*Write the output file.*

## Public Member Functions inherited from [gdcm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*) const
- [Command](#) \* [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

## Additional Inherited Members

## Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()



## 10.126.1 Detailed Description

[FileAnonymizer](#).

This [Anonymizer](#) is a file-based [Anonymizer](#). It requires a valid DICOM file and will use the [Value](#) Length to skip over any information.

It will not load the DICOM dataset taken from [SetInputFileName\(\)](#) into memory and should consume much less memory than [Anonymizer](#).

### Warning

: Each time you call [Replace\(\)](#) with a value. This value will be copied, and stored in memory. The behavior is not ideal for extremely large data (larger than memory size). This class is really meant to take a large DICOM input file and then only change some small attribute.

### caveats:

- This class will NOT work with unordered attributes in a DICOM [File](#),
- This class does neither recompute nor update the Group Length element,
- This class currently does not update the [File](#) Meta Information header.
- Only strict inplace Replace operation is supported when input and output file are the same.

### Examples

[FileAnonymize.cs](#), and [MakeTemplate.cxx](#).

## 10.126.2 Constructor & Destructor Documentation

### 10.126.2.1 FileAnonymizer()

```
gdcm::FileAnonymizer::FileAnonymizer ( )
```

### 10.126.2.2 ~FileAnonymizer()

```
gdcm::FileAnonymizer::~~FileAnonymizer ( ) [override]
```

## 10.126.3 Member Function Documentation

### 10.126.3.1 Empty()

```
void gdcm::FileAnonymizer::Empty (
    Tag const & t )
```

Make [Tag](#) t empty Warning: does not handle SQ element

### Examples

[FileAnonymize.cs](#), and [MakeTemplate.cxx](#).

### 10.126.3.2 Remove()

```
void gdcM::FileAnonymizer::Remove (
    Tag const & t )
```

remove a tag (even a SQ can be removed)

#### Examples

[FileAnonymize.cs](#).

### 10.126.3.3 Replace() [1/2]

```
void gdcM::FileAnonymizer::Replace (
    Tag const & t,
    const char * value_data,
    VL const & vl )
```

when the value contains \0, it is a good idea to specify the length. This function is required when dealing with VRBINARY tag

### 10.126.3.4 Replace() [2/2]

```
void gdcM::FileAnonymizer::Replace (
    Tag const & t,
    const char * value_str )
```

Replace tag with another value, if tag is not found it will be created: WARNING: this function can only execute if tag is a VRASCII WARNING: Do not ever try to write a value in a SQ Data [Element](#) !

#### Examples

[FileAnonymize.cs](#).

### 10.126.3.5 SetInputFileName()

```
void gdcM::FileAnonymizer::SetInputFileName (
    const char * filename_native )
```

Set input filename.

#### Examples

[FileAnonymize.cs](#), and [MakeTemplate.cxx](#).

### 10.126.3.6 SetOutputFileName()

```
void gdcm::FileAnonymizer::SetOutputFileName (
    const char * filename_native )
```

Set output filename.

#### Examples

[FileAnonymize.cs](#), and [MakeTemplate.cxx](#).

### 10.126.3.7 Write()

```
bool gdcm::FileAnonymizer::Write ( )
```

Write the output file.

#### Examples

[FileAnonymize.cs](#), and [MakeTemplate.cxx](#).

The documentation for this class was generated from the following file:

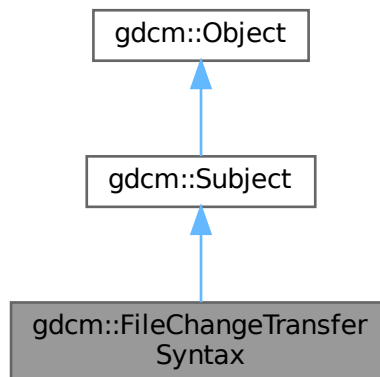
- [gdcmFileAnonymizer.h](#)

## 10.127 gdcm::FileChangeTransferSyntax Class Reference

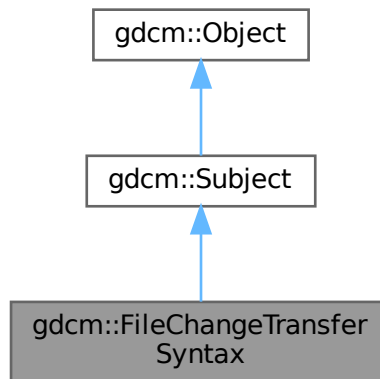
[FileChangeTransferSyntax](#).

```
#include <gdcmFileChangeTransferSyntax.h>
```

Inheritance diagram for gdcm::FileChangeTransferSyntax:



Collaboration diagram for `gdcm::FileChangeTransferSyntax`:



### Public Member Functions

- [FileChangeTransferSyntax](#) ()
- [~FileChangeTransferSyntax](#) () override
- bool [Change](#) ()  
*Change the transfer syntax.*
- [ImageCodec](#) \* [GetCodec](#) ()
- void [SetInputFileName](#) (const char \*filename\_native)  
*Set input filename (raw DICOM)*
- void [SetOutputFileName](#) (const char \*filename\_native)  
*Set output filename (target compressed DICOM)*
- void [SetTransferSyntax](#) ([TransferSyntax](#) const &ts)  
*Specify the Target Transfer Syntax.*

### Public Member Functions inherited from [gdcm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*) const
- [Command](#) \* [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

## Static Public Member Functions

- static [SmartPointer](#)< [FileChangeTransferSyntax](#) > [New](#) ()  
*for wrapped language: instantiate a reference counted object*

## Additional Inherited Members

## Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

### 10.127.1 Detailed Description

[FileChangeTransferSyntax](#).

This class is a file-based (limited) replacement of the in-memory [ImageChangeTransferSyntax](#).

This class provide a file-based compression-only mechanism. It will take in an uncompressed DICOM image file (Pixel Data element). Then produced as output a compressed DICOM file (Transfer Syntax will be updated).

Currently it supports the following transfer syntax:

- JPEGLosslessProcess14\_1

#### Examples

[FileChangeTS.cs](#), and [FileChangeTSLossy.cs](#).

### 10.127.2 Constructor & Destructor Documentation

#### 10.127.2.1 [FileChangeTransferSyntax](#)()

```
gdcm::FileChangeTransferSyntax::FileChangeTransferSyntax ( )
```

### 10.127.2.2 ~FileChangeTransferSyntax()

```
gdcm::FileChangeTransferSyntax::~~FileChangeTransferSyntax ( ) [override]
```

## 10.127.3 Member Function Documentation

### 10.127.3.1 Change()

```
bool gdcm::FileChangeTransferSyntax::Change ( )
```

Change the transfer syntax.

#### Examples

[FileChangeTS.cs](#), and [FileChangeTSLossy.cs](#).

### 10.127.3.2 GetCodec()

```
ImageCodec * gdcm::FileChangeTransferSyntax::GetCodec ( )
```

Retrieve the actual codec (valid after calling SetTransferSyntax) Only advanced users should call this function.

#### Examples

[FileChangeTSLossy.cs](#).

### 10.127.3.3 New()

```
static SmartPointer< FileChangeTransferSyntax > gdcm::FileChangeTransferSyntax::New ( ) [inline],  
[static]
```

for wrapped language: instantiate a reference counted object

#### Examples

[FileChangeTS.cs](#), and [FileChangeTSLossy.cs](#).

### 10.127.3.4 SetInputFileName()

```
void gdcm::FileChangeTransferSyntax::SetInputFileName (  
    const char * filename_native )
```

Set input filename (raw DICOM)

#### Examples

[FileChangeTS.cs](#), and [FileChangeTSLossy.cs](#).

### 10.127.3.5 SetOutputFileName()

```
void gdcm::FileChangeTransferSyntax::SetOutputFileName (
    const char * filename_native )
```

Set output filename (target compressed DICOM)

#### Examples

[FileChangeTS.cs](#), and [FileChangeTSLossy.cs](#).

### 10.127.3.6 SetTransferSyntax()

```
void gdcm::FileChangeTransferSyntax::SetTransferSyntax (
    TransferSyntax const & ts )
```

Specify the Target Transfer Syntax.

#### Examples

[FileChangeTS.cs](#), and [FileChangeTSLossy.cs](#).

The documentation for this class was generated from the following file:

- [gdcmFileChangeTransferSyntax.h](#)

## 10.128 gdcm::FileDecompressLookupTable Class Reference

[FileDecompressLookupTable](#) class.

```
#include <gdcmFileDecompressLookupTable.h>
```

Inheritance diagram for gdcm::FileDecompressLookupTable:



Collaboration diagram for `gdcm::FileDecompressLookupTable`:



### Public Member Functions

- `FileDecompressLookupTable ()`=default
- `~FileDecompressLookupTable ()` override=default
- `bool Change ()`  
*Decompress.*
- `File & GetFile ()`
- `Pixmap & GetPixmap ()`
- `const Pixmap & GetPixmap () const`
- `void SetFile (const File &f)`  
*Set/Get File.*
- `void SetPixmap (Pixmap const &img)`

### Public Member Functions inherited from `gdcm::Subject`

- `Subject ()`
- `~Subject ()` override
- `unsigned long AddObserver (const Event &event, Command *)`
- `unsigned long AddObserver (const Event &event, Command *) const`
- `Command * GetCommand (unsigned long tag)`
- `bool HasObserver (const Event &event) const`
- `void InvokeEvent (const Event &)`
- `void InvokeEvent (const Event &) const`
- `void RemoveAllObservers ()`
- `void RemoveObserver (unsigned long tag)`



## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
- *Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

## Additional Inherited Members

## Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

### 10.128.1 Detailed Description

[FileDecompressLookupTable](#) class.

It decompress the segmented LUT into linearized one (only PALETTE\_COLOR images) Output will be a [PhotometricInterpretation](#)=RGB image

### 10.128.2 Constructor & Destructor Documentation

#### 10.128.2.1 [FileDecompressLookupTable](#)()

```
gdcm::FileDecompressLookupTable::FileDecompressLookupTable ( ) [default]
```

#### 10.128.2.2 [~FileDecompressLookupTable](#)()

```
gdcm::FileDecompressLookupTable::~~FileDecompressLookupTable ( ) [override], [default]
```

### 10.128.3 Member Function Documentation

#### 10.128.3.1 [Change](#)()

```
bool gdcm::FileDecompressLookupTable::Change ( )
```

Decompress.

### 10.128.3.2 GetFile()

```
File & gdcm::FileDecompressLookupTable::GetFile ( ) [inline]
```

### 10.128.3.3 GetPixmap() [1/2]

```
Pixmap & gdcm::FileDecompressLookupTable::GetPixmap ( ) [inline]
```

### 10.128.3.4 GetPixmap() [2/2]

```
const Pixmap & gdcm::FileDecompressLookupTable::GetPixmap ( ) const [inline]
```

### 10.128.3.5 SetFile()

```
void gdcm::FileDecompressLookupTable::SetFile (
    const File & f ) [inline]
```

Set/Get [File](#).

### 10.128.3.6 SetPixmap()

```
void gdcm::FileDecompressLookupTable::SetPixmap (
    Pixmap const & img ) [inline]
```

The documentation for this class was generated from the following file:

- [gdcmFileDecompressLookupTable.h](#)

## 10.129 gdcm::FileDerivation Class Reference

[FileDerivation](#) class.

```
#include <gdcmFileDerivation.h>
```

**Public Member Functions**

- [FileDerivation](#) ()
- [~FileDerivation](#) ()
- bool [AddReference](#) (const char \*referencedsopclassuid, const char \*referencedsopinstanceuid)
- bool [Derive](#) ()  
*Change.*
- [File](#) & [GetFile](#) ()
- const [File](#) & [GetFile](#) () const
- void [SetAppendDerivationHistory](#) (bool b)
- void [SetDerivationCodeSequenceCodeValue](#) (unsigned int codevalue)  
*Specify the Derivation Code Sequence Code Value. Eg 113040.*
- void [SetDerivationDescription](#) (const char \*dd)  
*Specify the Derivation Description. Eg "lossy conversion".*
- void [SetFile](#) (const [File](#) &f)  
*Set/Get File.*
- void [SetPurposeOfReferenceCodeSequenceCodeValue](#) (unsigned int codevalue)  
*Specify the Purpose Of Reference Code Value. Eg. 121320.*

**Protected Member Functions**

- bool [AddDerivationDescription](#) ()
- bool [AddPurposeOfReferenceCodeSequence](#) ([DataSet](#) &ds)
- bool [AddSourceImageSequence](#) ()

**10.129.1 Detailed Description**

[FileDerivation](#) class.

See PS 3.16 - 2008 For the list of Code [Value](#) that can be used for in Derivation Code Sequence

URL: [http://medical.nema.org/medical/dicom/2008/08\\_16pu.pdf](http://medical.nema.org/medical/dicom/2008/08_16pu.pdf)

DICOM Part 16 has two Context Groups CID 7202 and CID 7203 which contain a set of codes defining reason for a source image reference (ie. reason code for referenced image sequence) and a coded description of the derivation applied to the new image data from the original. Both these context groups are extensible.

[File](#) Derivation is compulsory when creating a lossy derived image.

**Examples**

[DeriveSeries.cxx](#), [GenFakelImage.cxx](#), and [ReformatFile.cs](#).

**10.129.2 Constructor & Destructor Documentation****10.129.2.1 FileDerivation()**

```
gdcm::FileDerivation::FileDerivation ( )
```

### 10.129.2.2 ~FileDerivation()

```
gdcm::FileDerivation::~~FileDerivation ( )
```

## 10.129.3 Member Function Documentation

### 10.129.3.1 AddDerivationDescription()

```
bool gdcm::FileDerivation::AddDerivationDescription ( ) [protected]
```

### 10.129.3.2 AddPurposeOfReferenceCodeSequence()

```
bool gdcm::FileDerivation::AddPurposeOfReferenceCodeSequence (
    DataSet & ds ) [protected]
```

### 10.129.3.3 AddReference()

```
bool gdcm::FileDerivation::AddReference (
    const char * referencedsopclassuid,
    const char * referencedsopinstanceuid )
```

Create the proper reference. Need to pass the original SOP Class UID and the original SOP Instance UID, so that those value can be used as Reference.

#### Warning

referencedsopclassuid and referencedsopinstanceuid needs to be \0 padded. This is not compatible with how ByteValue->GetPointer works.

#### Examples

[DeriveSeries.cxx](#), [GenFakelImage.cxx](#), and [ReformatFile.cs](#).

### 10.129.3.4 AddSourceImageSequence()

```
bool gdcm::FileDerivation::AddSourceImageSequence ( ) [protected]
```

### 10.129.3.5 Derive()

```
bool gdcm::FileDerivation::Derive ( )
```

Change.

#### Examples

[DeriveSeries.cxx](#), [GenFakelImage.cxx](#), and [ReformatFile.cs](#).

### 10.129.3.6 GetFile() [1/2]

```
File & gdcmm::FileDerivation::GetFile ( ) [inline]
```

#### Examples

[GenFakelImage.cxx](#), and [ReformatFile.cs](#).

### 10.129.3.7 GetFile() [2/2]

```
const File & gdcmm::FileDerivation::GetFile ( ) const [inline]
```

### 10.129.3.8 SetAppendDerivationHistory()

```
void gdcmm::FileDerivation::SetAppendDerivationHistory (
    bool b )
```

Specify if Derivation history should be appended (default false) When false, this is an error if input already has a derivation history When true, both Purpose of Reference Code [Value](#) and Derivation Code Sequence Code [Value](#) can have their history appended.

### 10.129.3.9 SetDerivationCodeSequenceCodeValue()

```
void gdcmm::FileDerivation::SetDerivationCodeSequenceCodeValue (
    unsigned int codevalue )
```

Specify the Derivation Code Sequence Code [Value](#). Eg 113040.

#### Examples

[DeriveSeries.cxx](#), [GenFakelImage.cxx](#), and [ReformatFile.cs](#).

### 10.129.3.10 SetDerivationDescription()

```
void gdcmm::FileDerivation::SetDerivationDescription (
    const char * dd )
```

Specify the Derivation Description. Eg "lossy conversion".

### 10.129.3.11 SetFile()

```
void gdcM::FileDerivation::SetFile (
    const File & f ) [inline]
```

Set/Get [File](#).

#### Examples

[DeriveSeries.cxx](#), [GenFakelImage.cxx](#), and [ReformatFile.cs](#).

### 10.129.3.12 SetPurposeOfReferenceCodeSequenceCodeValue()

```
void gdcM::FileDerivation::SetPurposeOfReferenceCodeSequenceCodeValue (
    unsigned int codevalue )
```

Specify the Purpose Of Reference Code [Value](#). Eg. 121320.

#### Examples

[DeriveSeries.cxx](#), [GenFakelImage.cxx](#), and [ReformatFile.cs](#).

The documentation for this class was generated from the following file:

- [gdcMFileDerivation.h](#)

## 10.130 gdcM::FileExplicitFilter Class Reference

[FileExplicitFilter](#) class.

```
#include <gdcMFileExplicitFilter.h>
```

### Public Member Functions

- [FileExplicitFilter](#) ()
- [~FileExplicitFilter](#) ()=default
- bool [Change](#) ()  
*Set FMI Transfer Syntax.*
- [File](#) & [GetFile](#) ()
- void [SetChangePrivateTags](#) (bool b)  
*Decide whether or not to [VR](#)ify private tags.*
- void [SetFile](#) (const [File](#) &f)  
*Set/Get [File](#).*
- void [SetRecomputeItemLength](#) (bool b)  
*By default set Sequence & [Item](#) length to Undefined to avoid recomputing length:*
- void [SetRecomputeSequenceLength](#) (bool b)
- void [SetUseVRUN](#) (bool b)  
*When [VR](#)=16bits in explicit but Implicit has a 32bits length, use [VR](#)=UN.*

## Protected Member Functions

- bool [ChangeFMI](#) ()
- bool [ProcessDataSet](#) ([DataSet](#) &ds, [Dicts](#) const &dicts)

### 10.130.1 Detailed Description

[FileExplicitFilter](#) class.

After changing a file from Implicit to Explicit representation (see [ImageChangeTransferSyntax](#)) one operation is to make sure the [VR](#) of each DICOM attribute are accurate and do match the one from PS 3.6. Indeed when a file is written in Implicit representation, the [VR](#) is not stored directly in the file.

#### Warning

changing an implicit dataset to an explicit dataset is NOT a trivial task of simply changing the [VR](#) to the dict one:

- One has to make sure SQ is properly set
- One has to recompute the explicit length SQ
- One has to make sure that [VR](#) is valid for the encoding
- One has to make sure that [VR](#) 16bits can store the original value length

#### Examples

[GenAllVR.cxx](#), and [LargeVRDSExplicit.cxx](#).

### 10.130.2 Constructor & Destructor Documentation

#### 10.130.2.1 FileExplicitFilter()

```
gdcm::FileExplicitFilter::FileExplicitFilter ( ) [inline]
```

#### 10.130.2.2 ~FileExplicitFilter()

```
gdcm::FileExplicitFilter::~~FileExplicitFilter ( ) [default]
```

### 10.130.3 Member Function Documentation

#### 10.130.3.1 Change()

```
bool gdcm::FileExplicitFilter::Change ( )
```

Set FMI Transfer Syntax.

Change

#### Examples

[GenAllVR.cxx](#), and [LargeVRDSExplicit.cxx](#).

### 10.130.3.2 ChangeFMI()

```
bool gdcm::FileExplicitFilter::ChangeFMI ( ) [protected]
```

### 10.130.3.3 GetFile()

```
File & gdcm::FileExplicitFilter::GetFile ( ) [inline]
```

### 10.130.3.4 ProcessDataSet()

```
bool gdcm::FileExplicitFilter::ProcessDataSet (
    DataSet & ds,
    Dicts const & dicts ) [protected]
```

### 10.130.3.5 SetChangePrivateTags()

```
void gdcm::FileExplicitFilter::SetChangePrivateTags (
    bool b ) [inline]
```

Decide whether or not to [VR](#)ify private tags.

### 10.130.3.6 SetFile()

```
void gdcm::FileExplicitFilter::SetFile (
    const File & f ) [inline]
```

Set/Get [File](#).

#### Examples

[GenAllVR.cxx](#), and [LargeVRDSExplicit.cxx](#).

### 10.130.3.7 SetRecomputeItemLength()

```
void gdcm::FileExplicitFilter::SetRecomputeItemLength (
    bool b )
```

By default set Sequence & [Item](#) length to Undefined to avoid recomputing length:

### 10.130.3.8 SetRecomputeSequenceLength()

```
void gdcm::FileExplicitFilter::SetRecomputeSequenceLength (
    bool b )
```



### 10.130.3.9 SetUseVRUN()

```
void gdcm::FileExplicitFilter::SetUseVRUN (
    bool b ) [inline]
```

When [VR](#)=16bits in explicit but Implicit has a 32bits length, use [VR](#)=UN.

The documentation for this class was generated from the following file:

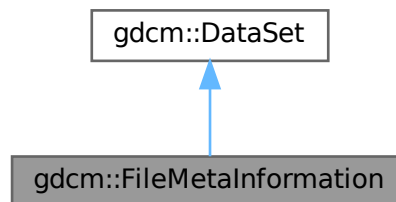
- [gdcmFileExplicitFilter.h](#)

## 10.131 gdcm::FileMetaInformation Class Reference

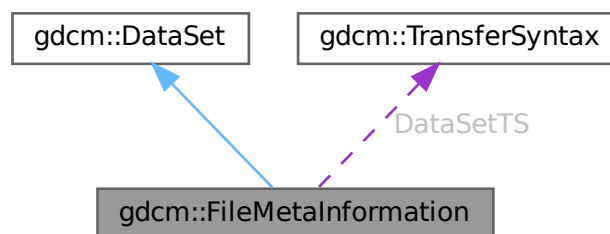
Class to represent a [File](#) Meta Information.

```
#include <gdcmFileMetaInformation.h>
```

Inheritance diagram for gdcm::FileMetaInformation:



Collaboration diagram for gdcm::FileMetaInformation:



## Public Member Functions

- [FileMetaInformation](#) ()
- [FileMetaInformation](#) ([FileMetaInformation](#) const &fmi)=default
- [~FileMetaInformation](#) ()
- void [FillFromDataSet](#) ([DataSet](#) const &ds)  
*Construct a [FileMetaInformation](#) from an already existing [DataSet](#):*
- const [TransferSyntax](#) & [GetDataSetTransferSyntax](#) () const
- [VL](#) [GetFullLength](#) () const
- [MediaStorage](#) [GetMediaStorage](#) () const
- std::string [GetMediaStorageAsString](#) () const
- [TransferSyntax::NegociatedType](#) [GetMetaInformationTS](#) () const
- [Preamble](#) & [GetPreamble](#) ()
- const [Preamble](#) & [GetPreamble](#) () const  
*Get [Preamble](#).*
- void [Insert](#) (const [DataElement](#) &de)
- bool [IsValid](#) () const
- [FileMetaInformation](#) & [operator=](#) (const [FileMetaInformation](#) &fmi)=default
- std::istream & [Read](#) (std::istream &is)  
*Read.*
- std::istream & [ReadCompat](#) (std::istream &is)
- void [Replace](#) (const [DataElement](#) &de)
- void [SetDataSetTransferSyntax](#) (const [TransferSyntax](#) &ts)
- void [SetPreamble](#) (const [Preamble](#) &p)
- std::ostream & [Write](#) (std::ostream &os) const  
*Write.*

## Public Member Functions inherited from [gdcm::DataSet](#)

- [Iterator](#) [Begin](#) ()
- [ConstIterator](#) [Begin](#) () const
- void [Clear](#) ()
- template<typename TDE >  
 unsigned int [ComputeGroupLength](#) ([Tag](#) const &tag) const
- [Iterator](#) [End](#) ()
- [ConstIterator](#) [End](#) () const
- bool [FindDataElement](#) (const [PrivateTag](#) &t) const  
*Look up if private tag 't' is present in the dataset:*
- bool [FindDataElement](#) (const [Tag](#) &t) const
- const [DataElement](#) & [FindNextDataElement](#) (const [Tag](#) &t) const
- const [DataElement](#) & [GetDataElement](#) (const [PrivateTag](#) &t) const  
*Return the dataelement.*
- const [DataElement](#) & [GetDataElement](#) (const [Tag](#) &t) const
- [DataSet](#) & [GetDES](#) ()
- const [DataSet](#) & [GetDES](#) () const
- template<typename TDE >  
[VL](#) [GetLength](#) () const
- [MediaStorage](#) [GetMediaStorage](#) () const
- std::string [GetPrivateCreator](#) (const [Tag](#) &t) const

- [PrivateTag GetPrivateTag](#) (const [Tag](#) &t) const  
*Return the private tag of the private tag 't', private creator will be set to empty if not found.*
- void [Insert](#) (const [DataElement](#) &de)
- bool [IsEmpty](#) () const  
*Returns if the dataset is empty.*
- const [DataElement](#) & [operator\(\)](#) (uint16\_t group, uint16\_t element) const
- [DataSet](#) & [operator=](#) ([DataSet](#) const &)=default
- const [DataElement](#) & [operator\[\]](#) (const [Tag](#) &t) const
- void [Print](#) (std::ostream &os, std::string const &indent="") const
- template<typename TDE , typename TSwap >  
std::istream & [Read](#) (std::istream &is)
- template<typename TDE , typename TSwap >  
std::istream & [ReadNested](#) (std::istream &is)
- template<typename TDE , typename TSwap >  
std::istream & [ReadSelectedPrivateTags](#) (std::istream &is, const std::set< [PrivateTag](#) > &tags, bool readvalues=true)
- template<typename TDE , typename TSwap >  
std::istream & [ReadSelectedPrivateTagsWithLength](#) (std::istream &is, const std::set< [PrivateTag](#) > &tags, [VL](#) &length, bool readvalues=true)
- template<typename TDE , typename TSwap >  
std::istream & [ReadSelectedTags](#) (std::istream &is, const std::set< [Tag](#) > &tags, bool readvalues=true)
- template<typename TDE , typename TSwap >  
std::istream & [ReadSelectedTagsWithLength](#) (std::istream &is, const std::set< [Tag](#) > &tags, [VL](#) &length, bool readvalues=true)
- template<typename TDE , typename TSwap >  
std::istream & [ReadUpToTag](#) (std::istream &is, const [Tag](#) &t, std::set< [Tag](#) > const &skiptags)
- template<typename TDE , typename TSwap >  
std::istream & [ReadUpToTagWithLength](#) (std::istream &is, const [Tag](#) &t, std::set< [Tag](#) > const &skiptags, [VL](#) &length)
- template<typename TDE , typename TSwap >  
std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)
- [SizeType Remove](#) (const [Tag](#) &tag)  
*Completely remove a dataelement from the dataset.*
- void [Replace](#) (const [DataElement](#) &de)  
*Replace a dataelement with another one.*
- void [ReplaceEmpty](#) (const [DataElement](#) &de)  
*Only replace a DICOM attribute when it is missing or empty.*
- [SizeType Size](#) () const
- template<typename TDE , typename TSwap >  
std::ostream const & [Write](#) (std::ostream &os) const

### Static Public Member Functions

- static void [AppendImplementationClassUID](#) (const char \*imp)
- static const char \* [GetImplementationClassUID](#) ()
- static const char \* [GetImplementationVersionName](#) ()
- static const char \* [GetSourceApplicationEntityTitle](#) ()
- static void [SetImplementationClassUID](#) (const char \*imp)  
*Override the GDCM default values:*
- static void [SetImplementationVersionName](#) (const char \*version)
- static void [SetSourceApplicationEntityTitle](#) (const char \*title)

**Protected Member Functions**

- void [ComputeDataSetMediaStorageSOPClass](#) ()
- void [ComputeDataSetTransferSyntax](#) ()
- void [Default](#) ()
- template<typename TSwap >  
std::istream & [ReadCompatInternal](#) (std::istream &is)

**Protected Member Functions inherited from [gdcm::DataSet](#)**

- [Tag ComputeDataElement](#) (const [PrivateTag](#) &t) const
- const [DataElement](#) & [GetDEEnd](#) () const
- void [InsertDataElement](#) (const [DataElement](#) &de)

**Static Protected Member Functions**

- static const char \* [GetFileMetaInformationVersion](#) ()
- static const char \* [GetGDCMImplementationClassUID](#) ()
- static const char \* [GetGDCMImplementationVersionName](#) ()
- static const char \* [GetGDCMSourceApplicationEntityTitle](#) ()

**Protected Attributes**

- [MediaStorage::MSType](#) [DataSetMS](#)
- [TransferSyntax](#) [DataSetTS](#)
- [TransferSyntax::NegociatedType](#) [MetaInformationTS](#)

**Friends**

- std::ostream & [operator<<](#) (std::ostream &\_os, const [FileMetaInformation](#) &\_val)

**Additional Inherited Members****Public Types inherited from [gdcm::DataSet](#)**

- typedef [DataSet::const\\_iterator](#) [ConstIterator](#)
- typedef std::set< [DataElement](#) > [DataElementSet](#)
- typedef [DataSet::iterator](#) [Iterator](#)
- typedef [DataSet::size\\_type](#) [SizeType](#)

## 10.131.1 Detailed Description

Class to represent a [File](#) Meta Information.

[FileMetaInformation](#) is a Explicit Structured Set. Whenever the file contains an [ImplicitDataElement DataSet](#), a conversion will take place.

Definition: The [File](#) Meta Information includes identifying information on the encapsulated Data Set. This header consists of a 128 byte [File Preamble](#), followed by a 4 byte DICOM prefix, followed by the [File](#) Meta Elements shown in [Table 7.1-1](#). This header shall be present in every DICOM file.

See also

[Writer Reader](#)

Examples

[ClinicalTrialIdentificationWorkflow.cs](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReformatFile.cs](#), and [StandardizeFiles.cs](#).

## 10.131.2 Constructor & Destructor Documentation

### 10.131.2.1 FileMetaInformation() [1/2]

```
gdcm::FileMetaInformation::FileMetaInformation ( )
```

### 10.131.2.2 ~FileMetaInformation()

```
gdcm::FileMetaInformation::~~FileMetaInformation ( )
```

References [gdcm::operator<<\(\)](#).

### 10.131.2.3 FileMetaInformation() [2/2]

```
gdcm::FileMetaInformation::FileMetaInformation (
    FileMetaInformation const & fmi ) [default]
```

## 10.131.3 Member Function Documentation

### 10.131.3.1 AppendImplementationClassUID()

```
static void gdcm::FileMetaInformation::AppendImplementationClassUID (
    const char * imp ) [static]
```

### 10.131.3.2 ComputeDataSetMediaStorageSOPClass()

```
void gdcm::FileMetaInformation::ComputeDataSetMediaStorageSOPClass ( ) [protected]
```

### 10.131.3.3 ComputeDataSetTransferSyntax()

```
void gdcm::FileMetaInformation::ComputeDataSetTransferSyntax ( ) [protected]
```

### 10.131.3.4 Default()

```
void gdcm::FileMetaInformation::Default ( ) [protected]
```

### 10.131.3.5 FillFromDataSet()

```
void gdcm::FileMetaInformation::FillFromDataSet (
    DataSet const & ds )
```

Construct a [FileMetaInformation](#) from an already existing [DataSet](#):

### 10.131.3.6 GetDataSetTransferSyntax()

```
const TransferSyntax & gdcm::FileMetaInformation::GetDataSetTransferSyntax ( ) const [inline]
```

#### Examples

[GetJPEGSamplePrecision.cxx](#), and [MergeTwoFiles.cxx](#).

### 10.131.3.7 GetFileMetaInformationVersion()

```
static const char * gdcm::FileMetaInformation::GetFileMetaInformationVersion ( ) [static], [protected]
```

### 10.131.3.8 GetFullLength()

```
VL gdcm::FileMetaInformation::GetFullLength ( ) const [inline]
```

References [gdcm::VL::GetLength\(\)](#).

### 10.131.3.9 GetGDCMImplementationClassUID()

```
static const char * gdcm::FileMetaInformation::GetGDCMImplementationClassUID ( ) [static], [protected]
```

**10.131.3.10 GetGDCMImplementationVersionName()**

```
static const char * gdcm::FileMetaInformation::GetGDCMImplementationVersionName ( ) [static],  
[protected]
```

**10.131.3.11 GetGDCMSourceApplicationEntityTitle()**

```
static const char * gdcm::FileMetaInformation::GetGDCMSourceApplicationEntityTitle ( ) [static],  
[protected]
```

**10.131.3.12 GetImplementationClassUID()**

```
static const char * gdcm::FileMetaInformation::GetImplementationClassUID ( ) [static]
```

**10.131.3.13 GetImplementationVersionName()**

```
static const char * gdcm::FileMetaInformation::GetImplementationVersionName ( ) [static]
```

**10.131.3.14 GetMediaStorage()**

```
MediaStorage gdcm::FileMetaInformation::GetMediaStorage ( ) const
```

**10.131.3.15 GetMediaStorageAsString()**

```
std::string gdcm::FileMetaInformation::GetMediaStorageAsString ( ) const
```

**10.131.3.16 GetMetaInformationTS()**

```
TransferSyntax::NegociatedType gdcm::FileMetaInformation::GetMetaInformationTS ( ) const [inline]
```

**10.131.3.17 GetPreamble() [1/2]**

```
Preamble & gdcm::FileMetaInformation::GetPreamble ( ) [inline]
```

**10.131.3.18 GetPreamble() [2/2]**

```
const Preamble & gdcm::FileMetaInformation::GetPreamble ( ) const [inline]
```

Get [Preamble](#).

#### 10.131.3.19 GetSourceApplicationEntityTitle()

```
static const char * gdcM::FileMetaInformation::GetSourceApplicationEntityTitle ( ) [static]
```

#### 10.131.3.20 Insert()

```
void gdcM::FileMetaInformation::Insert (
    const DataElement & de ) [inline]
```

References [gdcMErrorMacro](#), [gdcM::Tag::GetGroup\(\)](#), and [gdcM::DataElement::GetTag\(\)](#).

#### 10.131.3.21 IsValid()

```
bool gdcM::FileMetaInformation::IsValid ( ) const [inline]
```

#### 10.131.3.22 operator=()

```
FileMetaInformation & gdcM::FileMetaInformation::operator= (
    const FileMetaInformation & fmi ) [default]
```

#### 10.131.3.23 Read()

```
std::istream & gdcM::FileMetaInformation::Read (
    std::istream & is )
```

Read.

#### 10.131.3.24 ReadCompat()

```
std::istream & gdcM::FileMetaInformation::ReadCompat (
    std::istream & is )
```

#### 10.131.3.25 ReadCompatInternal()

```
template<typename TSwap >
std::istream & gdcM::FileMetaInformation::ReadCompatInternal (
    std::istream & is ) [protected]
```



### 10.131.3.26 Replace()

```
void gdcmm::FileMetaInformation::Replace (
    const DataElement & de ) [inline]
```

#### Examples

[LargeVRDSExplicit.cxx](#).

References [gdcmm::DataElement::GetTag\(\)](#).

### 10.131.3.27 SetDataSetTransferSyntax()

```
void gdcmm::FileMetaInformation::SetDataSetTransferSyntax (
    const TransferSyntax & ts )
```

#### Examples

[CreateJPIPDataSet.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [EncapsulateFileInRawData.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [MpegVideoInfo.cs](#), [QIDO-RS.cxx](#), [StreamImageReaderTest.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

### 10.131.3.28 SetImplementationClassUID()

```
static void gdcmm::FileMetaInformation::SetImplementationClassUID (
    const char * imp ) [static]
```

Override the GDCM default values:

### 10.131.3.29 SetImplementationVersionName()

```
static void gdcmm::FileMetaInformation::SetImplementationVersionName (
    const char * version ) [static]
```

### 10.131.3.30 SetPreamble()

```
void gdcmm::FileMetaInformation::SetPreamble (
    const Preamble & p ) [inline]
```

### 10.131.3.31 SetSourceApplicationEntityTitle()

```
static void gdcM::FileMetaInformation::SetSourceApplicationEntityTitle (
    const char * title ) [static]
```

#### Examples

[ClinicalTrialIdentificationWorkflow.cs](#), [FixJAIBugJPEGLS.cxx](#), [GenerateDICOMDIR.cs](#), [ReformatFile.cs](#), and [StandardizeFiles.cs](#).

### 10.131.3.32 Write()

```
std::ostream & gdcM::FileMetaInformation::Write (
    std::ostream & os ) const
```

Write.

## 10.131.4 Friends And Related Symbol Documentation

### 10.131.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const FileMetaInformation & _val ) [friend]
```

## 10.131.5 Member Data Documentation

### 10.131.5.1 DataSetMS

[MediaStorage::MSType](#) gdcM::FileMetaInformation::DataSetMS [protected]

### 10.131.5.2 DataSetTS

[TransferSyntax](#) gdcM::FileMetaInformation::DataSetTS [protected]

### 10.131.5.3 MetaInformationTS

[TransferSyntax::NegociatedType](#) gdcM::FileMetaInformation::MetaInformationTS [protected]

The documentation for this class was generated from the following file:

- [gdcMFileMetaInformation.h](#)

## 10.132 gdcm::Filename Class Reference

Class to manipulate file name's.

```
#include <gdcmFilename.h>
```

### Public Member Functions

- [Filename](#) (const char \*filename="")
- bool [EndWith](#) (const char ending[]) const  
*Does the filename ends with a particular string ?*
- const char \* [GetExtension](#) ()  
*return only the extension part of a filename*
- const char \* [GetFileName](#) () const  
*Return the full filename.*
- const char \* [GetName](#) ()  
*return only the name part of a filename*
- const char \* [GetPath](#) ()  
*Return only the path component of a filename.*
- bool [IsEmpty](#) () const  
*return whether the filename is empty*
- bool [IsIdentical](#) ([Filename](#) const &fn) const
- [operator const char \\*](#) () const
- const char \* [ToUnixSlashes](#) ()  
*Convert backslash (windows style) to UNIX style slash.*
- const char \* [ToWindowsSlashes](#) ()  
*Convert forward slash (UNIX style) to windows style slash.*

### Static Public Member Functions

- static const char \* [Join](#) (const char \*path, const char \*filename)

### 10.132.1 Detailed Description

Class to manipulate file name's.

#### Note

OS independent representation of a filename (to query path, name and extension from a filename)

#### Examples

[ClinicalTrialIdentificationWorkflow.cs.](#)

## 10.132.2 Constructor & Destructor Documentation

### 10.132.2.1 Filename()

```
gdcM::Filename::Filename (
    const char * filename = "" ) [inline]
```

## 10.132.3 Member Function Documentation

### 10.132.3.1 EndWith()

```
bool gdcM::Filename::EndWith (
    const char ending[] ) const
```

Does the filename ends with a particular string ?

### 10.132.3.2 GetExtension()

```
const char * gdcM::Filename::GetExtension ( )
```

return only the extension part of a filename

### 10.132.3.3 GetFileName()

```
const char * gdcM::Filename::GetFileName ( ) const [inline]
```

Return the full filename.

### 10.132.3.4 GetName()

```
const char * gdcM::Filename::GetName ( )
```

return only the name part of a filename

### 10.132.3.5 GetPath()

```
const char * gdcM::Filename::GetPath ( )
```

Return only the path component of a filename.

### Examples

[ClinicalTrialIdentificationWorkflow.cs](#).

### 10.132.3.6 IsEmpty()

```
bool gdcm::Filename::IsEmpty ( ) const [inline]
```

return whether the filename is empty

### 10.132.3.7 IsIdentical()

```
bool gdcm::Filename::IsIdentical (
    Filename const & fn ) const
```

### 10.132.3.8 Join()

```
static const char * gdcm::Filename::Join (
    const char * path,
    const char * filename ) [static]
```

Join two paths NOT THREAD SAFE

#### Examples

[BasicAnonymizer.cs](#), and [ClinicalTrialIdentificationWorkflow.cs](#).

### 10.132.3.9 operator const char \*()

```
gdcm::Filename::operator const char * ( ) const [inline]
```

Simple operator to allow `Filename myfilename( "...")`; `const char * s = myfilename`;

### 10.132.3.10 ToUnixSlashes()

```
const char * gdcm::Filename::ToUnixSlashes ( )
```

Convert backslash (windows style) to UNIX style slash.

### 10.132.3.11 ToWindowsSlashes()

```
const char * gdcm::Filename::ToWindowsSlashes ( )
```

Convert forward slash (UNIX style) to windows style slash.

The documentation for this class was generated from the following file:

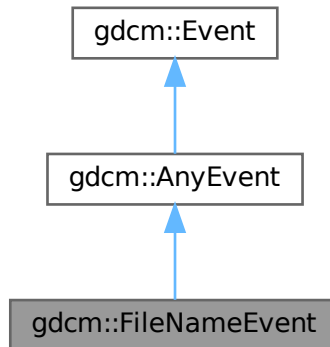
- [gdcmFilename.h](#)

## 10.133 gdcm::FileNameEvent Class Reference

[FileNameEvent](#).

```
#include <gdcmFileNameEvent.h>
```

Inheritance diagram for gdcm::FileNameEvent:



Collaboration diagram for gdcm::FileNameEvent:



### Public Types

- typedef [FileNameEvent](#) Self
- typedef [AnyEvent](#) Superclass

## Public Member Functions

- [FileNameEvent](#) (const char \*s="")
- [FileNameEvent](#) (const [Self](#) &s)
- [~FileNameEvent](#) () override=default
- bool [CheckEvent](#) (const [::gdcm::Event](#) \*e) const override
- const char \* [GetEventName](#) () const override
- const char \* [GetFileName](#) () const
- [::gdcm::Event](#) \* [MakeObject](#) () const override
- void [operator=](#) (const [Self](#) &)=delete
- void [SetFileName](#) (const char \*f)

## Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()
- [Event](#) (const [Event](#) &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) \*) const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

### 10.133.1 Detailed Description

[FileNameEvent](#).

Special type of event triggered during processing of [FileSet](#)

See also

[AnyEvent](#)

Examples

[ScanDirectory.cs](#), and [SimpleScanner.cxx](#).

### 10.133.2 Member Typedef Documentation

#### 10.133.2.1 Self

```
typedef FileNameEvent gdcm::FileNameEvent::Self
```

#### 10.133.2.2 Superclass

```
typedef AnyEvent gdcm::FileNameEvent::Superclass
```

### 10.133.3 Constructor & Destructor Documentation

#### 10.133.3.1 FileNameEvent() [1/2]

```
gdcM::FileNameEvent::FileNameEvent (
    const char * s = "" ) [inline]
```

#### 10.133.3.2 ~FileNameEvent()

```
gdcM::FileNameEvent::~~FileNameEvent ( ) [override], [default]
```

#### 10.133.3.3 FileNameEvent() [2/2]

```
gdcM::FileNameEvent::FileNameEvent (
    const Self & s ) [inline]
```

### 10.133.4 Member Function Documentation

#### 10.133.4.1 CheckEvent()

```
bool gdcM::FileNameEvent::CheckEvent (
    const ::gdcM::Event * e ) const [inline], [override]
```

#### 10.133.4.2 GetEventName()

```
const char * gdcM::FileNameEvent::GetEventName ( ) const [inline], [override], [virtual]
```

Return the StringName associated with the event.

Implements [gdcM::Event](#).

#### 10.133.4.3 GetFileName()

```
const char * gdcM::FileNameEvent::GetFileName ( ) const [inline]
```

#### Examples

[ScanDirectory.cs](#), and [SimpleScanner.cxx](#).



#### 10.133.4.4 MakeObject()

```
::gdcm::Event * gdcm::FileNameEvent::MakeObject ( ) const [inline], [override], [virtual]
```

Create an [Event](#) of this type This method work as a Factory for creating events of each particular type.

Implements [gdcm::Event](#).

#### 10.133.4.5 operator=()

```
void gdcm::FileNameEvent::operator= (
    const Self & ) [delete]
```

#### 10.133.4.6 SetFileName()

```
void gdcm::FileNameEvent::SetFileName (
    const char * f ) [inline]
```

The documentation for this class was generated from the following file:

- [gdcmFileNameEvent.h](#)

## 10.134 gdcm::FilenameGenerator Class Reference

[FilenameGenerator](#).

```
#include <gdcmFilenameGenerator.h>
```

### Public Types

- typedef std::vector< [FilenameType](#) > [FileNamesType](#)
- typedef std::string [FilenameType](#)
- typedef FileNamesType::size\_type [SizeType](#)

## Public Member Functions

- [FilenameGenerator](#) ()
- [~FilenameGenerator](#) ()=default
- bool [Generate](#) ()
  - Generate (return success)*
- const char \* [GetFilename](#) ([SizeType](#) n) const
  - Get a particular filename (call after Generate)*
- [FileNamesType](#) const & [GetFileNames](#) () const
  - Return all filenames.*
- [SizeType](#) [GetNumberOfFileNames](#) () const
- const char \* [GetPattern](#) () const
- const char \* [GetPrefix](#) () const
- void [SetNumberOfFileNames](#) ([SizeType](#) nfiles)
  - Set/Get the number of filenames to generate.*
- void [SetPattern](#) (const char \*pattern)
  - Set/Get pattern.*
- void [SetPrefix](#) (const char \*prefix)
  - Set/Get prefix.*

### 10.134.1 Detailed Description

[FilenameGenerator](#).

class to generate filenames based on a pattern (C-style)

Output will be:

for  $i = 0$ , number of filenames: `outfilename[i] = prefix + (pattern % i)`

where `pattern % i` means C-style `sprintf` of `Pattern` using value 'i'

Examples

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

### 10.134.2 Member Typedef Documentation

#### 10.134.2.1 FileNamesType

```
typedef std::vector<FilenameType> gdcm::FilenameGenerator::FileNamesType
```

#### 10.134.2.2 FilenameType

```
typedef std::string gdcm::FilenameGenerator::FilenameType
```

### 10.134.2.3 SizeType

```
typedef FilenamesType::size_type gdcm::FilenameGenerator::SizeType
```

## 10.134.3 Constructor & Destructor Documentation

### 10.134.3.1 FilenameGenerator()

```
gdcm::FilenameGenerator::FilenameGenerator ( ) [inline]
```

### 10.134.3.2 ~FilenameGenerator()

```
gdcm::FilenameGenerator::~~FilenameGenerator ( ) [default]
```

## 10.134.4 Member Function Documentation

### 10.134.4.1 Generate()

```
bool gdcm::FilenameGenerator::Generate ( )
```

Generate (return success)

#### Examples

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

### 10.134.4.2 GetFilename()

```
const char * gdcm::FilenameGenerator::GetFilename (
    SizeType n ) const
```

Get a particular filename (call after Generate)

#### Examples

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

### 10.134.4.3 GetFilenames()

```
FilenamesType const & gdcm::FilenameGenerator::GetFilenames ( ) const [inline]
```

Return all filenames.

#### 10.134.4.4 GetNumberOfFileNames()

```
SizeType gdcm::FilenameGenerator::GetNumberOfFileNames ( ) const
```

##### Examples

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

#### 10.134.4.5 GetPattern()

```
const char * gdcm::FilenameGenerator::GetPattern ( ) const [inline]
```

#### 10.134.4.6 GetPrefix()

```
const char * gdcm::FilenameGenerator::GetPrefix ( ) const [inline]
```

#### 10.134.4.7 SetNumberOfFileNames()

```
void gdcm::FilenameGenerator::SetNumberOfFileNames (
    SizeType nfiles )
```

Set/Get the number of filenames to generate.

##### Examples

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

#### 10.134.4.8 SetPattern()

```
void gdcm::FilenameGenerator::SetPattern (
    const char * pattern ) [inline]
```

Set/Get pattern.

##### Examples

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

#### 10.134.4.9 SetPrefix()

```
void gdcm::FilenameGenerator::SetPrefix (
    const char * prefix ) [inline]
```

Set/Get prefix.

The documentation for this class was generated from the following file:

- [gdcmFilenameGenerator.h](#)

## 10.135 gdcm::FileSet Class Reference

```
#include <gdcmFileSet.h>
```

### Public Types

- typedef std::vector< [FileType](#) > [FilesType](#)
- typedef std::string [FileType](#)

### Public Member Functions

- [FileSet](#) ()
- bool [AddFile](#) (const char \*filename)
- void [AddFile](#) ([File](#) const &)
- [FilesType](#) const & [GetFiles](#) () const
- void [SetFiles](#) ([FilesType](#) const &files)

### Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [FileSet](#) &d)

### 10.135.1 Detailed Description

File-set: A File-set is a collection of DICOM Files (and possibly non-DICOM Files) that share a common naming space within which [File](#) IDs are unique.

### 10.135.2 Member Typedef Documentation

#### 10.135.2.1 FilesType

```
typedef std::vector<FileType> gdcm::FileSet::FilesType
```

### 10.135.2.2 FileType

```
typedef std::string gdcm::FileSet::FileType
```

## 10.135.3 Constructor & Destructor Documentation

### 10.135.3.1 FileSet()

```
gdcm::FileSet::FileSet ( ) [inline]
```

## 10.135.4 Member Function Documentation

### 10.135.4.1 AddFile() [1/2]

```
bool gdcm::FileSet::AddFile (
    const char * filename )
```

Add a file 'filename' to the list of files. Return true on success, false in case filename could not be found on system.

### 10.135.4.2 AddFile() [2/2]

```
void gdcm::FileSet::AddFile (
    File const & ) [inline]
```

**Deprecated** . Does nothing

### 10.135.4.3 GetFiles()

```
FileType const & gdcm::FileSet::GetFiles ( ) const [inline]
```

### 10.135.4.4 SetFiles()

```
void gdcm::FileSet::SetFiles (
    FileType const & files )
```

## 10.135.5 Friends And Related Symbol Documentation

### 10.135.5.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & _os,  
    const FileSet & d ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmFileSet.h](#)

## 10.136 gdcm::FileStreamer Class Reference

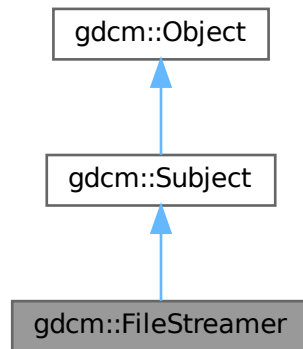
[FileStreamer.](#)

```
#include <gdcmFileStreamer.h>
```

Inheritance diagram for gdcm::FileStreamer:



Collaboration diagram for `gdcm::FileStreamer`:



### Public Member Functions

- [FileStreamer](#) ()
- [~FileStreamer](#) () override
- bool [AppendToDataElement](#) (const [Tag](#) &t, const char \*array, size\_t len)  
*Append to previously started [Tag](#) t.*
- bool [AppendToGroupDataElement](#) (const [PrivateTag](#) &pt, const char \*array, size\_t len)  
*Append to previously started private creator.*
- bool [CheckDataElement](#) (const [Tag](#) &t)
- void [CheckTemplateFileName](#) (bool check)
- bool [ReserveDataElement](#) (size\_t len)
- bool [ReserveGroupDataElement](#) (unsigned short ndataelement)
- void [SetOutputFileName](#) (const char \*filename\_native)  
*Set output filename (target file)*
- void [SetTemplateFileName](#) (const char \*filename\_native)  
*Set input DICOM template filename.*
- bool [StartDataElement](#) (const [Tag](#) &t)
- bool [StartGroupDataElement](#) (const [PrivateTag](#) &pt, size\_t maxsize=0, uint8\_t startoffset=0)
- bool [StopDataElement](#) (const [Tag](#) &t)  
*Stop appending to tag t. This will compute the proper attribute length.*
- bool [StopGroupDataElement](#) (const [PrivateTag](#) &pt)  
*Stop appending to private creator.*

### Public Member Functions inherited from [gdcm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*)



- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*) const
- [Command](#) \* [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

### Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

### Static Public Member Functions

- static [SmartPointer](#)< [FileStreamer](#) > [New](#) ()  
*for wrapped language: instantiate a reference counted object*

### Additional Inherited Members

### Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

## 10.136.1 Detailed Description

[FileStreamer](#).

This class let a user create a massive DICOM [DataSet](#) from a template DICOM file, by appending chunks of data.

This class support two mode of operation:

1. Creating a single [DataElement](#) by appending chunk after chunk of data.
2. Creating a set of [DataElement](#) within the same group, using a private creator for start. New [DataElement](#) are added any time the user defined maximum size for data element is reached.

#### Warning

any existing [DataElement](#) is removed, pick carefully which [DataElement](#) to add.

#### Examples

[FileStreaming.cs](#).

## 10.136.2 Constructor & Destructor Documentation

### 10.136.2.1 FileStreamer()

```
gdcm::FileStreamer::FileStreamer ( )
```

### 10.136.2.2 ~FileStreamer()

```
gdcm::FileStreamer::~~FileStreamer ( ) [override]
```

## 10.136.3 Member Function Documentation

### 10.136.3.1 AppendToDataElement()

```
bool gdcm::FileStreamer::AppendToDataElement (
    const Tag & t,
    const char * array,
    size_t len )
```

Append to previously started [Tag](#) t.

### 10.136.3.2 AppendToGroupDataElement()

```
bool gdcm::FileStreamer::AppendToGroupDataElement (
    const PrivateTag & pt,
    const char * array,
    size_t len )
```

Append to previously started private creator.

#### Examples

[FileStreaming.cs](#).

### 10.136.3.3 CheckDataElement()

```
bool gdcm::FileStreamer::CheckDataElement (
    const Tag & t )
```

Decide to check the Data [Element](#) to be written (default: off) The implementation has default strategy for checking validity of [DataElement](#). Currently it only support checking for the following tags:

- (7fe0,0010) Pixel Data

#### 10.136.3.4 CheckTemplateFileName()

```
void gdcm::FileStreamer::CheckTemplateFileName (
    bool check )
```

Instead of simply blindly copying the input DICOM Template file, GDCM will be used to check the input file, and correct any issues recognized within the file. Only use if you do not have control over the input template file.

#### 10.136.3.5 New()

```
static SmartPointer< FileStreamer > gdcm::FileStreamer::New ( ) [inline], [static]
```

for wrapped language: instantiate a reference counted object

#### 10.136.3.6 ReserveDataElement()

```
bool gdcm::FileStreamer::ReserveDataElement (
    size_t len )
```

Add a hint on the final size of the dataelement. When optimally chosen, this reduce the number of file in-place copying. Should be called before StartDataElement

#### 10.136.3.7 ReserveGroupDataElement()

```
bool gdcm::FileStreamer::ReserveGroupDataElement (
    unsigned short ndataelement )
```

Optimisation: pre-allocate the number of dataelement within the private group (ndataelement <= 256). Should be called before StartGroupDataElement

#### 10.136.3.8 SetOutputFileName()

```
void gdcm::FileStreamer::SetOutputFileName (
    const char * filename_native )
```

Set output filename (target file)

#### Examples

[FileStreaming.cs](#).

### 10.136.3.9 SetTemplateFileName()

```
void gdcM::FileStreamer::SetTemplateFileName (
    const char * filename_native )
```

Set input DICOM template filename.

#### Examples

[FileStreaming.cs](#).

### 10.136.3.10 StartDataElement()

```
bool gdcM::FileStreamer::StartDataElement (
    const Tag & t )
```

Start Single Data [Element](#) Operation This will delete any existing [Tag](#) t. Need to call it only once.

### 10.136.3.11 StartGroupDataElement()

```
bool gdcM::FileStreamer::StartGroupDataElement (
    const PrivateTag & pt,
    size_t maxsize = 0,
    uint8_t startoffset = 0 )
```

Start Private Group (multiple [DataElement](#)) Operation. Each newly added [DataElement](#) will have a length lower than

#### Parameters

<i>maxsize</i>	. When not specified, maxsize is set to maximum size allowed by DICOM ( $= 2^{32}$ ). startoffset can be used to specify the very first element you want to start with (instead of the first possible). <a href="#">Value</a> should be in [0x0, 0xff] This will find the first available private creator.
----------------	--

**Bug** maxsize should be a value lower than the actual total size of the buffer to be copied

#### Examples

[FileStreaming.cs](#).

### 10.136.3.12 StopDataElement()

```
bool gdcM::FileStreamer::StopDataElement (
    const Tag & t )
```

Stop appending to tag t. This will compute the proper attribute length.

### 10.136.3.13 StopGroupDataElement()

```
bool gdcm::FileStreamer::StopGroupDataElement (
    const PrivateTag & pt )
```

Stop appending to private creator.

#### Examples

[FileStreaming.cs](#).

The documentation for this class was generated from the following file:

- [gdcmFileStreamer.h](#)

## 10.137 gdcm::FileWithName Class Reference

[FileWithName](#).

```
#include <gdcmSerieHelper.h>
```

Inheritance diagram for gdcm::FileWithName:



Collaboration diagram for `gdcm::FileWithName`:



### Public Member Functions

- [FileWithName](#) ([File](#) &[f](#))

### Public Member Functions inherited from [gdcm::File](#)

- [File](#) ()
- [~File](#) () override
- [DataSet](#) & [GetDataSet](#) ()  
*Get Data Set.*
- const [DataSet](#) & [GetDataSet](#) () const  
*Get Data Set.*
- [FileMetaInformation](#) & [GetHeader](#) ()  
*Get File Meta Information.*
- const [FileMetaInformation](#) & [GetHeader](#) () const  
*Get File Meta Information.*
- `std::istream` & [Read](#) (`std::istream` &`is`)  
*Read.*
- void [SetDataSet](#) (const [DataSet](#) &`ds`)  
*Set Data Set.*
- void [SetHeader](#) (const [FileMetaInformation](#) &`fmi`)  
*Set File Meta Information.*
- `std::ostream` const & [Write](#) (`std::ostream` &`os`) const  
*Write.*

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

## Public Attributes

- std::string [filename](#)

## Additional Inherited Members

## Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

### 10.137.1 Detailed Description

[FileWithName](#).

Backward only class do not use in newer code

### 10.137.2 Constructor & Destructor Documentation

#### 10.137.2.1 FileWithName()

```
gdcm::FileWithName::FileWithName (  
    File & f ) [inline]
```

### 10.137.3 Member Data Documentation

#### 10.137.3.1 filename

```
std::string gdcm::FileWithName::filename
```

The documentation for this class was generated from the following file:

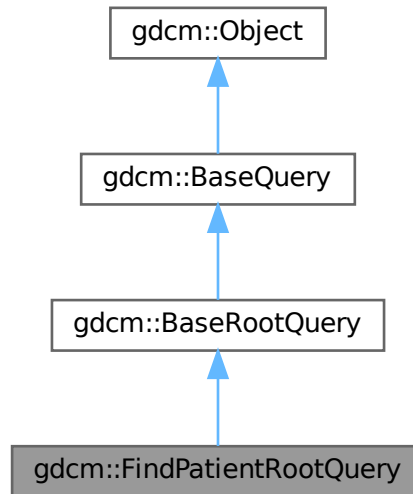
- [gdcmSerieHelper.h](#)

## 10.138 gdcm::FindPatientRootQuery Class Reference

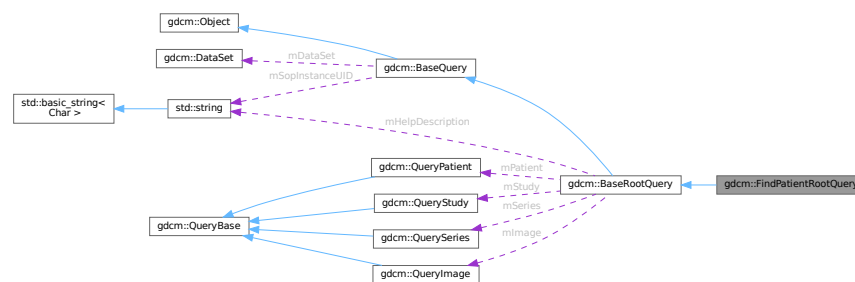
PatientRootQuery.

```
#include <gdcmFindPatientRootQuery.h>
```

Inheritance diagram for gdcm::FindPatientRootQuery:



Collaboration diagram for gdcm::FindPatientRootQuery:



### Public Member Functions

- [FindPatientRootQuery](#) ()
- [UIDs::TSName GetAbstractSyntaxUID](#) () const override
- `std::vector< Tag >` [GetTagListByLevel](#) (const [EQueryLevel](#) &inQueryLevel) override
- void [InitializeDataSet](#) (const [EQueryLevel](#) &inQueryLevel) override
- bool [ValidateQuery](#) (bool inStrict=true) const override



## Public Member Functions inherited from [gdcm::BaseRootQuery](#)

- [~BaseRootQuery](#) () override=default
- [EQueryLevel](#) [GetQueryLevelFromQueryRoot](#) ([ERootType](#) roottype)

## Public Member Functions inherited from [gdcm::BaseQuery](#)

- [~BaseQuery](#) () override
  - void [AddQueryDataSet](#) (const [DataSet](#) &ds)
  - [DataSet](#) & [GetQueryDataSet](#) ()
  - [DataSet](#) const & [GetQueryDataSet](#) () const
- Set/Get the internal representation of the query as a [DataSet](#).*
- std::string [GetSOPInstanceUID](#) () const
  - void [Print](#) (std::ostream &os) const override
  - void [SetSearchParameter](#) (const std::string &inKeyword, const std::string &inValue)
  - void [SetSearchParameter](#) (const [Tag](#) &inTag, const std::string &inValue)
  - void [SetSOPInstanceUID](#) (const std::string &iSopInstanceUID)
  - const std::ostream & [WriteHelpFile](#) (std::ostream &os)
  - bool [WriteQuery](#) (const std::string &inFileName)

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
  - [Object](#) (const [Object](#) &)
- Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
  - void [operator=](#) (const [Object](#) &)

## Friends

- class [QueryFactory](#)

## Additional Inherited Members

## Static Public Member Functions inherited from [gdcm::BaseRootQuery](#)

- static [QueryBase](#) \* [Construct](#) ([ERootType](#) inRootType, [EQueryLevel](#) qllevel)
- static int [GetQueryLevelFromString](#) (const char \*str)
- static const char \* [GetQueryLevelString](#) ([EQueryLevel](#) ql)

## Protected Member Functions inherited from [gdcm::BaseRootQuery](#)

- [BaseRootQuery](#) ()

### Protected Member Functions inherited from [gdcm::BaseQuery](#)

- [BaseQuery](#) ()
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const [DictEntry](#) &inDictEntry, const std::string &inValue)
- bool [ValidDataSet](#) (const [DataSet](#) &dataSetToValid, const [DataSet](#) &dataSetReference) const

### Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

### Protected Attributes inherited from [gdcm::BaseRootQuery](#)

- std::string [mHelpDescription](#)
- [QueryImage](#) [mImage](#)
- [QueryPatient](#) [mPatient](#)
- [ERootType](#) [mRootType](#)
- [QuerySeries](#) [mSeries](#)
- [QueryStudy](#) [mStudy](#)

### Protected Attributes inherited from [gdcm::BaseQuery](#)

- [DataSet](#) [mDataSet](#)
- std::string [mSopInstanceUID](#)

## 10.138.1 Detailed Description

PatientRootQuery.

contains: the class which will produce a dataset for c-find with patient root

## 10.138.2 Constructor & Destructor Documentation

### 10.138.2.1 FindPatientRootQuery()

```
gdcm::FindPatientRootQuery::FindPatientRootQuery ( )
```

## 10.138.3 Member Function Documentation

### 10.138.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::FindPatientRootQuery::GetAbstractSyntaxUID ( ) const [override], [virtual]
```

Implements [gdcm::BaseQuery](#).

### 10.138.3.2 GetTagListByLevel()

```
std::vector< Tag > gdcm::FindPatientRootQuery::GetTagListByLevel (
    const EQueryLevel & inQueryLevel ) [override], [virtual]
```

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implements [gdcm::BaseRootQuery](#).

### 10.138.3.3 InitializeDataSet()

```
void gdcm::FindPatientRootQuery::InitializeDataSet (
    const EQueryLevel & inQueryLevel ) [override], [virtual]
```

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmTk

Implements [gdcm::BaseRootQuery](#).

### 10.138.3.4 ValidateQuery()

```
bool gdcm::FindPatientRootQuery::ValidateQuery (
    bool inStrict = true ) const [override], [virtual]
```

have to be able to ensure that 0x8,0x52 is set (which will be true if InitializeDataSet is called...) that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional) by default, this function checks to see if the query is for finding, which is more permissive than for moving. For moving, only the unique tags are allowed. 10 Jan 2011: adding in the 'strict' mode. according to the standard (at least, how I've read it), only tags for a particular level should be allowed in a particular query (ie, just series level tags in a series level query). However, it seems that dcm4chee doesn't share that interpretation. So, if 'inStrict' is false, then tags from the current level and all higher levels are now considered valid. So, if you're doing a non-strict series-level query, tags from the patient and study level can be passed along as well.

Implements [gdcm::BaseRootQuery](#).

## 10.138.4 Friends And Related Symbol Documentation

### 10.138.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

The documentation for this class was generated from the following file:

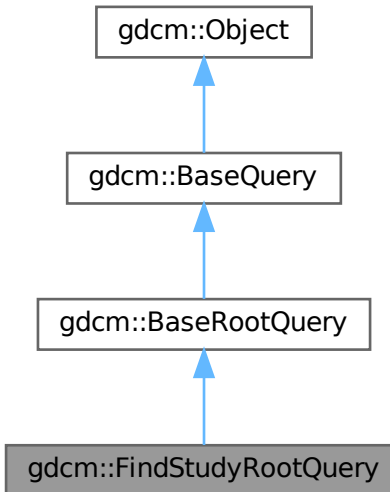
- [gdcmFindPatientRootQuery.h](#)

## 10.139 gdcm::FindStudyRootQuery Class Reference

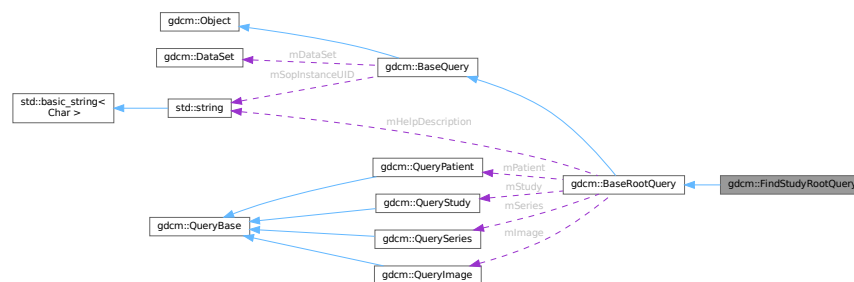
[FindStudyRootQuery](#).

```
#include <gdcmFindStudyRootQuery.h>
```

Inheritance diagram for gdcm::FindStudyRootQuery:



Collaboration diagram for gdcm::FindStudyRootQuery:



### Public Member Functions

- [FindStudyRootQuery](#) ()
- `UIDs::TSName GetAbstractSyntaxUID ()` const override
- `std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)` override
- `void InitializeDataSet (const EQueryLevel &inQueryLevel)` override
- `bool ValidateQuery (bool inStrict=true)` const override

## Public Member Functions inherited from [gdcm::BaseRootQuery](#)

- [~BaseRootQuery](#) () override=default
- [EQueryLevel](#) [GetQueryLevelFromQueryRoot](#) ([ERootType](#) roottype)

## Public Member Functions inherited from [gdcm::BaseQuery](#)

- [~BaseQuery](#) () override
  - void [AddQueryDataSet](#) (const [DataSet](#) &ds)
  - [DataSet](#) & [GetQueryDataSet](#) ()
  - [DataSet](#) const & [GetQueryDataSet](#) () const
- Set/Get the internal representation of the query as a [DataSet](#).*
- std::string [GetSOPInstanceUID](#) () const
  - void [Print](#) (std::ostream &os) const override
  - void [SetSearchParameter](#) (const std::string &inKeyword, const std::string &inValue)
  - void [SetSearchParameter](#) (const [Tag](#) &inTag, const std::string &inValue)
  - void [SetSOPInstanceUID](#) (const std::string &iSopInstanceUID)
  - const std::ostream & [WriteHelpFile](#) (std::ostream &os)
  - bool [WriteQuery](#) (const std::string &inFileName)

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
  - [Object](#) (const [Object](#) &)
- Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
  - void [operator=](#) (const [Object](#) &)

## Friends

- class [QueryFactory](#)

## Additional Inherited Members

## Static Public Member Functions inherited from [gdcm::BaseRootQuery](#)

- static [QueryBase](#) \* [Construct](#) ([ERootType](#) inRootType, [EQueryLevel](#) qllevel)
- static int [GetQueryLevelFromString](#) (const char \*str)
- static const char \* [GetQueryLevelString](#) ([EQueryLevel](#) ql)

## Protected Member Functions inherited from [gdcm::BaseRootQuery](#)

- [BaseRootQuery](#) ()

### Protected Member Functions inherited from [gdcm::BaseQuery](#)

- [BaseQuery](#) ()
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const [DictEntry](#) &inDictEntry, const std::string &inValue)
- bool [ValidDataSet](#) (const [DataSet](#) &dataSetToValid, const [DataSet](#) &dataSetReference) const

### Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

### Protected Attributes inherited from [gdcm::BaseRootQuery](#)

- std::string [mHelpDescription](#)
- [QueryImage](#) [mImage](#)
- [QueryPatient](#) [mPatient](#)
- [ERootType](#) [mRootType](#)
- [QuerySeries](#) [mSeries](#)
- [QueryStudy](#) [mStudy](#)

### Protected Attributes inherited from [gdcm::BaseQuery](#)

- [DataSet](#) [mDataSet](#)
- std::string [mSopInstanceUID](#)

## 10.139.1 Detailed Description

[FindStudyRootQuery](#).

contains: the class which will produce a dataset for C-FIND with study root

## 10.139.2 Constructor & Destructor Documentation

### 10.139.2.1 [FindStudyRootQuery](#)()

```
gdcm::FindStudyRootQuery::FindStudyRootQuery ( )
```

## 10.139.3 Member Function Documentation

### 10.139.3.1 [GetAbstractSyntaxUID](#)()

```
UIDs::TSName gdcm::FindStudyRootQuery::GetAbstractSyntaxUID ( ) const [override], [virtual]
```

Implements [gdcm::BaseQuery](#).

### 10.139.3.2 GetTagListByLevel()

```
std::vector< Tag > gdcm::FindStudyRootQuery::GetTagListByLevel (
    const EQueryLevel & inQueryLevel ) [override], [virtual]
```

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implements [gdcm::BaseRootQuery](#).

### 10.139.3.3 InitializeDataSet()

```
void gdcm::FindStudyRootQuery::InitializeDataSet (
    const EQueryLevel & inQueryLevel ) [override], [virtual]
```

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmtk

Implements [gdcm::BaseRootQuery](#).

### 10.139.3.4 ValidateQuery()

```
bool gdcm::FindStudyRootQuery::ValidateQuery (
    bool inStrict = true ) const [override], [virtual]
```

have to be able to ensure that (0008,0052) is set that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional)

Implements [gdcm::BaseRootQuery](#).

## 10.139.4 Friends And Related Symbol Documentation

### 10.139.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

The documentation for this class was generated from the following file:

- [gdcmFindStudyRootQuery.h](#)

## 10.140 gdcm::Fragment Class Reference

Class to represent a [Fragment](#).

```
#include <gdcmFragment.h>
```

Inheritance diagram for gdcm::Fragment:



Collaboration diagram for gdcm::Fragment:





## Public Member Functions

- [Fragment](#) ()
- [VL ComputeLength](#) () const
- [VL GetLength](#) () const
- template<typename TSwap >  
std::istream & [Read](#) (std::istream &is)
- template<typename TSwap >  
std::istream & [ReadBacktrack](#) (std::istream &is)
- template<typename TSwap >  
std::istream & [ReadPreValue](#) (std::istream &is)
- template<typename TSwap >  
std::istream & [ReadValue](#) (std::istream &is)
- template<typename TSwap >  
std::ostream & [Write](#) (std::ostream &os) const

## Public Member Functions inherited from [gdcm::DataElement](#)

- [DataElement](#) (const [DataElement](#) &\_val)
- [DataElement](#) (const [Tag](#) &t=[Tag](#)(0), const [VL](#) &vl=0, const [VR](#) &vr=[VR::INVALID](#))
- void [Clear](#) ()  
*Clear Data [Element](#) (make [Value](#) empty and invalidate [Tag](#) & [VR](#))*
- void [Empty](#) ()  
*Make Data [Element](#) empty (no [Value](#))*
- const [ByteValue](#) \* [GetByteValue](#) () const
- template<typename TDE >  
[VL GetLength](#) () const
- [SequenceOfFragments](#) \* [GetSequenceOfFragments](#) ()
- const [SequenceOfFragments](#) \* [GetSequenceOfFragments](#) () const
- [Tag](#) & [GetTag](#) ()
- const [Tag](#) & [GetTag](#) () const  
*Get [Tag](#).*
- [Value](#) & [GetValue](#) ()
- [Value](#) const & [GetValue](#) () const  
*Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):*
- [SmartPointer](#)< [SequenceOfItems](#) > [GetValueAsSQ](#) () const
- [VL](#) & [GetVL](#) ()
- const [VL](#) & [GetVL](#) () const  
*Get [VL](#).*
- [VR](#) const & [GetVR](#) () const
- bool [IsEmpty](#) () const  
*Check if Data [Element](#) is empty.*
- bool [IsUndefinedLength](#) () const  
*return if [Value](#) Length if of undefined length*
- bool [operator<](#) (const [DataElement](#) &de) const
- [DataElement](#) & [operator=](#) (const [DataElement](#) &)=default
- bool [operator==](#) (const [DataElement](#) &de) const
- template<typename TDE , typename TSwap >  
std::istream & [Read](#) (std::istream &is)

- `template<typename TDE , typename TSwap >`  
`std::istream & ReadOrSkip (std::istream &is, std::set< Tag > const &skiptags)`
- `template<typename TDE , typename TSwap >`  
`std::istream & ReadPreValue (std::istream &is, std::set< Tag > const &skiptags)`
- `template<typename TDE , typename TSwap >`  
`std::istream & ReadValue (std::istream &is, std::set< Tag > const &skiptags)`
- `template<typename TDE , typename TSwap >`  
`std::istream & ReadValueWithLength (std::istream &is, VL &length, std::set< Tag > const &skiptags)`
- `template<typename TDE , typename TSwap >`  
`std::istream & ReadWithLength (std::istream &is, VL &length)`
- `void SetByteValue (const char *array, VL length)`
- `void SetTag (const Tag &t)`
- `void SetValue (Value const &vl)`
- `void SetVL (const VL &vl)`
- `void SetVLToUndefined ()`
- `void SetVR (VR const &vr)`
- `template<typename TDE , typename TSwap >`  
`const std::ostream & Write (std::ostream &os) const`

## Friends

- `std::ostream & operator<< (std::ostream &os, const Fragment &val)`

## Additional Inherited Members

## Protected Types inherited from [gdcm::DataElement](#)

- `typedef SmartPointer< Value > ValuePtr`

## Protected Member Functions inherited from [gdcm::DataElement](#)

- `void SetValueFieldLength (VL vl, bool readvalues)`

## Protected Attributes inherited from [gdcm::DataElement](#)

- [Tag](#) [TagField](#)
- [ValuePtr](#) [ValueField](#)
- [VL](#) [ValueLengthField](#)
- [VR](#) [VRField](#)

## 10.140.1 Detailed Description

Class to represent a [Fragment](#).

### Examples

[DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), and [MpegVideoInfo.cs](#).

## 10.140.2 Constructor & Destructor Documentation

### 10.140.2.1 Fragment()

```
gdcm::Fragment::Fragment ( ) [inline]
```

## 10.140.3 Member Function Documentation

### 10.140.3.1 ComputeLength()

```
VL gdcm::Fragment::ComputeLength ( ) const
```

### 10.140.3.2 GetLength()

```
VL gdcm::Fragment::GetLength ( ) const
```

### 10.140.3.3 Read()

```
template<typename TSwap >  
std::istream & gdcm::Fragment::Read (  
    std::istream & is ) [inline]
```

Referenced by [gdcm::SequenceOfFragments::ReadValue\(\)](#).

### 10.140.3.4 ReadBacktrack()

```
template<typename TSwap >  
std::istream & gdcm::Fragment::ReadBacktrack (  
    std::istream & is ) [inline]
```

References [gdcmErrorMacro](#), [gdcmWarningMacro](#), and [gdcm::ParseException::SetLastElement\(\)](#).

Referenced by [gdcm::SequenceOfFragments::ReadValue\(\)](#).

### 10.140.3.5 ReadPreValue()

```
template<typename TSwap >  
std::istream & gdcm::Fragment::ReadPreValue (  
    std::istream & is ) [inline]
```

### 10.140.3.6 ReadValue()

```
template<typename TSwap >
std::istream & gdcm::Fragment::ReadValue (
    std::istream & is ) [inline]
```

References [gdcmWarningMacro](#), and [gdcm::ParseException::SetLastElement\(\)](#).

### 10.140.3.7 Write()

```
template<typename TSwap >
std::ostream & gdcm::Fragment::Write (
    std::ostream & os ) const [inline]
```

References [gdcm::ByteValue::ComputeLength\(\)](#), [gdcm::ByteValue::GetLength\(\)](#), [gdcm::ByteValue::Write\(\)](#), and [gdcm::VL::Write\(\)](#).

## 10.140.4 Friends And Related Symbol Documentation

### 10.140.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & os,
    const Fragment & val ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmFragment.h](#)

## 10.141 gdcm::Global Class Reference

[Global](#).

```
#include <gdcmGlobal.h>
```

### Public Member Functions

- [Global](#) ()
- [Global](#) (const [Global](#) &\_val)=delete
- [~Global](#) ()
- bool [Append](#) (const char \*path)
- [Defs](#) const & [GetDefs](#) () const
- [Dicts](#) & [GetDicts](#) ()
- [Dicts](#) const & [GetDicts](#) () const
- bool [LoadResourcesFiles](#) ()
- [Global](#) & [operator=](#) (const [Global](#) &\_val)=delete
- bool [Prepend](#) (const char \*path)

### Static Public Member Functions

- static [Global](#) & [GetInstance](#) ()  
*return the singleton instance*

### Protected Member Functions

- const char \* [Locate](#) (const char \*resfile) const  
*Locate a resource file.*

### Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [Global](#) &g)

## 10.141.1 Detailed Description

[Global](#).

#### Note

[Global](#) should be included in any translation unit that will use [Dict](#) or that implements the singleton pattern. It makes sure that the [Dict](#) singleton is created before and destroyed after all other singletons in GDCM.

#### Examples

[BasicAnonymizer.cs](#), [Cleaner.cs](#), [ClinicalTrialIdentificationWorkflow.cs](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenerateStandardSOPClasses.cxx](#), [PublicDict.cxx](#), [ReadAndPrintAttributes.cxx](#), and [TraverseModules.cxx](#).

## 10.141.2 Constructor & Destructor Documentation

### 10.141.2.1 [Global\(\)](#) [1/2]

```
gdcm::Global::Global ( )
```

### 10.141.2.2 [~Global\(\)](#)

```
gdcm::Global::~~Global ( )
```

### 10.141.2.3 [Global\(\)](#) [2/2]

```
gdcm::Global::Global (
    const Global & _val ) [delete]
```

### 10.141.3 Member Function Documentation

#### 10.141.3.1 Append()

```
bool gdcM::Global::Append (
    const char * path )
```

Append path at the end of the path list

##### Warning

not thread safe !

#### 10.141.3.2 GetDefs()

```
Defs const & gdcM::Global::GetDefs ( ) const
```

retrieve the default/internal (Part 3) You need to explicitly call LoadResourcesFiles before

##### Examples

[GenerateStandardSOPClasses.cxx](#), and [TraverseModules.cxx](#).

#### 10.141.3.3 GetDicts() [1/2]

```
Dicts & gdcM::Global::GetDicts ( )
```

#### 10.141.3.4 GetDicts() [2/2]

```
Dicts const & gdcM::Global::GetDicts ( ) const
```

retrieve the default/internal dicts (Part 6) This dict is filled up at load time

##### Examples

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [MrProtocol.cxx](#), [PublicDict.cxx](#), [ReadAndPrintAttributes.cxx](#), and [TraverseModules.cxx](#).

#### 10.141.3.5 GetInstance()

```
static Global & gdcM::Global::GetInstance ( ) [static]
```

return the singleton instance

##### Examples

[BasicAnonymizer.cs](#), [Cleaner.cs](#), [ClinicalTrialIdentificationWorkflow.cs](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenerateStandardSOPClasses.cxx](#), [MrProtocol.cxx](#), [PublicDict.cxx](#), and [ReadAndPrintAttributes.cxx](#).

### 10.141.3.6 LoadResourcesFiles()

```
bool gdcm::Global::LoadResourcesFiles ( )
```

Load all internal XML files, resource path need to have been set before calling this member function (see [Append/Prepend members func](#))

#### Warning

not thread safe !

#### Examples

[BasicAnonymizer.cs](#), [Cleaner.cs](#), [ClinicalTrialIdentificationWorkflow.cs](#), [GenerateStandardSOPClasses.cxx](#), and [TraverseModules.cxx](#).

### 10.141.3.7 Locate()

```
const char * gdcm::Global::Locate (
    const char * resfile ) const [protected]
```

Locate a resource file.

### 10.141.3.8 operator=()

```
Global & gdcm::Global::operator= (
    const Global & _val ) [delete]
```

### 10.141.3.9 Prepend()

```
bool gdcm::Global::Prepend (
    const char * path )
```

Prepend path at the beginning of the path list

#### Warning

not thread safe !

## 10.141.4 Friends And Related Symbol Documentation

### 10.141.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const Global & g ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmGlobal.h](#)

## 10.142 gdcm::GroupDict Class Reference

Class to represent the mapping from group number to its abbreviation and name.

```
#include <gdcmGroupDict.h>
```

### Public Types

- typedef std::vector< std::string > [GroupStringVector](#)

### Public Member Functions

- [GroupDict](#) ()
- [~GroupDict](#) ()=default
- std::string const & [GetAbbreviation](#) (uint16\_t num) const
- std::string const & [GetName](#) (uint16\_t num) const
- size\_t [Size](#) () const

### Protected Member Functions

- void [Add](#) (std::string const &abbreviation, std::string const &name)
- void [Insert](#) (uint16\_t num, std::string const &abbreviation, std::string const &name)

### Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [GroupDict](#) &\_val)

### 10.142.1 Detailed Description

Class to represent the mapping from group number to its abbreviation and name.

#### Note

Should I rewrite this class to use a std::map instead of std::vector for problem of memory consumption ?

### 10.142.2 Member Typedef Documentation

#### 10.142.2.1 GroupStringVector

```
typedef std::vector<std::string> gdcm::GroupDict::GroupStringVector
```



## 10.142.3 Constructor & Destructor Documentation

### 10.142.3.1 GroupDict()

```
gdcmm::GroupDict::GroupDict ( ) [inline]
```

### 10.142.3.2 ~GroupDict()

```
gdcmm::GroupDict::~GroupDict ( ) [default]
```

References [gdcmm::operator<<\(\)](#).

## 10.142.4 Member Function Documentation

### 10.142.4.1 Add()

```
void gdcmm::GroupDict::Add (
    std::string const & abbreviation,
    std::string const & name ) [protected]
```

### 10.142.4.2 GetAbbreviation()

```
std::string const & gdcmm::GroupDict::GetAbbreviation (
    uint16_t num ) const
```

### 10.142.4.3 GetName()

```
std::string const & gdcmm::GroupDict::GetName (
    uint16_t num ) const
```

### 10.142.4.4 Insert()

```
void gdcmm::GroupDict::Insert (
    uint16_t num,
    std::string const & abbreviation,
    std::string const & name ) [protected]
```

### 10.142.4.5 Size()

```
size_t gdcmm::GroupDict::Size ( ) const [inline]
```

## 10.142.5 Friends And Related Symbol Documentation

### 10.142.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const GroupDict & _val ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmGroupDict.h](#)

## 10.143 gdcm::IconImageFilter Class Reference

[IconImageFilter](#).

```
#include <gdcmIconImageFilter.h>
```

### Public Member Functions

- [IconImageFilter](#) ()
- [~IconImageFilter](#) ()
- bool [Extract](#) ()  
*Extract all Icon found in File.*
- [File](#) & [GetFile](#) ()
- const [File](#) & [GetFile](#) () const
- [IconImage](#) & [GetIconImage](#) (unsigned int i) const
- unsigned int [GetNumberOfIconImages](#) () const  
*Retrieve extract IconImage (need to call Extract first)*
- void [SetFile](#) (const [File](#) &f)  
*Set/Get File.*

### Protected Member Functions

- void [ExtractIconImages](#) ()
- void [ExtractVeprolIconImages](#) ()

### 10.143.1 Detailed Description

[IconImageFilter](#).

This filter will extract icons from a [File](#) This filter will loop over all known sequence (public and private) that may contains an IconImage and retrieve them. The filter will fails with a value of false if no icon can be found Since it handle both public and private icon type, one should not assume the icon is in uncompress form, some private vendor store private icon in JPEG8/JPEG12

Implementation details: This filter supports the following Icons:

- (0088,0200) Icon [Image](#) Sequence
- (0009,10,GEIIS) GE IIS Thumbnail Sequence
- (6003,10,GEMS\_Ultrasound\_ImageGroup\_001) GEMS [Image](#) Thumbnail Sequence
- (0055,30,VEPRO VIF 3.0 DATA) Icon Data
- (0055,30,VEPRO VIM 5.0 DATA) ICONDATA2

#### Warning

the icon stored in those private attribute do not conform to definition of Icon [Image](#) Sequence (do not simply copy/paste). For example some private icon can be expressed as 12bits pixel, while the DICOM standard only allow 8bits icons.

#### See also

[ImageReader](#)

#### Examples

[ExtractIconFromFile.cxx](#).

### 10.143.2 Constructor & Destructor Documentation

#### 10.143.2.1 IconImageFilter()

```
gdcm::IconImageFilter::IconImageFilter ( )
```

#### 10.143.2.2 ~IconImageFilter()

```
gdcm::IconImageFilter::~~IconImageFilter ( )
```

### 10.143.3 Member Function Documentation

#### 10.143.3.1 Extract()

```
bool gdcm::IconImageFilter::Extract ( )
```

Extract all Icon found in [File](#).

##### Examples

[ExtractIconFromFile.cxx](#).

#### 10.143.3.2 ExtractIconImages()

```
void gdcm::IconImageFilter::ExtractIconImages ( ) [protected]
```

#### 10.143.3.3 ExtractVeprolIconImages()

```
void gdcm::IconImageFilter::ExtractVeproIconImages ( ) [protected]
```

#### 10.143.3.4 GetFile() [1/2]

```
File & gdcm::IconImageFilter::GetFile ( ) [inline]
```

#### 10.143.3.5 GetFile() [2/2]

```
const File & gdcm::IconImageFilter::GetFile ( ) const [inline]
```

#### 10.143.3.6 GetIconImage()

```
IconImage & gdcm::IconImageFilter::GetIconImage (
    unsigned int i ) const
```

##### Examples

[ExtractIconFromFile.cxx](#).

### 10.143.3.7 GetNumberOfIconImages()

```
unsigned int gdcm::IconImageFilter::GetNumberOfIconImages ( ) const
```

Retrieve extract IconImage (need to call Extract first)

#### Examples

[ExtractIconFromFile.cxx](#).

### 10.143.3.8 SetFile()

```
void gdcm::IconImageFilter::SetFile (
    const File & f ) [inline]
```

Set/Get [File](#).

#### Examples

[ExtractIconFromFile.cxx](#).

The documentation for this class was generated from the following file:

- [gdcmIconImageFilter.h](#)

## 10.144 gdcm::IconImageGenerator Class Reference

[IconImageGenerator](#).

```
#include <gdcmIconImageGenerator.h>
```

### Public Member Functions

- [IconImageGenerator](#) ()
- [~IconImageGenerator](#) ()
- void [AutoPixelMinMax](#) (bool b)
- void [ConvertRGBToPaletteColor](#) (bool b)
- bool [Generate](#) ()
  - Generate Icon.*
- const [IconImage](#) & [GetIconImage](#) () const
  - Retrieve generated Icon.*
- [Pixmap](#) & [GetPixmap](#) ()
- const [Pixmap](#) & [GetPixmap](#) () const
- void [SetOutputDimensions](#) (const unsigned int dims[2])
  - Set Target dimension of output Icon.*
- void [SetOutsideValuePixel](#) (double v)
- void [SetPixelMinMax](#) (double min, double max)
- void [SetPixmap](#) (const [Pixmap](#) &p)
  - Set/Get File.*

### 10.144.1 Detailed Description

[IconImageGenerator](#).

This filter will generate a valid Icon from the Pixel Data element (an instance of [Pixmap](#)). To generate a valid Icon, one is only allowed the following Photometric Interpretation:

- MONOCHROME1
- MONOCHROME2
- PALETTE\_COLOR

The Pixel Bits Allocated is restricted to 8bits, therefore 16 bits image needs to be rescaled. By default the filter will use the full scalar range of 16bits image to rescale to unsigned 8bits. This may not be ideal for some situation, in which case the API `SetPixelMinMax` can be used to overwrite the default min,max interval used.

See also

[ImageReader](#)

Examples

[ExtractIconFromFile.cxx](#).

### 10.144.2 Constructor & Destructor Documentation

#### 10.144.2.1 IconImageGenerator()

```
gdcm::IconImageGenerator::IconImageGenerator ( )
```

#### 10.144.2.2 ~IconImageGenerator()

```
gdcm::IconImageGenerator::~~IconImageGenerator ( )
```

### 10.144.3 Member Function Documentation

#### 10.144.3.1 AutoPixelMinMax()

```
void gdcm::IconImageGenerator::AutoPixelMinMax (
    bool b )
```

Instead of explicitly specifying the min/max value for the rescale operation, let the internal mechanism compute the min/max of icon and rescale to best appropriate.

Examples

[ExtractIconFromFile.cxx](#).

### 10.144.3.2 ConvertRGBToPaletteColor()

```
void gdcm::IconImageGenerator::ConvertRGBToPaletteColor (
    bool b )
```

Converting from RGB to PALETTE\_COLOR can be a slow operation. However DICOM standard requires that color icon be described as palette. Set this boolean to false only if you understand the consequences. default value is true, false generates invalid Icon [Image](#) Sequence

### 10.144.3.3 Generate()

```
bool gdcm::IconImageGenerator::Generate ( )
```

Generate Icon.

#### Examples

[ExtractIconFromFile.cxx](#).

### 10.144.3.4 GetIconImage()

```
const IconImage & gdcm::IconImageGenerator::GetIconImage ( ) const [inline]
```

Retrieve generated Icon.

#### Examples

[ExtractIconFromFile.cxx](#).

### 10.144.3.5 GetPixmap() [1/2]

```
Pixmap & gdcm::IconImageGenerator::GetPixmap ( ) [inline]
```

### 10.144.3.6 GetPixmap() [2/2]

```
const Pixmap & gdcm::IconImageGenerator::GetPixmap ( ) const [inline]
```

### 10.144.3.7 SetOutputDimensions()

```
void gdcm::IconImageGenerator::SetOutputDimensions (
    const unsigned int dims[2] )
```

Set Target dimension of output Icon.

#### Examples

[ExtractIconFromFile.cxx](#).

#### 10.144.3.8 SetOutsideValuePixel()

```
void gdcm::IconImageGenerator::SetOutsideValuePixel (
    double v )
```

Set a pixel value that should be discarded. This happen typically for CT image, where a pixel has been used to pad outside the image (see Pixel Padding [Value](#)). Requires AutoPixelMinMax(true)

#### 10.144.3.9 SetPixelMinMax()

```
void gdcm::IconImageGenerator::SetPixelMinMax (
    double min,
    double max )
```

Override default min/max to compute best rescale for 16bits -> 8bits downscale. Typically those value can be read from the SmallestImagePixelValue LargestImagePixelValue DICOM attribute.

#### 10.144.3.10 SetPixmap()

```
void gdcm::IconImageGenerator::SetPixmap (
    const Pixmap & p ) [inline]
```

Set/Get [File](#).

#### Examples

[ExtractIconFromFile.cxx](#).

The documentation for this class was generated from the following file:

- [gdcmIconImageGenerator.h](#)

### 10.145 gdcm::ignore\_char Struct Reference

```
#include <gdcmElement.h>
```

#### Public Member Functions

- [ignore\\_char](#) (char c)

#### Public Attributes

- char [m\\_char](#)



## 10.145.1 Constructor & Destructor Documentation

### 10.145.1.1 ignore\_char()

```
gdcm::ignore_char::ignore_char (  
    char c ) [inline]
```

## 10.145.2 Member Data Documentation

### 10.145.2.1 m\_char

```
char gdcm::ignore_char::m_char
```

Referenced by [gdcm::operator>>\(\)](#).

The documentation for this struct was generated from the following file:

- [gdcmElement.h](#)

## 10.146 gdcm::Image Class Reference

[Image](#).

```
#include <gdcmImage.h>
```

Inheritance diagram for gdcm::Image:





## Public Member Functions inherited from [gdcm::Pixmap](#)

- [Pixmap](#) ()
- [~Pixmap](#) () override
- bool [AreOverlaysInPixelData](#) () const override  
*returns if Overlays are stored in the unused bit of the pixel data:*
- [Curve](#) & [GetCurve](#) (size\_t i=0)  
*Curve: group 50xx.*
- const [Curve](#) & [GetCurve](#) (size\_t i=0) const
- [IconImage](#) & [GetIconImage](#) ()
- const [IconImage](#) & [GetIconImage](#) () const  
*Set/Get Icon Image.*
- size\_t [GetNumberOfCurves](#) () const
- size\_t [GetNumberOfOverlays](#) () const
- [Overlay](#) & [GetOverlay](#) (size\_t i=0)  
*Overlay: group 60xx.*
- const [Overlay](#) & [GetOverlay](#) (size\_t i=0) const
- void [Print](#) (std::ostream &) const override
- void [RemoveOverlay](#) (size\_t i)
- void [SetIconImage](#) ([IconImage](#) const &ii)
- void [SetNumberOfCurves](#) (size\_t n)
- void [SetNumberOfOverlays](#) (size\_t n)
- bool [UnusedBitsPresentInPixelData](#) () const override  
*returns if there are unused bits in the pixel data*

## Public Member Functions inherited from [gdcm::Bitmap](#)

- [Bitmap](#) ()
- [~Bitmap](#) () override
- void [Clear](#) ()
- bool [GetBuffer](#) (char \*buffer) const  
*Access the raw data.*
- unsigned long [GetBufferLength](#) () const
- unsigned int [GetColumns](#) () const
- [DataElement](#) & [GetDataElement](#) ()
- const [DataElement](#) & [GetDataElement](#) () const
- unsigned int [GetDimension](#) (unsigned int idx) const
- const unsigned int \* [GetDimensions](#) () const  
*Return the dimension of the pixel data, first dimension (x), then 2nd (y), then 3rd (z)...*
- [LookupTable](#) & [GetLUT](#) ()
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const  
*INTERNAL do not use.*
- unsigned int [GetNumberOfDimensions](#) () const  
*Return the number of dimension of the pixel data bytes; for example 2 for a 2D matrices of values.*
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const  
*return the photometric interpretation*
- [PixelFormat](#) & [GetPixelFormat](#) ()

- const [PixelFormat](#) & [GetPixelFormat](#) () const  
*Get/Set [PixelFormat](#).*
- unsigned int [GetPlanarConfiguration](#) () const  
*return the planar configuration*
- unsigned int [GetRows](#) () const
- const [TransferSyntax](#) & [GetTransferSyntax](#) () const
- bool [IsEmpty](#) () const
- bool [IsLossy](#) () const  
*Return whether or not the image was compressed using a lossy compressor or not.*
- bool [IsTransferSyntaxCompatible](#) ([TransferSyntax](#) const &ts) const
- void [SetColumns](#) (unsigned int col)
- void [SetDataElement](#) ([DataElement](#) const &de)
- void [SetDimension](#) (unsigned int idx, unsigned int dim)
- void [SetDimensions](#) (const unsigned int dims[3])
- void [SetLossyFlag](#) (bool f)  
*Specifically set that the image was compressed using a lossy compression mechanism.*
- void [SetLUT](#) ([LookupTable](#) const &lut)  
*Set/Get LUT.*
- void [SetNeedByteSwap](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)
- void [SetRows](#) (unsigned int rows)
- void [SetTransferSyntax](#) ([TransferSyntax](#) const &ts)  
*Transfer syntax.*

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

## Additional Inherited Members

## Protected Types inherited from [gdcm::Bitmap](#)

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

### Protected Member Functions inherited from [gdcm::Bitmap](#)

- bool [ComputeLossyFlag](#) ()
- bool [GetBuffer2](#) (std::ostream &os) const
- bool [TryJPEG2000Codec](#) (char \*buffer, bool &lossyflag) const
- bool [TryJPEG2000Codec2](#) (std::ostream &os) const
- bool [TryJPEGCodec](#) (char \*buffer, bool &lossyflag) const
- bool [TryJPEGCodec2](#) (std::ostream &os) const
- bool [TryJPEGLSCCodec](#) (char \*buffer, bool &lossyflag) const
- bool [TryKAKADUCodec](#) (char \*buffer, bool &lossyflag) const
- bool [TryPVRGCodec](#) (char \*buffer, bool &lossyflag) const
- bool [TryRAWCodec](#) (char \*buffer, bool &lossyflag) const
- bool [TryRLECodec](#) (char \*buffer, bool &lossyflag) const

### Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

### Protected Attributes inherited from [gdcm::Pixmap](#)

- std::vector< [Curve](#) > [Curves](#)
- [SmartPointer](#)< [IconImage](#) > [Icon](#)
- std::vector< [Overlay](#) > [Overlays](#)

### Protected Attributes inherited from [gdcm::Bitmap](#)

- std::vector< unsigned int > [Dimensions](#)
- bool [LossyFlag](#)
- [LUTPtr](#) [LUT](#)
- bool [NeedByteSwap](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) [PF](#)
- [PhotometricInterpretation](#) [PI](#)
- [DataElement](#) [PixelData](#)
- unsigned int [PlanarConfiguration](#)
- [TransferSyntax](#) [TS](#)

### 10.146.1 Detailed Description

[Image](#).

This is the container for an [Image](#) in the general sense. From this container you should be able to request information like:

- Origin
- Dimension
- [PixelFormat](#) ... But also to retrieve the image as a raw buffer (char \*) Since we have to deal with both RAW data and JPEG stream (which internally encode all the above information) this API might seems redundant. One way to solve that would be to subclass [Image](#) with [JPEGImage](#) which would from the stream extract the header info and fill it to please [Image](#)...well except origin for instance

Basically you can see it as a storage for the Pixel Data element (7fe0,0010).

#### Warning

This class does some heuristics to guess the [Spacing](#) but is not compatible with DICOM CP-586. In case of doubt use [PixmapReader](#) instead

#### See also

[ImageReader](#) [PixmapReader](#)

#### Examples

[BasicImageAnonymizer.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [ConvertToQImage.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImage.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [ExtractIconFromFile.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenFakeImage.cxx](#), [GetArray.cs](#), [GetJPEGSamplePrecision.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [MpegVideoInfo.cs](#), [PatchFile.cxx](#), [PrintLUT.cxx](#), [ReadMultiTimesException.cxx](#), [RescaleImage.cs](#), [TemplateEmptyImage.cxx](#), [csa2img.cxx](#), [iU22tomultisc.cxx](#), and [threadgdcm.cxx](#).

### 10.146.2 Constructor & Destructor Documentation

#### 10.146.2.1 Image()

```
gdcm::Image::Image ( ) [inline]
```

#### 10.146.2.2 ~Image()

```
gdcm::Image::~Image ( ) [override], [default]
```

## 10.146.3 Member Function Documentation

### 10.146.3.1 GetDirectionCosines() [1/2]

```
const double * gdcm::Image::GetDirectionCosines ( ) const
```

Return a 6-tuples specifying the direction cosines A default value of (1,0,0,0,1,0) will be return when the direction cosines was not specified.

### 10.146.3.2 GetDirectionCosines() [2/2]

```
double gdcm::Image::GetDirectionCosines (
    unsigned int idx ) const
```

### 10.146.3.3 GetIntercept()

```
double gdcm::Image::GetIntercept ( ) const [inline]
```

### 10.146.3.4 GetOrigin() [1/2]

```
const double * gdcm::Image::GetOrigin ( ) const
```

Return a 3-tuples specifying the origin Will return (0,0,0) if the origin was not specified.

#### Examples

[HelloVizWorld.cxx](#).

### 10.146.3.5 GetOrigin() [2/2]

```
double gdcm::Image::GetOrigin (
    unsigned int idx ) const
```

### 10.146.3.6 GetSlope()

```
double gdcm::Image::GetSlope ( ) const [inline]
```

### 10.146.3.7 GetSpacing() [1/2]

```
const double * gdcm::Image::GetSpacing ( ) const
```

Return a 3-tuples specifying the spacing NOTE: 3rd value can be an arbitrary 1 value when the spacing was not specified (ex. 2D image). WARNING: when the spacing is not specifier, a default value of 1 will be returned

#### 10.146.3.8 GetSpacing() [2/2]

```
double gdcM::Image::GetSpacing (
    unsigned int idx ) const
```

#### 10.146.3.9 Print()

```
void gdcM::Image::Print (
    std::ostream & os ) const [override], [virtual]
```

print

Reimplemented from [gdcM::Bitmap](#).

#### Examples

[CompressImage.cxx](#), and [PatchFile.cxx](#).

#### 10.146.3.10 SetDirectionCosines() [1/3]

```
void gdcM::Image::SetDirectionCosines (
    const double dircos[6] )
```

#### 10.146.3.11 SetDirectionCosines() [2/3]

```
void gdcM::Image::SetDirectionCosines (
    const float dircos[6] )
```

#### 10.146.3.12 SetDirectionCosines() [3/3]

```
void gdcM::Image::SetDirectionCosines (
    unsigned int idx,
    double dircos )
```

#### 10.146.3.13 SetIntercept()

```
void gdcM::Image::SetIntercept (
    double intercept ) [inline]
```

intercept

#### Examples

[TemplateEmptyImage.cxx](#).



**10.146.3.14 SetOrigin() [1/3]**

```
void gdcm::Image::SetOrigin (
    const double origin[3] )
```

**10.146.3.15 SetOrigin() [2/3]**

```
void gdcm::Image::SetOrigin (
    const float origin[3] )
```

**10.146.3.16 SetOrigin() [3/3]**

```
void gdcm::Image::SetOrigin (
    unsigned int idx,
    double ori )
```

**10.146.3.17 SetSlope()**

```
void gdcm::Image::SetSlope (
    double slope ) [inline]
```

*slope*

**Examples**

[TemplateEmptyImage.cxx](#).

**10.146.3.18 SetSpacing() [1/2]**

```
void gdcm::Image::SetSpacing (
    const double spacing[3] )
```

**Examples**

[csa2img.cxx](#), and [iU22tomultisc.cxx](#).

**10.146.3.19 SetSpacing() [2/2]**

```
void gdcm::Image::SetSpacing (
    unsigned int idx,
    double spacing )
```

The documentation for this class was generated from the following file:

- [gdcmImage.h](#)

## 10.147 gdcm::ImageApplyLookupTable Class Reference

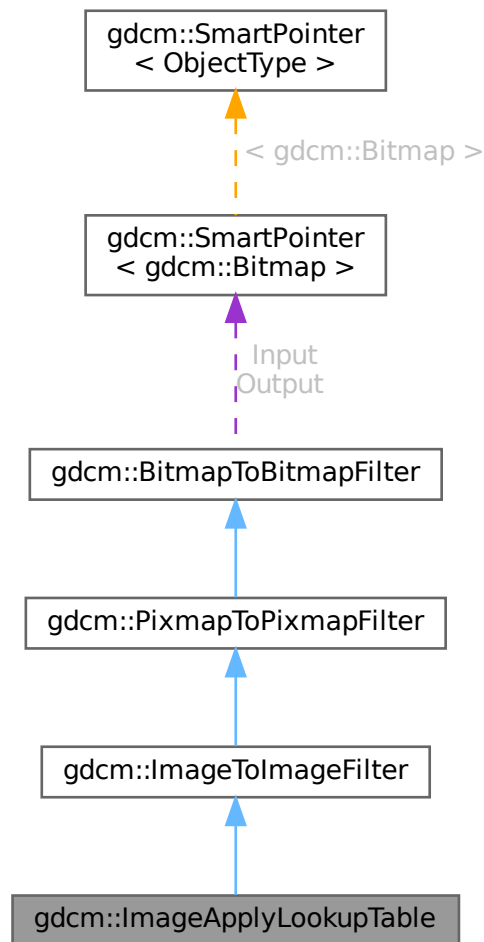
[ImageApplyLookupTable](#) class.

```
#include <gdcmImageApplyLookupTable.h>
```

Inheritance diagram for gdcm::ImageApplyLookupTable:



Collaboration diagram for gdcm::ImageApplyLookupTable:



### Public Member Functions

- [ImageApplyLookupTable](#) ()
- [~ImageApplyLookupTable](#) ()
- bool [Apply](#) ()  
    *Apply.*
- void [SetRGB8](#) (bool b)  
    *RGB8 ?*

### Public Member Functions inherited from [gdcm::ImageToImageFilter](#)

- [ImageToImageFilter](#) ()

- [~ImageToImageFilter](#) ()=default
- [Image](#) & [GetInput](#) ()
- const [Image](#) & [GetOutput](#) () const  
*Get Output image.*

### Public Member Functions inherited from [gdcm::PixmapToPixmapFilter](#)

- [PixmapToPixmapFilter](#) ()
- [~PixmapToPixmapFilter](#) ()=default
- [Pixmap](#) & [GetInput](#) ()
- const [Pixmap](#) & [GetOutput](#) () const  
*Get Output image.*
- const [Pixmap](#) & [GetOutputAsPixmap](#) () const

### Public Member Functions inherited from [gdcm::BitmapToBitmapFilter](#)

- [BitmapToBitmapFilter](#) ()
- [~BitmapToBitmapFilter](#) ()=default
- const [Bitmap](#) & [GetOutput](#) () const  
*Get Output image.*
- const [Bitmap](#) & [GetOutputAsBitmap](#) () const
- void [SetInput](#) (const [Bitmap](#) &image)  
*Set input image.*

### Additional Inherited Members

### Protected Attributes inherited from [gdcm::BitmapToBitmapFilter](#)

- [SmartPointer](#)< [Bitmap](#) > [Input](#)
- [SmartPointer](#)< [Bitmap](#) > [Output](#)

## 10.147.1 Detailed Description

[ImageApplyLookupTable](#) class.

It applies the LUT the PixelData (only PALETTE\_COLOR images) Output will be a [PhotometricInterpretation=RGB](#) image

## 10.147.2 Constructor & Destructor Documentation

### 10.147.2.1 ImageApplyLookupTable()

```
gdcm::ImageApplyLookupTable::ImageApplyLookupTable ( )
```

### 10.147.2.2 ~ImageApplyLookupTable()

```
gdcm::ImageApplyLookupTable::~ImageApplyLookupTable ( )
```

## 10.147.3 Member Function Documentation

### 10.147.3.1 Apply()

```
bool gdcm::ImageApplyLookupTable::Apply ( )
```

Apply.

### 10.147.3.2 SetRGB8()

```
void gdcm::ImageApplyLookupTable::SetRGB8 (
    bool b )
```

RGB8 ?

The documentation for this class was generated from the following file:

- [gdcmImageApplyLookupTable.h](#)

## 10.148 gdcm::ImageChangePhotometricInterpretation Class Reference

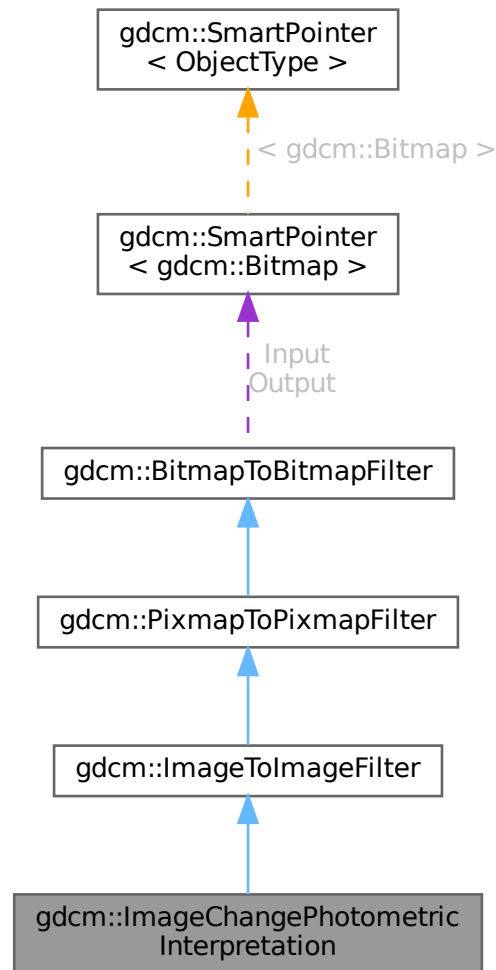
[ImageChangePhotometricInterpretation](#) class.

```
#include <gdcmImageChangePhotometricInterpretation.h>
```

Inheritance diagram for gdcm::ImageChangePhotometricInterpretation:



Collaboration diagram for `gdcM::ImageChangePhotometricInterpretation`:



### Public Member Functions

- `ImageChangePhotometricInterpretation ()`
- `~ImageChangePhotometricInterpretation ()=default`
- `bool Change ()`  
*Change.*
- `const PhotometricInterpretation & GetPhotometricInterpretation () const`
- `void SetPhotometricInterpretation (PhotometricInterpretation const &pi)`  
*Set/Get requested PhotometricInterpretation.*

**Public Member Functions inherited from [gdcm::ImageToImageFilter](#)**

- [ImageToImageFilter](#) ()
- [~ImageToImageFilter](#) ()=default
- [Image](#) & [GetInput](#) ()
- const [Image](#) & [GetOutput](#) () const

*Get Output image.*

**Public Member Functions inherited from [gdcm::PixmapToPixmapFilter](#)**

- [PixmapToPixmapFilter](#) ()
- [~PixmapToPixmapFilter](#) ()=default
- [Pixmap](#) & [GetInput](#) ()
- const [Pixmap](#) & [GetOutput](#) () const
- const [Pixmap](#) & [GetOutputAsPixmap](#) () const

*Get Output image.*

**Public Member Functions inherited from [gdcm::BitmapToBitmapFilter](#)**

- [BitmapToBitmapFilter](#) ()
- [~BitmapToBitmapFilter](#) ()=default
- const [Bitmap](#) & [GetOutput](#) () const
- const [Bitmap](#) & [GetOutputAsBitmap](#) () const
- void [SetInput](#) (const [Bitmap](#) &image)

*Get Output image.*

*Set input image.*

**Static Public Member Functions**

- template<typename T >  
static void [RGB2YBR](#) (T ybr[3], const T rgb[3], unsigned short storedbits=8)
- template<typename T >  
static void [YBR2RGB](#) (T rgb[3], const T ybr[3], unsigned short storedbits=8)

**Protected Member Functions**

- bool [ChangeMonochrome](#) ()
- bool [ChangeRGB2YBR](#) ()
- bool [ChangeYBR2RGB](#) ()

**Additional Inherited Members****Protected Attributes inherited from [gdcm::BitmapToBitmapFilter](#)**

- [SmartPointer](#)< [Bitmap](#) > [Input](#)
- [SmartPointer](#)< [Bitmap](#) > [Output](#)

### 10.148.1 Detailed Description

[ImageChangePhotometricInterpretation](#) class.

Class to change the Photometric Interpretation of an input DICOM

### 10.148.2 Constructor & Destructor Documentation

#### 10.148.2.1 ImageChangePhotometricInterpretation()

```
gdcm::ImageChangePhotometricInterpretation::ImageChangePhotometricInterpretation ( ) [inline]
```

#### 10.148.2.2 ~ImageChangePhotometricInterpretation()

```
gdcm::ImageChangePhotometricInterpretation::~~ImageChangePhotometricInterpretation ( ) [default]
```

### 10.148.3 Member Function Documentation

#### 10.148.3.1 Change()

```
bool gdcm::ImageChangePhotometricInterpretation::Change ( )
```

Change.

#### 10.148.3.2 ChangeMonochrome()

```
bool gdcm::ImageChangePhotometricInterpretation::ChangeMonochrome ( ) [protected]
```

#### 10.148.3.3 ChangeRGB2YBR()

```
bool gdcm::ImageChangePhotometricInterpretation::ChangeRGB2YBR ( ) [protected]
```

#### 10.148.3.4 ChangeYBR2RGB()

```
bool gdcm::ImageChangePhotometricInterpretation::ChangeYBR2RGB ( ) [protected]
```

#### 10.148.3.5 GetPhotometricInterpretation()

```
const PhotometricInterpretation & gdcm::ImageChangePhotometricInterpretation::GetPhotometric↵  
Interpretation ( ) const [inline]
```



### 10.148.3.6 RGB2YBR()

```
template<typename T >
void gdcm::ImageChangePhotometricInterpretation::RGB2YBR (
    T ybr[3],
    const T rgb[3],
    unsigned short storedbits = 8 ) [static]
```

colorspace conversion (based on CCIR Recommendation 601-2) -> T.871

References [gdcm::Round\(\)](#).

### 10.148.3.7 SetPhotometricInterpretation()

```
void gdcm::ImageChangePhotometricInterpretation::SetPhotometricInterpretation (
    PhotometricInterpretation const & pi ) [inline]
```

Set/Get requested [PhotometricInterpretation](#).

### 10.148.3.8 YBR2RGB()

```
template<typename T >
void gdcm::ImageChangePhotometricInterpretation::YBR2RGB (
    T rgb[3],
    const T ybr[3],
    unsigned short storedbits = 8 ) [static]
```

References [gdcm::Round\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmImageChangePhotometricInterpretation.h](#)

## 10.149 gdcm::ImageChangePlanarConfiguration Class Reference

[ImageChangePlanarConfiguration](#) class.

```
#include <gdcmImageChangePlanarConfiguration.h>
```

Inheritance diagram for `gdcm::ImageChangePlanarConfiguration`:



Collaboration diagram for gdcm::ImageChangePlanarConfiguration:



### Public Member Functions

- `ImageChangePlanarConfiguration ()`
- `~ImageChangePlanarConfiguration ()=default`
- `bool Change ()`  
*Change.*
- `unsigned int GetPlanarConfiguration () const`
- `void SetPlanarConfiguration (unsigned int pc)`  
*Set/Get requested PlanarConfiguration.*

### Public Member Functions inherited from [gdcm::ImageToImageFilter](#)

- [ImageToImageFilter](#) ()
- [~ImageToImageFilter](#) ()=default
- [Image](#) & [GetInput](#) ()
- const [Image](#) & [GetOutput](#) () const

*Get Output image.*

### Public Member Functions inherited from [gdcm::PixmapToPixmapFilter](#)

- [PixmapToPixmapFilter](#) ()
  - [~PixmapToPixmapFilter](#) ()=default
  - [Pixmap](#) & [GetInput](#) ()
  - const [Pixmap](#) & [GetOutput](#) () const
- Get Output image.*
- const [Pixmap](#) & [GetOutputAsPixmap](#) () const

### Public Member Functions inherited from [gdcm::BitmapToBitmapFilter](#)

- [BitmapToBitmapFilter](#) ()
  - [~BitmapToBitmapFilter](#) ()=default
  - const [Bitmap](#) & [GetOutput](#) () const
- Get Output image.*
- const [Bitmap](#) & [GetOutputAsBitmap](#) () const
  - void [SetInput](#) (const [Bitmap](#) &image)
- Set input image.*

### Static Public Member Functions

- template<typename T >  
static size\_t [RGBPixelsToRGBPlanes](#) (T \*r, T \*g, T \*b, const T \*rgb, size\_t s)
- template<typename T >  
static size\_t [RGBPlanesToRGBPixels](#) (T \*out, const T \*r, const T \*g, const T \*b, size\_t s)

### Additional Inherited Members

### Protected Attributes inherited from [gdcm::BitmapToBitmapFilter](#)

- [SmartPointer](#)< [Bitmap](#) > [Input](#)
- [SmartPointer](#)< [Bitmap](#) > [Output](#)

## 10.149.1 Detailed Description

[ImageChangePlanarConfiguration](#) class.

Class to change the Planar configuration of an input DICOM By default it will change into the more usual representation: PlanarConfiguration = 0

## 10.149.2 Constructor & Destructor Documentation

### 10.149.2.1 ImageChangePlanarConfiguration()

```
gdcm::ImageChangePlanarConfiguration::ImageChangePlanarConfiguration ( ) [inline]
```

### 10.149.2.2 ~ImageChangePlanarConfiguration()

```
gdcm::ImageChangePlanarConfiguration::~~ImageChangePlanarConfiguration ( ) [default]
```

## 10.149.3 Member Function Documentation

### 10.149.3.1 Change()

```
bool gdcm::ImageChangePlanarConfiguration::Change ( )
```

Change.

### 10.149.3.2 GetPlanarConfiguration()

```
unsigned int gdcm::ImageChangePlanarConfiguration::GetPlanarConfiguration ( ) const [inline]
```

### 10.149.3.3 RGBPixelsToRGBPlanes()

```
template<typename T >
size_t gdcm::ImageChangePlanarConfiguration::RGBPixelsToRGBPlanes (
    T * r,
    T * g,
    T * b,
    const T * rgb,
    size_t s ) [static]
```

Convert a regular RGB pixel image (R,G,B,R,G,B...) into a planar R,G,B image (R,R...,G,G...B,B)

#### Warning

this works on a frame basis, you need to loop over all frames in multiple frames image to apply this function

#### 10.149.3.4 RGBPlanesToRGBPixels()

```
template<typename T >
size_t gdcm::ImageChangePlanarConfiguration::RGBPlanesToRGBPixels (
    T * out,
    const T * r,
    const T * g,
    const T * b,
    size_t s ) [static]
```

s is the size of one plane (r,g or b). Thus the output buffer needs to be at least 3\*s bytes long s can be seen as the number of RGB pixels in the output

#### 10.149.3.5 SetPlanarConfiguration()

```
void gdcm::ImageChangePlanarConfiguration::SetPlanarConfiguration (
    unsigned int pc ) [inline]
```

Set/Get requested PlanarConfiguration.

The documentation for this class was generated from the following file:

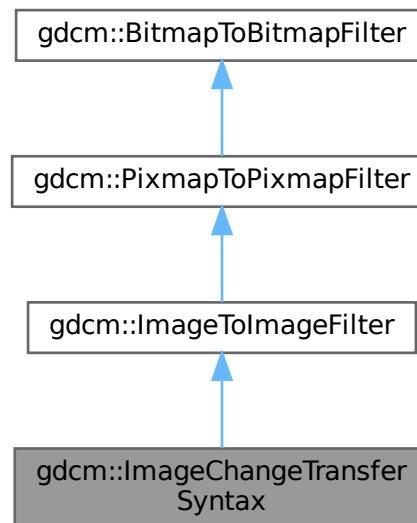
- [gdcmImageChangePlanarConfiguration.h](#)

### 10.150 gdcm::ImageChangeTransferSyntax Class Reference

[ImageChangeTransferSyntax](#) class.

```
#include <gdcmImageChangeTransferSyntax.h>
```

Inheritance diagram for gdcm::ImageChangeTransferSyntax:



Collaboration diagram for gdcm::ImageChangeTransferSyntax:



### Public Member Functions

- `ImageChangeTransferSyntax ()`
- `~ImageChangeTransferSyntax ()=default`
- `bool Change ()`  
*Change.*
- `const TransferSyntax & GetTransferSyntax () const`  
*Get Transfer Syntax.*
- `void SetCompressIconImage (bool b)`
- `void SetForce (bool f)`
- `void SetTransferSyntax (const TransferSyntax &ts)`  
*Set target Transfer Syntax.*
- `void SetUserCodec (ImageCodec *ic)`

### Public Member Functions inherited from [gdcm::ImageToImageFilter](#)

- [ImageToImageFilter](#) ()
- [~ImageToImageFilter](#) ()=default
- [Image](#) & [GetInput](#) ()
- const [Image](#) & [GetOutput](#) () const

*Get Output image.*

### Public Member Functions inherited from [gdcm::PixmapToPixmapFilter](#)

- [PixmapToPixmapFilter](#) ()
  - [~PixmapToPixmapFilter](#) ()=default
  - [Pixmap](#) & [GetInput](#) ()
  - const [Pixmap](#) & [GetOutput](#) () const
- Get Output image.*
- const [Pixmap](#) & [GetOutputAsPixmap](#) () const

### Public Member Functions inherited from [gdcm::BitmapToBitmapFilter](#)

- [BitmapToBitmapFilter](#) ()
  - [~BitmapToBitmapFilter](#) ()=default
  - const [Bitmap](#) & [GetOutput](#) () const
- Get Output image.*
- const [Bitmap](#) & [GetOutputAsBitmap](#) () const
  - void [SetInput](#) (const [Bitmap](#) &image)
- Set input image.*

### Protected Member Functions

- bool [TryJPEG2000Codec](#) (const [DataElement](#) &pixelde, [Bitmap](#) const &input, [Bitmap](#) &output)
- bool [TryJPEGCodec](#) (const [DataElement](#) &pixelde, [Bitmap](#) const &input, [Bitmap](#) &output)
- bool [TryJPEGLSCCodec](#) (const [DataElement](#) &pixelde, [Bitmap](#) const &input, [Bitmap](#) &output)
- bool [TryRAWCodec](#) (const [DataElement](#) &pixelde, [Bitmap](#) const &input, [Bitmap](#) &output)
- bool [TryRLECodec](#) (const [DataElement](#) &pixelde, [Bitmap](#) const &input, [Bitmap](#) &output)

### Additional Inherited Members

### Protected Attributes inherited from [gdcm::BitmapToBitmapFilter](#)

- [SmartPointer](#)< [Bitmap](#) > [Input](#)
- [SmartPointer](#)< [Bitmap](#) > [Output](#)



## 10.150.1 Detailed Description

[ImageChangeTransferSyntax](#) class.

Class to change the transfer syntax of an input DICOM

If only Force param is set but no input [TransferSyntax](#) is set, it is assumed that user only wants to inspect encapsulated stream (advanced dev. option).

When using UserCodec it is very important that the [TransferSyntax](#) (as set in SetTransferSyntax) is actually understood by UserCodec (ie. UserCodec->CanCode( TransferSyntax ) ). Otherwise the behavior is to use a default codec.

See also

[JPEGCodec](#) [JPEGLSCodec](#) [JPEG2000Codec](#)

Examples

[BasicImageAnonymizer.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), and [StandardizeFiles.cs](#).

## 10.150.2 Constructor & Destructor Documentation

### 10.150.2.1 ImageChangeTransferSyntax()

```
gdcm::ImageChangeTransferSyntax::ImageChangeTransferSyntax ( ) [inline]
```

### 10.150.2.2 ~ImageChangeTransferSyntax()

```
gdcm::ImageChangeTransferSyntax::~~ImageChangeTransferSyntax ( ) [default]
```

## 10.150.3 Member Function Documentation

### 10.150.3.1 Change()

```
bool gdcm::ImageChangeTransferSyntax::Change ( )
```

Change.

Examples

[BasicImageAnonymizer.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), and [StandardizeFiles.cs](#).

### 10.150.3.2 GetTransferSyntax()

```
const TransferSyntax & gdcm::ImageChangeTransferSyntax::GetTransferSyntax ( ) const [inline]
```

Get Transfer Syntax.

### 10.150.3.3 SetCompressIconImage()

```
void gdcm::ImageChangeTransferSyntax::SetCompressIconImage (
    bool b ) [inline]
```

Decide whether or not to also compress the Icon [Image](#) using the same Transfer Syntax. Default is to simply decompress icon image

#### Examples

[StandardizeFiles.cs](#).

### 10.150.3.4 SetForce()

```
void gdcm::ImageChangeTransferSyntax::SetForce (
    bool f ) [inline]
```

When target Transfer Syntax is identical to input target syntax, no operation is actually done. This is an issue when someone wants to re-compress using GDCM internal implementation a JPEG (for example) image

#### Examples

[StandardizeFiles.cs](#).

### 10.150.3.5 SetTransferSyntax()

```
void gdcm::ImageChangeTransferSyntax::SetTransferSyntax (
    const TransferSyntax & ts ) [inline]
```

Set target Transfer Syntax.

#### Examples

[BasicImageAnonymizer.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), and [StandardizeFiles.cs](#).

### 10.150.3.6 SetUserCodec()

```
void gdcm::ImageChangeTransferSyntax::SetUserCodec (
    ImageCodec * ic ) [inline]
```

Allow user to specify exactly which codec to use. this is needed to specify special qualities or compression option.

#### Warning

if the codec 'ic' is not compatible with the [TransferSyntax](#) requested, it will not be used. It is the user responsibility to check that `UserCodec->CanCode( TransferSyntax )`

#### Examples

[CompressLossyJPEG.cs](#).

### 10.150.3.7 TryJPEG2000Codec()

```
bool gdcm::ImageChangeTransferSyntax::TryJPEG2000Codec (
    const DataElement & pixelde,
    Bitmap const & input,
    Bitmap & output ) [protected]
```

### 10.150.3.8 TryJPEGCodec()

```
bool gdcm::ImageChangeTransferSyntax::TryJPEGCodec (
    const DataElement & pixelde,
    Bitmap const & input,
    Bitmap & output ) [protected]
```

### 10.150.3.9 TryJPEGLSCodec()

```
bool gdcm::ImageChangeTransferSyntax::TryJPEGLSCodec (
    const DataElement & pixelde,
    Bitmap const & input,
    Bitmap & output ) [protected]
```

### 10.150.3.10 TryRAWCodec()

```
bool gdcm::ImageChangeTransferSyntax::TryRAWCodec (
    const DataElement & pixelde,
    Bitmap const & input,
    Bitmap & output ) [protected]
```

### 10.150.3.11 TryRLECodec()

```
bool gdcmm::ImageChangeTransferSyntax::TryRLECodec (
    const DataElement & pixelde,
    Bitmap const & input,
    Bitmap & output ) [protected]
```

The documentation for this class was generated from the following file:

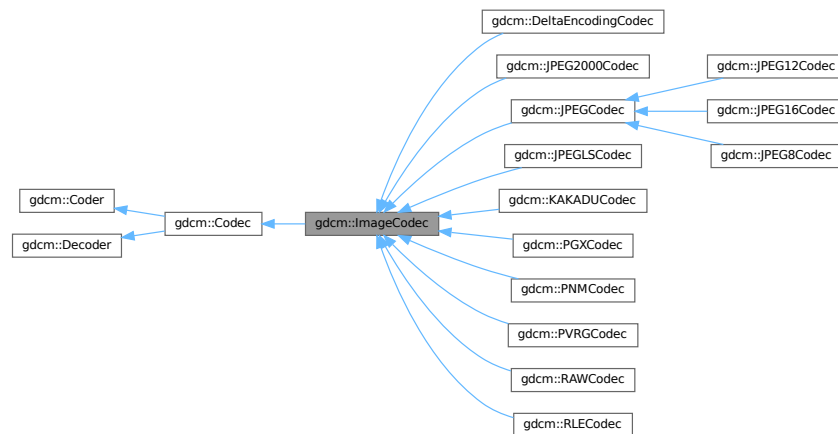
- [gdcmmImageChangeTransferSyntax.h](#)

## 10.151 gdcmm::ImageCodec Class Reference

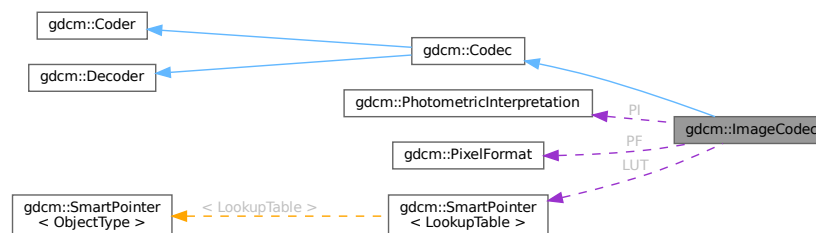
[ImageCodec.](#)

```
#include <gdcmmImageCodec.h>
```

Inheritance diagram for gdcmm::ImageCodec:



Collaboration diagram for gdcmm::ImageCodec:



**Public Member Functions**

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &) const override  
*Return whether this coder support this transfer syntax (can code it)*
- bool [CanDecode](#) ([TransferSyntax](#) const &) const override  
*Return whether this decoder support this transfer syntax (can decode it)*
- bool [CleanupUnusedBits](#) (char \*data, size\_t datalen)
- virtual [ImageCodec](#) \* [Clone](#) () const =0
- bool [Decode](#) ([DataElement](#) const &is\_, [DataElement](#) &os) override  
*Decode.*
- const unsigned int \* [GetDimensions](#) () const
- virtual bool [GetHeaderInfo](#) (std::istream &is\_, [TransferSyntax](#) &ts)
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- virtual void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)

**Public Member Functions inherited from [gdcm::Coder](#)**

- virtual [~Coder](#) ()=default
- virtual bool [Code](#) ([DataElement](#) const &in\_, [DataElement](#) &out\_)  
*Code.*

**Public Member Functions inherited from [gdcm::Decoder](#)**

- virtual [~Decoder](#) ()=default

**Protected Types**

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

### Protected Member Functions

- virtual bool [AppendFrameEncode](#) (std::ostream &out, const char \*data, size\_t datalen)
- virtual bool [AppendRowEncode](#) (std::ostream &out, const char \*data, size\_t datalen)
- bool [DecodeByStreams](#) (std::istream &is\_, std::ostream &os) override
- bool [DoByteSwap](#) (std::istream &is\_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is\_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is\_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is\_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is\_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is\_, std::ostream &os)
- virtual bool [IsFrameEncoder](#) ()
- virtual bool [IsRowEncoder](#) ()
- virtual bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)
- virtual bool [StartEncode](#) (std::ostream &os)
- virtual bool [StopEncode](#) (std::ostream &os)

### Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char \*bv, unsigned long len, std::ostream &os)

### Protected Attributes

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) LUT
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) PF
- [PhotometricInterpretation](#) PI
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

### Friends

- class [FileChangeTransferSyntax](#)
- class [ImageChangePhotometricInterpretation](#)

### 10.151.1 Detailed Description

[ImageCodec](#).

#### Note

Main codec, this is a central place for all implementation

#### Examples

[FileChangeTSLossy.cs](#).

### 10.151.2 Member Typedef Documentation

#### 10.151.2.1 LUTPtr

```
typedef SmartPointer<LookupTable> gdcm::ImageCodec::LUTPtr [protected]
```

### 10.151.3 Constructor & Destructor Documentation

#### 10.151.3.1 ImageCodec()

```
gdcm::ImageCodec::ImageCodec ( )
```

#### 10.151.3.2 ~ImageCodec()

```
gdcm::ImageCodec::~~ImageCodec ( ) [override]
```

### 10.151.4 Member Function Documentation

#### 10.151.4.1 AppendFrameEncode()

```
virtual bool gdcm::ImageCodec::AppendFrameEncode (  
    std::ostream & out,  
    const char * data,  
    size_t datalen ) [protected], [virtual]
```

Reimplemented in [gdcm::JPEG2000Codec](#), [gdcm::JPEGCodec](#), [gdcm::JPEGLSCodec](#), and [gdcm::RLECodec](#).

#### 10.151.4.2 AppendRowEncode()

```
virtual bool gdcm::ImageCodec::AppendRowEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [protected], [virtual]
```

Reimplemented in [gdcm::JPEG2000Codec](#), [gdcm::JPEGCodec](#), [gdcm::JPEGLSCodec](#), and [gdcm::RLECodec](#).

#### 10.151.4.3 CanCode()

```
bool gdcm::ImageCodec::CanCode (
    TransferSyntax const & ) const [inline], [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Implements [gdcm::Coder](#).

Reimplemented in [gdcm::JPEG2000Codec](#), [gdcm::JPEGCodec](#), [gdcm::JPEGLSCodec](#), [gdcm::KAKADUCodec](#), [gdcm::PGXCodec](#), [gdcm::PNMCodec](#), [gdcm::PVRGCodec](#), [gdcm::RAWCodec](#), and [gdcm::RLECodec](#).

#### 10.151.4.4 CanDecode()

```
bool gdcm::ImageCodec::CanDecode (
    TransferSyntax const & ) const [inline], [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Implements [gdcm::Decoder](#).

Reimplemented in [gdcm::JPEG2000Codec](#), [gdcm::JPEGCodec](#), [gdcm::JPEGLSCodec](#), [gdcm::KAKADUCodec](#), [gdcm::PGXCodec](#), [gdcm::PNMCodec](#), [gdcm::PVRGCodec](#), [gdcm::RAWCodec](#), and [gdcm::RLECodec](#).

#### 10.151.4.5 CleanupUnusedBits()

```
bool gdcm::ImageCodec::CleanupUnusedBits (
    char * data,
    size_t datalen )
```

#### 10.151.4.6 Clone()

```
virtual ImageCodec * gdcm::ImageCodec::Clone ( ) const [pure virtual]
```

Implemented in [gdcm::JPEG2000Codec](#), [gdcm::JPEGCodec](#), [gdcm::JPEGLSCodec](#), [gdcm::KAKADUCodec](#), [gdcm::PGXCodec](#), [gdcm::PNMCodec](#), [gdcm::PVRGCodec](#), [gdcm::RAWCodec](#), and [gdcm::RLECodec](#).



#### 10.151.4.7 Decode()

```
bool gdcm::ImageCodec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcm::Decoder](#).

Reimplemented in [gdcm::JPEG2000Codec](#), [gdcm::JPEGCodec](#), [gdcm::JPEGLSCCodec](#), [gdcm::KAKADUCodec](#), [gdcm::PVRGCodec](#), [gdcm::RAWCodec](#), and [gdcm::RLECodec](#).

#### 10.151.4.8 DecodeByStreams()

```
bool gdcm::ImageCodec::DecodeByStreams (
    std::istream & is_,
    std::ostream & os ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::Decoder](#).

Reimplemented in [gdcm::JPEG12Codec](#), [gdcm::JPEG16Codec](#), [gdcm::JPEG2000Codec](#), [gdcm::JPEG8Codec](#), [gdcm::JPEGCodec](#), [gdcm::RAWCodec](#), and [gdcm::RLECodec](#).

#### 10.151.4.9 DoByteSwap()

```
bool gdcm::ImageCodec::DoByteSwap (
    std::istream & is_,
    std::ostream & os ) [protected]
```

#### 10.151.4.10 DoInvertMonochrome()

```
bool gdcm::ImageCodec::DoInvertMonochrome (
    std::istream & is_,
    std::ostream & os ) [protected]
```

#### 10.151.4.11 DoOverlayCleanup()

```
bool gdcm::ImageCodec::DoOverlayCleanup (
    std::istream & is_,
    std::ostream & os ) [protected]
```

#### 10.151.4.12 DoPaddedCompositePixelCode()

```
bool gdcm::ImageCodec::DoPaddedCompositePixelCode (
    std::istream & is_,
    std::ostream & os ) [protected]
```

#### 10.151.4.13 DoPlanarConfiguration()

```
bool gdcM::ImageCodec::DoPlanarConfiguration (
    std::istream & is_,
    std::ostream & os ) [protected]
```

#### 10.151.4.14 DoSimpleCopy()

```
bool gdcM::ImageCodec::DoSimpleCopy (
    std::istream & is_,
    std::ostream & os ) [protected]
```

#### 10.151.4.15 DoYBR()

```
bool gdcM::ImageCodec::DoYBR (
    std::istream & is_,
    std::ostream & os ) [protected]
```

#### 10.151.4.16 DoYBRFull422()

```
bool gdcM::ImageCodec::DoYBRFull422 (
    std::istream & is_,
    std::ostream & os ) [protected]
```

#### 10.151.4.17 GetDimensions()

```
const unsigned int * gdcM::ImageCodec::GetDimensions ( ) const [inline]
```

#### 10.151.4.18 GetHeaderInfo()

```
virtual bool gdcM::ImageCodec::GetHeaderInfo (
    std::istream & is_,
    TransferSyntax & ts ) [virtual]
```

Reimplemented in [gdcM::JPEG12Codec](#), [gdcM::JPEG16Codec](#), [gdcM::JPEG2000Codec](#), [gdcM::JPEG8Codec](#), [gdcM::JPEGCodec](#), [gdcM::JPEGLSCCodec](#), [gdcM::PGXCodec](#), [gdcM::PNMCodec](#), [gdcM::RAWCodec](#), and [gdcM::RLECodec](#).

#### 10.151.4.19 GetLossyFlag()

```
bool gdcM::ImageCodec::GetLossyFlag ( ) const
```

#### 10.151.4.20 GetLUT()

```
const LookupTable & gdcm::ImageCodec::GetLUT ( ) const [inline]
```

#### 10.151.4.21 GetNeedByteSwap()

```
bool gdcm::ImageCodec::GetNeedByteSwap ( ) const [inline]
```

#### 10.151.4.22 GetNumberOfDimensions()

```
unsigned int gdcm::ImageCodec::GetNumberOfDimensions ( ) const
```

#### 10.151.4.23 GetPhotometricInterpretation()

```
const PhotometricInterpretation & gdcm::ImageCodec::GetPhotometricInterpretation ( ) const
```

#### 10.151.4.24 GetPixelFormat() [1/2]

```
PixelFormat & gdcm::ImageCodec::GetPixelFormat ( ) [inline]
```

#### Examples

[GetJPEGSamplePrecision.cxx](#).

#### 10.151.4.25 GetPixelFormat() [2/2]

```
const PixelFormat & gdcm::ImageCodec::GetPixelFormat ( ) const [inline]
```

#### 10.151.4.26 GetPlanarConfiguration()

```
unsigned int gdcm::ImageCodec::GetPlanarConfiguration ( ) const [inline]
```

#### 10.151.4.27 IsFrameEncoder()

```
virtual bool gdcm::ImageCodec::IsFrameEncoder ( ) [protected], [virtual]
```

Reimplemented in [gdcm::JPEG2000Codec](#), [gdcm::JPEGCodec](#), [gdcm::JPEGLSCodec](#), and [gdcm::RLECodec](#).

#### 10.151.4.28 IsLossy()

```
bool gdcM::ImageCodec::IsLossy ( ) const
```

#### 10.151.4.29 IsRowEncoder()

```
virtual bool gdcM::ImageCodec::IsRowEncoder ( ) [protected], [virtual]
```

Reimplemented in [gdcM::JPEG2000Codec](#), [gdcM::JPEGCodec](#), [gdcM::JPEGLSCodec](#), and [gdcM::RLECodec](#).

#### 10.151.4.30 IsValid()

```
virtual bool gdcM::ImageCodec::IsValid (
    PhotometricInterpretation const & pi ) [protected], [virtual]
```

Reimplemented in [gdcM::JPEGCodec](#).

#### 10.151.4.31 SetDimensions() [1/2]

```
void gdcM::ImageCodec::SetDimensions (
    const std::vector< unsigned int > & d )
```

#### 10.151.4.32 SetDimensions() [2/2]

```
void gdcM::ImageCodec::SetDimensions (
    const unsigned int d[3] )
```

#### Examples

[ExtractIconFromFile.cxx](#).

#### 10.151.4.33 SetLossyFlag()

```
void gdcM::ImageCodec::SetLossyFlag (
    bool l )
```

#### 10.151.4.34 SetLUT()

```
void gdcM::ImageCodec::SetLUT (
    LookupTable const & lut ) [inline]
```

#### Examples

[ExtractIconFromFile.cxx](#).

**10.151.4.35 SetNeedByteSwap()**

```
void gdcm::ImageCodec::SetNeedByteSwap (
    bool b ) [inline]
```

**10.151.4.36 SetNeedOverlayCleanup()**

```
void gdcm::ImageCodec::SetNeedOverlayCleanup (
    bool b ) [inline]
```

**10.151.4.37 SetNumberOfDimensions()**

```
void gdcm::ImageCodec::SetNumberOfDimensions (
    unsigned int dim )
```

**10.151.4.38 SetPhotometricInterpretation()**

```
void gdcm::ImageCodec::SetPhotometricInterpretation (
    PhotometricInterpretation const & pi )
```

**Examples**

[ExtractIconFromFile.cxx](#).

**10.151.4.39 SetPixelFormat()**

```
virtual void gdcm::ImageCodec::SetPixelFormat (
    PixelFormat const & pf ) [inline], [virtual]
```

Reimplemented in [gdcm::JPEGCodec](#).

**Examples**

[ExtractIconFromFile.cxx](#).

**10.151.4.40 SetPlanarConfiguration()**

```
void gdcm::ImageCodec::SetPlanarConfiguration (
    unsigned int pc ) [inline]
```

#### 10.151.4.41 StartEncode()

```
virtual bool gdcm::ImageCodec::StartEncode (
    std::ostream & os ) [protected], [virtual]
```

Reimplemented in [gdcm::JPEG2000Codec](#), [gdcm::JPEGCodec](#), [gdcm::JPEGLSCodec](#), and [gdcm::RLECodec](#).

#### 10.151.4.42 StopEncode()

```
virtual bool gdcm::ImageCodec::StopEncode (
    std::ostream & os ) [protected], [virtual]
```

Reimplemented in [gdcm::JPEG2000Codec](#), [gdcm::JPEGCodec](#), [gdcm::JPEGLSCodec](#), and [gdcm::RLECodec](#).

### 10.151.5 Friends And Related Symbol Documentation

#### 10.151.5.1 FileChangeTransferSyntax

```
friend class FileChangeTransferSyntax [friend]
```

This is a high level API to encode in a streaming fashion. Each plugin will handle differently the caching mechanism so that a limited memory is used when compressing dataset. [Codec](#) will fall into two categories:

- Full row encoder: only a single scanline (row) of data is needed to be loaded at a time;
- Full frame encoder (default): a complete frame (row x col) is needed to be loaded at a time

#### 10.151.5.2 ImageChangePhotometricInterpretation

```
friend class ImageChangePhotometricInterpretation [friend]
```

### 10.151.6 Member Data Documentation

#### 10.151.6.1 Dimensions

```
unsigned int gdcm::ImageCodec::Dimensions[3] [protected]
```

#### 10.151.6.2 LossyFlag

```
bool gdcm::ImageCodec::LossyFlag [protected]
```

### 10.151.6.3 LUT

`LUTPtr` `gdcm::ImageCodec::LUT` [protected]

### 10.151.6.4 NeedByteSwap

`bool` `gdcm::ImageCodec::NeedByteSwap` [protected]

### 10.151.6.5 NeedOverlayCleanup

`bool` `gdcm::ImageCodec::NeedOverlayCleanup` [protected]

### 10.151.6.6 NumberOfDimensions

`unsigned int` `gdcm::ImageCodec::NumberOfDimensions` [protected]

### 10.151.6.7 PF

`PixelFormat` `gdcm::ImageCodec::PF` [protected]

### 10.151.6.8 PI

`PhotometricInterpretation` `gdcm::ImageCodec::PI` [protected]

### 10.151.6.9 PlanarConfiguration

`unsigned int` `gdcm::ImageCodec::PlanarConfiguration` [protected]

### 10.151.6.10 RequestPaddedCompositePixelCode

`bool` `gdcm::ImageCodec::RequestPaddedCompositePixelCode` [protected]

### 10.151.6.11 RequestPlanarConfiguration

`bool` `gdcm::ImageCodec::RequestPlanarConfiguration` [protected]

The documentation for this class was generated from the following file:

- [gdcmImageCodec.h](#)

## 10.152 gdcm::ImageConverter Class Reference

[Image](#) Converter.

```
#include <gdcmImageConverter.h>
```

### Public Member Functions

- [ImageConverter](#) ()
- [~ImageConverter](#) ()
- void [Convert](#) ()
- const [Image](#) & [GetOutput](#) () const
- void [SetInput](#) ([Image](#) const &input)

### 10.152.1 Detailed Description

[Image](#) Converter.

#### Note

This is the class used to convert from on [Image](#) to another This is typically used to convert let say YBR JPEG compressed [Image](#) to a RAW RGB [Image](#). So that the buffer can be directly pass to third party application. This filter is application level and not integrated directly in GDCM

### 10.152.2 Constructor & Destructor Documentation

#### 10.152.2.1 ImageConverter()

```
gdcm::ImageConverter::ImageConverter ( )
```

#### 10.152.2.2 ~ImageConverter()

```
gdcm::ImageConverter::~ImageConverter ( )
```

### 10.152.3 Member Function Documentation

#### 10.152.3.1 Convert()

```
void gdcm::ImageConverter::Convert ( )
```



### 10.152.3.2 GetOutput()

```
const Image & gdcm::ImageConverter::GetOutput ( ) const
```

### 10.152.3.3 SetInput()

```
void gdcm::ImageConverter::SetInput (
    Image const & input )
```

The documentation for this class was generated from the following file:

- [gdcmImageConverter.h](#)

## 10.153 gdcm::ImageFragmentSplitter Class Reference

[ImageFragmentSplitter](#) class.

```
#include <gdcmImageFragmentSplitter.h>
```

Inheritance diagram for gdcm::ImageFragmentSplitter:



Collaboration diagram for `gdcm::ImageFragmentSplitter`:



### Public Member Functions

- `ImageFragmentSplitter ()`
- `~ImageFragmentSplitter ()=default`
- `unsigned int GetFragmentSizeMax () const`
- `void SetForce (bool f)`
- `void SetFragmentSizeMax (unsigned int fragsize)`  
*FragmentSizeMax needs to be an even number.*
- `bool Split ()`  
*Split.*

**Public Member Functions inherited from [gdcm::ImageToImageFilter](#)**

- [ImageToImageFilter](#) ()
- [~ImageToImageFilter](#) ()=default
- [Image](#) & [GetInput](#) ()
- const [Image](#) & [GetOutput](#) () const

*Get Output image.*

**Public Member Functions inherited from [gdcm::PixmapToPixmapFilter](#)**

- [PixmapToPixmapFilter](#) ()
- [~PixmapToPixmapFilter](#) ()=default
- [Pixmap](#) & [GetInput](#) ()
- const [Pixmap](#) & [GetOutput](#) () const
- const [Pixmap](#) & [GetOutputAsPixmap](#) () const

*Get Output image.*

**Public Member Functions inherited from [gdcm::BitmapToBitmapFilter](#)**

- [BitmapToBitmapFilter](#) ()
- [~BitmapToBitmapFilter](#) ()=default
- const [Bitmap](#) & [GetOutput](#) () const
- const [Bitmap](#) & [GetOutputAsBitmap](#) () const
- void [SetInput](#) (const [Bitmap](#) &image)

*Get Output image.*

*Set input image.*

**Additional Inherited Members****Protected Attributes inherited from [gdcm::BitmapToBitmapFilter](#)**

- [SmartPointer](#)< [Bitmap](#) > [Input](#)
- [SmartPointer](#)< [Bitmap](#) > [Output](#)

**10.153.1 Detailed Description**

[ImageFragmentSplitter](#) class.

For single frame image, DICOM standard allow splitting the frame into multiple fragments

**10.153.2 Constructor & Destructor Documentation****10.153.2.1 [ImageFragmentSplitter](#)()**

```
gdcm::ImageFragmentSplitter::ImageFragmentSplitter ( ) [inline]
```

### 10.153.2.2 ~ImageFragmentSplitter()

```
gdcmm::ImageFragmentSplitter::~ImageFragmentSplitter ( ) [default]
```

## 10.153.3 Member Function Documentation

### 10.153.3.1 GetFragmentSizeMax()

```
unsigned int gdcmm::ImageFragmentSplitter::GetFragmentSizeMax ( ) const [inline]
```

### 10.153.3.2 SetForce()

```
void gdcmm::ImageFragmentSplitter::SetForce (
    bool f ) [inline]
```

When file already has all its segment < FragmentSizeMax there is not need to run the filter. Unless the user explicitly say 'force' recomputation !

### 10.153.3.3 SetFragmentSizeMax()

```
void gdcmm::ImageFragmentSplitter::SetFragmentSizeMax (
    unsigned int fragsize )
```

FragmentSizeMax needs to be an even number.

### 10.153.3.4 Split()

```
bool gdcmm::ImageFragmentSplitter::Split ( )
```

Split.

The documentation for this class was generated from the following file:

- [gdcmmImageFragmentSplitter.h](#)

## 10.154 gdcmm::ImageHelper Class Reference

[ImageHelper](#) (internal class, not intended for user level)

```
#include <gdcmmImageHelper.h>
```

## Static Public Member Functions

- static [MediaStorage](#) [ComputeMediaStorageFromModality](#) (const char \*modality, unsigned int dimension=2, [PixelFormat](#) const &pf=[PixelFormat](#)(), [PhotometricInterpretation](#) const &pi=[PhotometricInterpretation](#)(), double rescaleintercept=0, double rescaleslope=1)

*Moved from [MediaStorage](#) here, since we need extra info stored in [PixelFormat](#) & [PhotometricInterpretation](#).*

- static bool [ComputeSpacingFromImagePositionPatient](#) (const std::vector< double > &imageposition, std::vector< double > &spacing)

*DO NOT USE.*

- static std::vector< unsigned int > [GetDimensionsValue](#) (const [File](#) &f)
- static bool [GetDirectionCosinesFromDataSet](#) ([DataSet](#) const &ds, std::vector< double > &dircos)
- static std::vector< double > [GetDirectionCosinesValue](#) ([File](#) const &f)
- static bool [GetForcePixelSpacing](#) ()
- static bool [GetForceRescaleInterceptSlope](#) ()
- static [SmartPointer< LookupTable >](#) [GetLUT](#) ([File](#) const &f)

*returns the lookup table of an image file*

- static std::vector< double > [GetOriginValue](#) ([File](#) const &f)

*Set/Get Origin (IPP) from/to a file.*

- static [PhotometricInterpretation](#) [GetPhotometricInterpretationValue](#) ([File](#) const &f)
- static [PixelFormat](#) [GetPixelFormatValue](#) (const [File](#) &f)
- static unsigned int [GetPlanarConfigurationValue](#) (const [File](#) &f)
- static bool [GetPMSRescaleInterceptSlope](#) ()
- static const [ByteValue](#) \* [GetPointerFromElement](#) ([Tag](#) const &tag, [File](#) const &f)
- static bool [GetRealWorldValueMappingContent](#) ([File](#) const &f, [RealWorldValueMappingContent](#) &rwvmc)
- static std::vector< double > [GetRescaleInterceptSlopeValue](#) ([File](#) const &f)
- static bool [GetSecondaryCaptureImagePlaneModule](#) ()
- static std::vector< double > [GetSpacingValue](#) ([File](#) const &f)

*Set/Get [Spacing](#) from/to a [File](#).*

- static void [SetDimensionsValue](#) ([File](#) &f, const [Pixmap](#) &img)
- static void [SetDirectionCosinesValue](#) ([DataSet](#) &ds, const std::vector< double > &dircos)
- static void [SetForcePixelSpacing](#) (bool)
- static void [SetForceRescaleInterceptSlope](#) (bool)
- static void [SetOriginValue](#) ([DataSet](#) &ds, const [Image](#) &img)
- static void [SetPMSRescaleInterceptSlope](#) (bool)
- static void [SetRescaleInterceptSlopeValue](#) ([File](#) &f, const [Image](#) &img)
- static void [SetSecondaryCaptureImagePlaneModule](#) (bool)
- static void [SetSpacingValue](#) ([DataSet](#) &ds, const std::vector< double > &spacing)

## Static Protected Member Functions

- static [Tag](#) [GetSpacingTagFromMediaStorage](#) ([MediaStorage](#) const &ms)
- static [Tag](#) [GetZSpacingTagFromMediaStorage](#) ([MediaStorage](#) const &ms)

### 10.154.1 Detailed Description

[ImageHelper](#) (internal class, not intended for user level)

Helper for writing World images in DICOM. DICOM has a 'template' approach to image where MR [Image](#) Storage are distinct object from Enhanced MR [Image](#) Storage. For example the Pixel [Spacing](#) in one object is not at the same position (ie [Tag](#)) as in the other this class is the central (read: fragile) place where all the dispatching is done from a unified view of a world image (typically VTK or ITK point of view) down to the low level DICOM point of view.

#### Warning

: do not expect the API of this class to be maintained at any point, since as Modalities are added the API might have to be augmented or behavior changed to cope with new modalities.

#### Examples

[ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), and [ExtractOneFrame.cs](#).

### 10.154.2 Member Function Documentation

#### 10.154.2.1 ComputeMediaStorageFromModality()

```
static MediaStorage gdcm::ImageHelper::ComputeMediaStorageFromModality (
    const char * modality,
    unsigned int dimension = 2,
    PixelFormat const & pf = PixelFormat (),
    PhotometricInterpretation const & pi = PhotometricInterpretation (),
    double rescaleintercept = 0,
    double rescaleslope = 1 ) [static]
```

Moved from [MediaStorage](#) here, since we need extra info stored in [PixelFormat](#) & [PhotometricInterpretation](#).

#### 10.154.2.2 ComputeSpacingFromImagePositionPatient()

```
static bool gdcm::ImageHelper::ComputeSpacingFromImagePositionPatient (
    const std::vector< double > & imageposition,
    std::vector< double > & spacing ) [static]
```

DO NOT USE.

#### 10.154.2.3 GetDimensionsValue()

```
static std::vector< unsigned int > gdcm::ImageHelper::GetDimensionsValue (
    const File & f ) [static]
```

This function checks tags (0x0028, 0x0010) and (0x0028, 0x0011) for the rows and columns of the image in pixels (as opposed to actual distances). The output is {col , row}

#### Examples

[ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), [ExtractOneFrame.cs](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

#### 10.154.2.4 GetDirectionCosinesFromDataSet()

```
static bool gdcm::ImageHelper::GetDirectionCosinesFromDataSet (
    DataSet const & ds,
    std::vector< double > & dircos ) [static]
```

#### 10.154.2.5 GetDirectionCosinesValue()

```
static std::vector< double > gdcm::ImageHelper::GetDirectionCosinesValue (
    File const & f ) [static]
```

Get Direction Cosines (IOP) from/to a file Requires a file because mediastorage must be known

#### 10.154.2.6 GetForcePixelSpacing()

```
static bool gdcm::ImageHelper::GetForcePixelSpacing ( ) [static]
```

#### 10.154.2.7 GetForceRescaleInterceptSlope()

```
static bool gdcm::ImageHelper::GetForceRescaleInterceptSlope ( ) [static]
```

#### 10.154.2.8 GetLUT()

```
static SmartPointer< LookupTable > gdcm::ImageHelper::GetLUT (
    File const & f ) [static]
```

returns the lookup table of an image file

#### 10.154.2.9 GetOriginValue()

```
static std::vector< double > gdcm::ImageHelper::GetOriginValue (
    File const & f ) [static]
```

Set/Get Origin (IPP) from/to a file.

#### 10.154.2.10 GetPhotometricInterpretationValue()

```
static PhotometricInterpretation gdcm::ImageHelper::GetPhotometricInterpretationValue (
    File const & f ) [static]
```

#### Examples

[ExtractImageRegion.cs](#).

#### 10.154.2.11 GetPixelFormatValue()

```
static PixelFormat gdcm::ImageHelper::GetPixelFormatValue (
    const File & f ) [static]
```

This function returns pixel information about an image from its dataset That includes samples per pixel and bit depth (in that order)

#### 10.154.2.12 GetPlanarConfigurationValue()

```
static unsigned int gdcm::ImageHelper::GetPlanarConfigurationValue (
    const File & f ) [static]
```

#### 10.154.2.13 GetPMSRescaleInterceptSlope()

```
static bool gdcm::ImageHelper::GetPMSRescaleInterceptSlope ( ) [static]
```

#### 10.154.2.14 GetPointerFromElement()

```
static const ByteValue * gdcm::ImageHelper::GetPointerFromElement (
    Tag const & tag,
    File const & f ) [static]
```

#### 10.154.2.15 GetRealWorldValueMappingContent()

```
static bool gdcm::ImageHelper::GetRealWorldValueMappingContent (
    File const & f,
    RealWorldValueMappingContent & rwvmc ) [static]
```

#### 10.154.2.16 GetRescaleInterceptSlopeValue()

```
static std::vector< double > gdcm::ImageHelper::GetRescaleInterceptSlopeValue (
    File const & f ) [static]
```

Set/Get shift/scale from/to a file

#### Warning

this function reads/sets the Slope/Intercept in appropriate class storage, but also Grid Scaling in RT Dose Storage  
Can't take a dataset because the mediastorage of the file must be known



### 10.154.2.17 GetSecondaryCaptureImagePlaneModule()

```
static bool gdcm::ImageHelper::GetSecondaryCaptureImagePlaneModule ( ) [static]
```

### 10.154.2.18 GetSpacingTagFromMediaStorage()

```
static Tag gdcm::ImageHelper::GetSpacingTagFromMediaStorage (
    MediaStorage const & ms ) [static], [protected]
```

### 10.154.2.19 GetSpacingValue()

```
static std::vector< double > gdcm::ImageHelper::GetSpacingValue (
    File const & f ) [static]
```

Set/Get [Spacing](#) from/to a [File](#).

### 10.154.2.20 GetZSpacingTagFromMediaStorage()

```
static Tag gdcm::ImageHelper::GetZSpacingTagFromMediaStorage (
    MediaStorage const & ms ) [static], [protected]
```

### 10.154.2.21 SetDimensionsValue()

```
static void gdcm::ImageHelper::SetDimensionsValue (
    File & f,
    const Pixmap & img ) [static]
```

### 10.154.2.22 SetDirectionCosinesValue()

```
static void gdcm::ImageHelper::SetDirectionCosinesValue (
    DataSet & ds,
    const std::vector< double > & dircos ) [static]
```

Set Direction Cosines (IOP) from/to a file When [IOD](#) does not defines what is IOP (eg. typically Secondary Capture [Image](#) Storage) this call will simply remove the IOP attribute. Else in case of MR/CT image storage, this call will properly lookup the correct attribute to store the IOP.

### 10.154.2.23 SetForcePixelSpacing()

```
static void gdcm::ImageHelper::SetForcePixelSpacing (
    bool ) [static]
```

GDCM 1.x compatibility issue: When using ReWrite an MR [Image](#) Storage would be rewritten as Secondary Capture [Object](#) while still having a Pixel [Spacing](#) tag (0028,0030). If you have deal with those files, use this very special flag to handle them Unless explicitly set elsewhere by the standard, it will use value from 0028,0030 / 0018,0088 for the Pixel [Spacing](#) of the [Image](#)

#### 10.154.2.24 SetForceRescaleInterceptSlope()

```
static void gdcm::ImageHelper::SetForceRescaleInterceptSlope (
    bool ) [static]
```

GDCM 1.x compatibility issue: Do not use anymore. This hack was used for some MR [Image](#) Storage generated by Philips Modality. When "Combine MR Rescaling" is set to TRUE, rescaling is removed. But when set to FALSE, the Modality LUT was exported. Internally GDCM now handles this gracefully.

#### 10.154.2.25 SetOriginValue()

```
static void gdcm::ImageHelper::SetOriginValue (
    DataSet & ds,
    const Image & img ) [static]
```

#### 10.154.2.26 SetPMSRescaleInterceptSlope()

```
static void gdcm::ImageHelper::SetPMSRescaleInterceptSlope (
    bool ) [static]
```

Since GDCM 2.6.1 Philips Medical [System](#) are read using the Private Field For Rescale Slope/Intercept by default. This mechanism can be deactivated using the following API: This option has no effect when ForceRescaleInterceptSlope is set to true GDCM will only read those private attribute but never write them out.

#### 10.154.2.27 SetRescaleInterceptSlopeValue()

```
static void gdcm::ImageHelper::SetRescaleInterceptSlopeValue (
    File & f,
    const Image & img ) [static]
```

#### 10.154.2.28 SetSecondaryCaptureImagePlaneModule()

```
static void gdcm::ImageHelper::SetSecondaryCaptureImagePlaneModule (
    bool ) [static]
```

Opt into [Image Plane Module](#) for Secondary Capture [Image](#) Storage Enable reading [Image](#) Position [Patient](#) (IPP), [Image Orientation Patient](#) (IOP) and Pixel [Spacing](#) (0028,0030) This is a custom extension for some existing dataset (academic)

### 10.154.2.29 SetSpacingValue()

```
static void gdcm::ImageHelper::SetSpacingValue (
    DataSet & ds,
    const std::vector< double > & spacing ) [static]
```

#### Warning

You need to call SetSpacingValue after SetOriginValue / SetDirectionCosinesValue

The documentation for this class was generated from the following file:

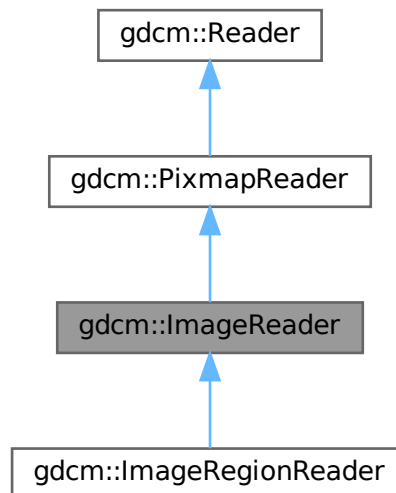
- [gdcmImageHelper.h](#)

## 10.155 gdcm::ImageReader Class Reference

[ImageReader](#).

```
#include <gdcmImageReader.h>
```

Inheritance diagram for gdcm::ImageReader:



Collaboration diagram for `gdcm::ImageReader`:



### Public Member Functions

- [ImageReader](#) ()
- [~ImageReader](#) () override
- [Image](#) & [GetImage](#) ()
- const [Image](#) & [GetImage](#) () const  
*Return the read image.*
- bool [Read](#) () override

### Public Member Functions inherited from [gdcm::PixmapReader](#)

- [PixmapReader](#) ()
- [~PixmapReader](#) () override
- [Pixmap](#) & [GetPixmap](#) ()
- const [Pixmap](#) & [GetPixmap](#) () const  
*Return the read image (need to call [Read\(\)](#) first)*
- bool [Read](#) () override

## Public Member Functions inherited from [gdcm::Reader](#)

- [Reader](#) ()
- virtual [~Reader](#) ()
- bool [CanRead](#) () const
- [File](#) & [GetFile](#) ()  
*Set/Get File.*
- const [File](#) & [GetFile](#) () const  
*Set/Get File.*
- size\_t [GetStreamCurrentPosition](#) () const
- bool [ReadSelectedPrivateTags](#) (std::set< [PrivateTag](#) > const &ptags, bool readvalues=true)  
*Will only read the specified selected private tags.*
- bool [ReadSelectedTags](#) (std::set< [Tag](#) > const &tags, bool readvalues=true)  
*Will only read the specified selected tags.*
- bool [ReadUpToTag](#) (const [Tag](#) &tag, std::set< [Tag](#) > const &skiptags=std::set< [Tag](#) >())
- void [SetFile](#) ([File](#) &file)  
*Set/Get File.*
- void [SetFileName](#) (const char \*filename\_native)
- void [SetStream](#) (std::istream &input\_stream)  
*Set the open-ed stream directly.*

## Protected Member Functions

- bool [ReadACRNEMAImage](#) () override
- bool [ReadImage](#) ([MediaStorage](#) const &ms) override

## Protected Member Functions inherited from [gdcm::PixmapReader](#)

- bool [ReadImageInternal](#) ([MediaStorage](#) const &ms, bool handlepixeldata=true)

## Protected Member Functions inherited from [gdcm::Reader](#)

- std::istream \* [GetStreamPtr](#) () const
- bool [ReadDataSet](#) ()
- bool [ReadMetaInformation](#) ()
- bool [ReadPreamble](#) ()

## Additional Inherited Members

## Protected Attributes inherited from [gdcm::PixmapReader](#)

- [SmartPointer](#)< [Pixmap](#) > [PixelData](#)

## Protected Attributes inherited from [gdcm::Reader](#)

- [SmartPointer](#)< [File](#) > [F](#)

### 10.155.1 Detailed Description

[ImageReader](#).

#### Note

its role is to convert the DICOM [DataSet](#) into a [Image](#) representation [Image](#) is different from [Pixmap](#) has it has a position and a direction in Space.

#### See also

[Image](#)

#### Examples

[BasicImageAnonymizer.cs](#), [CheckBigEndianBug.cxx](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [ConvertToQImage.cxx](#), [DecompressImage.cs](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetArray.cs](#), [GetJPEGSamplePrecision.cxx](#), [HelloVizWorld.cxx](#), [MergeTwoFiles.cxx](#), [MpegVideoInfo.cs](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [PrintLUT.cxx](#), [ReadMultiTimesException.cxx](#), [RescaleImage.cs](#), and [threadgdcm.cxx](#).

### 10.155.2 Constructor & Destructor Documentation

#### 10.155.2.1 [ImageReader\(\)](#)

```
gdcm::ImageReader::ImageReader ( )
```

#### 10.155.2.2 [~ImageReader\(\)](#)

```
gdcm::ImageReader::~ImageReader ( ) [override]
```

### 10.155.3 Member Function Documentation

#### 10.155.3.1 [GetImage\(\)](#) [1/2]

```
Image & gdcm::ImageReader::GetImage ( )
```

### 10.155.3.2 GetImage() [2/2]

```
const Image & gdcm::ImageReader::GetImage ( ) const
```

Return the read image.

#### Examples

[BasicImageAnonymizer.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [ConvertToQImage.cxx](#), [DecompressImage.cs](#), [ExtractIconFromFile.cxx](#), [ExtractImageRegionWithLUT.cs](#), [FixJAIBugJPEGLS.cxx](#), [GetArray.cs](#), [GetJPEGSamplePrecision.cxx](#), [HelloVizWorld.cxx](#), [MergeTwoFiles.cxx](#), [MpegVideoInfo.cs](#), [PatchFile.cxx](#), [PrintLUT.cxx](#), [ReadMultiTimesException.cxx](#), [RescaleImage.cs](#), [TemplateEmptyImage.cxx](#), and [threadgdcm.cxx](#).

### 10.155.3.3 Read()

```
bool gdcm::ImageReader::Read ( ) [override], [virtual]
```

Read the DICOM image. There are two reason for failure:

1. The input filename is not DICOM
2. The input DICOM file does not contains an [Image](#).

Reimplemented from [gdcm::Reader](#).

Reimplemented in [gdcm::ImageRegionReader](#).

#### Examples

[BasicImageAnonymizer.cs](#), [CheckBigEndianBug.cxx](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [ConvertToQImage.cxx](#), [DecompressImage.cs](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetArray.cs](#), [GetJPEGSamplePrecision.cxx](#), [HelloVizWorld.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [PrintLUT.cxx](#), [ReadMultiTimesException.cxx](#), [RescaleImage.cs](#), and [threadgdcm.cxx](#).

### 10.155.3.4 ReadACRNEMAIImage()

```
bool gdcm::ImageReader::ReadACRNEMAIImage ( ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::PixmapReader](#).

### 10.155.3.5 ReadImage()

```
bool gdcm::ImageReader::ReadImage (
    MediaStorage const & ms ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::PixmapReader](#).

The documentation for this class was generated from the following file:

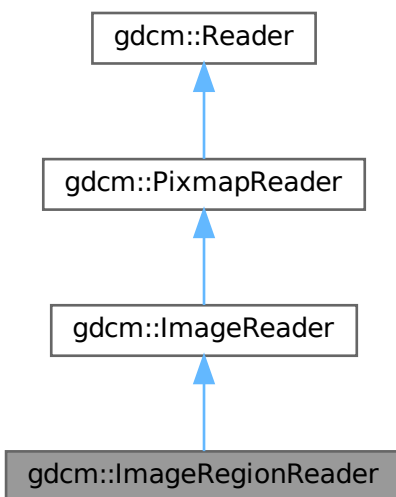
- [gdcmImageReader.h](#)

## 10.156 gdcm::ImageRegionReader Class Reference

[ImageRegionReader](#).

```
#include <gdcmImageRegionReader.h>
```

Inheritance diagram for gdcm::ImageRegionReader:





Collaboration diagram for gdcm::ImageRegionReader:



### Public Member Functions

- [ImageRegionReader](#) ()
- [~ImageRegionReader](#) () override
- [size\\_t ComputeBufferLength](#) () const
- [Region](#) const & [GetRegion](#) () const
- [bool ReadInformation](#) ()
- [bool ReadIntoBuffer](#) (char \*inreadbuffer, size\_t buflen)
- [void SetRegion](#) ([Region](#) const &region)

*Set/Get [Region](#) to be read.*

## Public Member Functions inherited from [gdcm::ImageReader](#)

- [ImageReader](#) ()
- [~ImageReader](#) () override
- [Image](#) & [GetImage](#) ()
- const [Image](#) & [GetImage](#) () const

*Return the read image.*

## Public Member Functions inherited from [gdcm::PixmapReader](#)

- [PixmapReader](#) ()
  - [~PixmapReader](#) () override
  - [Pixmap](#) & [GetPixmap](#) ()
  - const [Pixmap](#) & [GetPixmap](#) () const
- Return the read image (need to call [Read\(\)](#) first)*
- bool [Read](#) () override

## Public Member Functions inherited from [gdcm::Reader](#)

- [Reader](#) ()
  - virtual [~Reader](#) ()
  - bool [CanRead](#) () const
  - [File](#) & [GetFile](#) ()
- Set/Get File.*
- const [File](#) & [GetFile](#) () const
- Set/Get File.*
- size\_t [GetStreamCurrentPosition](#) () const
  - bool [ReadSelectedPrivateTags](#) (std::set< [PrivateTag](#) > const &ptags, bool readvalues=true)
- Will only read the specified selected private tags.*
- bool [ReadSelectedTags](#) (std::set< [Tag](#) > const &tags, bool readvalues=true)
- Will only read the specified selected tags.*
- bool [ReadUpToTag](#) (const [Tag](#) &tag, std::set< [Tag](#) > const &skiptags=std::set< [Tag](#) >())
  - void [SetFile](#) ([File](#) &file)
- Set/Get File.*
- void [SetFileName](#) (const char \*filename\_native)
  - void [SetStream](#) (std::istream &input\_stream)
- Set the open-ed stream directly.*

## Protected Member Functions

- bool [Read](#) () override
- To prevent user from calling super class [Read\(\)](#) function.*

## Protected Member Functions inherited from [gdcm::ImageReader](#)

- bool [ReadACRNEMAIImage](#) () override
- bool [ReadImage](#) ([MediaStorage](#) const &ms) override

**Protected Member Functions inherited from [gdcm::PixmapReader](#)**

- bool [ReadImageInternal](#) ([MediaStorage](#) const &ms, bool handlepixeldata=true)

**Protected Member Functions inherited from [gdcm::Reader](#)**

- std::istream \* [GetStreamPtr](#) () const
- bool [ReadDataSet](#) ()
- bool [ReadMetaInformation](#) ()
- bool [ReadPreamble](#) ()

**Additional Inherited Members****Protected Attributes inherited from [gdcm::PixmapReader](#)**

- [SmartPointer](#)< [Pixmap](#) > [PixelData](#)

**Protected Attributes inherited from [gdcm::Reader](#)**

- [SmartPointer](#)< [File](#) > [F](#)

**10.156.1 Detailed Description**[ImageRegionReader](#).

This class is able to read a region from a DICOM file containing an image. This implementation requires that the information stored in the DICOM header are consistent with what is in the encapsulated Pixel Data. This is technically not required by DICOM standard, which makes this implementation illegal with regards to the famous JPEG note: [http://dicom.nema.org/medical/dicom/current/output/chtml/part05/sect\\_8.2.html#para\\_4bcb841e-c6bf-4e26-82a5-3fad3c942da0](http://dicom.nema.org/medical/dicom/current/output/chtml/part05/sect_8.2.html#para_4bcb841e-c6bf-4e26-82a5-3fad3c942da0)

**See also**

[ImageReader](#)

**Examples**

[ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), and [TemplateEmptyImage.cxx](#).

**10.156.2 Constructor & Destructor Documentation****10.156.2.1 [ImageRegionReader](#)()**

```
gdcm::ImageRegionReader::ImageRegionReader ( )
```

### 10.156.2.2 ~ImageRegionReader()

```
gdcm::ImageRegionReader::~~ImageRegionReader ( ) [override]
```

## 10.156.3 Member Function Documentation

### 10.156.3.1 ComputeBufferLength()

```
size_t gdcm::ImageRegionReader::ComputeBufferLength ( ) const
```

Explicit call which will compute the minimal buffer length that can hold the whole uncompressed image as defined by [Region](#) region.

#### Returns

0 upon error

### 10.156.3.2 GetRegion()

```
Region const & gdcm::ImageRegionReader::GetRegion ( ) const
```

### 10.156.3.3 Read()

```
bool gdcm::ImageRegionReader::Read ( ) [override], [protected], [virtual]
```

To prevent user from calling super class [Read\(\)](#) function.

Reimplemented from [gdcm::ImageReader](#).

### 10.156.3.4 ReadInformation()

```
bool gdcm::ImageRegionReader::ReadInformation ( )
```

Read meta information (not Pixel Data) from the DICOM file.

#### Returns

false upon error

#### Examples

[ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), and [TemplateEmptyImage.cxx](#).

### 10.156.3.5 ReadIntoBuffer()

```
bool gdcm::ImageRegionReader::ReadIntoBuffer (
    char * inreadbuffer,
    size_t buflen )
```

Read into buffer: For Python, the `buflen` param is deduced directly from the input bytearray passed as parameter (function only takes one param).

#### Returns

false upon error

#### Examples

[ExtractImageRegion.cs](#), and [ExtractImageRegionWithLUT.cs](#).

### 10.156.3.6 SetRegion()

```
void gdcm::ImageRegionReader::SetRegion (
    Region const & region )
```

Set/Get [Region](#) to be read.

#### Examples

[ExtractImageRegion.cs](#), and [ExtractImageRegionWithLUT.cs](#).

The documentation for this class was generated from the following file:

- [gdcmImageRegionReader.h](#)

## 10.157 gdcm::ImageToImageFilter Class Reference

[ImageToImageFilter](#) class.

```
#include <gdcmImageToImageFilter.h>
```

Inheritance diagram for `gdcm::ImageToImageFilter`:



Collaboration diagram for `gdcm::ImageToImageFilter`:



### Public Member Functions

- [ImageToImageFilter](#) ()
- [~ImageToImageFilter](#) ()=default
- [Image](#) & [GetInput](#) ()
- const [Image](#) & [GetOutput](#) () const  
*Get Output image.*

### Public Member Functions inherited from [gdcm::PixmapToPixmapFilter](#)

- [PixmapToPixmapFilter](#) ()
- [~PixmapToPixmapFilter](#) ()=default
- [Pixmap](#) & [GetInput](#) ()
- const [Pixmap](#) & [GetOutput](#) () const  
*Get Output image.*
- const [Pixmap](#) & [GetOutputAsPixmap](#) () const

**Public Member Functions inherited from [gdcm::BitmapToBitmapFilter](#)**

- [BitmapToBitmapFilter](#) ()
- [~BitmapToBitmapFilter](#) ()=default
- const [Bitmap](#) & [GetOutput](#) () const  
*Get Output image.*
- const [Bitmap](#) & [GetOutputAsBitmap](#) () const
- void [SetInput](#) (const [Bitmap](#) &image)  
*Set input image.*

**Additional Inherited Members****Protected Attributes inherited from [gdcm::BitmapToBitmapFilter](#)**

- [SmartPointer](#)< [Bitmap](#) > [Input](#)
- [SmartPointer](#)< [Bitmap](#) > [Output](#)

**10.157.1 Detailed Description**

[ImageToImageFilter](#) class.

Super class for all filter taking an image and producing an output image

**10.157.2 Constructor & Destructor Documentation****10.157.2.1 [ImageToImageFilter](#)()**

```
gdcm::ImageToImageFilter::ImageToImageFilter ( )
```

**10.157.2.2 [~ImageToImageFilter](#)()**

```
gdcm::ImageToImageFilter::~ImageToImageFilter ( ) [default]
```

**10.157.3 Member Function Documentation****10.157.3.1 [GetInput](#)()**

```
Image & gdcm::ImageToImageFilter::GetInput ( )
```

### 10.157.3.2 GetOutput()

```
const Image & gdcM::ImageToImageFilter::GetOutput ( ) const
```

Get Output image.

#### Examples

[BasicImageAnonymizer.cs](#), [CompressImage.cxx](#), and [CompressLossyJPEG.cs](#).

The documentation for this class was generated from the following file:

- [gdcMImageToImageFilter.h](#)

## 10.158 gdcM::ImageWriter Class Reference

[ImageWriter](#).

```
#include <gdcMImageWriter.h>
```

Inheritance diagram for gdcM::ImageWriter:





Collaboration diagram for gdcm::ImageWriter:



## Public Member Functions

- `ImageWriter ()`
- `~ImageWriter ()` override
- `MediaStorage ComputeTargetMediaStorage ()`
- `const Image & GetImage ()` const override
- `Image & GetImage ()` override
- `bool Write ()` override

*Write.*

## Public Member Functions inherited from [gdcm::PixmapWriter](#)

- [PixmapWriter](#) ()
  - [~PixmapWriter](#) () override
  - [Pixmap](#) & [GetPixmap](#) ()
  - const [Pixmap](#) & [GetPixmap](#) () const
  - virtual void [SetImage](#) ([Pixmap](#) const &img)
  - void [SetPixmap](#) ([Pixmap](#) const &img)
  - bool [Write](#) () override
- Write.*

## Public Member Functions inherited from [gdcm::Writer](#)

- [Writer](#) ()
  - virtual [~Writer](#) ()
  - void [CheckFileMetaInformationOff](#) ()
  - void [CheckFileMetaInformationOn](#) ()
  - [File](#) & [GetFile](#) ()
  - void [SetCheckFileMetaInformation](#) (bool b)
- Undocumented function, do not use (= leave default)*
- void [SetFile](#) (const [File](#) &f)
- Set/Get the DICOM file ([DataSet](#) + Header)*
- void [SetFileName](#) (const char \*filename\_native)
- Set the filename of DICOM file to write:*
- void [SetStream](#) (std::ostream &output\_stream)
- Set user ostream buffer.*

## Additional Inherited Members

## Protected Member Functions inherited from [gdcm::PixmapWriter](#)

- void [DolconImage](#) ([DataSet](#) &ds, [Pixmap](#) const &image)
- bool [PrepareWrite](#) ([MediaStorage](#) const &refms)

## Protected Member Functions inherited from [gdcm::Writer](#)

- bool [GetCheckFileMetaInformation](#) () const
- std::ostream \* [GetStreamPtr](#) () const
- void [SetWriteDataSetOnly](#) (bool b)

## Protected Attributes inherited from [gdcm::PixmapWriter](#)

- [SmartPointer](#)< [Pixmap](#) > [PixelData](#)

## Protected Attributes inherited from [gdcm::Writer](#)

- `std::ofstream` \* [Ofstream](#)
- `std::ostream` \* [Stream](#)

### 10.158.1 Detailed Description

[ImageWriter](#).

This is an extended version of the [PixmapWriter](#). Pay attention that:

1. It will populate missing attribute for Secondary Capture [Image](#) Storage instances,
2. It may also change an input MR [Image](#) Storage instance into a pseudo Enhanced MR [Image](#) Storage instance whenever Modality LUT is required.
3. Some [DataElement](#) related to [gdcm::Image](#) may be slightly altered.

#### Examples

[BasicImageAnonymizer.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImage.cs](#), [GenFakelImage.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [MergeTwoFiles.cxx](#), [MpegVideoInfo.cs](#), [TemplateEmptyImage.cxx](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

### 10.158.2 Constructor & Destructor Documentation

#### 10.158.2.1 ImageWriter()

```
gdcm::ImageWriter::ImageWriter ( )
```

#### 10.158.2.2 ~ImageWriter()

```
gdcm::ImageWriter::~~ImageWriter ( ) [override]
```

### 10.158.3 Member Function Documentation

#### 10.158.3.1 ComputeTargetMediaStorage()

```
MediaStorage gdcm::ImageWriter::ComputeTargetMediaStorage ( )
```

internal function used to compute a target [MediaStorage](#) the most appropriate User may want to call this function ahead of time (before Write)

#### Examples

[TemplateEmptyImage.cxx](#).

**10.158.3.2 GetImage() [1/2]**

```
const Image & gdcm::ImageWriter::GetImage ( ) const [inline], [override], [virtual]
```

Set/Get [Image](#) to be written It will overwrite anything [Image](#) infos found in [DataSet](#) (see parent class to see how to pass dataset)

Reimplemented from [gdcm::PixmapWriter](#).

**Examples**

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

**10.158.3.3 GetImage() [2/2]**

```
Image & gdcm::ImageWriter::GetImage ( ) [inline], [override], [virtual]
```

Reimplemented from [gdcm::PixmapWriter](#).

**10.158.3.4 Write()**

```
bool gdcm::ImageWriter::Write ( ) [override], [virtual]
```

Write.

Reimplemented from [gdcm::Writer](#).

**Examples**

[BasicImageAnonymizer.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImage.cs](#), [GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [MergeTwoFiles.cxx](#), [MpegVideoInfo.cs](#), [TemplateEmptyImage.cxx](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

The documentation for this class was generated from the following file:

- [gdcmImageWriter.h](#)

**10.159 gdcm::network::ImplementationClassUIDSub Class Reference**

[ImplementationClassUIDSub](#).

```
#include <gdcmImplementationClassUIDSub.h>
```

## Public Member Functions

- [ImplementationClassUIDSub](#) ()
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- size\_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

### 10.159.1 Detailed Description

[ImplementationClassUIDSub](#).

PS 3.7 [Table D.3-1](#) IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

### 10.159.2 Constructor & Destructor Documentation

#### 10.159.2.1 ImplementationClassUIDSub()

```
gdcmm::network::ImplementationClassUIDSub::ImplementationClassUIDSub ( )
```

### 10.159.3 Member Function Documentation

#### 10.159.3.1 Print()

```
void gdcmm::network::ImplementationClassUIDSub::Print (
    std::ostream & os ) const
```

#### 10.159.3.2 Read()

```
std::istream & gdcmm::network::ImplementationClassUIDSub::Read (
    std::istream & is )
```

#### 10.159.3.3 Size()

```
size_t gdcmm::network::ImplementationClassUIDSub::Size ( ) const
```

#### 10.159.3.4 Write()

```
const std::ostream & gdcmm::network::ImplementationClassUIDSub::Write (
    std::ostream & os ) const
```

The documentation for this class was generated from the following file:

- [gdcmmImplementationClassUIDSub.h](#)

## 10.160 gdcm::network::ImplementationUIDSub Class Reference

[ImplementationUIDSub.](#)

```
#include <gdcmImplementationUIDSub.h>
```

### Public Member Functions

- [ImplementationUIDSub](#) ()
- const std::ostream & [Write](#) (std::ostream &os) const

### 10.160.1 Detailed Description

[ImplementationUIDSub.](#)

[Table D.3-2](#) IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)

### 10.160.2 Constructor & Destructor Documentation

#### 10.160.2.1 ImplementationUIDSub()

```
gdcm::network::ImplementationUIDSub::ImplementationUIDSub ( )
```

### 10.160.3 Member Function Documentation

#### 10.160.3.1 Write()

```
const std::ostream & gdcm::network::ImplementationUIDSub::Write (
    std::ostream & os ) const
```

The documentation for this class was generated from the following file:

- [gdcmImplementationUIDSub.h](#)

## 10.161 gdcm::network::ImplementationVersionNameSub Class Reference

[ImplementationVersionNameSub.](#)

```
#include <gdcmImplementationVersionNameSub.h>
```

## Public Member Functions

- [ImplementationVersionNameSub](#) ()
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- size\_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

### 10.161.1 Detailed Description

[ImplementationVersionNameSub](#).

[Table](#) D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

### 10.161.2 Constructor & Destructor Documentation

#### 10.161.2.1 ImplementationVersionNameSub()

```
gdcm::network::ImplementationVersionNameSub::ImplementationVersionNameSub ( )
```

### 10.161.3 Member Function Documentation

#### 10.161.3.1 Print()

```
void gdcm::network::ImplementationVersionNameSub::Print (
    std::ostream & os ) const
```

#### 10.161.3.2 Read()

```
std::istream & gdcm::network::ImplementationVersionNameSub::Read (
    std::istream & is )
```

#### 10.161.3.3 Size()

```
size_t gdcm::network::ImplementationVersionNameSub::Size ( ) const
```

#### 10.161.3.4 Write()

```
const std::ostream & gdcm::network::ImplementationVersionNameSub::Write (
    std::ostream & os ) const
```

The documentation for this class was generated from the following file:

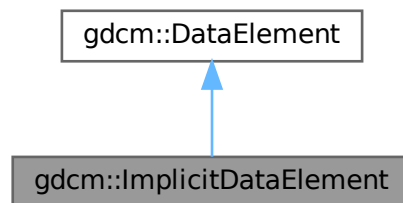
- [gdcmImplementationVersionNameSub.h](#)

## 10.162 gdcm::ImplicitDataElement Class Reference

Class to represent an *Implicit VR Data Element*.

```
#include <gdcmImplicitDataElement.h>
```

Inheritance diagram for gdcm::ImplicitDataElement:



Collaboration diagram for gdcm::ImplicitDataElement:



### Public Member Functions

- [VL GetLength](#) () const
- template<typename TSwap >  
std::istream & [Read](#) (std::istream &is)



- template<typename TSwap >  
std::istream & [ReadPreValue](#) (std::istream &is)
- template<typename TSwap >  
std::istream & [ReadValue](#) (std::istream &is, bool readvalues=true)
- template<typename TSwap >  
std::istream & [ReadValueWithLength](#) (std::istream &is, [VL](#) &length, bool readvalues=true)
- template<typename TSwap >  
std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length, bool readvalues=true)
- template<typename TSwap >  
const std::ostream & [Write](#) (std::ostream &os) const

## Public Member Functions inherited from [gdcm::DataElement](#)

- [DataElement](#) (const [DataElement](#) &\_val)
- [DataElement](#) (const [Tag](#) &t=[Tag](#)(0), const [VL](#) &vl=0, const [VR](#) &vr=[VR::INVALID](#))
- void [Clear](#) ()  
*Clear Data [Element](#) (make [Value](#) empty and invalidate [Tag](#) & [VR](#))*
- void [Empty](#) ()  
*Make Data [Element](#) empty (no [Value](#))*
- const [ByteValue](#) \* [GetByteValue](#) () const
- template<typename TDE >  
[VL](#) [GetLength](#) () const
- [SequenceOfFragments](#) \* [GetSequenceOfFragments](#) ()
- const [SequenceOfFragments](#) \* [GetSequenceOfFragments](#) () const
- [Tag](#) & [GetTag](#) ()
- const [Tag](#) & [GetTag](#) () const  
*Get [Tag](#).*
- [Value](#) & [GetValue](#) ()
- [Value](#) const & [GetValue](#) () const  
*Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):*
- [SmartPointer](#)< [SequenceOfItems](#) > [GetValueAsSQ](#) () const
- [VL](#) & [GetVL](#) ()
- const [VL](#) & [GetVL](#) () const  
*Get [VL](#).*
- [VR](#) const & [GetVR](#) () const
- bool [IsEmpty](#) () const  
*Check if Data [Element](#) is empty.*
- bool [IsUndefinedLength](#) () const  
*return if [Value](#) Length if of undefined length*
- bool [operator<](#) (const [DataElement](#) &de) const
- [DataElement](#) & [operator=](#) (const [DataElement](#) &)=default
- bool [operator==](#) (const [DataElement](#) &de) const
- template<typename TDE , typename TSwap >  
std::istream & [Read](#) (std::istream &is)
- template<typename TDE , typename TSwap >  
std::istream & [ReadOrSkip](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE , typename TSwap >  
std::istream & [ReadPreValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)

- `template<typename TDE , typename TSwap >`  
`std::istream & ReadValue (std::istream &is, std::set< Tag > const &skiptags)`
- `template<typename TDE , typename TSwap >`  
`std::istream & ReadValueWithLength (std::istream &is, VL &length, std::set< Tag > const &skiptags)`
- `template<typename TDE , typename TSwap >`  
`std::istream & ReadWithLength (std::istream &is, VL &length)`
- `void SetByteValue (const char *array, VL length)`
- `void SetTag (const Tag &t)`
- `void SetValue (Value const &vl)`
- `void SetVL (const VL &vl)`
- `void SetVLToUndefined ()`
- `void SetVR (VR const &vr)`
- `template<typename TDE , typename TSwap >`  
`const std::ostream & Write (std::ostream &os) const`

### Additional Inherited Members

### Protected Types inherited from [gdcm::DataElement](#)

- `typedef SmartPointer< Value > ValuePtr`

### Protected Member Functions inherited from [gdcm::DataElement](#)

- `void SetValueFieldLength (VL vl, bool readvalues)`

### Protected Attributes inherited from [gdcm::DataElement](#)

- `Tag TagField`
- `ValuePtr ValueField`
- `VL ValueLengthField`
- `VR VRField`

## 10.162.1 Detailed Description

Class to represent an *Implicit VR* Data Element.

#### Note

bla

#### Examples

[ReadExplicitLengthSQIVR.cxx](#).

## 10.162.2 Member Function Documentation

### 10.162.2.1 GetLength()

```
VL gdcm::ImplicitDataElement::GetLength ( ) const
```

### 10.162.2.2 Read()

```
template<typename TSwap >  
std::istream & gdcm::ImplicitDataElement::Read (  
    std::istream & is )
```

### 10.162.2.3 ReadPreValue()

```
template<typename TSwap >  
std::istream & gdcm::ImplicitDataElement::ReadPreValue (  
    std::istream & is )
```

### 10.162.2.4 ReadValue()

```
template<typename TSwap >  
std::istream & gdcm::ImplicitDataElement::ReadValue (  
    std::istream & is,  
    bool readvalues = true )
```

### 10.162.2.5 ReadValueWithLength()

```
template<typename TSwap >  
std::istream & gdcm::ImplicitDataElement::ReadValueWithLength (  
    std::istream & is,  
    VL & length,  
    bool readvalues = true )
```

### 10.162.2.6 ReadWithLength()

```
template<typename TSwap >  
std::istream & gdcm::ImplicitDataElement::ReadWithLength (  
    std::istream & is,  
    VL & length,  
    bool readvalues = true )
```

### 10.162.2.7 Write()

```
template<typename TSwap >  
const std::ostream & gdcM::ImplicitDataElement::Write (  
    std::ostream & os ) const
```

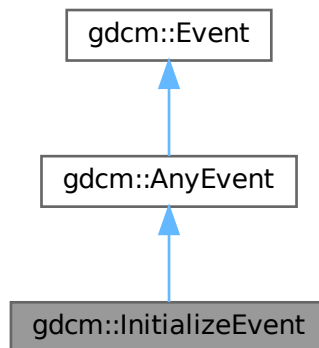
The documentation for this class was generated from the following file:

- [gdcMImplicitDataElement.h](#)

## 10.163 gdcM::InitializeEvent Class Reference

```
#include <gdcMEvent.h>
```

Inheritance diagram for gdcM::InitializeEvent:



Collaboration diagram for gdcm::InitializeEvent:



#### Additional Inherited Members

#### Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()
- [Event](#) (const [Event](#) &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) \*) const =0
- virtual const char \* [GetEventName](#) () const =0
- virtual [Event](#) \* [MakeObject](#) () const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

The documentation for this class was generated from the following file:

- [gdcmEvent.h](#)

## 10.164 gdcm::IOD Class Reference

Class for representing a [IOD](#).

```
#include <gdcmIOD.h>
```

#### Public Types

- typedef std::vector< [IODEntry](#) > [MapIODEntry](#)
- typedef MapIODEntry::size\_type [SizeType](#)

## Public Member Functions

- [IOD](#) ()=default
- void [AddIODEntry](#) (const [IODEntry](#) &iode)
- void [Clear](#) ()
- const [IODEntry](#) & [GetIODEntry](#) ([SizeType](#) idx) const
- [SizeType](#) [GetNumberOfIODs](#) () const
- [Type](#) [GetTypeFromTag](#) (const [Defs](#) &defs, const [Tag](#) &tag) const

## Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [IOD](#) &\_val)

## 10.164.1 Detailed Description

Class for representing a [IOD](#).

### Note

bla

### See also

[Dict](#)

### Examples

[TraverseModules.cxx](#).

## 10.164.2 Member Typedef Documentation

### 10.164.2.1 MapIODEntry

```
typedef std::vector<IODEntry> gdcm::IOD::MapIODEntry
```

### 10.164.2.2 SizeType

```
typedef MapIODEntry::size\_type gdcm::IOD::SizeType
```

## 10.164.3 Constructor & Destructor Documentation

### 10.164.3.1 IOD()

```
gdcm::IOD::IOD ( ) [default]
```

References [gdcm::operator<<\(\)](#).

## 10.164.4 Member Function Documentation

### 10.164.4.1 AddIODEntry()

```
void gdcm::IOD::AddIODEntry (
    const IODEntry & iode ) [inline]
```

### 10.164.4.2 Clear()

```
void gdcm::IOD::Clear ( ) [inline]
```

### 10.164.4.3 GetIODEntry()

```
const IODEntry & gdcm::IOD::GetIODEntry (
    SizeType idx ) const [inline]
```

#### Examples

[TraverseModules.cxx](#).

### 10.164.4.4 GetNumberOfIODs()

```
SizeType gdcm::IOD::GetNumberOfIODs ( ) const [inline]
```

#### Examples

[TraverseModules.cxx](#).

### 10.164.4.5 GetTypeFromTag()

```
Type gdcm::IOD::GetTypeFromTag (
    const Defs & defs,
    const Tag & tag ) const
```

## 10.164.5 Friends And Related Symbol Documentation

### 10.164.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const IOD & _val ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmIOD.h](#)

## 10.165 gdcm::IODEntry Class Reference

Class for representing a [IODEntry](#).

```
#include <gdcmIODEntry.h>
```

### Public Member Functions

- [IODEntry](#) (const char \*name="", const char \*ref="", const char \*inUsage="")
- const char \* [GetIE](#) () const
- const char \* [GetName](#) () const
- const char \* [GetRef](#) () const
- const char \* [GetUsage](#) () const
- [Usage::UsageType](#) [GetUsageType](#) () const
- void [SetIE](#) (const char \*ie)
- void [SetName](#) (const char \*name)
- void [SetRef](#) (const char \*ref)
- void [SetUsage](#) (const char \*inUsage)

### Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [IODEntry](#) &\_val)

### 10.165.1 Detailed Description

Class for representing a [IODEntry](#).

#### Note

A.1.3 [IOD Module Table](#) and Functional Group [Macro Table](#) This Section of each [IOD](#) defines in a tabular form the [Modules](#) comprising the [IOD](#). The following information must be specified for each [Module](#) in the table:

- The name of the [Module](#) or Functional Group
- A reference to the Section in Annex C which defines the [Module](#) or Functional Group
- The usage of the [Module](#) or Functional Group; whether it is:
  - Mandatory (see A.1.3.1) , abbreviated M
  - Conditional (see A.1.3.2) , abbreviated C
  - User Option (see A.1.3.3) , abbreviated U
- The [Modules](#) referenced are defined in Annex C. A.1.3.1 MANDATORY MODULES For each [IOD](#), Mandatory [Modules](#) shall be supported per the definitions, semantics and requirements defined in Annex C. PS 3.3 - 2008 Page 96
- Standard - A.1.3.2 CONDITIONAL MODULES Conditional [Modules](#) are Mandatory [Modules](#) if specific conditions are met. If the specified conditions are not met, this [Module](#) shall not be supported; that is, no information defined in that [Module](#) shall be sent. A.1.3.3 USER OPTION MODULES User Option [Modules](#) may or may not be supported. If an optional [Module](#) is supported, the [Attribute](#) Types specified in the [Modules](#) in Annex C shall be supported.

See also

[DictEntry](#)

Examples

[TraverseModules.cxx](#).



## 10.165.2 Constructor & Destructor Documentation

### 10.165.2.1 IODEntry()

```
gdcm::IODEntry::IODEntry (
    const char * name = "",
    const char * ref = "",
    const char * inUsage = "" ) [inline]
```

## 10.165.3 Member Function Documentation

### 10.165.3.1 GetIE()

```
const char * gdcm::IODEntry::GetIE ( ) const [inline]
```

### 10.165.3.2 GetName()

```
const char * gdcm::IODEntry::GetName ( ) const [inline]
```

### 10.165.3.3 GetRef()

```
const char * gdcm::IODEntry::GetRef ( ) const [inline]
```

#### Examples

[TraverseModules.cxx](#).

### 10.165.3.4 GetUsage()

```
const char * gdcm::IODEntry::GetUsage ( ) const [inline]
```

### 10.165.3.5 GetUsageType()

```
Usage::UsageType gdcm::IODEntry::GetUsageType ( ) const
```

### 10.165.3.6 SetIE()

```
void gdcm::IODEntry::SetIE (
    const char * ie ) [inline]
```

### 10.165.3.7 SetName()

```
void gdcmm::IODEntry::SetName (
    const char * name )    [inline]
```

### 10.165.3.8 SetRef()

```
void gdcmm::IODEntry::SetRef (
    const char * ref )    [inline]
```

### 10.165.3.9 SetUsage()

```
void gdcmm::IODEntry::SetUsage (
    const char * inUsage )    [inline]
```

## 10.165.4 Friends And Related Symbol Documentation

### 10.165.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const IODEntry & _val )    [friend]
```

The documentation for this class was generated from the following file:

- [gdcmmIODEntry.h](#)

## 10.166 gdcmm::IODs Class Reference

Class for representing a [IODs](#).

```
#include <gdcmmIODs.h>
```

### Public Types

- typedef std::map< [IODName](#), IOD > IODMapType
- typedef IODMapType::const\_iterator IODMapTypeConstIterator
- typedef std::string [IODName](#)

## Public Member Functions

- [IODs](#) ()=default
- void [AddIOD](#) (const char \*name, const [IOD](#) &module)
- [IODMapTypeConstIterator Begin](#) () const
- void [Clear](#) ()
- [IODMapTypeConstIterator End](#) () const
- const [IOD](#) & [GetIOD](#) (const char \*name) const

## Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [IODs](#) &\_val)

## 10.166.1 Detailed Description

Class for representing a [IODs](#).

### Note

bla

### See also

[IOD](#)

### Examples

[TraverseModules.cxx](#).

## 10.166.2 Member Typedef Documentation

### 10.166.2.1 IODMapType

```
typedef std::map<IODName, IOD> gdcm::IODs::IODMapType
```

### 10.166.2.2 IODMapTypeConstIterator

```
typedef IODMapType::const\_iterator gdcm::IODs::IODMapTypeConstIterator
```

### 10.166.2.3 IODName

```
typedef std::string gdcm::IODs::IODName
```

### 10.166.3 Constructor & Destructor Documentation

#### 10.166.3.1 IODs()

```
gdcm::IODs::IODs ( ) [default]
```

References [gdcm::operator<<\(\)](#).

### 10.166.4 Member Function Documentation

#### 10.166.4.1 AddIOD()

```
void gdcm::IODs::AddIOD (
    const char * name,
    const IOD & module ) [inline]
```

#### 10.166.4.2 Begin()

```
IODMapTypeConstIterator gdcm::IODs::Begin ( ) const [inline]
```

##### Examples

[TraverseModules.cxx](#).

#### 10.166.4.3 Clear()

```
void gdcm::IODs::Clear ( ) [inline]
```

#### 10.166.4.4 End()

```
IODMapTypeConstIterator gdcm::IODs::End ( ) const [inline]
```

##### Examples

[TraverseModules.cxx](#).

#### 10.166.4.5 GetIOD()

```
const IOD & gdcm::IODs::GetIOD (
    const char * name ) const [inline]
```

## 10.166.5 Friends And Related Symbol Documentation

### 10.166.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const IODs & _val ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmIODs.h](#)

## 10.167 gdcm::IPPSorter Class Reference

[IPPSorter.](#)

```
#include <gdcmIPPSorter.h>
```

Inheritance diagram for gdcm::IPPSorter:



Collaboration diagram for gdcm::IPPSorter:



## Public Member Functions

- [IPPSorter](#) ()
- double [GetDirectionCosinesTolerance](#) () const
- double [GetZSpacing](#) () const
- double [GetZSpacingTolerance](#) () const
- void [SetComputeZSpacing](#) (bool b)
- void [SetDirectionCosinesTolerance](#) (double tol)
- void [SetDropDuplicatePositions](#) (bool b)
- void [SetZSpacingTolerance](#) (double tol)
- bool [Sort](#) (std::vector< std::string > const &filenames) override

## Public Member Functions inherited from [gdcm::Sorter](#)

- [Sorter](#) ()
- virtual [~Sorter](#) ()
- bool [AddSelect](#) ([Tag](#) const &tag, const char \*value)  
*UNSUPPORTED FOR NOW.*
- const std::vector< std::string > & [GetFileNames](#) () const
- void [Print](#) (std::ostream &os) const  
*Print.*
- void [SetSortFunction](#) ([SortFunction](#) f)
- void [SetTagsToRead](#) (std::set< [Tag](#) > const &tags)
- virtual bool [StableSort](#) (std::vector< std::string > const &filenames)

## Protected Attributes

- bool [ComputeZSpacing](#)
- double [DirCosTolerance](#)
- bool [DropDuplicatePositions](#)
- double [ZSpacing](#)
- double [ZTolerance](#)

## Protected Attributes inherited from [gdcm::Sorter](#)

- std::vector< std::string > [FileNames](#)
- std::map< [Tag](#), std::string > [Selection](#)
- [SortFunction](#) [SortFunc](#)
- std::set< [Tag](#) > [TagsToRead](#)

## Additional Inherited Members

## Public Types inherited from [gdcm::Sorter](#)

- typedef bool(\* [SortFunction](#)) ([DataSet](#) const &, [DataSet](#) const &)  
*Set the sort function which compares one dataset to the other.*

## Protected Types inherited from [gdcm::Sorter](#)

- typedef std::map< [Tag](#), std::string > [SelectionMap](#)

### 10.167.1 Detailed Description

#### [IPPSorter](#).

Implement a simple [Image](#) Position ([Patient](#)) sorter, along the [Image Orientation](#) ([Patient](#)) direction. This algorithm does NOT support duplicate and will FAIL in case of duplicate IPP.

#### Warning

See special note for `SetZSpacingTolerance` when computing the ZSpacing from the IPP of each DICOM files (default tolerance for consistent spacing is: 1e-6mm)

For more information on [Spacing](#), and how it is defined in DICOM, advanced users may refers to:

[http://gdcm.sourceforge.net/wiki/index.php/Imager\\_Pixel\\_Spacing](http://gdcm.sourceforge.net/wiki/index.php/Imager_Pixel_Spacing)

**Bug** There are currently a couple of bugs in this implementation:

- Gantry Tilt is not considered (always an error)
- Application programmer should only sort valid [DataSet](#) (eg. [MRImageStorage](#), [CTImageStorage](#), [PETImageStorage](#))

#### Examples

[Compute3DSpacing.cxx](#), [VolumeSorter.cxx](#), [gdcmorthoplanes.cxx](#), and [reslicesphere.cxx](#).

## 10.167.2 Constructor & Destructor Documentation

### 10.167.2.1 IPPSorter()

```
gdcm::IPPSorter::IPPSorter ( )
```

## 10.167.3 Member Function Documentation

### 10.167.3.1 GetDirectionCosinesTolerance()

```
double gdcm::IPPSorter::GetDirectionCosinesTolerance ( ) const [inline]
```

### 10.167.3.2 GetZSpacing()

```
double gdcM::IPPSorter::GetZSpacing ( ) const [inline]
```

Read-only function to provide access to the computed value for the Z-Spacing. The `ComputeZSpacing` must have been set to true before execution of sort algorithm. Call this function *after* calling `Sort()`; Z-Spacing will be 0 on 2 occasions:

- Sorting simply failed, potentially duplicate IPP => ZSpacing = 0
- ZSpacing could not be computed (Z-Spacing is not constant, or ZTolerance is too low)

#### Examples

[Compute3DSpacing.cxx](#), [gdcMorthoplanes.cxx](#), and [reslicesphere.cxx](#).

### 10.167.3.3 GetZSpacingTolerance()

```
double gdcM::IPPSorter::GetZSpacingTolerance ( ) const [inline]
```

### 10.167.3.4 SetComputeZSpacing()

```
void gdcM::IPPSorter::SetComputeZSpacing (
    bool b ) [inline]
```

Functions related to Z-Spacing computation. Set to true when sort algorithm should also perform a regular Z-Spacing computation using the [Image](#) Position ([Patient](#)). Potential reason for failure:

1. ALL slices are taken into account, if one slice is missing then ZSpacing will be set to 0 since the spacing will not be found to be regular along the [Series](#)

#### Examples

[Compute3DSpacing.cxx](#), [VolumeSorter.cxx](#), [gdcMorthoplanes.cxx](#), and [reslicesphere.cxx](#).

### 10.167.3.5 SetDirectionCosinesTolerance()

```
void gdcM::IPPSorter::SetDirectionCosinesTolerance (
    double tol ) [inline]
```

Sometimes IOP along a series is slightly changing for example: "0.999081\\0.0426953\\0.00369272\\-0.0419025\\0.↔955059\\0.293439", "0.999081\\0.0426953\\0.00369275\\-0.0419025\\0.955059\\0.293439", "0.999081\\0.0426952\\0.↔00369272\\-0.0419025\\0.955059\\0.293439", We need an API to define the tolerance which is allowed. Internally the cross vector of each direction cosines is computed. The tolerance then defines the distance in between 1.0 to the dot product of those cross vectors. In a perfect world this dot product is of course 1.0 which implies a [DirectionCosines](#) tolerance of exactly 0.0 (default).



### 10.167.3.6 SetDropDuplicatePositions()

```
void gdcm::IPPSorter::SetDropDuplicatePositions (
    bool b ) [inline]
```

Makes the [IPPSorter](#) ignore multiple images located at the same position. Only the first occurrence will be kept. DropDuplicatePositions defaults to false.

### 10.167.3.7 SetZSpacingTolerance()

```
void gdcm::IPPSorter::SetZSpacingTolerance (
    double tol ) [inline]
```

1. Another reason for failure is that that Z-Spacing is only slightly changing (eg 1e-3) along the series, a human can determine that this is ok and change the tolerance from its default value: 1e-6

#### Examples

[Compute3DSpacing.cxx](#), [gdcmorthoplanes.cxx](#), and [reslicesphere.cxx](#).

### 10.167.3.8 Sort()

```
bool gdcm::IPPSorter::Sort (
    std::vector< std::string > const & filenames ) [override], [virtual]
```

Main entry point to the sorter. It will execute the filter, option should be set before running this function (SetZSpacingTolerance, ...) Return value indicate if sorting could be achieved,. Warning this does *NOT* imply that spacing is consistent, it only means the file are sorted according to IPP You should check if ZSpacing is 0 or not to deduce if file are actually a 3D volume

Reimplemented from [gdcm::Sorter](#).

#### Examples

[Compute3DSpacing.cxx](#), [VolumeSorter.cxx](#), [gdcmorthoplanes.cxx](#), and [reslicesphere.cxx](#).

## 10.167.4 Member Data Documentation

### 10.167.4.1 ComputeZSpacing

```
bool gdcm::IPPSorter::ComputeZSpacing [protected]
```

### 10.167.4.2 DirCosTolerance

```
double gdcm::IPPSorter::DirCosTolerance [protected]
```

#### 10.167.4.3 DropDuplicatePositions

```
bool gdcM::IPPSorter::DropDuplicatePositions [protected]
```

#### 10.167.4.4 ZSpacing

```
double gdcM::IPPSorter::ZSpacing [protected]
```

#### 10.167.4.5 ZTolerance

```
double gdcM::IPPSorter::ZTolerance [protected]
```

The documentation for this class was generated from the following file:

- [gdcMIPPSorter.h](#)

### 10.168 gdcM::Item Class Reference

Class to represent an [Item](#).

```
#include <gdcMItem.h>
```

Inheritance diagram for gdcM::Item:



Collaboration diagram for gdcm::Item:



### Public Member Functions

- [Item](#) ()
- [Item](#) ([Item](#) const &val)
- void [Clear](#) ()
- bool [FindDataElement](#) (const [Tag](#) &t) const
- const [DataElement](#) & [GetDataElement](#) (const [Tag](#) &t) const
- template<typename TDE >  
  [VL](#) [GetLength](#) () const
- [DataSet](#) & [GetNestedDataSet](#) ()
- const [DataSet](#) & [GetNestedDataSet](#) () const
- void [InsertDataElement](#) (const [DataElement](#) &de)
- template<typename TDE , typename TSwap >  
  std::istream & [Read](#) (std::istream &is)
- void [SetNestedDataSet](#) (const [DataSet](#) &nested)
- template<typename TDE , typename TSwap >  
  const std::ostream & [Write](#) (std::ostream &os) const

### Public Member Functions inherited from [gdcm::DataElement](#)

- [DataElement](#) (const [DataElement](#) &\_val)
- [DataElement](#) (const [Tag](#) &t=[Tag](#)(0), const [VL](#) &vl=0, const [VR](#) &vr=[VR](#)::INVALID)
- void [Clear](#) ()  
  Clear Data [Element](#) (make [Value](#) empty and invalidate [Tag](#) & [VR](#))
- void [Empty](#) ()  
  Make Data [Element](#) empty (no [Value](#))
- const [ByteValue](#) \* [GetByteValue](#) () const

- `template<typename TDE >`  
`VL GetLength () const`
- `SequenceOfFragments * GetSequenceOfFragments ()`
- `const SequenceOfFragments * GetSequenceOfFragments () const`
- `Tag & GetTag ()`
- `const Tag & GetTag () const`  
*Get Tag.*
- `Value & GetValue ()`
- `Value const & GetValue () const`  
*Set/Get Value (bytes array, SQ of items, SQ of fragments):*
- `SmartPointer< SequenceOfItems > GetValueAsSQ () const`
- `VL & GetVL ()`
- `const VL & GetVL () const`  
*Get VL.*
- `VR const & GetVR () const`
- `bool IsEmpty () const`  
*Check if Data Element is empty.*
- `bool IsUndefinedLength () const`  
*return if Value Length if of undefined length*
- `bool operator< (const DataElement &de) const`
- `DataElement & operator= (const DataElement &)=default`
- `bool operator== (const DataElement &de) const`
- `template<typename TDE , typename TSwap >`  
`std::istream & Read (std::istream &is)`
- `template<typename TDE , typename TSwap >`  
`std::istream & ReadOrSkip (std::istream &is, std::set< Tag > const &skiptags)`
- `template<typename TDE , typename TSwap >`  
`std::istream & ReadPreValue (std::istream &is, std::set< Tag > const &skiptags)`
- `template<typename TDE , typename TSwap >`  
`std::istream & ReadValue (std::istream &is, std::set< Tag > const &skiptags)`
- `template<typename TDE , typename TSwap >`  
`std::istream & ReadValueWithLength (std::istream &is, VL &length, std::set< Tag > const &skiptags)`
- `template<typename TDE , typename TSwap >`  
`std::istream & ReadWithLength (std::istream &is, VL &length)`
- `void SetByteValue (const char *array, VL length)`
- `void SetTag (const Tag &t)`
- `void SetValue (Value const &vl)`
- `void SetVL (const VL &vl)`
- `void SetVLToUndefined ()`
- `void SetVR (VR const &vr)`
- `template<typename TDE , typename TSwap >`  
`const std::ostream & Write (std::ostream &os) const`

## Friends

- `std::ostream & operator<< (std::ostream &os, const Item &val)`

## Additional Inherited Members

## Protected Types inherited from [gdcm::DataElement](#)

- typedef [SmartPointer](#)< [Value](#) > [ValuePtr](#)

## Protected Member Functions inherited from [gdcm::DataElement](#)

- void [SetValueFieldLength](#) ([VL](#) vl, bool readvalues)

## Protected Attributes inherited from [gdcm::DataElement](#)

- [Tag](#) [TagField](#)
- [ValuePtr](#) [ValueField](#)
- [VL](#) [ValueLengthField](#)
- [VR](#) [VRField](#)

## 10.168.1 Detailed Description

Class to represent an [Item](#).

A component of the value of a Data [Element](#) that is of [Value](#) Representation Sequence of Items. An [Item](#) contains a Data Set . See PS 3.5 7.5.1 [Item](#) Encoding Rules Each [Item](#) of a Data [Element](#) of [VR](#) SQ shall be encoded as a DICOM Standard Data [Element](#) with a specific Data [Element](#) [Tag](#) of [Value](#) (FFFE,E000). The [Item](#) [Tag](#) is followed by a 4 byte [Item](#) Length field encoded in one of the following two ways Explicit/ Implicit

### Note

ITEM: A component of the [Value](#) of a Data [Element](#) that is of [Value](#) Representation Sequence of Items. An [Item](#) contains a Data Set.

### Examples

[ChangeSequenceUltrasound.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [NewSequence.cs](#), [SimplePrint.cs](#), [gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

## 10.168.2 Constructor & Destructor Documentation

### 10.168.2.1 [Item\(\)](#) [1/2]

```
gdcm::Item::Item ( ) [inline]
```

**10.168.2.2 Item() [2/2]**

```
gdcmm::Item::Item (
    Item const & val ) [inline]
```

**10.168.3 Member Function Documentation****10.168.3.1 Clear()**

```
void gdcmm::Item::Clear ( ) [inline]
```

Referenced by [gdcmm::SequenceOfItems::Read\(\)](#).

**10.168.3.2 FindDataElement()**

```
bool gdcmm::Item::FindDataElement (
    const Tag & t ) const [inline]
```

**10.168.3.3 GetDataElement()**

```
const DataElement & gdcmm::Item::GetDataElement (
    const Tag & t ) const [inline]
```

**10.168.3.4 GetLength()**

```
template<typename TDE >
VL gdcmm::Item::GetLength ( ) const
```

**10.168.3.5 GetNestedDataSet() [1/2]**

```
DataSet & gdcmm::Item::GetNestedDataSet ( ) [inline]
```

**10.168.3.6 GetNestedDataSet() [2/2]**

```
const DataSet & gdcmm::Item::GetNestedDataSet ( ) const [inline]
```

**Examples**

[ChangeSequenceUltrasound.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [NewSequence.cs](#), [SimplePrint.cs](#), [gdcmmrtionplan.cxx](#), and [gdcmmrtplan.cxx](#).

Referenced by [gdcmm::SequenceOfItems::Read\(\)](#).

### 10.168.3.7 InsertDataElement()

```
void gdcm::Item::InsertDataElement (
    const DataElement & de ) [inline]
```

### 10.168.3.8 Read()

```
template<typename TDE , typename TSwap >
std::istream & gdcm::Item::Read (
    std::istream & is ) [inline]
```

References [gdcm::ByteSwapFilter::ByteSwap\(\)](#), [gdcm::DataSet::Clear\(\)](#), [gdcmDebugMacro](#), [gdcmErrorMacro](#), [gdcmWarningMacro](#), [gdcm::DataSet::IsEmpty\(\)](#), and [gdcm::ByteSwapFilter::SetByteSwapTag\(\)](#).

Referenced by [gdcm::SequenceOfItems::Read\(\)](#).

### 10.168.3.9 SetNestedDataSet()

```
void gdcm::Item::SetNestedDataSet (
    const DataSet & nested ) [inline]
```

### 10.168.3.10 Write()

```
template<typename TDE , typename TSwap >
const std::ostream & gdcm::Item::Write (
    std::ostream & os ) const [inline]
```

References [gdcmWarningMacro](#), [gdcm::VL::GetLength\(\)](#), [gdcm::Tag::Write\(\)](#), and [gdcm::VL::Write\(\)](#).

## 10.168.4 Friends And Related Symbol Documentation

### 10.168.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & os,
    const Item & val ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmItem.h](#)

## 10.169 gdcm::IterationEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::IterationEvent:



Collaboration diagram for gdcm::IterationEvent:



### Additional Inherited Members

### Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()



- [Event](#) (const [Event](#) &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) \*) const =0
- virtual const char \* [GetEventName](#) () const =0
- virtual [Event](#) \* [MakeObject](#) () const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

The documentation for this class was generated from the following file:

- [gdcmEvent.h](#)

## 10.170 gdcm::JPEG12Codec Class Reference

Class to do JPEG 12bits (lossy & lossless)

```
#include <gdcmJPEG12Codec.h>
```

Inheritance diagram for gdcm::JPEG12Codec:



Collaboration diagram for `gdcm::JPEG12Codec`:



## Public Member Functions

- [JPEG12Codec](#) ()
- [~JPEG12Codec](#) () override
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os) override
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override
- bool [InternalCode](#) (const char \*input, unsigned long len, std::ostream &os) override

## Public Member Functions inherited from [gdcm::JPEGCodec](#)

- [JPEGCodec](#) ()
- [~JPEGCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const override  
*Return whether this coder support this transfer syntax (can code it)*
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const override  
*Return whether this decoder support this transfer syntax (can decode it)*
- [ImageCodec](#) \* [Clone](#) () const override
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out) override  
*Compress into JPEG.*
- void [ComputeOffsetTable](#) (bool b)  
*Compute the offset table:*
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override  
*Decode.*
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override
- bool [GetLossless](#) () const
- double [GetQuality](#) () const
- void [SetLossless](#) (bool l)
- void [SetPixelFormat](#) ([PixelFormat](#) const &pf) override
- void [SetQuality](#) (double q)

**Public Member Functions inherited from [gdcm::ImageCodec](#)**

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CleanupUnusedBits](#) (char \*data, size\_t datalen)
- const unsigned int \* [GetDimensions](#) () const
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- void [SetPlanarConfiguration](#) (unsigned int pc)

**Public Member Functions inherited from [gdcm::Coder](#)**

- virtual [~Coder](#) ()=default

**Public Member Functions inherited from [gdcm::Decoder](#)**

- virtual [~Decoder](#) ()=default

**Protected Member Functions**

- bool [EncodeBuffer](#) (std::ostream &os, const char \*data, size\_t datalen) override
- bool [IsStateSuspension](#) () const override

**Protected Member Functions inherited from [gdcm::JPEGCodec](#)**

- bool [AppendFrameEncode](#) (std::ostream &out, const char \*data, size\_t datalen) override
- bool [AppendRowEncode](#) (std::ostream &out, const char \*data, size\_t datalen) override
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os) override
- bool [DecodeExtent](#) (char \*buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is)
- bool [IsFrameEncoder](#) () override
- bool [IsRowEncoder](#) () override
- bool [IsValid](#) ([PhotometricInterpretation](#) const &pi) override
- void [SetBitSample](#) (int bit)
- bool [StartEncode](#) (std::ostream &) override
- bool [StopEncode](#) (std::ostream &) override

### Protected Member Functions inherited from [gdcm::ImageCodec](#)

- bool [DoByteSwap](#) (std::istream &is\_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is\_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is\_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is\_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is\_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is\_, std::ostream &os)

### Additional Inherited Members

### Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

### Protected Attributes inherited from [gdcm::JPEGCodec](#)

- int [BitSample](#)
- int [Quality](#)

### Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) [LUT](#)
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) [PF](#)
- [PhotometricInterpretation](#) [PI](#)
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

## 10.170.1 Detailed Description

Class to do JPEG 12bits (lossy & lossless)

#### Note

internal class

## 10.170.2 Constructor & Destructor Documentation

### 10.170.2.1 JPEG12Codec()

```
gdcm::JPEG12Codec::JPEG12Codec ( )
```

### 10.170.2.2 ~JPEG12Codec()

```
gdcm::JPEG12Codec::~~JPEG12Codec ( ) [override]
```

## 10.170.3 Member Function Documentation

### 10.170.3.1 DecodeByStreams()

```
bool gdcm::JPEG12Codec::DecodeByStreams (
    std::istream & is,
    std::ostream & os ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.170.3.2 EncodeBuffer()

```
bool gdcm::JPEG12Codec::EncodeBuffer (
    std::ostream & os,
    const char * data,
    size_t datalen ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

### 10.170.3.3 GetHeaderInfo()

```
bool gdcm::JPEG12Codec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.170.3.4 InternalCode()

```
bool gdcm::JPEG12Codec::InternalCode (
    const char * input,
    unsigned long len,
    std::ostream & os ) [override], [virtual]
```

Reimplemented from [gdcm::Coder](#).

### 10.170.3.5 IsStateSuspension()

```
bool gdcM::JPEG12Codec::IsStateSuspension ( ) const [override], [protected], [virtual]
```

Reimplemented from [gdcM::JPEGCodec](#).

The documentation for this class was generated from the following file:

- [gdcMJPEG12Codec.h](#)

## 10.171 gdcM::JPEG16Codec Class Reference

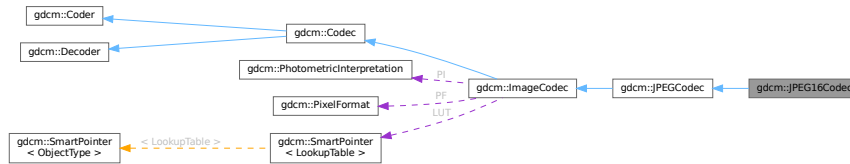
Class to do JPEG 16bits (lossless)

```
#include <gdcMJPEG16Codec.h>
```

Inheritance diagram for gdcM::JPEG16Codec:



Collaboration diagram for gdcm::JPEG16Codec:



## Public Member Functions

- [JPEG16Codec](#) ()
- [~JPEG16Codec](#) () override
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os) override
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override
- bool [InternalCode](#) (const char \*input, unsigned long len, std::ostream &os) override

## Public Member Functions inherited from [gdcm::JPEGCodec](#)

- [JPEGCodec](#) ()
- [~JPEGCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const override  
*Return whether this coder support this transfer syntax (can code it)*
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const override  
*Return whether this decoder support this transfer syntax (can decode it)*
- [ImageCodec](#) \* [Clone](#) () const override
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out) override  
*Compress into JPEG.*
- void [ComputeOffsetTable](#) (bool b)  
*Compute the offset table:*
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override  
*Decode.*
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override
- bool [GetLossless](#) () const
- double [GetQuality](#) () const
- void [SetLossless](#) (bool l)
- void [SetPixelFormat](#) ([PixelFormat](#) const &pf) override
- void [SetQuality](#) (double q)

### Public Member Functions inherited from [gdcm::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CleanupUnusedBits](#) (char \*data, size\_t datalen)
- const unsigned int \* [GetDimensions](#) () const
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- void [SetPlanarConfiguration](#) (unsigned int pc)

### Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default

### Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

### Protected Member Functions

- bool [EncodeBuffer](#) (std::ostream &os, const char \*data, size\_t datalen) override
- bool [IsStateSuspension](#) () const override

### Protected Member Functions inherited from [gdcm::JPEGCodec](#)

- bool [AppendFrameEncode](#) (std::ostream &out, const char \*data, size\_t datalen) override
- bool [AppendRowEncode](#) (std::ostream &out, const char \*data, size\_t datalen) override
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os) override
- bool [DecodeExtent](#) (char \*buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is)
- bool [IsFrameEncoder](#) () override
- bool [IsRowEncoder](#) () override
- bool [IsValid](#) ([PhotometricInterpretation](#) const &pi) override
- void [SetBitSample](#) (int bit)
- bool [StartEncode](#) (std::ostream &) override
- bool [StopEncode](#) (std::ostream &) override



## Protected Member Functions inherited from [gdcm::ImageCodec](#)

- bool [DoByteSwap](#) (std::istream &is\_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is\_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is\_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is\_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is\_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is\_, std::ostream &os)

## Additional Inherited Members

## Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

## Protected Attributes inherited from [gdcm::JPEGCodec](#)

- int [BitSample](#)
- int [Quality](#)

## Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) [LUT](#)
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) [PF](#)
- [PhotometricInterpretation](#) [PI](#)
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

### 10.171.1 Detailed Description

Class to do JPEG 16bits (lossless)

#### Note

internal class

## 10.171.2 Constructor & Destructor Documentation

### 10.171.2.1 JPEG16Codec()

```
gdcm::JPEG16Codec::JPEG16Codec ( )
```

### 10.171.2.2 ~JPEG16Codec()

```
gdcm::JPEG16Codec::~~JPEG16Codec ( ) [override]
```

## 10.171.3 Member Function Documentation

### 10.171.3.1 DecodeByStreams()

```
bool gdcm::JPEG16Codec::DecodeByStreams (
    std::istream & is,
    std::ostream & os ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.171.3.2 EncodeBuffer()

```
bool gdcm::JPEG16Codec::EncodeBuffer (
    std::ostream & os,
    const char * data,
    size_t datalen ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

### 10.171.3.3 GetHeaderInfo()

```
bool gdcm::JPEG16Codec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.171.3.4 InternalCode()

```
bool gdcm::JPEG16Codec::InternalCode (
    const char * input,
    unsigned long len,
    std::ostream & os ) [override], [virtual]
```

Reimplemented from [gdcm::Coder](#).

### 10.171.3.5 IsStateSuspension()

```
bool gdcm::JPEG16Codec::IsStateSuspension ( ) const [override], [protected], [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

The documentation for this class was generated from the following file:

- [gdcmJPEG16Codec.h](#)

## 10.172 gdcm::JPEG2000Codec Class Reference

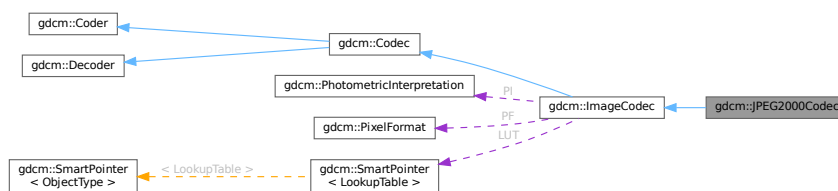
Class to do JPEG 2000.

```
#include <gdcmJPEG2000Codec.h>
```

Inheritance diagram for gdcm::JPEG2000Codec:



Collaboration diagram for gdcm::JPEG2000Codec:



## Public Member Functions

- [JPEG2000Codec](#) ()
- [~JPEG2000Codec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const override  
*Return whether this coder support this transfer syntax (can code it)*
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const override  
*Return whether this decoder support this transfer syntax (can decode it)*
- [ImageCodec](#) \* [Clone](#) () const override
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out) override  
*Code.*
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override  
*Decode.*
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override
- double [GetQuality](#) (unsigned int idx=0) const
- double [GetRate](#) (unsigned int idx=0) const
- void [SetMCT](#) (unsigned int mct)
- void [SetNumberOfResolutions](#) (unsigned int nres)
- void [SetNumberOfThreadsForDecompression](#) (int nThreads)
- void [SetQuality](#) (unsigned int idx, double q)
- void [SetRate](#) (unsigned int idx, double rate)
- void [SetReversible](#) (bool res)
- void [SetTileSize](#) (unsigned int tx, unsigned int ty)

## Public Member Functions inherited from [gdcm::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CleanupUnusedBits](#) (char \*data, size\_t datalen)
- const unsigned int \* [GetDimensions](#) () const
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- virtual void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)

**Public Member Functions inherited from [gdcm::Coder](#)**

- virtual [~Coder](#) ()=default

**Public Member Functions inherited from [gdcm::Decoder](#)**

- virtual [~Decoder](#) ()=default

**Protected Member Functions**

- bool [AppendFrameEncode](#) (std::ostream &out, const char \*data, size\_t datalen) override
- bool [AppendRowEncode](#) (std::ostream &out, const char \*data, size\_t datalen) override
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os) override
- bool [DecodeExtent](#) (char \*buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is)
- bool [IsFrameEncoder](#) () override
- bool [IsRowEncoder](#) () override
- bool [StartEncode](#) (std::ostream &) override
- bool [StopEncode](#) (std::ostream &) override

**Protected Member Functions inherited from [gdcm::ImageCodec](#)**

- bool [DoByteSwap](#) (std::istream &is\_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is\_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is\_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is\_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is\_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is\_, std::ostream &os)
- virtual bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)

**Protected Member Functions inherited from [gdcm::Coder](#)**

- virtual bool [InternalCode](#) (const char \*bv, unsigned long len, std::ostream &os)

**Friends**

- class [Bitmap](#)
- class [ImageRegionReader](#)

**Additional Inherited Members****Protected Types inherited from [gdcm::ImageCodec](#)**

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

## Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) LUT
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) PF
- [PhotometricInterpretation](#) PI
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

### 10.172.1 Detailed Description

Class to do JPEG 2000.

#### Note

the class will produce JPC (JPEG 2000 codestream), since some private implementor are using full jp2 file the decoder tolerate jp2 input this is an implementation of an [ImageCodec](#)

### 10.172.2 Constructor & Destructor Documentation

#### 10.172.2.1 JPEG2000Codec()

```
gdcm::JPEG2000Codec::JPEG2000Codec ( )
```

#### 10.172.2.2 ~JPEG2000Codec()

```
gdcm::JPEG2000Codec::~~JPEG2000Codec ( ) [override]
```

### 10.172.3 Member Function Documentation

#### 10.172.3.1 AppendFrameEncode()

```
bool gdcm::JPEG2000Codec::AppendFrameEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.172.3.2 AppendRowEncode()

```
bool gdcm::JPEG2000Codec::AppendRowEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.172.3.3 CanCode()

```
bool gdcm::JPEG2000Codec::CanCode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.172.3.4 CanDecode()

```
bool gdcm::JPEG2000Codec::CanDecode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.172.3.5 Clone()

```
ImageCodec * gdcm::JPEG2000Codec::Clone ( ) const [override], [virtual]
```

Implements [gdcm::ImageCodec](#).

### 10.172.3.6 Code()

```
bool gdcm::JPEG2000Codec::Code (
    DataElement const & in_,
    DataElement & out_ ) [override], [virtual]
```

Code.

Reimplemented from [gdcm::Coder](#).

### 10.172.3.7 Decode()

```
bool gdcM::JPEG2000Codec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcM::ImageCodec](#).

### 10.172.3.8 DecodeByStreams()

```
bool gdcM::JPEG2000Codec::DecodeByStreams (
    std::istream & is,
    std::ostream & os ) [override], [protected], [virtual]
```

Reimplemented from [gdcM::ImageCodec](#).

### 10.172.3.9 DecodeExtent()

```
bool gdcM::JPEG2000Codec::DecodeExtent (
    char * buffer,
    unsigned int xmin,
    unsigned int xmax,
    unsigned int ymin,
    unsigned int ymax,
    unsigned int zmin,
    unsigned int zmax,
    std::istream & is ) [protected]
```

### 10.172.3.10 GetHeaderInfo()

```
bool gdcM::JPEG2000Codec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [override], [virtual]
```

Reimplemented from [gdcM::ImageCodec](#).

### 10.172.3.11 GetQuality()

```
double gdcM::JPEG2000Codec::GetQuality (
    unsigned int idx = 0 ) const
```

### 10.172.3.12 GetRate()

```
double gdcM::JPEG2000Codec::GetRate (
    unsigned int idx = 0 ) const
```



### 10.172.3.13 IsFrameEncoder()

```
bool gdcm::JPEG2000Codec::IsFrameEncoder ( ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.172.3.14 IsRowEncoder()

```
bool gdcm::JPEG2000Codec::IsRowEncoder ( ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.172.3.15 SetMCT()

```
void gdcm::JPEG2000Codec::SetMCT (
    unsigned int mct )
```

### 10.172.3.16 SetNumberOfResolutions()

```
void gdcm::JPEG2000Codec::SetNumberOfResolutions (
    unsigned int nres )
```

### 10.172.3.17 SetNumberOfThreadsForDecompression()

```
void gdcm::JPEG2000Codec::SetNumberOfThreadsForDecompression (
    int nThreads )
```

Set Number of threads

#### Parameters

<i>nThreads</i>	: number of threads for decompression codec, if 0 or 1 decompression is done in current thread, if negative value is set determine how many virtual threads are available
-----------------	---

### 10.172.3.18 SetQuality()

```
void gdcm::JPEG2000Codec::SetQuality (
    unsigned int idx,
    double q )
```

#### 10.172.3.19 SetRate()

```
void gdcm::JPEG2000Codec::SetRate (
    unsigned int idx,
    double rate )
```

#### 10.172.3.20 SetReversible()

```
void gdcm::JPEG2000Codec::SetReversible (
    bool res )
```

#### 10.172.3.21 SetTileSize()

```
void gdcm::JPEG2000Codec::SetTileSize (
    unsigned int tx,
    unsigned int ty )
```

#### 10.172.3.22 StartEncode()

```
bool gdcm::JPEG2000Codec::StartEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### 10.172.3.23 StopEncode()

```
bool gdcm::JPEG2000Codec::StopEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.172.4 Friends And Related Symbol Documentation

#### 10.172.4.1 Bitmap

```
friend class Bitmap [friend]
```

#### 10.172.4.2 ImageRegionReader

```
friend class ImageRegionReader [friend]
```

The documentation for this class was generated from the following file:

- [gdcmJPEG2000Codec.h](#)

## 10.173 gdcm::JPEG8Codec Class Reference

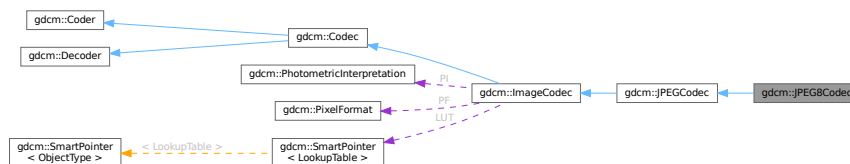
Class to do JPEG 8bits (lossy & lossless)

```
#include <gdcmJPEG8Codec.h>
```

Inheritance diagram for gdcm::JPEG8Codec:



Collaboration diagram for gdcm::JPEG8Codec:



### Public Member Functions

- [JPEG8Codec](#) ()
- [~JPEG8Codec](#) () override
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os) override
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override
- bool [InternalCode](#) (const char \*input, unsigned long len, std::ostream &os) override

## Public Member Functions inherited from [gdcm::JPEGCodec](#)

- [JPEGCodec](#) ()
- [~JPEGCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const override  
*Return whether this coder support this transfer syntax (can code it)*
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const override  
*Return whether this decoder support this transfer syntax (can decode it)*
- [ImageCodec](#) \* [Clone](#) () const override
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out) override  
*Compress into JPEG.*
- void [ComputeOffsetTable](#) (bool b)  
*Compute the offset table:*
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override  
*Decode.*
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override
- bool [GetLossless](#) () const
- double [GetQuality](#) () const
- void [SetLossless](#) (bool l)
- void [SetPixelFormat](#) ([PixelFormat](#) const &pf) override
- void [SetQuality](#) (double q)

## Public Member Functions inherited from [gdcm::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CleanupUnusedBits](#) (char \*data, size\_t datalen)
- const unsigned int \* [GetDimensions](#) () const
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- void [SetPlanarConfiguration](#) (unsigned int pc)

## Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default

## Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

## Protected Member Functions

- bool [EncodeBuffer](#) (std::ostream &os, const char \*data, size\_t datalen) override
- bool [IsStateSuspension](#) () const override

## Protected Member Functions inherited from [gdcm::JPEGCodec](#)

- bool [AppendFrameEncode](#) (std::ostream &out, const char \*data, size\_t datalen) override
- bool [AppendRowEncode](#) (std::ostream &out, const char \*data, size\_t datalen) override
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os) override
- bool [DecodeExtent](#) (char \*buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is)
- bool [IsFrameEncoder](#) () override
- bool [IsRowEncoder](#) () override
- bool [IsValid](#) ([PhotometricInterpretation](#) const &pi) override
- void [SetBitSample](#) (int bit)
- bool [StartEncode](#) (std::ostream &) override
- bool [StopEncode](#) (std::ostream &) override

## Protected Member Functions inherited from [gdcm::ImageCodec](#)

- bool [DoByteSwap](#) (std::istream &is\_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is\_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is\_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is\_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is\_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is\_, std::ostream &os)

## Additional Inherited Members

## Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

### Protected Attributes inherited from [gdcm::JPEGCodec](#)

- int [BitSample](#)
- int [Quality](#)

### Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) LUT
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) PF
- [PhotometricInterpretation](#) PI
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

## 10.173.1 Detailed Description

Class to do JPEG 8bits (lossy & lossless)

### Note

internal class

## 10.173.2 Constructor & Destructor Documentation

### 10.173.2.1 JPEG8Codec()

```
gdcm::JPEG8Codec::JPEG8Codec ( )
```

### 10.173.2.2 ~JPEG8Codec()

```
gdcm::JPEG8Codec::~~JPEG8Codec ( ) [override]
```

## 10.173.3 Member Function Documentation

### 10.173.3.1 DecodeByStreams()

```
bool gdcm::JPEG8Codec::DecodeByStreams (
    std::istream & is,
    std::ostream & os ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.173.3.2 EncodeBuffer()

```
bool gdcm::JPEG8Codec::EncodeBuffer (
    std::ostream & os,
    const char * data,
    size_t datalen ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

### 10.173.3.3 GetHeaderInfo()

```
bool gdcm::JPEG8Codec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.173.3.4 InternalCode()

```
bool gdcm::JPEG8Codec::InternalCode (
    const char * input,
    unsigned long len,
    std::ostream & os ) [override], [virtual]
```

Reimplemented from [gdcm::Coder](#).

### 10.173.3.5 IsStateSuspension()

```
bool gdcm::JPEG8Codec::IsStateSuspension ( ) const [override], [protected], [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

The documentation for this class was generated from the following file:

- [gdcmJPEG8Codec.h](#)

## 10.174 gdcm::JPEGCodec Class Reference

JPEG codec.

```
#include <gdcmJPEGCodec.h>
```

Inheritance diagram for gdcm::JPEGCodec:



Collaboration diagram for gdcm::JPEGCodec:



### Public Member Functions

- [JPEGCodec](#) ()
- [~JPEGCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const override  
*Return whether this coder support this transfer syntax (can code it)*
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const override



*Return whether this decoder support this transfer syntax (can decode it)*

- [ImageCodec](#) \* [Clone](#) () const override
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out) override

*Compress into JPEG.*

- void [ComputeOffsetTable](#) (bool b)

*Compute the offset table:*

- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override

*Decode.*

- virtual bool [EncodeBuffer](#) (std::ostream &out, const char \*inbuffer, size\_t inlen)
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override
- bool [GetLossless](#) () const
- double [GetQuality](#) () const
- void [SetLossless](#) (bool l)
- void [SetPixelFormat](#) ([PixelFormat](#) const &pf) override
- void [SetQuality](#) (double q)

## Public Member Functions inherited from [gdcm::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CleanupUnusedBits](#) (char \*data, size\_t datalen)
- const unsigned int \* [GetDimensions](#) () const
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- void [SetPlanarConfiguration](#) (unsigned int pc)

## Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default

## Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

### Protected Member Functions

- bool [AppendFrameEncode](#) (std::ostream &out, const char \*data, size\_t datalen) override
- bool [AppendRowEncode](#) (std::ostream &out, const char \*data, size\_t datalen) override
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os) override
- bool [DecodeExtent](#) (char \*buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is)
- bool [IsFrameEncoder](#) () override
- bool [IsRowEncoder](#) () override
- virtual bool [IsStateSuspension](#) () const
- bool [IsValid](#) (PhotometricInterpretation const &pi) override
- void [SetBitSample](#) (int bit)
- bool [StartEncode](#) (std::ostream &) override
- bool [StopEncode](#) (std::ostream &) override

### Protected Member Functions inherited from [gdcm::ImageCodec](#)

- bool [DoByteSwap](#) (std::istream &is\_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is\_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is\_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is\_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is\_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is\_, std::ostream &os)

### Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char \*bv, unsigned long len, std::ostream &os)

### Protected Attributes

- int [BitSample](#)
- int [Quality](#)

### Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) LUT
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) PF
- [PhotometricInterpretation](#) PI
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

## Friends

- class [ImageRegionReader](#)

## Additional Inherited Members

## Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > LUTPtr

### 10.174.1 Detailed Description

JPEG codec.

Class to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispach in between the different codec implementation: [JPEG8Codec](#), [JPEG12Codec](#) & [JPEG16Codec](#) It also support inconsistency in between DICOM header and JPEG compressed stream [ImageCodec](#) implementation for the JPEG case

#### Note

Things you should know if you ever want to dive into DICOM/JPEG world (among other):

- [http://groups.google.com/group/comp.protocols.dicom/browse\\_thread/thread/625e46919f208](http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/625e46919f208)
- [http://groups.google.com/group/comp.protocols.dicom/browse\\_thread/thread/75fdfccc65a62](http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/75fdfccc65a62)
- [http://groups.google.com/group/comp.protocols.dicom/browse\\_thread/thread/2d525ef6a2f09](http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/2d525ef6a2f09)
- [http://groups.google.com/group/comp.protocols.dicom/browse\\_thread/thread/6b93af410f8c](http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/6b93af410f8c)

#### Examples

[CompressLossyJPEG.cs](#), [FileChangeTSLossy.cs](#), and [GetJPEGSamplePrecision.cxx](#).

### 10.174.2 Constructor & Destructor Documentation

#### 10.174.2.1 JPEGCodec()

```
gdcm::JPEGCodec::JPEGCodec ( )
```

#### 10.174.2.2 ~JPEGCodec()

```
gdcm::JPEGCodec::~~JPEGCodec ( ) [override]
```

### 10.174.3 Member Function Documentation

#### 10.174.3.1 AppendFrameEncode()

```
bool gdcm::JPEGCodec::AppendFrameEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### 10.174.3.2 AppendRowEncode()

```
bool gdcm::JPEGCodec::AppendRowEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### 10.174.3.3 CanCode()

```
bool gdcm::JPEGCodec::CanCode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

#### Examples

[CompressLossyJPEG.cs](#).

#### 10.174.3.4 CanDecode()

```
bool gdcm::JPEGCodec::CanDecode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

#### 10.174.3.5 Clone()

```
ImageCodec * gdcm::JPEGCodec::Clone ( ) const [override], [virtual]
```

Implements [gdcm::ImageCodec](#).

### 10.174.3.6 Code()

```
bool gdcm::JPEGCodec::Code (
    DataElement const & in,
    DataElement & out ) [override], [virtual]
```

Compress into JPEG.

Reimplemented from [gdcm::Coder](#).

### 10.174.3.7 ComputeOffsetTable()

```
void gdcm::JPEGCodec::ComputeOffsetTable (
    bool b )
```

Compute the offset table:

### 10.174.3.8 Decode()

```
bool gdcm::JPEGCodec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcm::ImageCodec](#).

### 10.174.3.9 DecodeByStreams()

```
bool gdcm::JPEGCodec::DecodeByStreams (
    std::istream & is,
    std::ostream & os ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.174.3.10 DecodeExtent()

```
bool gdcm::JPEGCodec::DecodeExtent (
    char * buffer,
    unsigned int xmin,
    unsigned int xmax,
    unsigned int ymin,
    unsigned int ymax,
    unsigned int zmin,
    unsigned int zmax,
    std::istream & is ) [protected]
```

#### 10.174.3.11 EncodeBuffer()

```
virtual bool gdcm::JPEGCodec::EncodeBuffer (
    std::ostream & out,
    const char * inbuffer,
    size_t inlen ) [virtual]
```

Reimplemented in [gdcm::JPEG12Codec](#), [gdcm::JPEG16Codec](#), and [gdcm::JPEG8Codec](#).

#### 10.174.3.12 GetHeaderInfo()

```
bool gdcm::JPEGCodec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### Examples

[GetJPEGSamplePrecision.cxx](#).

#### 10.174.3.13 GetLossless()

```
bool gdcm::JPEGCodec::GetLossless ( ) const
```

#### 10.174.3.14 GetQuality()

```
double gdcm::JPEGCodec::GetQuality ( ) const
```

#### 10.174.3.15 IsFrameEncoder()

```
bool gdcm::JPEGCodec::IsFrameEncoder ( ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### 10.174.3.16 IsRowEncoder()

```
bool gdcm::JPEGCodec::IsRowEncoder ( ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.174.3.17 IsStateSuspension()

```
virtual bool gdcm::JPEGCodec::IsStateSuspension ( ) const [protected], [virtual]
```

Reimplemented in [gdcm::JPEG12Codec](#), [gdcm::JPEG16Codec](#), and [gdcm::JPEG8Codec](#).

### 10.174.3.18 IsValid()

```
bool gdcm::JPEGCodec::IsValid (
    PhotometricInterpretation const & pi ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.174.3.19 SetBitSample()

```
void gdcm::JPEGCodec::SetBitSample (
    int bit ) [protected]
```

### 10.174.3.20 SetLossless()

```
void gdcm::JPEGCodec::SetLossless (
    bool l )
```

#### Examples

[CompressLossyJPEG.cs](#), and [FileChangeTSLossy.cs](#).

### 10.174.3.21 SetPixelFormat()

```
void gdcm::JPEGCodec::SetPixelFormat (
    PixelFormat const & pf ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### Examples

[GetJPEGSamplePrecision.cxx](#).

### 10.174.3.22 SetQuality()

```
void gdcm::JPEGCodec::SetQuality (
    double q )
```

#### Examples

[CompressLossyJPEG.cs](#), and [FileChangeTSLossy.cs](#).

### 10.174.3.23 StartEncode()

```
bool gdcM::JPEGCodec::StartEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcM::ImageCodec](#).

### 10.174.3.24 StopEncode()

```
bool gdcM::JPEGCodec::StopEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcM::ImageCodec](#).

## 10.174.4 Friends And Related Symbol Documentation

### 10.174.4.1 ImageRegionReader

```
friend class ImageRegionReader [friend]
```

## 10.174.5 Member Data Documentation

### 10.174.5.1 BitSample

```
int gdcM::JPEGCodec::BitSample [protected]
```

### 10.174.5.2 Quality

```
int gdcM::JPEGCodec::Quality [protected]
```

The documentation for this class was generated from the following file:

- [gdcMJPEGCodec.h](#)



## 10.175 gdcm::JPEGLSCodec Class Reference

JPEG-LS.

```
#include <gdcmJPEGLSCodec.h>
```

Inheritance diagram for gdcm::JPEGLSCodec:



Collaboration diagram for gdcm::JPEGLSCodec:



### Public Member Functions

- [JPEGLSCodec](#) ()
- [~JPEGLSCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const override  
*Return whether this coder support this transfer syntax (can code it)*

- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const override  
*Return whether this decoder support this transfer syntax (can decode it)*
- [ImageCodec](#) \* [Clone](#) () const override
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out) override  
*Code.*
- bool [Decode](#) ([DataElement](#) const &in, char \*outBuffer, size\_t inBufferLength, uint32\_t inXMin, uint32\_t inXMax, uint32\_t inYMin, uint32\_t inYMax, uint32\_t inZMin, uint32\_t inZMax)  
*Decode.*
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override  
*Decode.*
- unsigned long [GetBufferLength](#) () const
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override
- bool [GetLossless](#) () const
- void [SetBufferLength](#) (unsigned long l)
- void [SetLossless](#) (bool l)
- void [SetLossyError](#) (int error)  
*[0-3] generally*

## Public Member Functions inherited from [gdcm::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CleanupUnusedBits](#) (char \*data, size\_t datalen)
- const unsigned int \* [GetDimensions](#) () const
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- virtual void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)

## Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default

## Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

## Protected Member Functions

- bool [AppendFrameEncode](#) (std::ostream &out, const char \*data, size\_t datalen) override
- bool [AppendRowEncode](#) (std::ostream &out, const char \*data, size\_t datalen) override
- bool [DecodeExtent](#) (char \*buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is)
- bool [IsFrameEncoder](#) () override
- bool [IsRowEncoder](#) () override
- bool [StartEncode](#) (std::ostream &) override
- bool [StopEncode](#) (std::ostream &) override

## Protected Member Functions inherited from [gdcm::ImageCodec](#)

- bool [DecodeByStreams](#) (std::istream &is\_, std::ostream &os) override
- bool [DoByteSwap](#) (std::istream &is\_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is\_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is\_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is\_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is\_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is\_, std::ostream &os)
- virtual bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)

## Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char \*bv, unsigned long len, std::ostream &os)

## Friends

- class [ImageRegionReader](#)

## Additional Inherited Members

## Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > LUTPtr

## Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) LUT
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) PF
- [PhotometricInterpretation](#) PI
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

### 10.175.1 Detailed Description

JPEG-LS.

Note

codec that implement the JPEG-LS compression this is an implementation of [ImageCodec](#) for JPEG-LS

It uses the CharLS JPEG-LS implementation <https://github.com/team-charls/charls>

### 10.175.2 Constructor & Destructor Documentation

#### 10.175.2.1 JPEGLSCodec()

```
gdcm::JPEGLSCodec::JPEGLSCodec ( )
```

#### 10.175.2.2 ~JPEGLSCodec()

```
gdcm::JPEGLSCodec::~~JPEGLSCodec ( ) [override]
```

### 10.175.3 Member Function Documentation

#### 10.175.3.1 AppendFrameEncode()

```
bool gdcm::JPEGLSCodec::AppendFrameEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.175.3.2 AppendRowEncode()

```
bool gdcm::JPEGLSCodec::AppendRowEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.175.3.3 CanCode()

```
bool gdcm::JPEGLSCodec::CanCode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.175.3.4 CanDecode()

```
bool gdcm::JPEGLSCodec::CanDecode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.175.3.5 Clone()

```
ImageCodec * gdcm::JPEGLSCodec::Clone ( ) const [override], [virtual]
```

Implements [gdcm::ImageCodec](#).

### 10.175.3.6 Code()

```
bool gdcm::JPEGLSCodec::Code (
    DataElement const & in_,
    DataElement & out_ ) [override], [virtual]
```

Code.

Reimplemented from [gdcm::Coder](#).

### 10.175.3.7 Decode() [1/2]

```
bool gdcm::JPEGLSCodec::Decode (
    DataElement const & in,
    char * outBuffer,
    size_t inBufferLength,
    uint32_t inXMin,
    uint32_t inXMax,
    uint32_t inYMin,
    uint32_t inYMax,
    uint32_t inZMin,
    uint32_t inZMax )
```

### 10.175.3.8 Decode() [2/2]

```
bool gdcm::JPEGLSCodec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcm::ImageCodec](#).

### 10.175.3.9 DecodeExtent()

```
bool gdcm::JPEGLSCodec::DecodeExtent (
    char * buffer,
    unsigned int xmin,
    unsigned int xmax,
    unsigned int ymin,
    unsigned int ymax,
    unsigned int zmin,
    unsigned int zmax,
    std::istream & is ) [protected]
```

### 10.175.3.10 GetBufferLength()

```
unsigned long gdcm::JPEGLSCodec::GetBufferLength ( ) const [inline]
```

### 10.175.3.11 GetHeaderInfo()

```
bool gdcm::JPEGLSCodec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.175.3.12 GetLossless()

```
bool gdcm::JPEGLSCodec::GetLossless ( ) const
```

### 10.175.3.13 IsFrameEncoder()

```
bool gdcm::JPEGLSCodec::IsFrameEncoder ( ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.175.3.14 IsRowEncoder()

```
bool gdcm::JPEGLSCodec::IsRowEncoder ( ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.175.3.15 SetBufferLength()

```
void gdcm::JPEGLSCodec::SetBufferLength (
    unsigned long l ) [inline]
```

### 10.175.3.16 SetLossless()

```
void gdcm::JPEGLSCodec::SetLossless (
    bool l )
```

### 10.175.3.17 SetLossyError()

```
void gdcm::JPEGLSCodec::SetLossyError (
    int error )
```

[0-3] generally

### 10.175.3.18 StartEncode()

```
bool gdcm::JPEGLSCodec::StartEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.175.3.19 StopEncode()

```
bool gdcmm::JPEGLSCodec::StopEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcmm::ImageCodec](#).

## 10.175.4 Friends And Related Symbol Documentation

### 10.175.4.1 ImageRegionReader

```
friend class ImageRegionReader [friend]
```

The documentation for this class was generated from the following file:

- [gdcmmJPEGLSCodec.h](#)

## 10.176 gdcmm::JSON Class Reference

```
#include <gdcmmJSON.h>
```

### Public Member Functions

- [JSON](#) ()
- [~JSON](#) ()
- bool [Code](#) ([DataSet](#) const &in, std::ostream &os)
- bool [Decode](#) (std::istream &is, [DataSet](#) &out)
- bool [GetPrettyPrint](#) () const
- void [PrettyPrintOff](#) ()
- void [PrettyPrintOn](#) ()
- void [SetPrettyPrint](#) (bool onoff)

### 10.176.1 Detailed Description

#### Examples

[QIDO-RS.cxx](#).

### 10.176.2 Constructor & Destructor Documentation

#### 10.176.2.1 JSON()

```
gdcmm::JSON::JSON ( )
```



### 10.176.2.2 ~JSON()

```
gdcm::JSON::~~JSON ( )
```

## 10.176.3 Member Function Documentation

### 10.176.3.1 Code()

```
bool gdcm::JSON::Code (
    DataSet const & in,
    std::ostream & os )
```

#### Examples

[QIDO-RS.cxx](#).

### 10.176.3.2 Decode()

```
bool gdcm::JSON::Decode (
    std::istream & is,
    DataSet & out )
```

#### Examples

[QIDO-RS.cxx](#).

### 10.176.3.3 GetPrettyPrint()

```
bool gdcm::JSON::GetPrettyPrint ( ) const
```

### 10.176.3.4 PrettyPrintOff()

```
void gdcm::JSON::PrettyPrintOff ( )
```

### 10.176.3.5 PrettyPrintOn()

```
void gdcm::JSON::PrettyPrintOn ( )
```

#### Examples

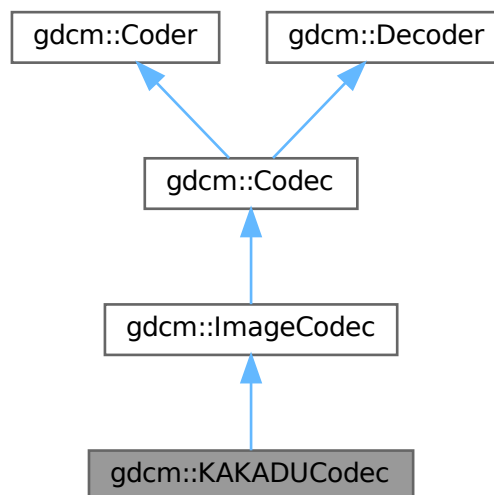
[QIDO-RS.cxx](#).

```
void gdcmm::JSON::SetPrettyPrint (
    bool onoff )
```

- [gdcmJSON.h](#)

## KAKADUCodec.

Inheritance diagram for gdcm::KAKADUCodec:



```

classDiagram
    class gdcmm_KAKADUCoder["gdcmm::KAKADUCoder"]
    class gdcmm_ImageCoder["gdcmm::ImageCoder"]
    class gdcmm_Codec["gdcmm::Codec"]
    class gdcmm_Coder["gdcmm::Coder"]
    class gdcmm_Decoder["gdcmm::Decoder"]
    class gdcmm_PhotometricInterpretation["gdcmm::PhotometricInterpretation"]
    class gdcmm_PixelFormat["gdcmm::PixelFormat"]
    class gdcmm_SmartPointer_LookupTable["gdcmm::SmartPointer < LookupTable >"]
    class gdcmm_SmartPointer_ObjectType["gdcmm::SmartPointer < Object type >"]

    gdcmm_KAKADUCoder <|-- gdcmm_ImageCoder
    gdcmm_ImageCoder <|-- gdcmm_Codec
    gdcmm_ImageCoder <|-- gdcmm_PixelFormat
    gdcmm_ImageCoder <|-- gdcmm_SmartPointer_LookupTable
    gdcmm_Codec <|-- gdcmm_Coder
    gdcmm_Codec <|-- gdcmm_Decoder
    gdcmm_ImageCoder --> gdcmm_PhotometricInterpretation : PI
    gdcmm_ImageCoder --> gdcmm_PixelFormat : PF
    gdcmm_ImageCoder --> gdcmm_SmartPointer_LookupTable : LUT
    gdcmm_SmartPointer_LookupTable --> gdcmm_SmartPointer_ObjectType : < LookupTable >
  
```

**Public Member Functions**

- [KAKADUCodec](#) ()
- [~KAKADUCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const override  
*Return whether this coder support this transfer syntax (can code it)*
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const override  
*Return whether this decoder support this transfer syntax (can decode it)*
- [ImageCodec](#) \* [Clone](#) () const override
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out) override  
*Code.*
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override  
*Decode.*

**Public Member Functions inherited from [gdcm::ImageCodec](#)**

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CleanupUnusedBits](#) (char \*data, size\_t datalen)
- const unsigned int \* [GetDimensions](#) () const
- virtual bool [GetHeaderInfo](#) (std::istream &is\_, [TransferSyntax](#) &ts)
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- virtual void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)

**Public Member Functions inherited from [gdcm::Coder](#)**

- virtual [~Coder](#) ()=default

**Public Member Functions inherited from [gdcm::Decoder](#)**

- virtual [~Decoder](#) ()=default

## Additional Inherited Members

### Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

### Protected Member Functions inherited from [gdcm::ImageCodec](#)

- virtual bool [AppendFrameEncode](#) (std::ostream &out, const char \*data, size\_t datalen)
- virtual bool [AppendRowEncode](#) (std::ostream &out, const char \*data, size\_t datalen)
- bool [DecodeByStreams](#) (std::istream &is\_, std::ostream &os) override
- bool [DoByteSwap](#) (std::istream &is\_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is\_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is\_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is\_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is\_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is\_, std::ostream &os)
- virtual bool [IsFrameEncoder](#) ()
- virtual bool [IsRowEncoder](#) ()
- virtual bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)
- virtual bool [StartEncode](#) (std::ostream &os)
- virtual bool [StopEncode](#) (std::ostream &os)

### Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char \*bv, unsigned long len, std::ostream &os)

### Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) [LUT](#)
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) [PF](#)
- [PhotometricInterpretation](#) [PI](#)
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

## 10.177.1 Detailed Description

[KAKADUCodec](#).

## 10.177.2 Constructor & Destructor Documentation

### 10.177.2.1 KAKADUCodec()

```
gdcm::KAKADUCodec::KAKADUCodec ( )
```

### 10.177.2.2 ~KAKADUCodec()

```
gdcm::KAKADUCodec::~~KAKADUCodec ( ) [override]
```

## 10.177.3 Member Function Documentation

### 10.177.3.1 CanCode()

```
bool gdcm::KAKADUCodec::CanCode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.177.3.2 CanDecode()

```
bool gdcm::KAKADUCodec::CanDecode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.177.3.3 Clone()

```
ImageCodec * gdcm::KAKADUCodec::Clone ( ) const [override], [virtual]
```

Implements [gdcm::ImageCodec](#).

### 10.177.3.4 Code()

```
bool gdcm::KAKADUCodec::Code (
    DataElement const & in_,
    DataElement & out_ ) [override], [virtual]
```

Code.

Reimplemented from [gdcm::Coder](#).

### 10.177.3.5 Decode()

```
bool gdcM::KAKADUCodec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcM::ImageCodec](#).

The documentation for this class was generated from the following file:

- [gdcMKAKADUCodec.h](#)

## 10.178 gdcM::LO Class Reference

[LO](#).

```
#include <gdcMLO.h>
```

Inheritance diagram for gdcM::LO:



Collaboration diagram for gdcM::LO:



## Public Types

- typedef [Superclass::const\\_iterator](#) [const\\_iterator](#)
- typedef [Superclass::const\\_reference](#) [const\\_reference](#)
- typedef [Superclass::const\\_reverse\\_iterator](#) [const\\_reverse\\_iterator](#)
- typedef [Superclass::difference\\_type](#) [difference\\_type](#)
- typedef [Superclass::iterator](#) [iterator](#)
- typedef [Superclass::pointer](#) [pointer](#)
- typedef [Superclass::reference](#) [reference](#)
- typedef [Superclass::reverse\\_iterator](#) [reverse\\_iterator](#)
- typedef [Superclass::size\\_type](#) [size\\_type](#)
- typedef [String<'\', 64 > Superclass](#)
- typedef [Superclass::value\\_type](#) [value\\_type](#)

## Public Member Functions

- [LO](#) ()
- [LO](#) (const [Superclass](#) &s, [size\\_type](#) pos=0, [size\\_type](#) n=npos)
- [LO](#) (const [value\\_type](#) \*s)
- [LO](#) (const [value\\_type](#) \*s, [size\\_type](#) n)
- bool [IsValid](#) () const

## 10.178.1 Detailed Description

[LO](#).

Note

TODO

## 10.178.2 Member Typedef Documentation

### 10.178.2.1 [const\\_iterator](#)

```
typedef Superclass::const\_iterator gdcm::LO::const_iterator
```

### 10.178.2.2 [const\\_reference](#)

```
typedef Superclass::const\_reference gdcm::LO::const_reference
```

### 10.178.2.3 [const\\_reverse\\_iterator](#)

```
typedef Superclass::const\_reverse\_iterator gdcm::LO::const_reverse_iterator
```

#### 10.178.2.4 difference\_type

```
typedef Superclass::difference_type gdc::LO::difference_type
```

#### 10.178.2.5 iterator

```
typedef Superclass::iterator gdc::LO::iterator
```

#### 10.178.2.6 pointer

```
typedef Superclass::pointer gdc::LO::pointer
```

#### 10.178.2.7 reference

```
typedef Superclass::reference gdc::LO::reference
```

#### 10.178.2.8 reverse\_iterator

```
typedef Superclass::reverse_iterator gdc::LO::reverse_iterator
```

#### 10.178.2.9 size\_type

```
typedef Superclass::size_type gdc::LO::size_type
```

#### 10.178.2.10 Superclass

```
typedef String<'\\', 64> gdc::LO::Superclass
```

#### 10.178.2.11 value\_type

```
typedef Superclass::value_type gdc::LO::value_type
```

### 10.178.3 Constructor & Destructor Documentation

#### 10.178.3.1 LO() [1/4]

```
gdc::LO::LO ( ) [inline]
```



### 10.178.3.2 LO() [2/4]

```
gdcm::LO::LO (
    const value_type * s ) [inline]
```

### 10.178.3.3 LO() [3/4]

```
gdcm::LO::LO (
    const value_type * s,
    size_type n ) [inline]
```

### 10.178.3.4 LO() [4/4]

```
gdcm::LO::LO (
    const Superclass & s,
    size_type pos = 0,
    size_type n = npos ) [inline]
```

## 10.178.4 Member Function Documentation

### 10.178.4.1 IsValid()

```
bool gdcm::LO::IsValid ( ) const [inline]
```

References [gdcm::String< TDelimiter, TMaxLength, TPadChar >::IsValid\(\)](#).

The documentation for this class was generated from the following file:

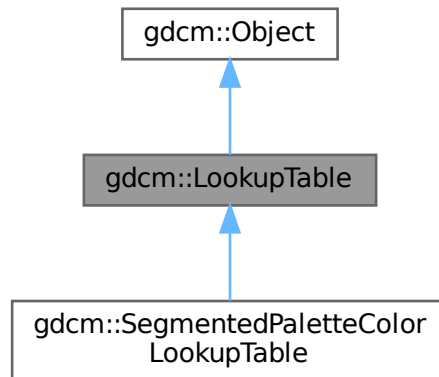
- [gdcmLO.h](#)

## 10.179 gdcm::LookupTable Class Reference

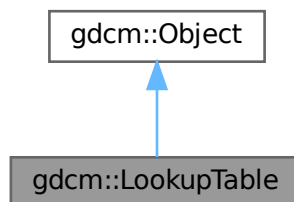
[LookupTable](#) class.

```
#include <gdcmLookupTable.h>
```

Inheritance diagram for `gdcm::LookupTable`:



Collaboration diagram for `gdcm::LookupTable`:



### Public Types

- enum `LookupTableType` {  
    `RED` = 0 ,  
    `GREEN` ,  
    `BLUE` ,  
    `GRAY` ,  
    `UNKNOWN` }

### Public Member Functions

- `LookupTable` ()

- [LookupTable](#) ([LookupTable](#) const &lut)
- [~LookupTable](#) () override
- void [Allocate](#) (unsigned short bitsample=8)  
*Allocate the LUT.*
- void [Clear](#) ()  
*Clear the LUT.*
- bool [Decode](#) (char \*outputbuffer, size\_t outlen, const char \*inputbuffer, size\_t inlen) const
- void [Decode](#) (std::istream &is, std::ostream &os) const  
*Decode the LUT.*
- bool [Decode8](#) (char \*outputbuffer, size\_t outlen, const char \*inputbuffer, size\_t inlen) const  
*Decode into RGB 8 bits space.*
- unsigned short [GetBitSample](#) () const  
*return the bit sample*
- bool [GetBufferAsRGBA](#) (unsigned char \*rgba) const  
*return the LUT as RGBA buffer*
- void [GetLUT](#) ([LookupTableType](#) type, unsigned char \*array, unsigned int &length) const
- void [GetLUTDescriptor](#) ([LookupTableType](#) type, unsigned short &length, unsigned short &subscript, unsigned short &bitsize) const
- unsigned int [GetLUTLength](#) ([LookupTableType](#) type) const
- const unsigned char \* [GetPointer](#) () const  
*return a raw pointer to the LUT*
- void [InitializeBlueLUT](#) (unsigned short length, unsigned short subscript, unsigned short bitsize)
- bool [Initialized](#) () const  
*return whether the LUT has been initialized*
- void [InitializeGreenLUT](#) (unsigned short length, unsigned short subscript, unsigned short bitsize)
- void [InitializeLUT](#) ([LookupTableType](#) type, unsigned short length, unsigned short subscript, unsigned short bitsize)  
*Generic interface:*
- void [InitializeRedLUT](#) (unsigned short length, unsigned short subscript, unsigned short bitsize)  
*RED / GREEN / BLUE specific:*
- bool [IsRGB8](#) () const  
*Return whether 16 bits LUT is in RGB 8 bits space.*
- void [Print](#) (std::ostream &) const override
- void [SetBlueLUT](#) (const unsigned char \*blue, unsigned int length)
- void [SetGreenLUT](#) (const unsigned char \*green, unsigned int length)
- virtual void [SetLUT](#) ([LookupTableType](#) type, const unsigned char \*array, unsigned int length)
- void [SetRedLUT](#) (const unsigned char \*red, unsigned int length)
- bool [WriteBufferAsRGBA](#) (const unsigned char \*rgba)  
*Write the LUT as RGBA.*

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

**Protected Attributes**

- unsigned short [BitSample](#)
- bool [IncompleteLUT](#):1
- LookupTableInternal \* [Internal](#)

**Additional Inherited Members****Protected Member Functions inherited from [gdcm::Object](#)**

- void [Register](#) ()
- void [UnRegister](#) ()

**10.179.1 Detailed Description**

[LookupTable](#) class.

**Examples**

[ExtractImageRegionWithLUT.cs](#), and [PrintLUT.cxx](#).

**10.179.2 Member Enumeration Documentation****10.179.2.1 LookupTableType**

```
enum gdcm::LookupTable::LookupTableType
```

**Enumerator**

RED	
GREEN	
BLUE	
GRAY	
UNKNOWN	

**10.179.3 Constructor & Destructor Documentation****10.179.3.1 LookupTable() [1/2]**

```
gdcm::LookupTable::LookupTable ( )
```

### 10.179.3.2 ~LookupTable()

```
gdcm::LookupTable::~~LookupTable ( ) [override]
```

### 10.179.3.3 LookupTable() [2/2]

```
gdcm::LookupTable::LookupTable (
    LookupTable const & lut ) [inline]
```

## 10.179.4 Member Function Documentation

### 10.179.4.1 Allocate()

```
void gdcm::LookupTable::Allocate (
    unsigned short bitsample = 8 )
```

Allocate the LUT.

### 10.179.4.2 Clear()

```
void gdcm::LookupTable::Clear ( )
```

Clear the LUT.

### 10.179.4.3 Decode() [1/2]

```
bool gdcm::LookupTable::Decode (
    char * outputbuffer,
    size_t outlen,
    const char * inputbuffer,
    size_t inlen ) const
```

Decode the LUT outputbuffer will contains the RGB decoded PALETTE COLOR input image of size inlen the outputbuffer should be at least 3 times the size of inlen

### 10.179.4.4 Decode() [2/2]

```
void gdcm::LookupTable::Decode (
    std::istream & is,
    std::ostream & os ) const
```

Decode the LUT.

### Examples

[ExtractImageRegionWithLUT.cs](#).

#### 10.179.4.5 Decode8()

```
bool gdc::LookupTable::Decode8 (
    char * outputbuffer,
    size_t outlen,
    const char * inputbuffer,
    size_t inlen ) const
```

Decode into RGB 8 bits space.

#### 10.179.4.6 GetBitSample()

```
unsigned short gdc::LookupTable::GetBitSample ( ) const [inline]
```

return the bit sample

#### 10.179.4.7 GetBufferAsRGBA()

```
bool gdc::LookupTable::GetBufferAsRGBA (
    unsigned char * rgba ) const
```

return the LUT as RGBA buffer

#### 10.179.4.8 GetLUT()

```
void gdc::LookupTable::GetLUT (
    LookupTableType type,
    unsigned char * array,
    unsigned int & length ) const
```

#### 10.179.4.9 GetLUTDescriptor()

```
void gdc::LookupTable::GetLUTDescriptor (
    LookupTableType type,
    unsigned short & length,
    unsigned short & subscript,
    unsigned short & bitsize ) const
```

#### 10.179.4.10 GetLUTLength()

```
unsigned int gdc::LookupTable::GetLUTLength (
    LookupTableType type ) const
```

#### 10.179.4.11 GetPointer()

```
const unsigned char * gdcm::LookupTable::GetPointer ( ) const
```

return a raw pointer to the LUT

#### 10.179.4.12 InitializeBlueLUT()

```
void gdcm::LookupTable::InitializeBlueLUT (
    unsigned short length,
    unsigned short subscript,
    unsigned short bitsize )
```

#### 10.179.4.13 Initialized()

```
bool gdcm::LookupTable::Initialized ( ) const
```

return whether the LUT has been initialized

#### 10.179.4.14 InitializeGreenLUT()

```
void gdcm::LookupTable::InitializeGreenLUT (
    unsigned short length,
    unsigned short subscript,
    unsigned short bitsize )
```

#### 10.179.4.15 InitializeLUT()

```
void gdcm::LookupTable::InitializeLUT (
    LookupTableType type,
    unsigned short length,
    unsigned short subscript,
    unsigned short bitsize )
```

Generic interface:

#### 10.179.4.16 InitializeRedLUT()

```
void gdcm::LookupTable::InitializeRedLUT (
    unsigned short length,
    unsigned short subscript,
    unsigned short bitsize )
```

RED / GREEN / BLUE specific:

#### 10.179.4.17 IsRGB8()

```
bool gdcm::LookupTable::IsRGB8 ( ) const
```

Return whether 16 bits LUT is in RGB 8 bits space.

#### 10.179.4.18 Print()

```
void gdcm::LookupTable::Print (
    std::ostream & ) const [override], [virtual]
```

Reimplemented from [gdcm::Object](#).

Reimplemented in [gdcm::SegmentedPaletteColorLookupTable](#).

#### Examples

[PrintLUT.cxx](#).

#### 10.179.4.19 SetBlueLUT()

```
void gdcm::LookupTable::SetBlueLUT (
    const unsigned char * blue,
    unsigned int length )
```

#### 10.179.4.20 SetGreenLUT()

```
void gdcm::LookupTable::SetGreenLUT (
    const unsigned char * green,
    unsigned int length )
```

#### 10.179.4.21 SetLUT()

```
virtual void gdcm::LookupTable::SetLUT (
    LookupTableType type,
    const unsigned char * array,
    unsigned int length ) [virtual]
```

Reimplemented in [gdcm::SegmentedPaletteColorLookupTable](#).

#### 10.179.4.22 SetRedLUT()

```
void gdcm::LookupTable::SetRedLUT (
    const unsigned char * red,
    unsigned int length )
```



### 10.179.4.23 WriteBufferAsRGBA()

```
bool gdcm::LookupTable::WriteBufferAsRGBA (
    const unsigned char * rgba )
```

Write the LUT as RGBA.

## 10.179.5 Member Data Documentation

### 10.179.5.1 BitSample

```
unsigned short gdcm::LookupTable::BitSample [protected]
```

### 10.179.5.2 IncompleteLUT

```
bool gdcm::LookupTable::IncompleteLUT [protected]
```

### 10.179.5.3 Internal

```
LookupTableInternal* gdcm::LookupTable::Internal [protected]
```

The documentation for this class was generated from the following file:

- [gdcmLookupTable.h](#)

## 10.180 gdcm::Scanner2::ltstr Struct Reference

```
#include <gdcmScanner2.h>
```

### Public Member Functions

- bool [operator\(\)](#) (const char \*s1, const char \*s2) const

## 10.180.1 Member Function Documentation

### 10.180.1.1 operator()()

```
bool gdcm::Scanner2::ltstr::operator() (
    const char * s1,
    const char * s2 ) const [inline]
```

The documentation for this struct was generated from the following file:

- [gdcmScanner2.h](#)

## 10.181 gdcm::Scanner::ltstr Struct Reference

```
#include <gdcmScanner.h>
```

### Public Member Functions

- bool [operator\(\)](#) (const char \*s1, const char \*s2) const

### 10.181.1 Member Function Documentation

#### 10.181.1.1 operator>()

```
bool gdcm::Scanner::ltstr::operator() (
    const char * s1,
    const char * s2 ) const [inline]
```

The documentation for this struct was generated from the following file:

- [gdcmScanner.h](#)

## 10.182 gdcm::StrictScanner2::ltstr Struct Reference

```
#include <gdcmStrictScanner2.h>
```

### Public Member Functions

- bool [operator\(\)](#) (const char \*s1, const char \*s2) const

### 10.182.1 Member Function Documentation

#### 10.182.1.1 operator>()

```
bool gdcm::StrictScanner2::ltstr::operator() (
    const char * s1,
    const char * s2 ) const [inline]
```

The documentation for this struct was generated from the following file:

- [gdcmStrictScanner2.h](#)

## 10.183 gdcm::StrictScanner::ltstr Struct Reference

```
#include <gdcmStrictScanner.h>
```

### Public Member Functions

- bool [operator\(\)](#) (const char \*s1, const char \*s2) const

### 10.183.1 Member Function Documentation

#### 10.183.1.1 operator>()

```
bool gdcm::StrictScanner::ltstr::operator() (
    const char * s1,
    const char * s2 ) const [inline]
```

The documentation for this struct was generated from the following file:

- [gdcmStrictScanner.h](#)

## 10.184 gdcm::Macro Class Reference

Class for representing a [Macro](#).

```
#include <gdcmMacro.h>
```

### Public Types

- typedef std::vector< std::string > [ArrayIncludeMacrosType](#)
- typedef std::map< [Tag](#), [MacroEntry](#) > [MapModuleEntry](#)

### Public Member Functions

- [Macro](#) ()=default
- void [AddMacroEntry](#) (const [Tag](#) &tag, const [MacroEntry](#) &module)  
*Will add a [ModuleEntry](#) directly at root-level. See [Macro](#) for nested-included level.*
- void [Clear](#) ()
- bool [FindMacroEntry](#) (const [Tag](#) &tag) const
- const [MacroEntry](#) & [GetMacroEntry](#) (const [Tag](#) &tag) const
- const char \* [GetName](#) () const
- void [SetName](#) (const char \*name)
- bool [Verify](#) (const [DataSet](#) &ds, [Usage](#) const &usage) const

## Friends

- `std::ostream & operator<< (std::ostream &_os, const Macro &_val)`

## 10.184.1 Detailed Description

Class for representing a [Macro](#).

### Note

[Attribute Macro](#): a set of Attributes that are described in a single table that is referenced by multiple [Module](#) or other tables.

### See also

[Module](#)

## 10.184.2 Member Typedef Documentation

### 10.184.2.1 `ArrayIncludeMacroType`

```
typedef std::vector<std::string> gdcmmacro::Macro::ArrayIncludeMacroType
```

### 10.184.2.2 `MapModuleEntry`

```
typedef std::map<Tag, MacroEntry> gdcmmacro::Macro::MapModuleEntry
```

## 10.184.3 Constructor & Destructor Documentation

### 10.184.3.1 `Macro()`

```
gdcmmacro::Macro::Macro ( ) [default]
```

References [gdcmmacro::operator<<\(\)](#).

## 10.184.4 Member Function Documentation

### 10.184.4.1 `AddMacroEntry()`

```
void gdcmmacro::Macro::AddMacroEntry (
    const Tag & tag,
    const MacroEntry & module ) [inline]
```

Will add a [ModuleEntry](#) directly at root-level. See [Macro](#) for nested-included level.

#### 10.184.4.2 Clear()

```
void gdcmmacro::Macro::Clear ( ) [inline]
```

#### 10.184.4.3 FindMacroEntry()

```
bool gdcmmacro::Macro::FindMacroEntry (
    const Tag & tag ) const
```

Find or Get a [ModuleEntry](#). [ModuleEntry](#) are either search are root-level or within nested-macro included in module.

#### 10.184.4.4 GetMacroEntry()

```
const MacroEntry & gdcmmacro::Macro::GetMacroEntry (
    const Tag & tag ) const
```

#### 10.184.4.5 GetName()

```
const char * gdcmmacro::Macro::GetName ( ) const [inline]
```

#### 10.184.4.6 SetName()

```
void gdcmmacro::Macro::SetName (
    const char * name ) [inline]
```

#### 10.184.4.7 Verify()

```
bool gdcmmacro::Macro::Verify (
    const DataSet & ds,
    Usage const & usage ) const
```

### 10.184.5 Friends And Related Symbol Documentation

#### 10.184.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const Macro & _val ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmmacro.h](#)

## 10.185 gdcm::Macros Class Reference

Class for representing a [Modules](#).

```
#include <gdcmMacros.h>
```

### Public Types

- typedef std::map< std::string, [Macro](#) > [ModuleMapType](#)

### Public Member Functions

- [Macros](#) ()=default
- void [AddMacro](#) (const char \*ref, const [Macro](#) &module)
- void [Clear](#) ()
- const [Macro](#) & [GetMacro](#) (const char \*name) const
- bool [IsEmpty](#) () const

### Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [Macros](#) &\_val)

### 10.185.1 Detailed Description

Class for representing a [Modules](#).

#### Note

bla

#### See also

[Module](#)

#### Examples

[TraverseModules.cxx](#).

### 10.185.2 Member Typedef Documentation

#### 10.185.2.1 ModuleMapType

```
typedef std::map<std::string, Macro> gdcm::Macros::ModuleMapType
```

## 10.185.3 Constructor & Destructor Documentation

### 10.185.3.1 Macros()

```
gdcm::Macros::Macros ( ) [default]
```

References [gdcm::operator<<\(\)](#).

## 10.185.4 Member Function Documentation

### 10.185.4.1 AddMacro()

```
void gdcm::Macros::AddMacro (
    const char * ref,
    const Macro & module ) [inline]
```

### 10.185.4.2 Clear()

```
void gdcm::Macros::Clear ( ) [inline]
```

### 10.185.4.3 GetMacro()

```
const Macro & gdcm::Macros::GetMacro (
    const char * name ) const [inline]
```

### 10.185.4.4 IsEmpty()

```
bool gdcm::Macros::IsEmpty ( ) const [inline]
```

## 10.185.5 Friends And Related Symbol Documentation

### 10.185.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const Macros & _val ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmMacros.h](#)

## 10.186 gdcm::network::MaximumLengthSub Class Reference

[MaximumLengthSub.](#)

```
#include <gdcmMaximumLengthSub.h>
```

### Public Member Functions

- [MaximumLengthSub](#) ()
- uint32\_t [GetMaximumLength](#) () const
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [SetMaximumLength](#) (uint32\_t maximumlength)
- size\_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

### 10.186.1 Detailed Description

[MaximumLengthSub.](#)

Annex D [Table](#) D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

or

[Table](#) D.1-2 Maximum length sub-item fields (A-ASSOCIATE-AC)

### 10.186.2 Constructor & Destructor Documentation

#### 10.186.2.1 MaximumLengthSub()

```
gdcm::network::MaximumLengthSub::MaximumLengthSub ( )
```

### 10.186.3 Member Function Documentation

#### 10.186.3.1 GetMaximumLength()

```
uint32_t gdcm::network::MaximumLengthSub::GetMaximumLength ( ) const [inline]
```

#### 10.186.3.2 Print()

```
void gdcm::network::MaximumLengthSub::Print (
    std::ostream & os ) const
```



### 10.186.3.3 Read()

```
std::istream & gdcm::network::MaximumLengthSub::Read (
    std::istream & is )
```

### 10.186.3.4 SetMaximumLength()

```
void gdcm::network::MaximumLengthSub::SetMaximumLength (
    uint32_t maximumlength )
```

### 10.186.3.5 Size()

```
size_t gdcm::network::MaximumLengthSub::Size ( ) const
```

### 10.186.3.6 Write()

```
const std::ostream & gdcm::network::MaximumLengthSub::Write (
    std::ostream & os ) const
```

The documentation for this class was generated from the following file:

- [gdcmMaximumLengthSub.h](#)

## 10.187 gdcm::MD5 Class Reference

Class for [MD5](#).

```
#include <gdcmMD5.h>
```

### Static Public Member Functions

- static bool [Compute](#) (const char \*buffer, size\_t buf\_len, char digest\_str[33])
- static bool [ComputeFile](#) (const char \*filename, char digest\_str[33])  
*Compute md5 from a file filename*

### 10.187.1 Detailed Description

Class for [MD5](#).

#### Warning

this class is able to pick from two implementations:

1. a lightweight md5 implementation (when GDCM\_BUILD\_TESTING is turned ON)
2. the one from OpenSSL (when GDCM\_USE\_SYSTEM\_OPENSSL is turned ON)

In all other cases it will return an error

## 10.187.2 Member Function Documentation

### 10.187.2.1 Compute()

```
static bool gdcM::MD5::Compute (
    const char * buffer,
    size_t buf_len,
    char digest_str[33] ) [static]
```

### 10.187.2.2 ComputeFile()

```
static bool gdcM::MD5::ComputeFile (
    const char * filename,
    char digest_str[33] ) [static]
```

Compute md5 from a file *filename*

The documentation for this class was generated from the following file:

- [gdcMMD5.h](#)

## 10.188 gdcM::MEC\_MR3 Class Reference

Class for [MEC\\_MR3](#).

```
#include <gdcMMEC_MR3.h>
```

### Static Public Member Functions

- static const [PrivateTag](#) & [GetCanonMECMR3Tag](#) ()
- static const [PrivateTag](#) & [GetPMTFInformationDataTag](#) ()
- static const [PrivateTag](#) & [GetToshibaMECMR3Tag](#) ()
- static bool [Print](#) (const char \*src, size\_t srclen)

### 10.188.1 Detailed Description

Class for [MEC\\_MR3](#).

## 10.188.2 Member Function Documentation

### 10.188.2.1 GetCanonMECMR3Tag()

```
static const PrivateTag & gdcM::MEC_MR3::GetCanonMECMR3Tag ( ) [static]
```

Return the private tag used by CANON to store the [MEC\\_MR3](#) data This is: [PrivateTag](#)(0x0029,0x90,"CANON\_MEC←\_MR3");

### 10.188.2.2 GetPMTFInformationDataTag()

```
static const PrivateTag & gdcm::MEC_MR3::GetPMTFInformationDataTag ( ) [static]
```

Return the private tag used by PMTF to store the [MEC\\_MR3](#) data This is: [PrivateTag](#)(0x0029,0x90,"PMTF INFORMATION DATA");

### 10.188.2.3 GetToshibaMECMR3Tag()

```
static const PrivateTag & gdcm::MEC_MR3::GetToshibaMECMR3Tag ( ) [static]
```

Return the private tag used by TOSHIBA to store the [MEC\\_MR3](#) data This is: [PrivateTag](#)(0x0029,0x90,"TOSHIBA\_MEC\_MR3");

### 10.188.2.4 Print()

```
static bool gdcm::MEC_MR3::Print (
    const char * src,
    size_t srclen ) [static]
```

The documentation for this class was generated from the following file:

- [gdcmMEC\\_MR3.h](#)

## 10.189 gdcm::MediaStorage Class Reference

[MediaStorage](#).

```
#include <gdcmMediaStorage.h>
```

### Public Types

- enum [MSType](#) {
 [MediaStorageDirectoryStorage](#) = 0 ,
 [ComputedRadiographyImageStorage](#) ,
 [DigitalXRayImageStorageForPresentation](#) ,
 [DigitalXRayImageStorageForProcessing](#) ,
 [DigitalMammographyImageStorageForPresentation](#) ,
 [DigitalMammographyImageStorageForProcessing](#) ,
 [DigitalIntraoralXrayImageStorageForPresentation](#) ,
 [DigitalIntraoralXrayImageStorageForProcessing](#) ,
 [CTImageStorage](#) ,
 [EnhancedCTImageStorage](#) ,
 [UltrasoundImageStorageRetired](#) ,
 [UltrasoundImageStorage](#) ,
 [UltrasoundMultiFrameImageStorageRetired](#) ,

[UltrasoundMultiFrameImageStorage](#) ,  
[MRIImageStorage](#) ,  
[EnhancedMRIImageStorage](#) ,  
[MRSpectroscopyStorage](#) ,  
[NuclearMedicineImageStorageRetired](#) ,  
[SecondaryCaptureImageStorage](#) ,  
[MultiframeSingleBitSecondaryCaptureImageStorage](#) ,  
[MultiframeGrayscaleByteSecondaryCaptureImageStorage](#) ,  
[MultiframeGrayscaleWordSecondaryCaptureImageStorage](#) ,  
[MultiframeTrueColorSecondaryCaptureImageStorage](#) ,  
[StandaloneOverlayStorage](#) ,  
[StandaloneCurveStorage](#) ,  
[LeadECGWaveformStorage](#) ,  
[GeneralECGWaveformStorage](#) ,  
[AmbulatoryECGWaveformStorage](#) ,  
[HemodynamicWaveformStorage](#) ,  
[CardiacElectrophysiologyWaveformStorage](#) ,  
[BasicVoiceAudioWaveformStorage](#) ,  
[StandaloneModalityLUTStorage](#) ,  
[StandaloneVOILUTStorage](#) ,  
[GrayscaleSoftcopyPresentationStateStorageSOPClass](#) ,  
[XRayAngiographicImageStorage](#) ,  
[XRayRadiofluoroscopicImageStorage](#) ,  
[XRayAngiographicBiPlaneImageStorageRetired](#) ,  
[NuclearMedicineImageStorage](#) ,  
[RawDataStorage](#) ,  
[SpatialRegistrationStorage](#) ,  
[SpatialFiducialsStorage](#) ,  
[PETImageStorage](#) ,  
[RTImageStorage](#) ,  
[RTDoseStorage](#) ,  
[RTStructureSetStorage](#) ,  
[RTPlanStorage](#) ,  
[CSANonImageStorage](#) ,  
[Philips3D](#) ,  
[EnhancedSR](#) ,  
[BasicTextSR](#) ,  
[HardcopyGrayscaleImageStorage](#) ,  
[ComprehensiveSR](#) ,  
[DetachedStudyManagementSOPClass](#) ,  
[EncapsulatedPDFStorage](#) ,  
[EncapsulatedCDASStorage](#) ,  
[StudyComponentManagementSOPClass](#) ,  
[DetachedVisitManagementSOPClass](#) ,  
[DetachedPatientManagementSOPClass](#) ,  
[VideoEndoscopicImageStorage](#) ,  
[GeneralElectricMagneticResonanceImageStorage](#) ,  
[GEPrivate3DModelStorage](#) ,  
[ToshibaPrivateDataStorage](#) ,  
[MammographyCADSR](#) ,  
[KeyObjectSelectionDocument](#) ,  
[HangingProtocolStorage](#) ,  
[ModalityPerformedProcedureStepSOPClass](#) ,  
[PhilipsPrivateMRSyntheticImageStorage](#) ,

```

    VLPhotographicImageStorage ,
    SegmentationStorage ,
    RTIonPlanStorage ,
    XRay3DAngiographicImageStorage ,
    EnhancedXAImageStorage ,
    RTIonBeamsTreatmentRecordStorage ,
    SurfaceSegmentationStorage ,
    VLWholeSlideMicroscopyImageStorage ,
    RTTreatmentSummaryRecordStorage ,
    EnhancedUSVolumeStorage ,
    XRayRadiationDoseSR ,
    VLEndoscopicImageStorage ,
    BreastTomosynthesisImageStorage ,
    FujiPrivateCRLImageStorage ,
    OphthalmicPhotography8BitImageStorage ,
    OphthalmicTomographyImageStorage ,
    VLMicroscopicImageStorage ,
    EnhancedPETImageStorage ,
    VideoPhotographicImageStorage ,
    XRay3DCraniofacialImageStorage ,
    IVOCTForPresentation ,
    IVOCTForProcessing ,
    LegacyConvertedEnhancedCTImageStorage ,
    LegacyConvertedEnhancedMRIImageStorage ,
    LegacyConvertedEnhancedPETImageStorage ,
    BreastProjectionXRayImageStorageForPresentation ,
    BreastProjectionXRayImageStorageForProcessing ,
    HardcopyColorImageStorage ,
    EnhancedMRColorImageStorage ,
    FujiPrivateMammoCRLImageStorage ,
    OphthalmicPhotography16BitImageStorage ,
    VideoMicroscopicImageStorage ,
    MS_END }
• enum ObjectType {
    NoObject = 0 ,
    Video ,
    Waveform ,
    Audio ,
    PDF ,
    URI ,
    Segmentation ,
    ObjectEnd }

```

## Public Member Functions

- [MediaStorage](#) (MSType type=MS\_END)
- const char \* [GetModality](#) () const
- unsigned int [GetModalityDimension](#) () const
- const char \* [GetString](#) () const
 

*Return the Media [String](#) of the object.*
- void [GuessFromModality](#) (const char \*modality, unsigned int dimension=2)
- bool [IsUndefined](#) () const

- [operator MType](#) () const
- bool [SetFromDataSet](#) ([DataSet](#) const &ds)
- bool [SetFromFile](#) ([File](#) const &file)
- bool [SetFromHeader](#) ([FileMetaInformation](#) const &fmi)
- bool [SetFromModality](#) ([DataSet](#) const &ds)

### Static Public Member Functions

- static const char \* [GetMSString](#) ([MType](#) ts)  
*Return the Media [String](#) associated. Will return NULL for MS\_END.*
- static [MType](#) [GetMSType](#) (const char \*str)
- static unsigned int [GetNumberOfModality](#) ()
- static unsigned int [GetNumberOfMSString](#) ()
- static unsigned int [GetNumberOfMSType](#) ()
- static bool [IsImage](#) ([MType](#) ts)

### Protected Member Functions

- void [SetFromSourceImageSequence](#) ([DataSet](#) const &ds)

### Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [MediaStorage](#) &ms)

## 10.189.1 Detailed Description

[MediaStorage](#).

### Note

FIXME There should not be any notion of [Image](#) and/or PDF at that point Only the codec can answer yes I support this Media Storage or not... For instance an [ImageCodec](#) will answer yes to most of them while a [PDFCodec](#) will answer only for the Encapsulated PDF

### See also

[UIDs](#)

### Examples

[CreateJPIPDataSet.cxx](#), [EncapsulateFileInRawData.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenerateStandardSOPClasses.cxx](#), [GetSubSequenceData.cxx](#), [MpegVideoInfo.cs](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), [TestReader.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), and [iU22tomultisc.cxx](#).

## 10.189.2 Member Enumeration Documentation

### 10.189.2.1 MType

```
enum gdcm::MediaStorage::MType
```

## Enumerator

MediaStorageDirectoryStorage	
ComputedRadiographyImageStorage	
DigitalXRayImageStorageForPresentation	
DigitalXRayImageStorageForProcessing	
DigitalMammographyImageStorageForPresentation	
DigitalMammographyImageStorageForProcessing	
DigitalIntraoralXrayImageStorageForPresentation	
DigitalIntraoralXRayImageStorageForProcessing	
CTImageStorage	
EnhancedCTImageStorage	
UltrasoundImageStorageRetired	
UltrasoundImageStorage	
UltrasoundMultiFrameImageStorageRetired	
UltrasoundMultiFrameImageStorage	
MRImageStorage	
EnhancedMRImageStorage	
MRSpectroscopyStorage	
NuclearMedicineImageStorageRetired	
SecondaryCaptureImageStorage	
MultiframeSingleBitSecondaryCaptureImageStorage	
MultiframeGrayscaleByteSecondaryCaptureImageStorage	
MultiframeGrayscaleWordSecondaryCaptureImageStorage	
MultiframeTrueColorSecondaryCaptureImageStorage	
StandaloneOverlayStorage	
StandaloneCurveStorage	
LeadECGWaveformStorage	
GeneralECGWaveformStorage	
AmbulatoryECGWaveformStorage	
HemodynamicWaveformStorage	
CardiacElectrophysiologyWaveformStorage	
BasicVoiceAudioWaveformStorage	
StandaloneModalityLUTStorage	
StandaloneVOILUTStorage	
GrayscaleSoftcopyPresentationStateStorageSOPClass	
XRayAngiographicImageStorage	
XRayRadiofluoroscopingImageStorage	
XRayAngiographicBiPlaneImageStorageRetired	
NuclearMedicineImageStorage	
RawDataStorage	
SpacialRegistrationStorage	
SpacialFiducialsStorage	
PETImageStorage	
RTImageStorage	

## Enumerator

RTDoseStorage	
RTStructureSetStorage	
RTPlanStorage	
CSANonImageStorage	
Philips3D	
EnhancedSR	
BasicTextSR	
HardcopyGrayscaleImageStorage	
ComprehensiveSR	
DetachedStudyManagementSOPClass	
EncapsulatedPDFStorage	
EncapsulatedCDASStorage	
StudyComponentManagementSOPClass	
DetachedVisitManagementSOPClass	
DetachedPatientManagementSOPClass	
VideoEndoscopicImageStorage	
GeneralElectricMagneticResonanceImageStorage	
GEPrivate3DModelStorage	
ToshibaPrivateDataStorage	
MammographyCADSR	
KeyObjectSelectionDocument	
HangingProtocolStorage	
ModalityPerformedProcedureStepSOPClass	
PhilipsPrivateMRSyntheticImageStorage	
VLPhotographicImageStorage	
SegmentationStorage	
RTIonPlanStorage	
XRay3DAngiographicImageStorage	
EnhancedXAImageStorage	
RTIonBeamsTreatmentRecordStorage	
SurfaceSegmentationStorage	
VLWholeSlideMicroscopyImageStorage	
RTTreatmentSummaryRecordStorage	
EnhancedUSVolumeStorage	
XRayRadiationDoseSR	
VLEndoscopicImageStorage	
BreastTomosynthesisImageStorage	
FujiPrivateCRImageStorage	
OphthalmicPhotography8BitImageStorage	
OphthalmicTomographyImageStorage	
VLMicroscopicImageStorage	
EnhancedPETImageStorage	
VideoPhotographicImageStorage	



## Enumerator

XRay3DCraniofacialImageStorage	
IVOCTForPresentation	
IVOCTForProcessing	
LegacyConvertedEnhancedCTImageStorage	
LegacyConvertedEnhancedMRIImageStorage	
LegacyConvertedEnhancedPETImageStorage	
BreastProjectionXRayImageStorageForPresentation	
BreastProjectionXRayImageStorageForProcessing	
HardcopyColorImageStorage	
EnhancedMRColorImageStorage	
FujiPrivateMammoCRImageStorage	
OphthalmicPhotography16BitImageStorage	
VideoMicroscopicImageStorage	
MS_END	

## Examples

[GenerateStandardSOPClasses.cxx](#), and [MpegVideoInfo.cs](#).

## 10.189.2.2 ObjectType

```
enum gdcm::MediaStorage::ObjectType
```

## Enumerator

NoObject	
Video	
Waveform	
Audio	
PDF	
URI	
Segmentation	
ObjectEnd	

## 10.189.3 Constructor &amp; Destructor Documentation

## 10.189.3.1 MediaStorage()

```
gdcm::MediaStorage::MediaStorage (
    MSType type = MS_END ) [inline]
```

## 10.189.4 Member Function Documentation

### 10.189.4.1 GetModality()

```
const char * gdcm::MediaStorage::GetModality ( ) const
```

### 10.189.4.2 GetModalityDimension()

```
unsigned int gdcm::MediaStorage::GetModalityDimension ( ) const
```

### 10.189.4.3 GetMSString()

```
static const char * gdcm::MediaStorage::GetMSString (
    MSType ts ) [static]
```

Return the Media [String](#) associated. Will return NULL for MS\_END.

#### Examples

[GenerateStandardSOPClasses.cxx](#).

### 10.189.4.4 GetMSType()

```
static MSType gdcm::MediaStorage::GetMSType (
    const char * str ) [static]
```

#### Examples

[MetaImageMD5Activiz.cs](#), and [TestReader.cxx](#).

### 10.189.4.5 GetNumberOfModality()

```
static unsigned int gdcm::MediaStorage::GetNumberOfModality ( ) [static]
```

### 10.189.4.6 GetNumberOfMSString()

```
static unsigned int gdcm::MediaStorage::GetNumberOfMSString ( ) [static]
```

### 10.189.4.7 GetNumberOfMSType()

```
static unsigned int gdcm::MediaStorage::GetNumberOfMSType ( ) [static]
```

#### 10.189.4.8 GetString()

```
const char * gdcm::MediaStorage::GetString ( ) const
```

Return the Media [String](#) of the object.

##### Examples

[CreateJPIPDataSet.cxx](#), [EncapsulateFileInRawData.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GetSubSequenceData.cxx](#), [MpegVideoInfo.cs](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), and [iU22tomultisc.cxx](#).

#### 10.189.4.9 GuessFromModality()

```
void gdcm::MediaStorage::GuessFromModality (
    const char * modality,
    unsigned int dimension = 2 )
```

References [gdcm::operator<<\(\)](#).

#### 10.189.4.10 IsImage()

```
static bool gdcm::MediaStorage::IsImage (
    MSType ts ) [static]
```

Returns whether DICOM has a Pixel Data element (7fe0,0010)

##### Warning

MRSpectroscopyStorage could be image but are not

##### Examples

[MetaImageMD5Activiz.cs](#).

#### 10.189.4.11 IsUndefined()

```
bool gdcm::MediaStorage::IsUndefined ( ) const [inline]
```

##### Examples

[TestReader.cxx](#).

#### 10.189.4.12 operator MType()

```
gdcm::MediaStorage::operator MType ( ) const [inline]
```

#### 10.189.4.13 SetFromDataSet()

```
bool gdcm::MediaStorage::SetFromDataSet (
    DataSet const & ds )
```

Advanced user only (functions should be protected level...) Those function are lower level than SetFromFile

#### 10.189.4.14 SetFromFile()

```
bool gdcm::MediaStorage::SetFromFile (
    File const & file )
```

Attempt to set the [MediaStorage](#) from a file: WARNING: When no [MediaStorage](#) & Modality are found BUT a PixelData element is found then [MediaStorage](#) is set to the default SecondaryCaptureImageStorage (return value is false in this case)

#### Examples

[ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [TestReader.cxx](#), [gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

#### 10.189.4.15 SetFromHeader()

```
bool gdcm::MediaStorage::SetFromHeader (
    FileMetaInformation const & fmi )
```

#### 10.189.4.16 SetFromModality()

```
bool gdcm::MediaStorage::SetFromModality (
    DataSet const & ds )
```

#### 10.189.4.17 SetFromSourceImageSequence()

```
void gdcm::MediaStorage::SetFromSourceImageSequence (
    DataSet const & ds ) [protected]
```

## 10.189.5 Friends And Related Symbol Documentation

### 10.189.5.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & os,  
    const MediaStorage & ms ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmMediaStorage.h](#)

## 10.190 gdcm::MemberCommand< T > Class Template Reference

[Command](#) subclass that calls a pointer to a member function.

```
#include <gdcmCommand.h>
```

Inheritance diagram for gdcm::MemberCommand< T >:



Collaboration diagram for `gdcm::MemberCommand< T >`:



### Public Types

- typedef `MemberCommand Self`
- typedef void(`T::* TConstMemberFunctionPointer`) (const `Subject *`, const `Event &`)
- typedef void(`T::* TMemberFunctionPointer`) (`Subject *`, const `Event &`)

### Public Member Functions

- `MemberCommand` (const `Self &`)=delete
- void `Execute` (const `Subject *`caller, const `Event &`event) override
- void `Execute` (`Subject *`caller, const `Event &`event) override
- void `operator=` (const `Self &`)=delete
- void `SetCallbackFunction` (`T *`object, `TConstMemberFunctionPointer` memberFunction)
- void `SetCallbackFunction` (`T *`object, `TMemberFunctionPointer` memberFunction)

### Public Member Functions inherited from `gdcm::Command`

- `Command` (const `Command &`)=delete
- void `operator=` (const `Command &`)=delete

**Public Member Functions inherited from [gdcmmembercommand::Subject](#)**

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*) const
- [Command](#) \* [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

**Public Member Functions inherited from [gdcmmembercommand::Object](#)**

- [Object](#) ()
- [Object](#) (const [Object](#) &)
- *Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

**Static Public Member Functions**

- static [SmartPointer](#)< [MemberCommand](#) > [New](#) ()

**Protected Member Functions**

- [MemberCommand](#) ()
- [~MemberCommand](#) () override=default

**Protected Member Functions inherited from [gdcmmembercommand::Command](#)**

- [Command](#) ()
- [~Command](#) () override

**Protected Member Functions inherited from [gdcmmembercommand::Object](#)**

- void [Register](#) ()
- void [UnRegister](#) ()

**Protected Attributes**

- [TConstMemberFunctionPointer](#) [m\\_ConstMemberFunction](#)
- [TMemberFunctionPointer](#) [m\\_MemberFunction](#)
- T \* [m\\_This](#)

### 10.190.1 Detailed Description

```
template<class T>
class gdcM::MemberCommand< T >
```

[Command](#) subclass that calls a pointer to a member function.

[MemberCommand](#) calls a pointer to a member function with the same arguments as `Execute` on [Command](#).

### 10.190.2 Member Typedef Documentation

#### 10.190.2.1 Self

```
template<class T >
typedef MemberCommand gdcM::MemberCommand< T >::Self
```

Standard class typedefs.

#### 10.190.2.2 TConstMemberFunctionPointer

```
template<class T >
typedef void(T::* gdcM::MemberCommand< T >::TConstMemberFunctionPointer) (const Subject *, const
Event &)
```

#### 10.190.2.3 TMemberFunctionPointer

```
template<class T >
typedef void(T::* gdcM::MemberCommand< T >::TMemberFunctionPointer) (Subject *, const Event &)
```

pointer to a member function that takes a `Subject*` and the event

### 10.190.3 Constructor & Destructor Documentation

#### 10.190.3.1 MemberCommand() [1/2]

```
template<class T >
gdcM::MemberCommand< T >::MemberCommand (
    const Self & ) [delete]
```

#### 10.190.3.2 MemberCommand() [2/2]

```
template<class T >
gdcM::MemberCommand< T >::MemberCommand ( ) [inline], [protected]
```

Referenced by [gdcM::MemberCommand< T >::New\(\)](#).



### 10.190.3.3 ~MemberCommand()

```
template<class T >
gdcmmembercommand< T >::~~MemberCommand ( ) [override], [protected], [default]
```

## 10.190.4 Member Function Documentation

### 10.190.4.1 Execute() [1/2]

```
template<class T >
void gdcmmembercommand< T >::Execute (
    const Subject * caller,
    const Event & event ) [inline], [override], [virtual]
```

Invoke the member function with a const object.

Implements [gdcmmembercommand](#).

References [gdcmmembercommand< T >::m\\_ConstMemberFunction](#).

### 10.190.4.2 Execute() [2/2]

```
template<class T >
void gdcmmembercommand< T >::Execute (
    Subject * caller,
    const Event & event ) [inline], [override], [virtual]
```

Invoke the member function.

Implements [gdcmmembercommand](#).

References [gdcmmembercommand< T >::m\\_MemberFunction](#).

### 10.190.4.3 New()

```
template<class T >
static SmartPointer< MemberCommand > gdcmmembercommand< T >::New ( ) [inline], [static]
```

Method for creation through the object factory.

References [gdcmmembercommand< T >::MemberCommand\(\)](#).

### 10.190.4.4 operator=()

```
template<class T >
void gdcmmembercommand< T >::operator= (
    const Self & ) [delete]
```

**10.190.4.5 SetCallbackFunction() [1/2]**

```
template<class T >
void gdcM::MemberCommand< T >::SetCallbackFunction (
    T * object,
    TConstMemberFunctionPointer memberFunction ) [inline]
```

References [gdcM::MemberCommand< T >::m\\_ConstMemberFunction](#), and [gdcM::MemberCommand< T >::m\\_This](#).

**10.190.4.6 SetCallbackFunction() [2/2]**

```
template<class T >
void gdcM::MemberCommand< T >::SetCallbackFunction (
    T * object,
    TMemberFunctionPointer memberFunction ) [inline]
```

Run-time type information (and related methods). Set the callback function along with the object that it will be invoked on.

References [gdcM::MemberCommand< T >::m\\_MemberFunction](#), and [gdcM::MemberCommand< T >::m\\_This](#).

**10.190.5 Member Data Documentation****10.190.5.1 m\_ConstMemberFunction**

```
template<class T >
TConstMemberFunctionPointer gdcM::MemberCommand< T >::m_ConstMemberFunction [protected]
```

Referenced by [gdcM::MemberCommand< T >::Execute\(\)](#), and [gdcM::MemberCommand< T >::SetCallbackFunction\(\)](#).

**10.190.5.2 m\_MemberFunction**

```
template<class T >
TMemberFunctionPointer gdcM::MemberCommand< T >::m_MemberFunction [protected]
```

Referenced by [gdcM::MemberCommand< T >::Execute\(\)](#), and [gdcM::MemberCommand< T >::SetCallbackFunction\(\)](#).

**10.190.5.3 m\_This**

```
template<class T >
T* gdcM::MemberCommand< T >::m_This [protected]
```

Referenced by [gdcM::MemberCommand< T >::SetCallbackFunction\(\)](#), and [gdcM::MemberCommand< T >::SetCallbackFunction\(\)](#).

The documentation for this class was generated from the following file:

- [gdcMCommand.h](#)

## 10.191 gdcM::MeshPrimitive Class Reference

This class defines surface mesh primitives.

```
#include <gdcMMeshPrimitive.h>
```

Inheritance diagram for gdcM::MeshPrimitive:



Collaboration diagram for gdcM::MeshPrimitive:



## Public Types

- enum [MPTType](#) {  
[VERTEX](#) = 0 ,  
[EDGE](#) ,  
[TRIANGLE](#) ,  
[TRIANGLE\\_STRIP](#) ,  
[TRIANGLE\\_FAN](#) ,  
[LINE](#) ,  
[FACET](#) ,  
[MPTType\\_END](#) }

*This enumeration defines primitive types.*

- typedef std::vector< [DataElement](#) > [PrimitivesData](#)

## Public Member Functions

- [MeshPrimitive](#) ()
- [~MeshPrimitive](#) () override
- void [AddPrimitiveData](#) ([DataElement](#) const &de)
- unsigned int [GetNumberOfPrimitivesData](#) () const
- [DataElement](#) & [GetPrimitiveData](#) ()
- const [DataElement](#) & [GetPrimitiveData](#) () const
- [DataElement](#) & [GetPrimitiveData](#) (const unsigned int idx)
- const [DataElement](#) & [GetPrimitiveData](#) (const unsigned int idx) const
- [PrimitivesData](#) & [GetPrimitivesData](#) ()
- const [PrimitivesData](#) & [GetPrimitivesData](#) () const
- [MPTType](#) [GetPrimitiveType](#) () const
- void [SetPrimitiveData](#) (const unsigned int idx, [DataElement](#) const &de)
- void [SetPrimitiveData](#) ([DataElement](#) const &de)
- void [SetPrimitivesData](#) ([PrimitivesData](#) const &DEs)
- void [SetPrimitiveType](#) (const [MPTType](#) type)

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
- *Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

## Static Public Member Functions

- static [MPTType](#) [GetMPTType](#) (const char \*type)
- static const char \* [GetMPTTypeString](#) (const [MPTType](#) type)

Protected Attributes

- [PrimitivesData](#) [PrimitiveData](#)
- [MPType](#) [PrimitiveType](#)

Additional Inherited Members

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

10.191.1 Detailed Description

This class defines surface mesh primitives.

It is designed from surface mesh primitives macro.

See also

PS 3.3 C.27.4

10.191.2 Member Typedef Documentation

10.191.2.1 PrimitivesData

```
typedef std::vector< DataElement > gdcm::MeshPrimitive::PrimitivesData
```

10.191.3 Member Enumeration Documentation

10.191.3.1 MPType

```
enum gdcm::MeshPrimitive::MPType
```

This enumeration defines primitive types.

See also

PS 3.3 C.27.4.1

Enumerator

VERTEX	
EDGE	
TRIANGLE	
TRIANGLE_STRIP	
TRIANGLE_FAN	
LINE	
FACET	

## 10.191.4 Constructor & Destructor Documentation

### 10.191.4.1 MeshPrimitive()

```
gdcM::MeshPrimitive::MeshPrimitive ( )
```

### 10.191.4.2 ~MeshPrimitive()

```
gdcM::MeshPrimitive::~~MeshPrimitive ( ) [override]
```

## 10.191.5 Member Function Documentation

### 10.191.5.1 AddPrimitiveData()

```
void gdcM::MeshPrimitive::AddPrimitiveData (
    DataElement const & de )
```

### 10.191.5.2 GetMPType()

```
static MPType gdcM::MeshPrimitive::GetMPType (
    const char * type ) [static]
```

### 10.191.5.3 GetMPTypeString()

```
static const char * gdcM::MeshPrimitive::GetMPTypeString (
    const MPType type ) [static]
```

### 10.191.5.4 GetNumberOfPrimitivesData()

```
unsigned int gdcM::MeshPrimitive::GetNumberOfPrimitivesData ( ) const
```

### 10.191.5.5 GetPrimitiveData() [1/4]

```
DataElement & gdcM::MeshPrimitive::GetPrimitiveData ( )
```

### 10.191.5.6 GetPrimitiveData() [2/4]

```
const DataElement & gdcM::MeshPrimitive::GetPrimitiveData ( ) const
```

**10.191.5.7 GetPrimitiveData() [3/4]**

```
DataElement & gdcm::MeshPrimitive::GetPrimitiveData (
    const unsigned int idx )
```

**10.191.5.8 GetPrimitiveData() [4/4]**

```
const DataElement & gdcm::MeshPrimitive::GetPrimitiveData (
    const unsigned int idx ) const
```

**10.191.5.9 GetPrimitivesData() [1/2]**

```
PrimitivesData & gdcm::MeshPrimitive::GetPrimitivesData ( )
```

**10.191.5.10 GetPrimitivesData() [2/2]**

```
const PrimitivesData & gdcm::MeshPrimitive::GetPrimitivesData ( ) const
```

**10.191.5.11 GetPrimitiveType()**

```
MPType gdcm::MeshPrimitive::GetPrimitiveType ( ) const
```

**10.191.5.12 SetPrimitiveData() [1/2]**

```
void gdcm::MeshPrimitive::SetPrimitiveData (
    const unsigned int idx,
    DataElement const & de )
```

**10.191.5.13 SetPrimitiveData() [2/2]**

```
void gdcm::MeshPrimitive::SetPrimitiveData (
    DataElement const & de )
```

**10.191.5.14 SetPrimitivesData()**

```
void gdcm::MeshPrimitive::SetPrimitivesData (
    PrimitivesData const & DEs )
```

### 10.191.5.15 SetPrimitiveType()

```
void gdcM::MeshPrimitive::SetPrimitiveType (
    const MPTypE type )
```

## 10.191.6 Member Data Documentation

### 10.191.6.1 PrimitiveData

```
PrimitivEsData gdcM::MeshPrimitive::PrimitiveData [protected]
```

### 10.191.6.2 PrimitiveType

```
MPTypE gdcM::MeshPrimitive::PrimitiveType [protected]
```

The documentation for this class was generated from the following file:

- [gdcMMeshPrimitive.h](#)

## 10.192 gdcM::ModalityPerformedProcedureStepCreateQuery Class Reference

[ModalityPerformedProcedureStepCreateQuery.](#)

```
#include <gdcMModalityPerformedProcedureStepCreateQuery.h>
```

Inheritance diagram for gdcM::ModalityPerformedProcedureStepCreateQuery:





Collaboration diagram for gdcm::ModalityPerformedProcedureStepCreateQuery:



### Public Member Functions

- [ModalityPerformedProcedureStepCreateQuery](#) (const std::string &iSopInstanceUID)
- [UIDs::TSName GetAbstractSyntaxUID](#) () const override
- [gdcm::DataSet GetRequiredDataSet](#) () const
- bool [ValidateQuery](#) (bool inStrict=true) const override

### Public Member Functions inherited from [gdcm::BaseQuery](#)

- [~BaseQuery](#) () override
- void [AddQueryDataSet](#) (const [DataSet](#) &ds)
- [DataSet](#) & [GetQueryDataSet](#) ()
- [DataSet](#) const & [GetQueryDataSet](#) () const
- *Set/Get the internal representation of the query as a [DataSet](#).*
- std::string [GetSOPInstanceUID](#) () const
- void [Print](#) (std::ostream &os) const override
- void [SetSearchParameter](#) (const std::string &inKeyword, const std::string &inValue)
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const std::string &inValue)
- void [SetSOPInstanceUID](#) (const std::string &iSopInstanceUID)
- const std::ostream & [WriteHelpFile](#) (std::ostream &os)
- bool [WriteQuery](#) (const std::string &inFileName)

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

## Friends

- class [QueryFactory](#)

## Additional Inherited Members

## Protected Member Functions inherited from [gdcm::BaseQuery](#)

- [BaseQuery](#) ()
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const [DictEntry](#) &inDictEntry, const std::string &inValue)
- bool [ValidDataSet](#) (const [DataSet](#) &dataSetToValid, const [DataSet](#) &dataSetReference) const

## Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

## Protected Attributes inherited from [gdcm::BaseQuery](#)

- [DataSet](#) [mDataSet](#)
- std::string [mSopInstanceUID](#)

### 10.192.1 Detailed Description

[ModalityPerformedProcedureStepCreateQuery](#).

contains: the class which will produce a dataset for n-create for Modality Performed Procedure Step sop class

### 10.192.2 Constructor & Destructor Documentation

#### 10.192.2.1 [ModalityPerformedProcedureStepCreateQuery](#)()

```
gdcm::ModalityPerformedProcedureStepCreateQuery::ModalityPerformedProcedureStepCreateQuery (
    const std::string & iSopInstanceUID )
```

### 10.192.3 Member Function Documentation

#### 10.192.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::ModalityPerformedProcedureStepCreateQuery::GetAbstractSyntaxUID ( ) const [override],  
[virtual]
```

Implements [gdcm::BaseQuery](#).

#### 10.192.3.2 GetRequiredDataSet()

```
gdcm::DataSet gdcm::ModalityPerformedProcedureStepCreateQuery::GetRequiredDataSet ( ) const
```

#### 10.192.3.3 ValidateQuery()

```
bool gdcm::ModalityPerformedProcedureStepCreateQuery::ValidateQuery (  
    bool inStrict = true ) const [override], [virtual]
```

Implements [gdcm::BaseQuery](#).

### 10.192.4 Friends And Related Symbol Documentation

#### 10.192.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

The documentation for this class was generated from the following file:

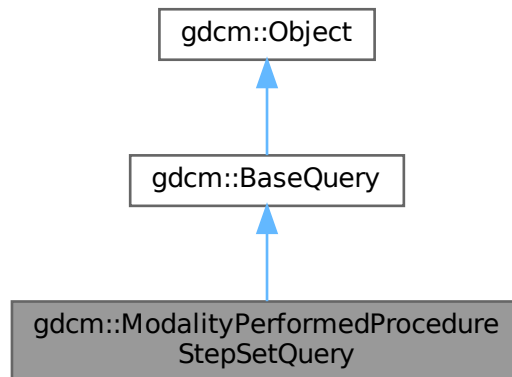
- [gdcmModalityPerformedProcedureStepCreateQuery.h](#)

## 10.193 gdcm::ModalityPerformedProcedureStepSetQuery Class Reference

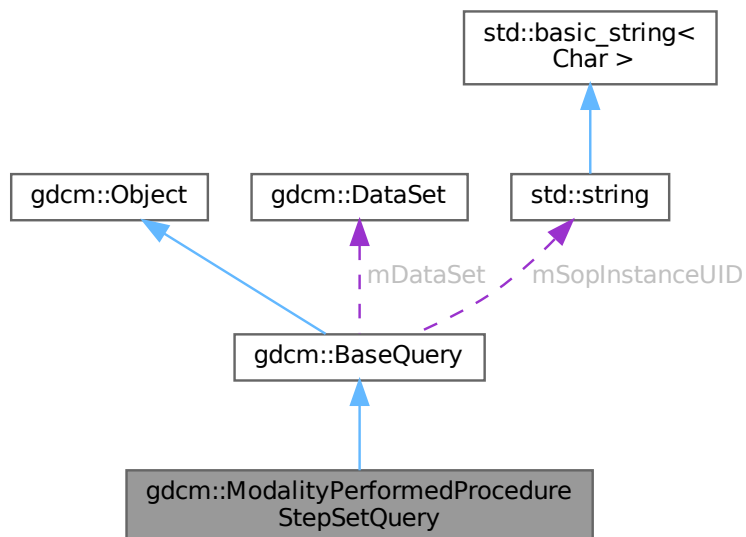
[ModalityPerformedProcedureStepSetQuery](#).

```
#include <gdcmModalityPerformedProcedureStepSetQuery.h>
```

Inheritance diagram for `gdcm::ModalityPerformedProcedureStepSetQuery`:



Collaboration diagram for `gdcm::ModalityPerformedProcedureStepSetQuery`:



## Public Member Functions

- [ModalityPerformedProcedureStepSetQuery](#) (const `std::string` &`iSopInstanceUID`)

- [UIDs::TSName GetAbstractSyntaxUID](#) () const override
- [gdcm::DataSet GetRequiredDataSet](#) () const
- bool [ValidateQuery](#) (bool inStrict=true) const override

### Public Member Functions inherited from [gdcm::BaseQuery](#)

- [~BaseQuery](#) () override
- void [AddQueryDataSet](#) (const [DataSet](#) &ds)
- [DataSet](#) & [GetQueryDataSet](#) ()
- [DataSet](#) const & [GetQueryDataSet](#) () const
- *Set/Get the internal representation of the query as a [DataSet](#).*
- std::string [GetSOPInstanceUID](#) () const
- void [Print](#) (std::ostream &os) const override
- void [SetSearchParameter](#) (const std::string &inKeyword, const std::string &inValue)
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const std::string &inValue)
- void [SetSOPInstanceUID](#) (const std::string &iSopInstanceUID)
- const std::ostream & [WriteHelpFile](#) (std::ostream &os)
- bool [WriteQuery](#) (const std::string &inFileName)

### Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
- *Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

### Friends

- class [QueryFactory](#)

### Additional Inherited Members

### Protected Member Functions inherited from [gdcm::BaseQuery](#)

- [BaseQuery](#) ()
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const [DictEntry](#) &inDictEntry, const std::string &inValue)
- bool [ValidDataSet](#) (const [DataSet](#) &dataSetToValid, const [DataSet](#) &dataSetReference) const

### Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

## Protected Attributes inherited from [gdcm::BaseQuery](#)

- [DataSet](#) `mDataSet`
- `std::string` `mSopInstanceUID`

### 10.193.1 Detailed Description

[ModalityPerformedProcedureStepSetQuery](#).

contains: the class which will produce a dataset for n-set for Modality Performed Procedure Step sop class

### 10.193.2 Constructor & Destructor Documentation

#### 10.193.2.1 [ModalityPerformedProcedureStepSetQuery\(\)](#)

```
gdcm::ModalityPerformedProcedureStepSetQuery::ModalityPerformedProcedureStepSetQuery (
    const std::string & iSopInstanceUID )
```

### 10.193.3 Member Function Documentation

#### 10.193.3.1 [GetAbstractSyntaxUID\(\)](#)

```
UIDs::TSName gdcm::ModalityPerformedProcedureStepSetQuery::GetAbstractSyntaxUID ( ) const [override],
[virtual]
```

Implements [gdcm::BaseQuery](#).

#### 10.193.3.2 [GetRequiredDataSet\(\)](#)

```
gdcm::DataSet gdcm::ModalityPerformedProcedureStepSetQuery::GetRequiredDataSet ( ) const
```

#### 10.193.3.3 [ValidateQuery\(\)](#)

```
bool gdcm::ModalityPerformedProcedureStepSetQuery::ValidateQuery (
    bool inStrict = true ) const [override], [virtual]
```

Implements [gdcm::BaseQuery](#).

## 10.193.4 Friends And Related Symbol Documentation

### 10.193.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

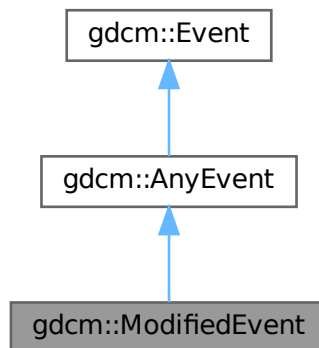
The documentation for this class was generated from the following file:

- [gdcmModalityPerformedProcedureStepSetQuery.h](#)

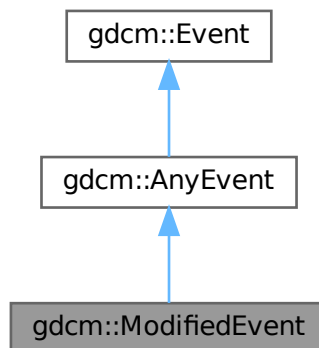
## 10.194 gdcm::ModifiedEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::ModifiedEvent:



Collaboration diagram for `gdcm::ModifiedEvent`:



#### Additional Inherited Members

#### Public Member Functions inherited from `gdcm::Event`

- `Event ()`
- `Event (const Event &)`
- `virtual ~Event ()`
- `virtual bool CheckEvent (const Event *) const =0`
- `virtual const char * GetEventName () const =0`
- `virtual Event * MakeObject () const =0`
- `void operator= (const Event &)=delete`
- `virtual void Print (std::ostream &os) const`

The documentation for this class was generated from the following file:

- `gdcmEvent.h`

## 10.195 `gdcm::Module` Class Reference

Class for representing a `Module`.

```
#include <gdcmModule.h>
```

#### Public Types

- `typedef std::vector< std::string > ArrayIncludeMacrosType`
- `typedef std::map< Tag, ModuleEntry > MapModuleEntry`



## Public Member Functions

- [Module](#) ()=default
- void [AddMacro](#) (const char \*include)
- void [AddModuleEntry](#) (const [Tag](#) &tag, const [ModuleEntry](#) &module)  
*Will add a [ModuleEntry](#) directly at root-level. See [Macro](#) for nested-included level.*
- void [Clear](#) ()
- bool [FindModuleEntryInMacros](#) ([Macros](#) const &macros, const [Tag](#) &tag) const
- const [ModuleEntry](#) & [GetModuleEntryInMacros](#) ([Macros](#) const &macros, const [Tag](#) &tag) const
- const char \* [GetName](#) () const
- void [SetName](#) (const char \*name)
- bool [Verify](#) (const [DataSet](#) &ds, [Usage](#) const &usage) const

## Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [Module](#) &\_val)

## 10.195.1 Detailed Description

Class for representing a [Module](#).

### Note

[Module](#): A set of Attributes within an Information Entity or Normalized [IOD](#) which are logically related to each other.

### See also

[Macro](#)

### Examples

[TraverseModules.cxx](#).

## 10.195.2 Member Typedef Documentation

### 10.195.2.1 ArrayIncludeMacroType

```
typedef std::vector<std::string> gdcm::Module::ArrayIncludeMacroType
```

### 10.195.2.2 MapModuleEntry

```
typedef std::map<Tag, ModuleEntry> gdcm::Module::MapModuleEntry
```

### 10.195.3 Constructor & Destructor Documentation

#### 10.195.3.1 Module()

```
gdcmmodule::Module::Module ( ) [default]
```

References [gdcmmodule::operator<<\(\)](#).

### 10.195.4 Member Function Documentation

#### 10.195.4.1 AddMacro()

```
void gdcmmodule::Module::AddMacro (
    const char * include ) [inline]
```

#### 10.195.4.2 AddModuleEntry()

```
void gdcmmodule::Module::AddModuleEntry (
    const Tag & tag,
    const ModuleEntry & module ) [inline]
```

Will add a [ModuleEntry](#) directly at root-level. See [Macro](#) for nested-included level.

#### 10.195.4.3 Clear()

```
void gdcmmodule::Module::Clear ( ) [inline]
```

#### 10.195.4.4 FindModuleEntryInMacros()

```
bool gdcmmodule::Module::FindModuleEntryInMacros (
    Macros const & macros,
    const Tag & tag ) const
```

Find or Get a [ModuleEntry](#). [ModuleEntry](#) are either search are root-level or within nested-macro included in module.

#### 10.195.4.5 GetModuleEntryInMacros()

```
const ModuleEntry & gdcmmodule::Module::GetModuleEntryInMacros (
    Macros const & macros,
    const Tag & tag ) const
```

#### 10.195.4.6 GetName()

```
const char * gdcm::Module::GetName ( ) const [inline]
```

#### 10.195.4.7 SetName()

```
void gdcm::Module::SetName (
    const char * name ) [inline]
```

#### 10.195.4.8 Verify()

```
bool gdcm::Module::Verify (
    const DataSet & ds,
    Usage const & usage ) const
```

### 10.195.5 Friends And Related Symbol Documentation

#### 10.195.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const Module & _val ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmModule.h](#)

## 10.196 gdcm::ModuleEntry Class Reference

Class for representing a [ModuleEntry](#).

```
#include <gdcmModuleEntry.h>
```

Inheritance diagram for gdcm::ModuleEntry:



Collaboration diagram for `gdcmmoduleentry::ModuleEntry`:



## Public Types

- typedef `std::string` [Description](#)

## Public Member Functions

- [ModuleEntry](#) (`const char *name=""`, `const char *type="3"`, `const char *description=""`)
- virtual `~ModuleEntry()`=default
- const [Description](#) & [GetDescription](#) () const
- const char \* [GetName](#) () const
- const [Type](#) & [GetType](#) () const
- void [SetDescription](#) (`const char *d`)
- void [SetName](#) (`const char *name`)
- void [SetType](#) (`const Type &type`)

## Protected Attributes

- [Type](#) [DataElementType](#)
- [Description](#) [DescriptionField](#)
- `std::string` [Name](#)

## Friends

- `std::ostream & operator<<` (`std::ostream &_os`, const [ModuleEntry](#) &\_val)

## 10.196.1 Detailed Description

Class for representing a [ModuleEntry](#).

Note

bla

See also

[DictEntry](#)

Examples

[TraverseModules.cxx](#).

## 10.196.2 Member Typedef Documentation

### 10.196.2.1 Description

```
typedef std::string gdcm::ModuleEntry::Description
```

## 10.196.3 Constructor & Destructor Documentation

### 10.196.3.1 ModuleEntry()

```
gdcm::ModuleEntry::ModuleEntry (
    const char * name = "",
    const char * type = "3",
    const char * description = "" ) [inline]
```

### 10.196.3.2 ~ModuleEntry()

```
virtual gdcm::ModuleEntry::~~ModuleEntry ( ) [virtual], [default]
```

References [gdcm::operator<<\(\)](#).

## 10.196.4 Member Function Documentation

### 10.196.4.1 GetDescription()

```
const Description & gdcm::ModuleEntry::GetDescription ( ) const [inline]
```

#### 10.196.4.2 GetName()

```
const char * gdcM::ModuleEntry::GetName ( ) const [inline]
```

#### 10.196.4.3 GetType()

```
const Type & gdcM::ModuleEntry::GetType ( ) const [inline]
```

#### Examples

[TraverseModules.cxx](#).

#### 10.196.4.4 SetDescription()

```
void gdcM::ModuleEntry::SetDescription (
    const char * d ) [inline]
```

#### 10.196.4.5 SetName()

```
void gdcM::ModuleEntry::SetName (
    const char * name ) [inline]
```

#### 10.196.4.6 SetType()

```
void gdcM::ModuleEntry::SetType (
    const Type & type ) [inline]
```

### 10.196.5 Friends And Related Symbol Documentation

#### 10.196.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const ModuleEntry & _val ) [friend]
```

### 10.196.6 Member Data Documentation

#### 10.196.6.1 DataElementType

```
Type gdcM::ModuleEntry::DataElementType [protected]
```

### 10.196.6.2 DescriptionField

`Description` gdcm::ModuleEntry::DescriptionField [protected]

### 10.196.6.3 Name

`std::string` gdcm::ModuleEntry::Name [protected]

The documentation for this class was generated from the following file:

- [gdcmModuleEntry.h](#)

## 10.197 gdcm::Modules Class Reference

Class for representing a [Modules](#).

```
#include <gdcmModules.h>
```

### Public Types

- typedef std::map< std::string, [Module](#) > [ModuleMapType](#)

### Public Member Functions

- [Modules](#) ()=default
- void [AddModule](#) (const char \*ref, const [Module](#) &module)
- void [Clear](#) ()
- const [Module](#) & [GetModule](#) (const char \*name) const
- bool [IsEmpty](#) () const

### Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [Modules](#) &\_val)

### 10.197.1 Detailed Description

Class for representing a [Modules](#).

#### Note

bla

#### See also

[Module](#)

#### Examples

[TraverseModules.cxx](#).

## 10.197.2 Member Typedef Documentation

### 10.197.2.1 ModuleMapType

```
typedef std::map<std::string, Module> gdcM::Modules::ModuleMapType
```

## 10.197.3 Constructor & Destructor Documentation

### 10.197.3.1 Modules()

```
gdcM::Modules::Modules ( ) [default]
```

References [gdcM::operator<<\(\)](#).

## 10.197.4 Member Function Documentation

### 10.197.4.1 AddModule()

```
void gdcM::Modules::AddModule (
    const char * ref,
    const Module & module ) [inline]
```

### 10.197.4.2 Clear()

```
void gdcM::Modules::Clear ( ) [inline]
```

### 10.197.4.3 GetModule()

```
const Module & gdcM::Modules::GetModule (
    const char * name ) const [inline]
```

### 10.197.4.4 IsEmpty()

```
bool gdcM::Modules::IsEmpty ( ) const [inline]
```

## 10.197.5 Friends And Related Symbol Documentation

### 10.197.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const Modules & _val ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcMModules.h](#)

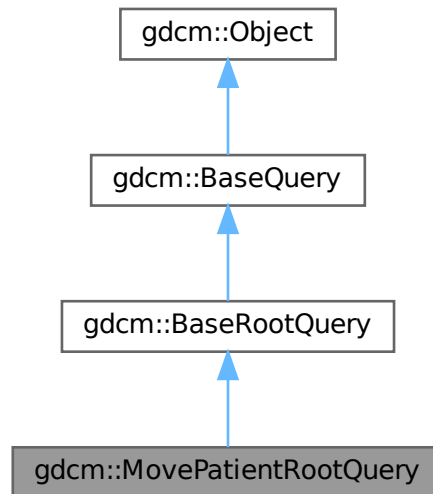


## 10.198 gdcm::MovePatientRootQuery Class Reference

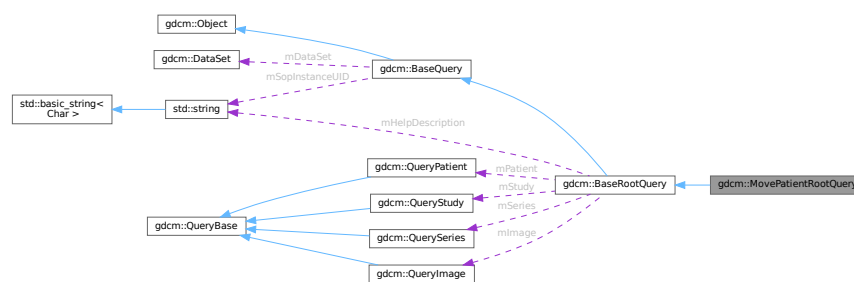
[MovePatientRootQuery](#).

```
#include <gdcmMovePatientRootQuery.h>
```

Inheritance diagram for gdcm::MovePatientRootQuery:



Collaboration diagram for gdcm::MovePatientRootQuery:



### Public Member Functions

- [MovePatientRootQuery](#) ()
- `UIDs::TSName GetAbstractSyntaxUID ()` const override
- `std::vector< Tag > GetTagListByLevel` (const [EQueryLevel](#) &inQueryLevel) override
- void [InitializeDataSet](#) (const [EQueryLevel](#) &inQueryLevel) override
- bool [ValidateQuery](#) (bool inStrict=true) const override

## Public Member Functions inherited from [gdcm::BaseRootQuery](#)

- [~BaseRootQuery](#) () override=default
- [EQueryLevel](#) [GetQueryLevelFromQueryRoot](#) ([ERootType](#) roottype)

## Public Member Functions inherited from [gdcm::BaseQuery](#)

- [~BaseQuery](#) () override
  - void [AddQueryDataSet](#) (const [DataSet](#) &ds)
  - [DataSet](#) & [GetQueryDataSet](#) ()
  - [DataSet](#) const & [GetQueryDataSet](#) () const
- Set/Get the internal representation of the query as a [DataSet](#).*
- std::string [GetSOPInstanceUID](#) () const
  - void [Print](#) (std::ostream &os) const override
  - void [SetSearchParameter](#) (const std::string &inKeyword, const std::string &inValue)
  - void [SetSearchParameter](#) (const [Tag](#) &inTag, const std::string &inValue)
  - void [SetSOPInstanceUID](#) (const std::string &iSopInstanceUID)
  - const std::ostream & [WriteHelpFile](#) (std::ostream &os)
  - bool [WriteQuery](#) (const std::string &inFileName)

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
  - [Object](#) (const [Object](#) &)
- Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
  - void [operator=](#) (const [Object](#) &)

## Friends

- class [QueryFactory](#)

## Additional Inherited Members

## Static Public Member Functions inherited from [gdcm::BaseRootQuery](#)

- static [QueryBase](#) \* [Construct](#) ([ERootType](#) inRootType, [EQueryLevel](#) qllevel)
- static int [GetQueryLevelFromString](#) (const char \*str)
- static const char \* [GetQueryLevelString](#) ([EQueryLevel](#) ql)

## Protected Member Functions inherited from [gdcm::BaseRootQuery](#)

- [BaseRootQuery](#) ()

**Protected Member Functions inherited from [gdcm::BaseQuery](#)**

- [BaseQuery](#) ()
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const [DictEntry](#) &inDictEntry, const std::string &inValue)
- bool [ValidDataSet](#) (const [DataSet](#) &dataSetToValid, const [DataSet](#) &dataSetReference) const

**Protected Member Functions inherited from [gdcm::Object](#)**

- void [Register](#) ()
- void [UnRegister](#) ()

**Protected Attributes inherited from [gdcm::BaseRootQuery](#)**

- std::string [mHelpDescription](#)
- [QueryImage](#) [mImage](#)
- [QueryPatient](#) [mPatient](#)
- [ERootType](#) [mRootType](#)
- [QuerySeries](#) [mSeries](#)
- [QueryStudy](#) [mStudy](#)

**Protected Attributes inherited from [gdcm::BaseQuery](#)**

- [DataSet](#) [mDataSet](#)
- std::string [mSopInstanceUID](#)

**10.198.1 Detailed Description**

[MovePatientRootQuery](#).

contains: the class which will produce a dataset for c-move with patient root

**10.198.2 Constructor & Destructor Documentation****10.198.2.1 [MovePatientRootQuery](#)()**

```
gdcm::MovePatientRootQuery::MovePatientRootQuery ( )
```

**10.198.3 Member Function Documentation****10.198.3.1 [GetAbstractSyntaxUID](#)()**

```
UIDs::TSName gdcm::MovePatientRootQuery::GetAbstractSyntaxUID ( ) const [override], [virtual]
```

Implements [gdcm::BaseQuery](#).

### 10.198.3.2 GetTagListByLevel()

```
std::vector< Tag > gdcm::MovePatientRootQuery::GetTagListByLevel (
    const EQueryLevel & inQueryLevel ) [override], [virtual]
```

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implements [gdcm::BaseRootQuery](#).

### 10.198.3.3 InitializeDataSet()

```
void gdcm::MovePatientRootQuery::InitializeDataSet (
    const EQueryLevel & inQueryLevel ) [override], [virtual]
```

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmTk

Implements [gdcm::BaseRootQuery](#).

### 10.198.3.4 ValidateQuery()

```
bool gdcm::MovePatientRootQuery::ValidateQuery (
    bool inStrict = true ) const [override], [virtual]
```

have to be able to ensure that 0x8,0x52 is set (which will be true if InitializeDataSet is called...) that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional) by default, this function checks to see if the query is for finding, which is more permissive than for moving. For moving, only the unique tags are allowed. 10 Jan 2011: adding in the 'strict' mode. according to the standard (at least, how I've read it), only tags for a particular level should be allowed in a particular query (ie, just series level tags in a series level query). However, it seems that dcm4chee doesn't share that interpretation. So, if 'inStrict' is false, then tags from the current level and all higher levels are now considered valid. So, if you're doing a non-strict series-level query, tags from the patient and study level can be passed along as well.

Implements [gdcm::BaseRootQuery](#).

## 10.198.4 Friends And Related Symbol Documentation

### 10.198.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

The documentation for this class was generated from the following file:

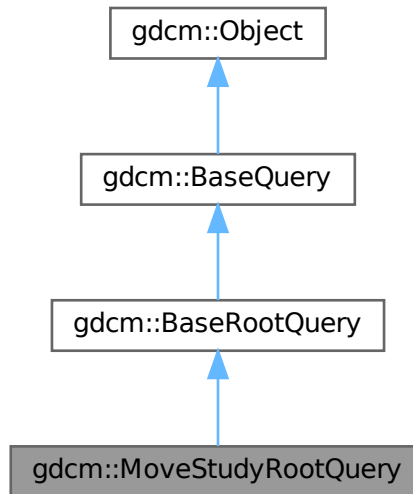
- [gdcmMovePatientRootQuery.h](#)

## 10.199 gdcm::MoveStudyRootQuery Class Reference

[MoveStudyRootQuery](#).

```
#include <gdcmMoveStudyRootQuery.h>
```

Inheritance diagram for gdcm::MoveStudyRootQuery:



Collaboration diagram for gdcm::MoveStudyRootQuery:



### Public Member Functions

- [MoveStudyRootQuery](#) ()
- `UIDs::TSName GetAbstractSyntaxUID ()` const override
- `std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)` override
- `void InitializeDataSet (const EQueryLevel &inQueryLevel)` override
- `bool ValidateQuery (bool inStrict=true)` const override

## Public Member Functions inherited from [gdcm::BaseRootQuery](#)

- [~BaseRootQuery](#) () override=default
- [EQueryLevel GetQueryLevelFromQueryRoot](#) (ERootType roottype)

## Public Member Functions inherited from [gdcm::BaseQuery](#)

- [~BaseQuery](#) () override
  - void [AddQueryDataSet](#) (const [DataSet](#) &ds)
  - [DataSet](#) & [GetQueryDataSet](#) ()
  - [DataSet](#) const & [GetQueryDataSet](#) () const
- Set/Get the internal representation of the query as a [DataSet](#).*
- std::string [GetSOPInstanceUID](#) () const
  - void [Print](#) (std::ostream &os) const override
  - void [SetSearchParameter](#) (const std::string &inKeyword, const std::string &inValue)
  - void [SetSearchParameter](#) (const [Tag](#) &inTag, const std::string &inValue)
  - void [SetSOPInstanceUID](#) (const std::string &iSopInstanceUID)
  - const std::ostream & [WriteHelpFile](#) (std::ostream &os)
  - bool [WriteQuery](#) (const std::string &inFileName)

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
  - [Object](#) (const [Object](#) &)
- Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
  - void [operator=](#) (const [Object](#) &)

## Friends

- class [QueryFactory](#)

## Additional Inherited Members

## Static Public Member Functions inherited from [gdcm::BaseRootQuery](#)

- static [QueryBase](#) \* [Construct](#) (ERootType inRootType, [EQueryLevel](#) qllevel)
- static int [GetQueryLevelFromString](#) (const char \*str)
- static const char \* [GetQueryLevelString](#) ([EQueryLevel](#) ql)

## Protected Member Functions inherited from [gdcm::BaseRootQuery](#)

- [BaseRootQuery](#) ()

**Protected Member Functions inherited from [gdcm::BaseQuery](#)**

- [BaseQuery](#) ()
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const [DictEntry](#) &inDictEntry, const std::string &inValue)
- bool [ValidDataSet](#) (const [DataSet](#) &dataSetToValid, const [DataSet](#) &dataSetReference) const

**Protected Member Functions inherited from [gdcm::Object](#)**

- void [Register](#) ()
- void [UnRegister](#) ()

**Protected Attributes inherited from [gdcm::BaseRootQuery](#)**

- std::string [mHelpDescription](#)
- [QueryImage](#) [mImage](#)
- [QueryPatient](#) [mPatient](#)
- [ERootType](#) [mRootType](#)
- [QuerySeries](#) [mSeries](#)
- [QueryStudy](#) [mStudy](#)

**Protected Attributes inherited from [gdcm::BaseQuery](#)**

- [DataSet](#) [mDataSet](#)
- std::string [mSopInstanceUID](#)

**10.199.1 Detailed Description**

[MoveStudyRootQuery](#).

contains: the class which will produce a dataset for C-MOVE with study root

**10.199.2 Constructor & Destructor Documentation****10.199.2.1 [MoveStudyRootQuery](#)()**

```
gdcm::MoveStudyRootQuery::MoveStudyRootQuery ( )
```

**10.199.3 Member Function Documentation****10.199.3.1 [GetAbstractSyntaxUID](#)()**

```
UIDs::TSName gdcm::MoveStudyRootQuery::GetAbstractSyntaxUID ( ) const [override], [virtual]
```

Implements [gdcm::BaseQuery](#).

### 10.199.3.2 GetTagListByLevel()

```
std::vector< Tag > gdcm::MoveStudyRootQuery::GetTagListByLevel (
    const EQueryLevel & inQueryLevel ) [override], [virtual]
```

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implements [gdcm::BaseRootQuery](#).

### 10.199.3.3 InitializeDataSet()

```
void gdcm::MoveStudyRootQuery::InitializeDataSet (
    const EQueryLevel & inQueryLevel ) [override], [virtual]
```

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmTk

Implements [gdcm::BaseRootQuery](#).

### 10.199.3.4 ValidateQuery()

```
bool gdcm::MoveStudyRootQuery::ValidateQuery (
    bool inStrict = true ) const [override], [virtual]
```

have to be able to ensure that 0x8,0x52 is set (which will be true if InitializeDataSet is called...) that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional) by default, this function checks to see if the query is for finding, which is more permissive than for moving. For moving, only the unique tags are allowed. 10 Jan 2011: adding in the 'strict' mode. according to the standard (at least, how I've read it), only tags for a particular level should be allowed in a particular query (ie, just series level tags in a series level query). However, it seems that dcm4chee doesn't share that interpretation. So, if 'inStrict' is false, then tags from the current level and all higher levels are now considered valid. So, if you're doing a non-strict series-level query, tags from the patient and study level can be passed along as well.

Implements [gdcm::BaseRootQuery](#).

## 10.199.4 Friends And Related Symbol Documentation

### 10.199.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

The documentation for this class was generated from the following file:

- [gdcmMoveStudyRootQuery.h](#)



## 10.200 gdcm::MrProtocol Class Reference

Class for [MrProtocol](#).

```
#include <gdcmMrProtocol.h>
```

### Classes

- struct [Slice](#)
- struct [SliceArray](#)
- struct [Vector3](#)

### Public Member Functions

- [MrProtocol](#) ()
- [~MrProtocol](#) ()
- bool [FindMrProtocolByName](#) (const char \*name) const
- const char \* [GetMrProtocolByName](#) (const char \*name) const
- bool [GetSliceArray](#) ([MrProtocol::SliceArray](#) &sa) const
- int [GetVersion](#) () const
- bool [Load](#) (const [ByteValue](#) \*bv, const char \*str, int version)
- void [Print](#) (std::ostream &os) const

### Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [MrProtocol](#) &d)

### 10.200.1 Detailed Description

Class for [MrProtocol](#).

#### Examples

[MrProtocol.cxx](#).

### 10.200.2 Constructor & Destructor Documentation

#### 10.200.2.1 MrProtocol()

```
gdcm::MrProtocol::MrProtocol ( )
```

#### 10.200.2.2 ~MrProtocol()

```
gdcm::MrProtocol::~~MrProtocol ( )
```

### 10.200.3 Member Function Documentation

#### 10.200.3.1 FindMrProtocolByName()

```
bool gdcM::MrProtocol::FindMrProtocolByName (
    const char * name ) const
```

#### 10.200.3.2 GetMrProtocolByName()

```
const char * gdcM::MrProtocol::GetMrProtocolByName (
    const char * name ) const
```

#### 10.200.3.3 GetSliceArray()

```
bool gdcM::MrProtocol::GetSliceArray (
    MrProtocol::SliceArray & sa ) const
```

#### 10.200.3.4 GetVersion()

```
int gdcM::MrProtocol::GetVersion ( ) const
```

#### 10.200.3.5 Load()

```
bool gdcM::MrProtocol::Load (
    const ByteValue * bv,
    const char * str,
    int version )
```

#### 10.200.3.6 Print()

```
void gdcM::MrProtocol::Print (
    std::ostream & os ) const
```

### 10.200.4 Friends And Related Symbol Documentation

#### 10.200.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const MrProtocol & d ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcMMrProtocol.h](#)

## 10.201 gdcmm::network::NActionRQ Class Reference

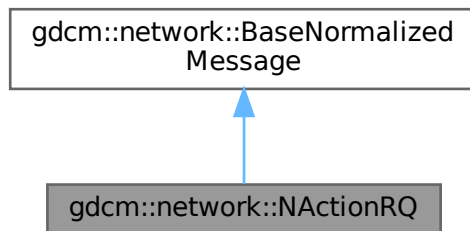
[NActionRQ](#).

```
#include <gdcmmNActionMessages.h>
```

Inheritance diagram for gdcmm::network::NActionRQ:



Collaboration diagram for gdcmm::network::NActionRQ:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseQuery](#) \*inQuery) override

### Public Member Functions inherited from [gdcmm::network::BaseNormalizedMessage](#)

- virtual `~BaseNormalizedMessage` ()=default

### 10.201.1 Detailed Description

[NActionRQ](#).

this file defines the messages for the NAction action

### 10.201.2 Member Function Documentation

#### 10.201.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcmm::network::NActionRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [override], [virtual]
```

Implements [gdcmm::network::BaseNormalizedMessage](#).

The documentation for this class was generated from the following file:

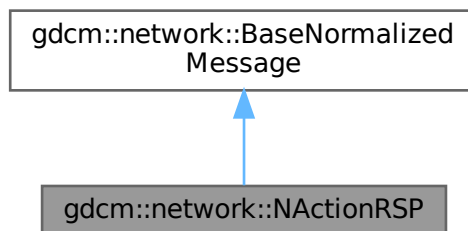
- [gdcmmNActionMessages.h](#)

## 10.202 gdcmm::network::NActionRSP Class Reference

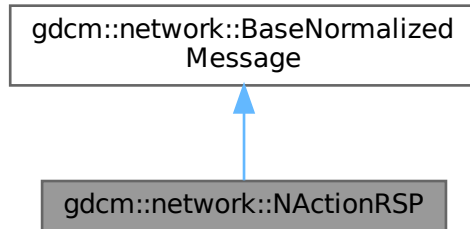
[NActionRSP](#) this file defines the messages for the NAction action.

```
#include <gdcmmNActionMessages.h>
```

Inheritance diagram for gdcmm::network::NActionRSP:



Collaboration diagram for gdcm::network::NActionRSP:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)`

### Public Member Functions inherited from [gdcm::network::BaseNormalizedMessage](#)

- `virtual ~BaseNormalizedMessage ()=default`
- `virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery)=0`

## 10.202.1 Detailed Description

[NActionRSP](#) this file defines the messages for the NAction action.

## 10.202.2 Member Function Documentation

### 10.202.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::NActionRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

The documentation for this class was generated from the following file:

- [gdcmNActionMessages.h](#)

## 10.203 gdcmm::network::NCreateRQ Class Reference

[NCreateRQ](#).

```
#include <gdcmmNCreateMessages.h>
```

Inheritance diagram for gdcmm::network::NCreateRQ:



Collaboration diagram for gdcmm::network::NCreateRQ:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseQuery](#) \*inQuery) override

### Public Member Functions inherited from [gdcmm::network::BaseNormalizedMessage](#)

- virtual `~BaseNormalizedMessage` ()=default

### 10.203.1 Detailed Description

[NCreateRQ](#).

this file defines the messages for the ncreate action

### 10.203.2 Member Function Documentation

#### 10.203.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::NCreateRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [override], [virtual]
```

Implements [gdcm::network::BaseNormalizedMessage](#).

The documentation for this class was generated from the following file:

- [gdcmNCreateMessages.h](#)

## 10.204 gdcm::network::NCreateRSP Class Reference

[NCreateRSP](#) this file defines the messages for the ncreate action.

```
#include <gdcmNCreateMessages.h>
```

Inheritance diagram for gdcm::network::NCreateRSP:



Collaboration diagram for `gdcm::network::NCreateRSP`:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet` (`const DataSet *inDataSet`)

### Public Member Functions inherited from [gdcm::network::BaseNormalizedMessage](#)

- `virtual ~BaseNormalizedMessage` ()=default
- `virtual std::vector< PresentationDataValue > ConstructPDV` (`const ULConnection &inConnection`, `const BaseQuery *inQuery`)=0

## 10.204.1 Detailed Description

[NCreateRSP](#) this file defines the messages for the ncreate action.

## 10.204.2 Member Function Documentation

### 10.204.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::NCreateRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

The documentation for this class was generated from the following file:

- [gdcmNCreateMessages.h](#)



## 10.205 gdcmm::network::NDeleteRQ Class Reference

[NDeleteRQ](#).

```
#include <gdcmmNDeleteMessages.h>
```

Inheritance diagram for gdcmm::network::NDeleteRQ:



Collaboration diagram for gdcmm::network::NDeleteRQ:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseQuery](#) \*inQuery) override

### Public Member Functions inherited from [gdcmm::network::BaseNormalizedMessage](#)

- virtual `~BaseNormalizedMessage` ()=default

### 10.205.1 Detailed Description

[NDeleteRQ](#).

this file defines the messages for the ndelete action

### 10.205.2 Member Function Documentation

#### 10.205.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcn::network::NDeleteRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [override], [virtual]
```

Implements [gdcn::network::BaseNormalizedMessage](#).

The documentation for this class was generated from the following file:

- [gdcnNDeleteMessages.h](#)

## 10.206 gdcn::network::NDeleteRSP Class Reference

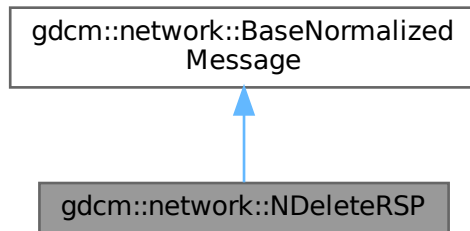
[NDeleteRSP](#) this file defines the messages for the ndelete action.

```
#include <gdcnNDeleteMessages.h>
```

Inheritance diagram for gdcn::network::NDeleteRSP:



Collaboration diagram for gdcm::network::NDeleteRSP:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)`

### Public Member Functions inherited from [gdcm::network::BaseNormalizedMessage](#)

- `virtual ~BaseNormalizedMessage ()=default`
- `virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery)=0`

## 10.206.1 Detailed Description

[NDeleteRSP](#) this file defines the messages for the ndelete action.

## 10.206.2 Member Function Documentation

### 10.206.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::NDeleteRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

The documentation for this class was generated from the following file:

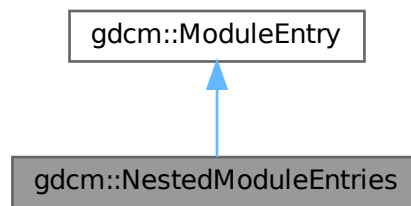
- [gdcmNDeleteMessages.h](#)

## 10.207 gdcm::NestedModuleEntries Class Reference

Class for representing a [NestedModuleEntries](#).

```
#include <gdcmNestedModuleEntries.h>
```

Inheritance diagram for gdcm::NestedModuleEntries:



Collaboration diagram for gdcm::NestedModuleEntries:



## Public Types

- typedef std::vector< [ModuleEntry](#) >::size\_type [SizeType](#)

## Public Types inherited from [gdcm::ModuleEntry](#)

- typedef std::string [Description](#)

## Public Member Functions

- [NestedModuleEntries](#) (const char \*name="", const char \*type="3", const char \*description="")
- void [AddModuleEntry](#) (const [ModuleEntry](#) &me)
- [ModuleEntry](#) & [GetModuleEntry](#) ([SizeType](#) idx)
- const [ModuleEntry](#) & [GetModuleEntry](#) ([SizeType](#) idx) const
- [SizeType](#) [GetNumberOfModuleEntries](#) ()

## Public Member Functions inherited from [gdcm::ModuleEntry](#)

- [ModuleEntry](#) (const char \*name="", const char \*type="3", const char \*description="")
- virtual [~ModuleEntry](#) ()=default
- const [Description](#) & [GetDescription](#) () const
- const char \* [GetName](#) () const
- const [Type](#) & [GetType](#) () const
- void [SetDescription](#) (const char \*d)
- void [SetName](#) (const char \*name)
- void [SetType](#) (const [Type](#) &type)

## Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [NestedModuleEntries](#) &\_val)

## Additional Inherited Members

## Protected Attributes inherited from [gdcm::ModuleEntry](#)

- [Type](#) [DataElementType](#)
- [Description](#) [DescriptionField](#)
- std::string [Name](#)

### 10.207.1 Detailed Description

Class for representing a [NestedModuleEntries](#).

#### Note

bla

#### See also

[ModuleEntry](#)

### 10.207.2 Member Typedef Documentation

#### 10.207.2.1 SizeType

```
typedef std::vector<ModuleEntry>::size_type gdcM::NestedModuleEntries::SizeType
```

### 10.207.3 Constructor & Destructor Documentation

#### 10.207.3.1 NestedModuleEntries()

```
gdcM::NestedModuleEntries::NestedModuleEntries (
    const char * name = "",
    const char * type = "3",
    const char * description = "" ) [inline]
```

### 10.207.4 Member Function Documentation

#### 10.207.4.1 AddModuleEntry()

```
void gdcM::NestedModuleEntries::AddModuleEntry (
    const ModuleEntry & me ) [inline]
```

#### 10.207.4.2 GetModuleEntry() [1/2]

```
ModuleEntry & gdcM::NestedModuleEntries::GetModuleEntry (
    SizeType idx ) [inline]
```

#### 10.207.4.3 GetModuleEntry() [2/2]

```
const ModuleEntry & gdcM::NestedModuleEntries::GetModuleEntry (
    SizeType idx ) const [inline]
```

#### 10.207.4.4 GetNumberOfModuleEntries()

```
SizeType gdcm::NestedModuleEntries::GetNumberOfModuleEntries ( ) [inline]
```

### 10.207.5 Friends And Related Symbol Documentation

#### 10.207.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const NestedModuleEntries & _val ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmNestedModuleEntries.h](#)

## 10.208 gdcm::network::NEventReportRQ Class Reference

[NEventReportRQ](#).

```
#include <gdcmNEventReportMessages.h>
```

Inheritance diagram for gdcm::network::NEventReportRQ:



Collaboration diagram for `gdcm::network::NEventReportRQ`:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (`const ULConnection &inConnection`, `const BaseQuery *inQuery`) override

### Public Member Functions inherited from [gdcm::network::BaseNormalizedMessage](#)

- `virtual ~BaseNormalizedMessage ()=default`

## 10.208.1 Detailed Description

[NEventReportRQ](#).

this file defines the messages for the neventreport action

## 10.208.2 Member Function Documentation

### 10.208.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::NEventReportRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [override], [virtual]
```

Implements [gdcm::network::BaseNormalizedMessage](#).

The documentation for this class was generated from the following file:

- [gdcmNEventReportMessages.h](#)

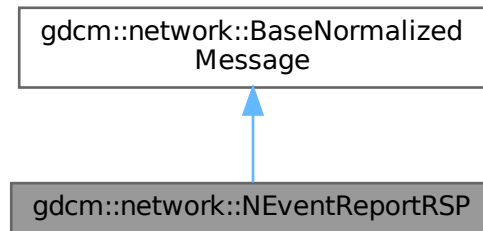


## 10.209 gdcm::network::NEventReportRSP Class Reference

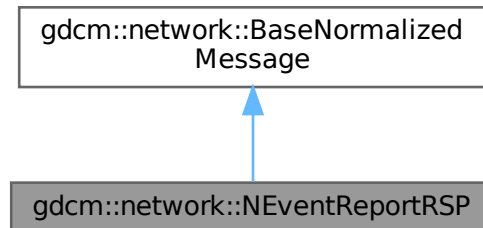
[NEventReportRSP](#) this file defines the messages for the neventreport action.

```
#include <gdcmNEventReportMessages.h>
```

Inheritance diagram for gdcm::network::NEventReportRSP:



Collaboration diagram for gdcm::network::NEventReportRSP:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet` (const [DataSet](#) \*inDataSet)

### Public Member Functions inherited from [gdcm::network::BaseNormalizedMessage](#)

- virtual `~BaseNormalizedMessage` ()=default
- virtual `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseQuery](#) \*inQuery)=0

### 10.209.1 Detailed Description

[NEventReportRSP](#) this file defines the messages for the neventreport action.

### 10.209.2 Member Function Documentation

#### 10.209.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcmm::network::NEventReportRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

The documentation for this class was generated from the following file:

- [gdcmmNEventReportMessages.h](#)

## 10.210 gdcmm::network::NGetRQ Class Reference

[NGetRQ](#).

```
#include <gdcmmNGetMessages.h>
```

Inheritance diagram for gdcmm::network::NGetRQ:



Collaboration diagram for gdcmm::network::NGetRQ:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &`inConnection`, const [BaseQuery](#) \*`inQuery`) override

### Public Member Functions inherited from [gdcmm::network::BaseNormalizedMessage](#)

- virtual `~BaseNormalizedMessage` ()=default

## 10.210.1 Detailed Description

[NGetRQ](#).

this file defines the messages for the nget action

## 10.210.2 Member Function Documentation

### 10.210.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcmm::network::NGetRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [override], [virtual]
```

Implements [gdcmm::network::BaseNormalizedMessage](#).

The documentation for this class was generated from the following file:

- [gdcmmNGetMessages.h](#)

## 10.211 gdcm::network::NGetRSP Class Reference

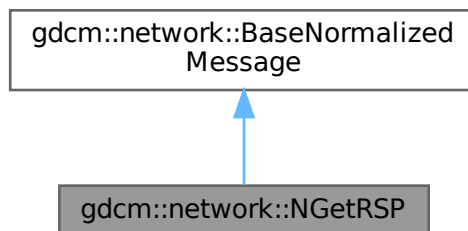
[NGetRSP](#) this file defines the messages for the nget action.

```
#include <gdcmNGetMessages.h>
```

Inheritance diagram for gdcm::network::NGetRSP:



Collaboration diagram for gdcm::network::NGetRSP:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet` (const [DataSet](#) \*inDataSet)

### Public Member Functions inherited from [gdcm::network::BaseNormalizedMessage](#)

- virtual `~BaseNormalizedMessage` ()=default
- virtual `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseQuery](#) \*inQuery)=0

### 10.211.1 Detailed Description

[NGetRSP](#) this file defines the messages for the nget action.

### 10.211.2 Member Function Documentation

#### 10.211.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::NGetRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

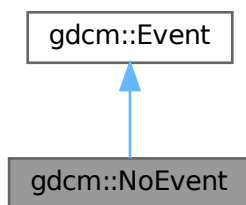
The documentation for this class was generated from the following file:

- [gdcmNGetMessages.h](#)

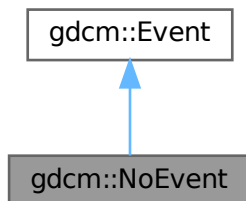
## 10.212 gdcm::NoEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::NoEvent:



Collaboration diagram for gdcm::NoEvent:



## Additional Inherited Members

### Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()
- [Event](#) (const [Event](#) &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) \*) const =0
- virtual const char \* [GetEventName](#) () const =0
- virtual [Event](#) \* [MakeObject](#) () const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

### 10.212.1 Detailed Description

Define some common GDCM events

The documentation for this class was generated from the following file:

- [gdcmEvent.h](#)

## 10.213 [gdcm::network::NormalizedMessageFactory](#) Class Reference

```
#include <gdcmNormalizedMessageFactory.h>
```

### Static Public Member Functions

- static std::vector< [PresentationDataValue](#) > [ConstructNAction](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) \*inQuery)
- static std::vector< [PresentationDataValue](#) > [ConstructNCreate](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) \*inQuery)
- static std::vector< [PresentationDataValue](#) > [ConstructNDelete](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) \*inQuery)
- static std::vector< [PresentationDataValue](#) > [ConstructNEventReport](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) \*inQuery)
- static std::vector< [PresentationDataValue](#) > [ConstructNGet](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) \*inQuery)
- static std::vector< [PresentationDataValue](#) > [ConstructNSet](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) \*inQuery)

### 10.213.1 Member Function Documentation

#### 10.213.1.1 [ConstructNAction\(\)](#)

```
static std::vector< PresentationDataValue > gdcm::network::NormalizedMessageFactory::ConstructNAction (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

### 10.213.1.2 ConstructNCreate()

```
static std::vector< PresentationDataValue > gdcm::network::NormalizedMessageFactory::Construct↵
NCreate (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

### 10.213.1.3 ConstructNDelete()

```
static std::vector< PresentationDataValue > gdcm::network::NormalizedMessageFactory::Construct↵
NDelete (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

### 10.213.1.4 ConstructNEventReport()

```
static std::vector< PresentationDataValue > gdcm::network::NormalizedMessageFactory::Construct↵
NEventReport (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

### 10.213.1.5 ConstructNGet()

```
static std::vector< PresentationDataValue > gdcm::network::NormalizedMessageFactory::Construct↵
NGet (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

### 10.213.1.6 ConstructNSet()

```
static std::vector< PresentationDataValue > gdcm::network::NormalizedMessageFactory::Construct↵
NSet (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

The documentation for this class was generated from the following file:

- [gdcmNormalizedMessageFactory.h](#)

## 10.214 gdcm::NormalizedNetworkFunctions Class Reference

Normalized Network Functions.

```
#include <gdcmNormalizedNetworkFunctions.h>
```

## Static Public Member Functions

- static [BaseQuery](#) \* [ConstructQuery](#) (const std::string &sopInstanceUID, const [DataSet](#) &queryds, [ENQueryType](#) queryType=[eCreateMMPS](#))
- static bool [NAction](#) (const char \*remote, uint16\_t portno, const [BaseQuery](#) \*query, std::vector< [DataSet](#) > &retDataSets, const char \*aetitle, const char \*call)
- static bool [NCreate](#) (const char \*remote, uint16\_t portno, [BaseQuery](#) \*query, std::vector< [DataSet](#) > &retDataSets, const char \*aetitle, const char \*call)
- static bool [NDelete](#) (const char \*remote, uint16\_t portno, const [BaseQuery](#) \*query, std::vector< [DataSet](#) > &retDataSets, const char \*aetitle, const char \*call)
- static bool [NEventReport](#) (const char \*remote, uint16\_t portno, const [BaseQuery](#) \*query, std::vector< [DataSet](#) > &retDataSets, const char \*aetitle, const char \*call)
- static bool [NGet](#) (const char \*remote, uint16\_t portno, const [BaseQuery](#) \*query, std::vector< [DataSet](#) > &retDataSets, const char \*aetitle, const char \*call)
- static bool [NSet](#) (const char \*remote, uint16\_t portno, const [BaseQuery](#) \*query, std::vector< [DataSet](#) > &retDataSets, const char \*aetitle, const char \*call)

### 10.214.1 Detailed Description

Normalized Network Functions.

These functions provide a generic API to the DICOM functions implemented in GDCM. Advanced users can use this code as a template for building their own versions of these functions (for instance, to provide progress bars or some other way of handling returned query information), but for most users, these functions should be sufficient to interface with a PACS to a local machine. Note that these functions are not contained within a static class or some other class-style interface, because multiple connections can be instantiated in the same program. The DICOM standard is much more function oriented rather than class oriented in this instance, so the design of this API reflects that functional approach. These functions implements the following SCU operations:

- N-EVENT-REPORT
- N-GET
- N-SET
- N-ACTION
- N-CREATE
- N-DELETE

### 10.214.2 Member Function Documentation

#### 10.214.2.1 ConstructQuery()

```
static BaseQuery * gdcmm::NormalizedNetworkFunctions::ConstructQuery (
    const std::string & sopInstanceUID,
    const DataSet & queryds,
    ENQueryType queryType = eCreateMMPS ) [static]
```



### 10.214.2.2 NAction()

```
static bool gdcmm::NormalizedNetworkFunctions::NAction (
    const char * remote,
    uint16_t portno,
    const BaseQuery * query,
    std::vector< DataSet > & retDataSets,
    const char * aetitle,
    const char * call ) [static]
```

### 10.214.2.3 NCreate()

```
static bool gdcmm::NormalizedNetworkFunctions::NCreate (
    const char * remote,
    uint16_t portno,
    BaseQuery * query,
    std::vector< DataSet > & retDataSets,
    const char * aetitle,
    const char * call ) [static]
```

### 10.214.2.4 NDelete()

```
static bool gdcmm::NormalizedNetworkFunctions::NDelete (
    const char * remote,
    uint16_t portno,
    const BaseQuery * query,
    std::vector< DataSet > & retDataSets,
    const char * aetitle,
    const char * call ) [static]
```

### 10.214.2.5 NEventReport()

```
static bool gdcmm::NormalizedNetworkFunctions::NEventReport (
    const char * remote,
    uint16_t portno,
    const BaseQuery * query,
    std::vector< DataSet > & retDataSets,
    const char * aetitle,
    const char * call ) [static]
```

### 10.214.2.6 NGet()

```
static bool gdcmm::NormalizedNetworkFunctions::NGet (
    const char * remote,
    uint16_t portno,
    const BaseQuery * query,
    std::vector< DataSet > & retDataSets,
    const char * aetitle,
    const char * call ) [static]
```

### 10.214.2.7 NSet()

```
static bool gdcmm::NormalizedNetworkFunctions::NSet (
    const char * remote,
    uint16_t portno,
    const BaseQuery * query,
    std::vector< DataSet > & retDataSets,
    const char * aetitle,
    const char * call ) [static]
```

The documentation for this class was generated from the following file:

- [gdcmmNormalizedNetworkFunctions.h](#)

## 10.215 gdcmm::network::NSetRQ Class Reference

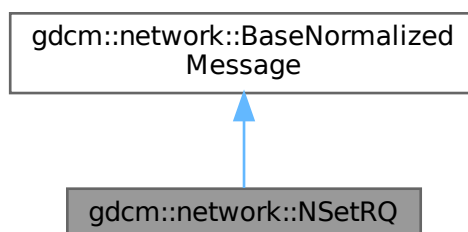
[NSetRQ](#).

```
#include <gdcmmNSetMessages.h>
```

Inheritance diagram for gdcmm::network::NSetRQ:



Collaboration diagram for gdcmm::network::NSetRQ:



**Public Member Functions**

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseQuery](#) \*inQuery) override

**Public Member Functions inherited from [gdcm::network::BaseNormalizedMessage](#)**

- virtual `~BaseNormalizedMessage` ()=default

**10.215.1 Detailed Description**

[NSetRQ](#).

this file defines the messages for the nset action

**10.215.2 Member Function Documentation****10.215.2.1 ConstructPDV()**

```
std::vector< PresentationDataValue > gdcm::network::NSetRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [override], [virtual]
```

Implements [gdcm::network::BaseNormalizedMessage](#).

The documentation for this class was generated from the following file:

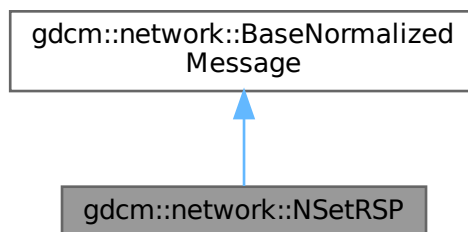
- [gdcmNSetMessages.h](#)

**10.216 gdcm::network::NSetRSP Class Reference**

[NSetRSP](#) this file defines the messages for the nset action.

```
#include <gdcmNSetMessages.h>
```

Inheritance diagram for `gdcm::network::NSetRSP`:



Collaboration diagram for `gdcm::network::NSetRSP`:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)`

### Public Member Functions inherited from [gdcm::network::BaseNormalizedMessage](#)

- `virtual ~BaseNormalizedMessage ()=default`
- `virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery)=0`

## 10.216.1 Detailed Description

[NSetRSP](#) this file defines the messages for the nset action.

## 10.216.2 Member Function Documentation

### 10.216.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::NSetRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

The documentation for this class was generated from the following file:

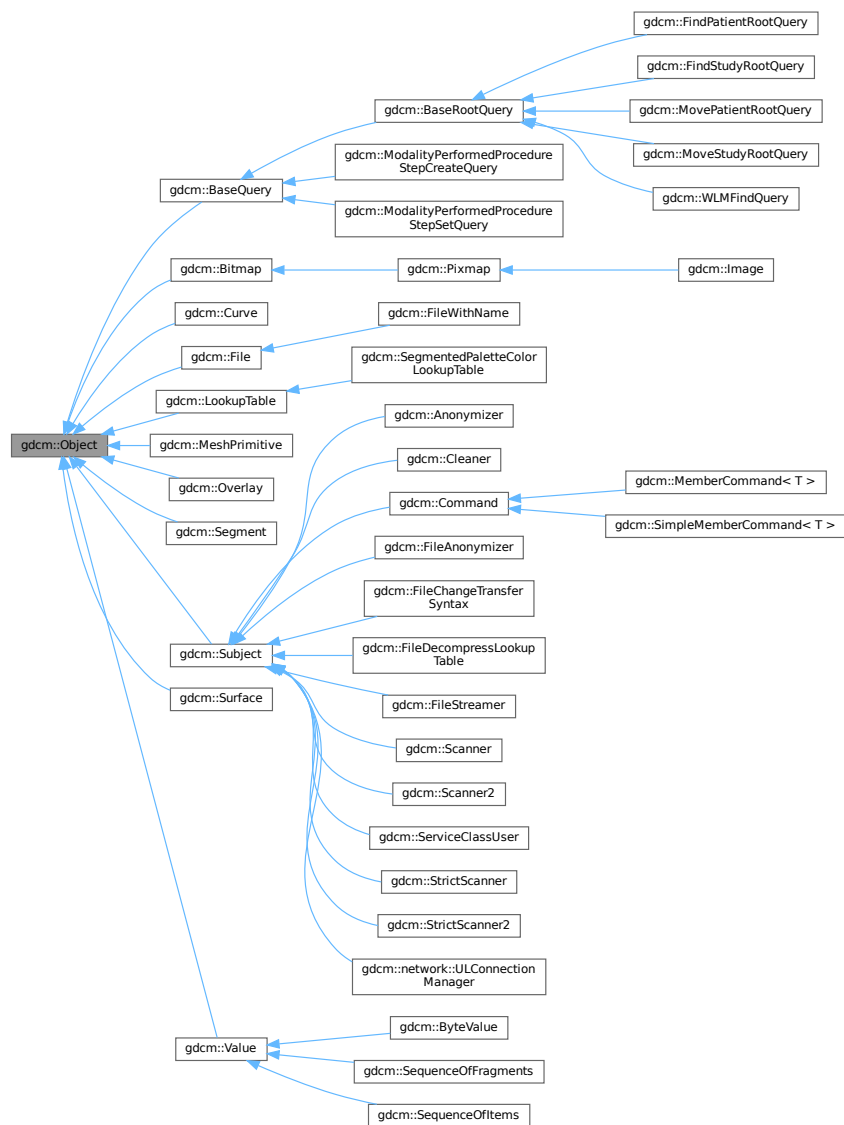
- [gdcmNSetMessages.h](#)

## 10.217 gdcm::Object Class Reference

Object.

```
#include <gdcmObject.h>
```

Inheritance diagram for gdcm::Object:



### Public Member Functions

- [Object](#) ()
- [Object](#) (const [Object](#) &)

*Special requirement for copy/cstor, assignment operator.*

- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

### Protected Member Functions

- void [Register](#) ()
- void [UnRegister](#) ()

### Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [Object](#) &obj)
- template<class ObjectType >  
class [SmartPointer](#)

## 10.217.1 Detailed Description

[Object](#).

### Note

main superclass for object that want to use [SmartPointer](#) invasive ref counting system

### See also

[SmartPointer](#)

## 10.217.2 Constructor & Destructor Documentation

### 10.217.2.1 [Object\(\)](#) [1/2]

```
gdcm::Object::Object ( ) [inline]
```

### 10.217.2.2 [~Object\(\)](#)

```
virtual gdcm::Object::~~Object ( ) [inline], [virtual]
```

### 10.217.2.3 [Object\(\)](#) [2/2]

```
gdcm::Object::Object (
    const Object & ) [inline]
```

Special requirement for copy/cstor, assignment operator.

## 10.217.3 Member Function Documentation

### 10.217.3.1 operator=()

```
void gdcm::Object::operator= (
    const Object & ) [inline]
```

### 10.217.3.2 Print()

```
virtual void gdcm::Object::Print (
    std::ostream & ) const [inline], [virtual]
```

Reimplemented in [gdcm::Bitmap](#), [gdcm::Curve](#), [gdcm::LookupTable](#), [gdcm::Overlay](#), [gdcm::Pixmap](#), [gdcm::SegmentedPaletteColorLookupTable](#), [gdcm::ByteValue](#), [gdcm::SequenceOfFragments](#), [gdcm::SequenceOfItems](#), [gdcm::Image](#), [gdcm::Scanner](#), [gdcm::Scanner2](#), [gdcm::StrictScanner](#), [gdcm::StrictScanner2](#), and [gdcm::BaseQuery](#).

#### Examples

[ReadAndDumpDICOMDIR.cxx](#).

### 10.217.3.3 Register()

```
void gdcm::Object::Register ( ) [inline], [protected]
```

### 10.217.3.4 UnRegister()

```
void gdcm::Object::UnRegister ( ) [inline], [protected]
```

## 10.217.4 Friends And Related Symbol Documentation

### 10.217.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & os,
    const Object & obj ) [friend]
```

### 10.217.4.2 SmartPointer

```
template<class ObjectType >
friend class SmartPointer [friend]
```

The documentation for this class was generated from the following file:

- [gdcmObject.h](#)

## 10.218 gdcmm::OpenSSLCryptoFactory Class Reference

```
#include <gdcmmOpenSSLCryptoFactory.h>
```

Inheritance diagram for gdcmm::OpenSSLCryptoFactory:



Collaboration diagram for gdcmm::OpenSSLCryptoFactory:



### Public Member Functions

- [OpenSSLCryptoFactory](#) ([CryptoLib](#) id)
- [CryptographicMessageSyntax](#) \* [CreateCMSProvider](#) ()

### Protected Member Functions

- void [InitOpenSSL](#) ()

### Protected Member Functions inherited from [gdcmm::CryptoFactory](#)

- [CryptoFactory](#) ()=default
- [CryptoFactory](#) ([CryptoLib](#) id)
- [~CryptoFactory](#) ()=default



## Additional Inherited Members

## Public Types inherited from [gdcm::CryptoFactory](#)

- enum [CryptoLib](#) {  
    [DEFAULT](#) = 0 ,  
    [OPENSSL](#) = 1 ,  
    [CAPI](#) = 2 ,  
    [OPENSSL7](#) = 3 }

## Static Public Member Functions inherited from [gdcm::CryptoFactory](#)

- static [CryptoFactory](#) \* [GetFactoryInstance](#) ([CryptoLib](#) id=[DEFAULT](#))

## 10.218.1 Constructor & Destructor Documentation

### 10.218.1.1 [OpenSSLCryptoFactory\(\)](#)

```
gdcm::OpenSSLCryptoFactory::OpenSSLCryptoFactory (  
    CryptoLib id )    [inline]
```

References [gdcmDebugMacro](#).

## 10.218.2 Member Function Documentation

### 10.218.2.1 [CreateCMSProvider\(\)](#)

```
CryptographicMessageSyntax * gdcm::OpenSSLCryptoFactory::CreateCMSProvider ( )    [inline], [virtual]
```

Implements [gdcm::CryptoFactory](#).

### 10.218.2.2 [InitOpenSSL\(\)](#)

```
void gdcm::OpenSSLCryptoFactory::InitOpenSSL ( )    [protected]
```

The documentation for this class was generated from the following file:

- [gdcmOpenSSLCryptoFactory.h](#)

## 10.219 gdcmm::OpenSSLCryptographicMessageSyntax Class Reference

```
#include <gdcmmOpenSSLCryptographicMessageSyntax.h>
```

Inheritance diagram for gdcmm::OpenSSLCryptographicMessageSyntax:



Collaboration diagram for gdcmm::OpenSSLCryptographicMessageSyntax:



### Public Member Functions

- [OpenSSLCryptographicMessageSyntax](#) ()
- [~OpenSSLCryptographicMessageSyntax](#) ()
- bool [Decrypt](#) (char \*output, size\_t &outlen, const char \*array, size\_t len) const  
*decrypt content from a PKCS#7 envelopedData structure*
- bool [Encrypt](#) (char \*output, size\_t &outlen, const char \*array, size\_t len) const  
*create a CMS envelopedData structure*
- [CipherTypes](#) [GetCipherType](#) () const
- bool [ParseCertificateFile](#) (const char \*filename)
- bool [ParseKeyFile](#) (const char \*filename)
- void [SetCipherType](#) ([CipherTypes](#) type)
- bool [SetPassword](#) (const char \*pass, size\_t passLen)

## Public Member Functions inherited from `gdcM::CryptographicMessageSyntax`

- `CryptographicMessageSyntax` ()=default
- `CryptographicMessageSyntax` (const `CryptographicMessageSyntax` &)=delete
- virtual `~CryptographicMessageSyntax` ()=default
- void `operator=` (const `CryptographicMessageSyntax` &)=delete

## Additional Inherited Members

## Public Types inherited from `gdcM::CryptographicMessageSyntax`

- enum `CipherTypes` {  
    `DES3_CIPHER` ,  
    `AES128_CIPHER` ,  
    `AES192_CIPHER` ,  
    `AES256_CIPHER` }

### 10.219.1 Constructor & Destructor Documentation

#### 10.219.1.1 `OpenSSLCryptographicMessageSyntax()`

```
gdcM::OpenSSLCryptographicMessageSyntax::OpenSSLCryptographicMessageSyntax ( )
```

#### 10.219.1.2 `~OpenSSLCryptographicMessageSyntax()`

```
gdcM::OpenSSLCryptographicMessageSyntax::~~OpenSSLCryptographicMessageSyntax ( )
```

### 10.219.2 Member Function Documentation

#### 10.219.2.1 `Decrypt()`

```
bool gdcM::OpenSSLCryptographicMessageSyntax::Decrypt (
    char * output,
    size_t & outlen,
    const char * array,
    size_t len ) const [virtual]
```

decrypt content from a PKCS#7 envelopedData structure

Implements `gdcM::CryptographicMessageSyntax`.

### 10.219.2.2 Encrypt()

```
bool gdcM::OpenSSLCryptographicMessageSyntax::Encrypt (
    char * output,
    size_t & outlen,
    const char * array,
    size_t len ) const [virtual]
```

create a CMS envelopedData structure

Implements [gdcM::CryptographicMessageSyntax](#).

### 10.219.2.3 GetCipherType()

```
CipherTypes gdcM::OpenSSLCryptographicMessageSyntax::GetCipherType ( ) const [virtual]
```

Implements [gdcM::CryptographicMessageSyntax](#).

### 10.219.2.4 ParseCertificateFile()

```
bool gdcM::OpenSSLCryptographicMessageSyntax::ParseCertificateFile (
    const char * filename ) [virtual]
```

Implements [gdcM::CryptographicMessageSyntax](#).

### 10.219.2.5 ParseKeyFile()

```
bool gdcM::OpenSSLCryptographicMessageSyntax::ParseKeyFile (
    const char * filename ) [virtual]
```

Implements [gdcM::CryptographicMessageSyntax](#).

### 10.219.2.6 SetCipherType()

```
void gdcM::OpenSSLCryptographicMessageSyntax::SetCipherType (
    CipherTypes type ) [virtual]
```

Set Cipher [Type](#). Default is: AES256\_CIPHER

Implements [gdcM::CryptographicMessageSyntax](#).

### 10.219.2.7 SetPassword()

```
bool gdcm::OpenSSLCryptographicMessageSyntax::SetPassword (
    const char * pass,
    size_t passLen ) [virtual]
```

Implements [gdcm::CryptographicMessageSyntax](#).

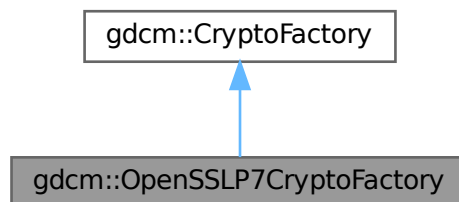
The documentation for this class was generated from the following file:

- [gdcmOpenSSLCryptographicMessageSyntax.h](#)

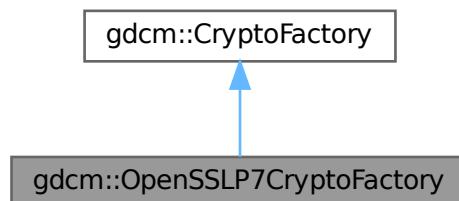
## 10.220 gdcm::OpenSSLP7CryptoFactory Class Reference

```
#include <gdcmOpenSSLP7CryptoFactory.h>
```

Inheritance diagram for gdcm::OpenSSLP7CryptoFactory:



Collaboration diagram for gdcm::OpenSSLP7CryptoFactory:



**Public Member Functions**

- [OpenSSLP7CryptoFactory](#) ([CryptoLib](#) id)
- [CryptographicMessageSyntax](#) \* [CreateCMSProvider](#) ()

**Additional Inherited Members****Public Types inherited from [gdcM::CryptoFactory](#)**

- enum [CryptoLib](#) {  
[DEFAULT](#) = 0 ,  
[OPENSSL](#) = 1 ,  
[CAPI](#) = 2 ,  
[OPENSSLP7](#) = 3 }

**Static Public Member Functions inherited from [gdcM::CryptoFactory](#)**

- static [CryptoFactory](#) \* [GetFactoryInstance](#) ([CryptoLib](#) id=DEFAULT)

**Protected Member Functions inherited from [gdcM::CryptoFactory](#)**

- [CryptoFactory](#) ()=default
- [CryptoFactory](#) ([CryptoLib](#) id)
- [~CryptoFactory](#) ()=default

**10.220.1 Constructor & Destructor Documentation****10.220.1.1 OpenSSLP7CryptoFactory()**

```
gdcM::OpenSSLP7CryptoFactory::OpenSSLP7CryptoFactory (
    CryptoLib id ) [inline]
```

References [gdcMDebugMacro](#).

**10.220.2 Member Function Documentation****10.220.2.1 CreateCMSProvider()**

```
CryptographicMessageSyntax * gdcM::OpenSSLP7CryptoFactory::CreateCMSProvider ( ) [inline], [virtual]
```

Implements [gdcM::CryptoFactory](#).

The documentation for this class was generated from the following file:

- [gdcMOpenSSLP7CryptoFactory.h](#)

## 10.221 gdcmm::OpenSSL7CryptographicMessageSyntax Class Reference

```
#include <gdcmmOpenSSL7CryptographicMessageSyntax.h>
```

Inheritance diagram for gdcmm::OpenSSL7CryptographicMessageSyntax:



Collaboration diagram for gdcmm::OpenSSL7CryptographicMessageSyntax:



### Public Member Functions

- [OpenSSL7CryptographicMessageSyntax](#) ()
- [~OpenSSL7CryptographicMessageSyntax](#) ()
- bool [Decrypt](#) (char \*output, size\_t &outlen, const char \*array, size\_t len) const  
*decrypt content from a PKCS#7 envelopedData structure*
- bool [Encrypt](#) (char \*output, size\_t &outlen, const char \*array, size\_t len) const  
*create a PKCS#7 envelopedData structure*
- [CipherTypes](#) [GetCipherType](#) () const
- bool [ParseCertificateFile](#) (const char \*filename)
- bool [ParseKeyFile](#) (const char \*filename)
- void [SetCipherType](#) ([CipherTypes](#) type)
- bool [SetPassword](#) (const char \*, size\_t)

## Public Member Functions inherited from [gdcm::CryptographicMessageSyntax](#)

- [CryptographicMessageSyntax](#) ()=default
- [CryptographicMessageSyntax](#) (const [CryptographicMessageSyntax](#) &)=delete
- virtual [~CryptographicMessageSyntax](#) ()=default
- void [operator=](#) (const [CryptographicMessageSyntax](#) &)=delete

## Additional Inherited Members

## Public Types inherited from [gdcm::CryptographicMessageSyntax](#)

- enum [CipherTypes](#) {  
[DES3\\_CIPHER](#) ,  
[AES128\\_CIPHER](#) ,  
[AES192\\_CIPHER](#) ,  
[AES256\\_CIPHER](#) }

### 10.221.1 Detailed Description

Class for [CryptographicMessageSyntax](#) encryption. This is just a simple wrapper around openssl PKCS7\_encrypt functionalities

See online documentation [http://www.openssl.org/docs/crypto/PKCS7\\_encrypt.html](http://www.openssl.org/docs/crypto/PKCS7_encrypt.html)

### 10.221.2 Constructor & Destructor Documentation

#### 10.221.2.1 OpenSSLP7CryptographicMessageSyntax()

```
gdcm::OpenSSLP7CryptographicMessageSyntax::OpenSSLP7CryptographicMessageSyntax ( )
```

#### 10.221.2.2 ~OpenSSLP7CryptographicMessageSyntax()

```
gdcm::OpenSSLP7CryptographicMessageSyntax::~~OpenSSLP7CryptographicMessageSyntax ( )
```

### 10.221.3 Member Function Documentation

#### 10.221.3.1 Decrypt()

```
bool gdcm::OpenSSLP7CryptographicMessageSyntax::Decrypt (
    char * output,
    size_t & outlen,
    const char * array,
    size_t len ) const [virtual]
```

decrypt content from a PKCS#7 envelopedData structure

Implements [gdcm::CryptographicMessageSyntax](#).



### 10.221.3.2 Encrypt()

```
bool gdcmm::OpenSSLP7CryptographicMessageSyntax::Encrypt (
    char * output,
    size_t & outlen,
    const char * array,
    size_t len ) const [virtual]
```

create a PKCS#7 envelopedData structure

Implements [gdcmm::CryptographicMessageSyntax](#).

### 10.221.3.3 GetCipherType()

```
CipherTypes gdcmm::OpenSSLP7CryptographicMessageSyntax::GetCipherType ( ) const [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

### 10.221.3.4 ParseCertificateFile()

```
bool gdcmm::OpenSSLP7CryptographicMessageSyntax::ParseCertificateFile (
    const char * filename ) [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

### 10.221.3.5 ParseKeyFile()

```
bool gdcmm::OpenSSLP7CryptographicMessageSyntax::ParseKeyFile (
    const char * filename ) [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

### 10.221.3.6 SetCipherType()

```
void gdcmm::OpenSSLP7CryptographicMessageSyntax::SetCipherType (
    CipherTypes type ) [virtual]
```

Set Cipher [Type](#). Default is: AES256\_CIPHER

Implements [gdcmm::CryptographicMessageSyntax](#).

### 10.221.3.7 SetPassword()

```
bool gdcM::OpenSSL7CryptographicMessageSyntax::SetPassword (
    const char * ,
    size_t ) [inline], [virtual]
```

Implements [gdcM::CryptographicMessageSyntax](#).

References [gdcMWarningMacro](#).

The documentation for this class was generated from the following file:

- [gdcMOpenSSL7CryptographicMessageSyntax.h](#)

## 10.222 gdcM::Orientation Class Reference

class to handle [Orientation](#)

```
#include <gdcMOrientation.h>
```

### Public Types

- enum [OrientationType](#) {  
[UNKNOWN](#) ,  
[AXIAL](#) ,  
[CORONAL](#) ,  
[SAGITTAL](#) ,  
[OBLIQUE](#) }

### Public Member Functions

- [Orientation](#) ()
- [~Orientation](#) ()=default
- void [Print](#) (std::ostream &) const  
*Print.*

### Static Public Member Functions

- static const char \* [GetLabel](#) ([OrientationType](#) type)  
*Return the label of an [Orientation](#).*
- static double [GetObliquityThresholdCosineValue](#) ()
- static [OrientationType](#) [GetType](#) (const double dircos[6])
- static void [SetObliquityThresholdCosineValue](#) (double val)  
*ObliquityThresholdCosineValue stuff.*

### Static Protected Member Functions

- static char [GetMajorAxisFromPatientRelativeDirectionCosine](#) (double x, double y, double z)

### Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [Orientation](#) &o)

## 10.222.1 Detailed Description

class to handle [Orientation](#)

## 10.222.2 Member Enumeration Documentation

### 10.222.2.1 OrientationType

```
enum gdcm::Orientation::OrientationType
```

#### Enumerator

UNKNOWN	
AXIAL	
CORONAL	
SAGITTAL	
OBLIQUE	

## 10.222.3 Constructor & Destructor Documentation

### 10.222.3.1 Orientation()

```
gdcm::Orientation::Orientation ( )
```

### 10.222.3.2 ~Orientation()

```
gdcm::Orientation::~~Orientation ( ) [default]
```

## 10.222.4 Member Function Documentation

### 10.222.4.1 GetLabel()

```
static const char * gdcm::Orientation::GetLabel (
    OrientationType type ) [static]
```

Return the label of an [Orientation](#).

#### Examples

[FixOrientation.cxx](#).

### 10.222.4.2 GetMajorAxisFromPatientRelativeDirectionCosine()

```
static char gdcm::Orientation::GetMajorAxisFromPatientRelativeDirectionCosine (
    double x,
    double y,
    double z ) [static], [protected]
```

### 10.222.4.3 GetObliquityThresholdCosineValue()

```
static double gdcm::Orientation::GetObliquityThresholdCosineValue ( ) [static]
```

### 10.222.4.4 GetType()

```
static OrientationType gdcm::Orientation::GetType (
    const double dircos[6] ) [static]
```

Return the type of orientation from a direction cosines Input is an array of 6 double

#### Examples

[FixOrientation.cxx](#).

### 10.222.4.5 Print()

```
void gdcm::Orientation::Print (
    std::ostream & ) const
```

Print.

#### 10.222.4.6 SetObliquityThresholdCosineValue()

```
static void gdcmm::Orientation::SetObliquityThresholdCosineValue (
    double val ) [static]
```

ObliquityThresholdCosineValue stuff.

### 10.222.5 Friends And Related Symbol Documentation

#### 10.222.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const Orientation & o ) [friend]
```

The documentation for this class was generated from the following file:

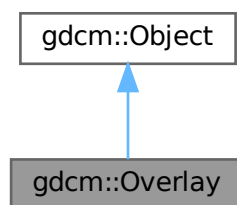
- [gdcmmOrientation.h](#)

## 10.223 gdcmm::Overlay Class Reference

[Overlay](#) class.

```
#include <gdcmmOverlay.h>
```

Inheritance diagram for gdcmm::Overlay:



Collaboration diagram for `gdcm::Overlay`:



### Public Types

- enum `OverlayType` {  
     `Invalid` = 0 ,  
     `Graphics` = 1 ,  
     `ROI` = 2 }

### Public Member Functions

- `Overlay ()`
- `Overlay (Overlay const &ov)`
- `~Overlay ()` override
- void `Decompress (std::ostream &os) const`  
     *Decode the internal OverlayData (packed bits) into unpacked representation.*
- unsigned short `GetBitPosition () const`  
     *return bit position*
- unsigned short `GetBitsAllocated () const`  
     *return bits allocated*
- unsigned short `GetColumns () const`  
     *get columns*
- const char \* `GetDescription () const`  
     *get description*
- unsigned short `GetGroup () const`  
     *Get Group number.*
- const signed short \* `GetOrigin () const`  
     *get origin*
- const `ByteValue` & `GetOverlayData () const`
- unsigned short `GetRows () const`  
     *get rows*
- const char \* `GetType () const`  
     *get type*
- `OverlayType` `GetTypeAsEnum () const`

- bool [GetUnpackBuffer](#) (char \*buffer, size\_t len) const
- size\_t [GetUnpackBufferLength](#) () const
- bool [GrabOverlayFromPixelData](#) ([DataSet](#) const &ds)
- bool [IsEmpty](#) () const
  - Return whether or not the [Overlay](#) is empty:*
- bool [IsInPixelData](#) () const
  - return if the [Overlay](#) is stored in the pixel data or not*
- void [IsInPixelData](#) (bool b)
  - Set whether or no the OverlayData is in the Pixel Data:*
- bool [IsZero](#) () const
  - return true if all bits are set to 0*
- [Overlay](#) & [operator=](#) ([Overlay](#) const &ov)
- void [Print](#) (std::ostream &) const override
  - Print.*
- void [SetBitPosition](#) (unsigned short bitposition)
  - set bit position*
- void [SetBitsAllocated](#) (unsigned short bitsallocated)
  - set bits allocated*
- void [SetColumns](#) (unsigned short columns)
  - set columns*
- void [SetDescription](#) (const char \*description)
  - set description*
- void [SetFrameOrigin](#) (unsigned short frameorigin)
  - set frame origin*
- void [SetGroup](#) (unsigned short group)
  - Set Group number.*
- void [SetNumberOfFrames](#) (unsigned int numberofframes)
  - set number of frames*
- void [SetOrigin](#) (const signed short origin[2])
  - set origin*
- void [SetOverlay](#) (const char \*array, size\_t length)
  - set overlay from byte array + length*
- void [SetRows](#) (unsigned short rows)
  - set rows*
- void [SetType](#) (const char \*type)
  - set type*
- void [Update](#) (const [DataElement](#) &de)
  - Update overlay from data element de:*

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
  - Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

### Static Public Member Functions

- static const char \* [GetOverlayTypeAsString](#) ([OverlayType](#) ot)
- static [OverlayType](#) [GetOverlayTypeFromString](#) (const char \*)

### Additional Inherited Members

### Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

## 10.223.1 Detailed Description

[Overlay](#) class.

#### Note

see [AreOverlaysInPixelData](#)

**Todo** Is there actually any way to recognize an overlay ? On images with multiple overlay I do not see any way to differentiate them (other than the group tag).

Example:

## 10.223.2 Member Enumeration Documentation

### 10.223.2.1 OverlayType

```
enum gdcm::Overlay::OverlayType
```

#### Enumerator

Invalid	
Graphics	
ROI	

## 10.223.3 Constructor & Destructor Documentation

### 10.223.3.1 Overlay() [1/2]

```
gdcm::Overlay::Overlay ( )
```



### 10.223.3.2 ~Overlay()

```
gdcm::Overlay::~~Overlay ( ) [override]
```

### 10.223.3.3 Overlay() [2/2]

```
gdcm::Overlay::Overlay (
    Overlay const & ov )
```

## 10.223.4 Member Function Documentation

### 10.223.4.1 Decompress()

```
void gdcm::Overlay::Decompress (
    std::ostream & os ) const
```

Decode the internal OverlayData (packed bits) into unpacked representation.

### 10.223.4.2 GetBitPosition()

```
unsigned short gdcm::Overlay::GetBitPosition ( ) const
```

return bit position

### 10.223.4.3 GetBitsAllocated()

```
unsigned short gdcm::Overlay::GetBitsAllocated ( ) const
```

return bits allocated

### 10.223.4.4 GetColumns()

```
unsigned short gdcm::Overlay::GetColumns ( ) const
```

get columns

### 10.223.4.5 GetDescription()

```
const char * gdcm::Overlay::GetDescription ( ) const
```

get description

#### 10.223.4.6 GetGroup()

```
unsigned short gdcm::Overlay::GetGroup ( ) const
```

Get Group number.

#### 10.223.4.7 GetOrigin()

```
const signed short * gdcm::Overlay::GetOrigin ( ) const
```

get origin

#### 10.223.4.8 GetOverlayData()

```
const ByteValue & gdcm::Overlay::GetOverlayData ( ) const
```

Return the [Overlay](#) Data as [ByteValue](#): Not thread safe

#### 10.223.4.9 GetOverlayTypeAsString()

```
static const char * gdcm::Overlay::GetOverlayTypeAsString (
    OverlayType ot ) [static]
```

#### 10.223.4.10 GetOverlayTypeFromString()

```
static OverlayType gdcm::Overlay::GetOverlayTypeFromString (
    const char * ) [static]
```

#### 10.223.4.11 GetRows()

```
unsigned short gdcm::Overlay::GetRows ( ) const
```

get rows

#### 10.223.4.12 GetType()

```
const char * gdcm::Overlay::GetType ( ) const
```

get type

#### 10.223.4.13 GetTypeAsEnum()

```
OverlayType gdcm::Overlay::GetTypeAsEnum ( ) const
```

#### 10.223.4.14 GetUnpackBuffer()

```
bool gdcm::Overlay::GetUnpackBuffer (
    char * buffer,
    size_t len ) const
```

Retrieve the unpack buffer for [Overlay](#). This is an error if the size is below [GetUnpackBufferLength\(\)](#)

#### 10.223.4.15 GetUnpackBufferLength()

```
size_t gdcm::Overlay::GetUnpackBufferLength ( ) const
```

Retrieve the size of the buffer needed to hold the [Overlay](#) as specified by Col & Row parameters

#### 10.223.4.16 GrabOverlayFromPixelData()

```
bool gdcm::Overlay::GrabOverlayFromPixelData (
    DataSet const & ds )
```

#### 10.223.4.17 IsEmpty()

```
bool gdcm::Overlay::IsEmpty ( ) const
```

Return whether or not the [Overlay](#) is empty:

#### 10.223.4.18 IsInPixelData() [1/2]

```
bool gdcm::Overlay::IsInPixelData ( ) const
```

return if the [Overlay](#) is stored in the pixel data or not

#### 10.223.4.19 IsInPixelData() [2/2]

```
void gdcm::Overlay::IsInPixelData (
    bool b )
```

Set whether or no the OverlayData is in the Pixel Data:

**10.223.4.20 IsZero()**

```
bool gdcM::Overlay::IsZero ( ) const
```

return true if all bits are set to 0

**10.223.4.21 operator=()**

```
Overlay & gdcM::Overlay::operator= (
    Overlay const & ov )
```

**10.223.4.22 Print()**

```
void gdcM::Overlay::Print (
    std::ostream & ) const [override], [virtual]
```

Print.

Reimplemented from [gdcM::Object](#).

**10.223.4.23 SetBitPosition()**

```
void gdcM::Overlay::SetBitPosition (
    unsigned short bitposition )
```

set bit position

**10.223.4.24 SetBitsAllocated()**

```
void gdcM::Overlay::SetBitsAllocated (
    unsigned short bitsallocated )
```

set bits allocated

**10.223.4.25 SetColumns()**

```
void gdcM::Overlay::SetColumns (
    unsigned short columns )
```

set columns

**10.223.4.26 SetDescription()**

```
void gdcm::Overlay::SetDescription (
    const char * description )
```

set description

**10.223.4.27 SetFrameOrigin()**

```
void gdcm::Overlay::SetFrameOrigin (
    unsigned short frameorigin )
```

set frame origin

**10.223.4.28 SetGroup()**

```
void gdcm::Overlay::SetGroup (
    unsigned short group )
```

Set Group number.

**10.223.4.29 SetNumberOfFrames()**

```
void gdcm::Overlay::SetNumberOfFrames (
    unsigned int numberofframes )
```

set number of frames

**10.223.4.30 SetOrigin()**

```
void gdcm::Overlay::SetOrigin (
    const signed short origin[2] )
```

set origin

**10.223.4.31 SetOverlay()**

```
void gdcm::Overlay::SetOverlay (
    const char * array,
    size_t length )
```

set overlay from byte array + length

#### 10.223.4.32 SetRows()

```
void gdcM::Overlay::SetRows (
    unsigned short rows )
```

set rows

#### 10.223.4.33 SetType()

```
void gdcM::Overlay::SetType (
    const char * type )
```

set type

#### 10.223.4.34 Update()

```
void gdcM::Overlay::Update (
    const DataElement & de )
```

Update overlay from data element de:

The documentation for this class was generated from the following file:

- [gdcMOverlay.h](#)

## 10.224 gdcM::ParseException Class Reference

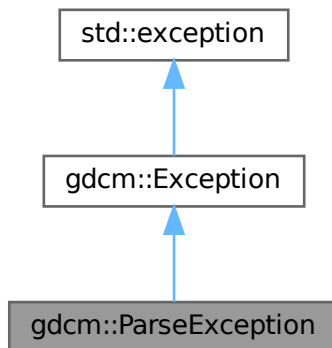
[ParseException](#) Standard exception handling object.

```
#include <gdcMParseException.h>
```

Inheritance diagram for gdcM::ParseException:



Collaboration diagram for gdcm::ParseException:



### Public Member Functions

- [ParseException](#) ()=default
- [ParseException](#) (const [ParseException](#) &orig)
- [~ParseException](#) () override throw ()
- const [DataElement](#) & [GetLastElement](#) () const
- [ParseException](#) & [operator=](#) (const [ParseException](#) &orig)
- void [SetLastElement](#) ([DataElement](#) &de)

### Public Member Functions inherited from [gdcm::Exception](#)

- [Exception](#) (const char \*desc="None", const char \*file=\_\_FILE\_\_, unsigned int lineNumber=\_\_LINE\_\_, const char \*func="")
- [~Exception](#) () override throw ()
- const char \* [GetDescription](#) () const  
*Return the Description.*
- const char \* [what](#) () const override throw ()  
*what implementation*

## 10.224.1 Detailed Description

[ParseException](#) Standard exception handling object.

## 10.224.2 Constructor & Destructor Documentation

### 10.224.2.1 ParseException() [1/2]

```
gdcm::ParseException::ParseException ( ) [default]
```

### 10.224.2.2 ~ParseException()

```
gdcm::ParseException::~~ParseException ( ) throw ( ) [inline], [override]
```

### 10.224.2.3 ParseException() [2/2]

```
gdcm::ParseException::ParseException (
    const ParseException & orig ) [inline]
```

## 10.224.3 Member Function Documentation

### 10.224.3.1 GetLastElement()

```
const DataElement & gdcm::ParseException::GetLastElement ( ) const [inline]
```

### 10.224.3.2 operator=()

```
ParseException & gdcm::ParseException::operator= (
    const ParseException & orig ) [inline]
```

Assignment operator.

### 10.224.3.3 SetLastElement()

```
void gdcm::ParseException::SetLastElement (
    DataElement & de ) [inline]
```

Equivalence operator.

Referenced by [gdcm::BasicOffsetTable::Read\(\)](#), [gdcm::Fragment::ReadBacktrack\(\)](#), and [gdcm::Fragment::ReadValue\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmParseException.h](#)

## 10.225 gdcm::Parser Class Reference

[Parser](#) ala XML\_Parser from expat (SAX)

```
#include <gdcmParser.h>
```



## Public Types

- typedef void(\* [EndElementHandler](#)) (void \*userData, const [Tag](#) &name)
- enum [ErrorType](#) {  
    [NoError](#) ,  
    [NoMemoryError](#) ,  
    [SyntaxError](#) ,  
    [NoElementsError](#) ,  
    [TagMismatchError](#) ,  
    [DuplicateAttributeError](#) ,  
    [JunkAfterDocElementError](#) ,  
    [UndefinedEntityError](#) ,  
    [UnexpectedStateError](#) }
- typedef void(\* [StartElementHandler](#)) (void \*userData, const [Tag](#) &tag, const char \*atts[])

## Public Member Functions

- [Parser](#) ()
- [~Parser](#) ()
- unsigned long [GetCurrentByteIndex](#) () const
- [ErrorType](#) [GetErrorCode](#) () const
- void \* [GetUserData](#) () const
- bool [Parse](#) (const char \*s, int len, bool isFinal)
- void [SetElementHandler](#) ([StartElementHandler](#) start, [EndElementHandler](#) end)
- void [SetUserData](#) (void \*userData)

## Static Public Member Functions

- static const char \* [GetErrorString](#) ([ErrorType](#) const &err)

## Protected Member Functions

- char \* [GetBuffer](#) (int len)
- bool [ParseBuffer](#) (int len, bool isFinal)
- [ErrorType](#) [Process](#) ()

### 10.225.1 Detailed Description

[Parser](#) ala XML\_Parser from expat (SAX)

Detailed description here

#### Note

Simple API for DICOM

## 10.225.2 Member Typedef Documentation

### 10.225.2.1 EndElementHandler

```
typedef void(* gdcM::Parser::EndElementHandler) (void *userData, const Tag &name)
```

### 10.225.2.2 StartElementHandler

```
typedef void(* gdcM::Parser::StartElementHandler) (void *userData, const Tag &tag, const char *atts[])
```

## 10.225.3 Member Enumeration Documentation

### 10.225.3.1 ErrorType

```
enum gdcM::Parser::ErrorType
```

Enumerator

NoError	
NoMemoryError	
SyntaxError	
NoElementsError	
TagMismatchError	
DuplicateAttributeError	
JunkAfterDocElementError	
UndefinedEntityError	
UnexpectedStateError	

## 10.225.4 Constructor & Destructor Documentation

### 10.225.4.1 Parser()

```
gdcM::Parser::Parser ( ) [inline]
```

### 10.225.4.2 ~Parser()

```
gdcM::Parser::~~Parser ( ) [inline]
```

## 10.225.5 Member Function Documentation

### 10.225.5.1 GetBuffer()

```
char * gdcm::Parser::GetBuffer (
    int len ) [protected]
```

### 10.225.5.2 GetCurrentByteIndex()

```
unsigned long gdcm::Parser::GetCurrentByteIndex ( ) const
```

### 10.225.5.3 GetErrorCode()

```
ErrorType gdcm::Parser::GetErrorCode ( ) const
```

### 10.225.5.4 GetErrorString()

```
static const char * gdcm::Parser::GetErrorString (
    ErrorType const & err ) [static]
```

### 10.225.5.5 GetUserData()

```
void * gdcm::Parser::GetUserData ( ) const
```

### 10.225.5.6 Parse()

```
bool gdcm::Parser::Parse (
    const char * s,
    int len,
    bool isFinal )
```

### 10.225.5.7 ParseBuffer()

```
bool gdcm::Parser::ParseBuffer (
    int len,
    bool isFinal ) [protected]
```

### 10.225.5.8 Process()

```
ErrorType gdcm::Parser::Process ( ) [protected]
```

#### 10.225.5.9 SetElementHandler()

```
void gdcM::Parser::SetElementHandler (
    StartElementHandler start,
    EndElementHandler end )
```

#### 10.225.5.10 SetUserData()

```
void gdcM::Parser::SetUserData (
    void * userData )
```

The documentation for this class was generated from the following file:

- [gdcMParser.h](#)

### 10.226 gdcM::Patient Class Reference

See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

```
#include <gdcMPatient.h>
```

#### Public Member Functions

- [Patient](#) ()=default

#### 10.226.1 Detailed Description

See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

#### 10.226.2 Constructor & Destructor Documentation

##### 10.226.2.1 Patient()

```
gdcM::Patient::Patient ( ) [default]
```

The documentation for this class was generated from the following file:

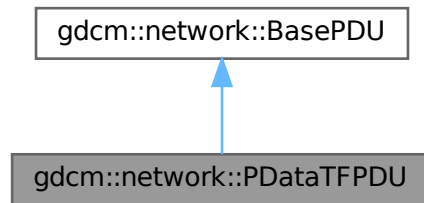
- [gdcMPatient.h](#)

## 10.227 gdcm::network::PDataTFPDU Class Reference

[PDataTFPDU](#).

```
#include <gdcmPDataTFPDU.h>
```

Inheritance diagram for gdcm::network::PDataTFPDU:



Collaboration diagram for gdcm::network::PDataTFPDU:



### Public Types

- typedef std::vector< [PresentationDataValue](#) >::size\_type [SizeType](#)

### Public Member Functions

- [PDataTFPDU](#) ()
- void [AddPresentationDataValue](#) ([PresentationDataValue](#) const &pdv)
- [SizeType](#) [GetNumberOfPresentationDataValues](#) () const
- [PresentationDataValue](#) const & [GetPresentationDataValue](#) ([SizeType](#) i) const
- bool [IsLastFragment](#) () const override
- void [Print](#) (std::ostream &os) const override
- std::istream & [Read](#) (std::istream &is) override
- size\_t [Size](#) () const override
- const std::ostream & [Write](#) (std::ostream &os) const override

## Public Member Functions inherited from [gdcm::network::BasePDU](#)

- virtual [~BasePDU](#) ()=default

## Protected Member Functions

- std::istream & [ReadInto](#) (std::istream &is, std::ostream &os)

### 10.227.1 Detailed Description

[PDataTFPDU](#).

[Table 9-22 P-DATA-TF PDU FIELDS](#)

### 10.227.2 Member Typedef Documentation

#### 10.227.2.1 SizeType

```
typedef std::vector<PresentationDataValue>::size_type gdcm::network::PDataTFPDU::SizeType
```

### 10.227.3 Constructor & Destructor Documentation

#### 10.227.3.1 PDataTFPDU()

```
gdcm::network::PDataTFPDU::PDataTFPDU ( )
```

### 10.227.4 Member Function Documentation

#### 10.227.4.1 AddPresentationDataValue()

```
void gdcm::network::PDataTFPDU::AddPresentationDataValue (
    PresentationDataValue const & pdv ) [inline]
```

#### 10.227.4.2 GetNumberOfPresentationDataValues()

```
SizeType gdcm::network::PDataTFPDU::GetNumberOfPresentationDataValues ( ) const [inline]
```

#### 10.227.4.3 GetPresentationDataValue()

```
PresentationDataValue const & gdcm::network::PDataTFPDU::GetPresentationDataValue (
    SizeType i ) const [inline]
```

#### 10.227.4.4 IsLastFragment()

```
bool gdcm::network::PDataTFPDU::IsLastFragment ( ) const [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

#### 10.227.4.5 Print()

```
void gdcm::network::PDataTFPDU::Print (
    std::ostream & os ) const [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

#### 10.227.4.6 Read()

```
std::istream & gdcm::network::PDataTFPDU::Read (
    std::istream & is ) [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

#### 10.227.4.7 ReadInto()

```
std::istream & gdcm::network::PDataTFPDU::ReadInto (
    std::istream & is,
    std::ostream & os ) [protected]
```

#### 10.227.4.8 Size()

```
size_t gdcm::network::PDataTFPDU::Size ( ) const [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

#### 10.227.4.9 Write()

```
const std::ostream & gdcm::network::PDataTFPDU::Write (
    std::ostream & os ) const [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

The documentation for this class was generated from the following file:

- [gdcmPDataTFPDU.h](#)

## 10.228 gdcm::PDBElement Class Reference

Class to represent a PDB [Element](#).

```
#include <gdcmPDBElement.h>
```

Collaboration diagram for gdcm::PDBElement:



### Public Member Functions

- `PDBElement()` = default
- `const char * GetName()` const  
*Set/Get Name.*
- `const char * GetValue()` const  
*Set/Get Value.*
- `bool operator== (const PDBElement &de)` const
- `void SetName (const char *name)`
- `void SetValue (const char *value)`

### Protected Attributes

- `std::string NameField`
- `std::string ValueField`

### Friends

- `std::ostream & operator<< (std::ostream &os, const PDBElement &val)`



## 10.228.1 Detailed Description

Class to represent a PDB [Element](#).

See also

[PDBHeader](#)

## 10.228.2 Constructor & Destructor Documentation

### 10.228.2.1 PDBelement()

```
gdcm::PDBelement::PDBelement ( ) [default]
```

References [gdcm::operator<<\(\)](#).

## 10.228.3 Member Function Documentation

### 10.228.3.1 GetName()

```
const char * gdcm::PDBelement::GetName ( ) const [inline]
```

Set/Get Name.

### 10.228.3.2 GetValue()

```
const char * gdcm::PDBelement::GetValue ( ) const [inline]
```

Set/Get [Value](#).

### 10.228.3.3 operator==( )

```
bool gdcm::PDBelement::operator==(   
    const PDBelement & de ) const [inline]
```

References [NameField](#), and [ValueField](#).

### 10.228.3.4 SetName()

```
void gdcm::PDBelement::SetName (   
    const char * name ) [inline]
```

### 10.228.3.5 SetValue()

```
void gdcM::PDBelement::SetValue (
    const char * value ) [inline]
```

## 10.228.4 Friends And Related Symbol Documentation

### 10.228.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & os,
    const PDBelement & val ) [friend]
```

## 10.228.5 Member Data Documentation

### 10.228.5.1 NameField

```
std::string gdcM::PDBelement::NameField [protected]
```

Referenced by [operator==\(\)](#).

### 10.228.5.2 ValueField

```
std::string gdcM::PDBelement::ValueField [protected]
```

Referenced by [operator==\(\)](#).

The documentation for this class was generated from the following file:

- [gdcMPDBelement.h](#)

## 10.229 gdcM::PDBHeader Class Reference

Class for [PDBHeader](#).

```
#include <gdcMPDBHeader.h>
```

## Public Member Functions

- [PDBHeader](#) ()=default
- [~PDBHeader](#) ()=default
- bool [FindPDBElementByName](#) (const char \*name)  
*Return true if the PDB element matching name is found or not.*
- const [PDBElement](#) & [GetPDBElementByName](#) (const char \*name)
- bool [LoadFromDataElement](#) ([DataElement](#) const &de)  
*Load the PDB Header from a [DataElement](#) of a [DataSet](#).*
- void [Print](#) (std::ostream &os) const  
*Print.*

## Static Public Member Functions

- static const [PrivateTag](#) & [GetPDBInfoTag](#) ()  
*Return the Private [Tag](#) where the PDB header is stored within a DICOM [DataSet](#).*

## Protected Member Functions

- const [PDBElement](#) & [GetPDBEEnd](#) () const

## Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [PDBHeader](#) &d)

## 10.229.1 Detailed Description

Class for [PDBHeader](#).

GEMS MR [Image](#) have an [Attribute](#) (0025,1b,GEMS\_SERS\_01) which store the Acquisition parameter of the MR [Image](#). It is compressed and can therefore not be used as is. This class de-encapsulated the Protocol Data Block and allow users to query element by name.

### Warning

Everything you do with this code is at your own risk, since decoding process was not written from specification documents.

: the API of this class might change.

: SEDESC is not always pure ASCII it can contains latin1

### See also

[CSAHeader](#)

## 10.229.2 Constructor & Destructor Documentation

### 10.229.2.1 PDBHeader()

```
gdcm::PDBHeader::PDBHeader ( ) [default]
```

### 10.229.2.2 ~PDBHeader()

```
gdcm::PDBHeader::~~PDBHeader ( ) [default]
```

## 10.229.3 Member Function Documentation

### 10.229.3.1 FindPDBElementByName()

```
bool gdcm::PDBHeader::FindPDBElementByName (
    const char * name )
```

Return true if the PDB element matching name is found or not.

### 10.229.3.2 GetPDBEEnd()

```
const PDBElement & gdcm::PDBHeader::GetPDBEEnd ( ) const [protected]
```

### 10.229.3.3 GetPDBElementByName()

```
const PDBElement & gdcm::PDBHeader::GetPDBElementByName (
    const char * name )
```

Lookup in the PDB header if a PDB element match the name 'name':

#### Warning

Case Sensitive

### 10.229.3.4 GetPDBInfoTag()

```
static const PrivateTag & gdcm::PDBHeader::GetPDBInfoTag ( ) [static]
```

Return the Private [Tag](#) where the PDB header is stored within a DICOM [DataSet](#).

### 10.229.3.5 LoadFromDataElement()

```
bool gdcm::PDBHeader::LoadFromDataElement (
    DataElement const & de )
```

Load the PDB Header from a [DataElement](#) of a [DataSet](#).

### 10.229.3.6 Print()

```
void gdcm::PDBHeader::Print (
    std::ostream & os ) const
```

Print.

## 10.229.4 Friends And Related Symbol Documentation

### 10.229.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const PDBHeader & d ) [friend]
```

The documentation for this class was generated from the following file:

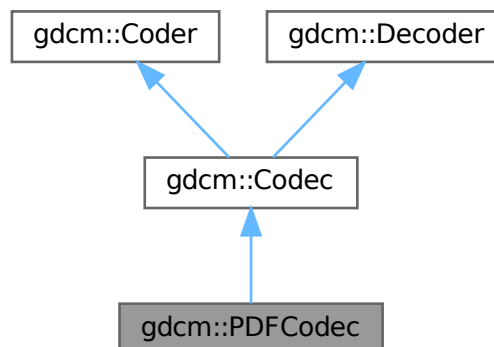
- [gdcmPDBHeader.h](#)

## 10.230 gdcm::PDFCodec Class Reference

[PDFCodec](#) class.

```
#include <gdcmPDFCodec.h>
```

Inheritance diagram for gdcm::PDFCodec:



Collaboration diagram for gdcm::PDFCodec:



### Public Member Functions

- [PDFCodec](#) ()
- [~PDFCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &) const override  
*Return whether this coder support this transfer syntax (can code it)*
- bool [CanDecode](#) ([TransferSyntax](#) const &) const override  
*Return whether this decoder support this transfer syntax (can decode it)*
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override  
*Decode.*

### Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default
- virtual bool [Code](#) ([DataElement](#) const &in\_, [DataElement](#) &out\_)  
*Code.*

### Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

### Additional Inherited Members

### Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char \*bv, unsigned long len, std::ostream &os)

## Protected Member Functions inherited from [gdcm::Decoder](#)

- virtual bool [DecodeByStreams](#) (std::istream &, std::ostream &)

### 10.230.1 Detailed Description

[PDFCodec](#) class.

### 10.230.2 Constructor & Destructor Documentation

#### 10.230.2.1 PDFCodec()

```
gdcm::PDFCodec::PDFCodec ( )
```

#### 10.230.2.2 ~PDFCodec()

```
gdcm::PDFCodec::~~PDFCodec ( ) [override]
```

### 10.230.3 Member Function Documentation

#### 10.230.3.1 CanCode()

```
bool gdcm::PDFCodec::CanCode (
    TransferSyntax const & ) const [inline], [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Implements [gdcm::Coder](#).

#### 10.230.3.2 CanDecode()

```
bool gdcm::PDFCodec::CanDecode (
    TransferSyntax const & ) const [inline], [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Implements [gdcm::Decoder](#).

### 10.230.3.3 Decode()

```
bool gdcmm::PDFCodec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcmm::Decoder](#).

The documentation for this class was generated from the following file:

- [gdcmmPDFCodec.h](#)

## 10.231 gdcmm::network::PDUFactory Class Reference

[PDUFactory](#) basically, given an initial byte, construct the.

```
#include <gdcmmPDUFactory.h>
```

### Static Public Member Functions

- static [BasePDU](#) \* [ConstructAbortPDU](#) ()
- static [BasePDU](#) \* [ConstructPDU](#) (uint8\_t itemtype)
- static [BasePDU](#) \* [ConstructReleasePDU](#) ()
- static std::vector< [BasePDU](#) \* > [CreateCEchoPDU](#) (const [ULConnection](#) &inConnection)
- static std::vector< [BasePDU](#) \* > [CreateCFindPDU](#) (const [ULConnection](#) &inConnection, const [BaseRootQuery](#) \*inRootQuery)
- static std::vector< [BasePDU](#) \* > [CreateCMovePDU](#) (const [ULConnection](#) &inConnection, const [BaseRootQuery](#) \*inRootQuery)
- static std::vector< [BasePDU](#) \* > [CreateCStoreRQPDU](#) (const [ULConnection](#) &inConnection, const [File](#) &file, bool writeDataSet=true)
- static std::vector< [BasePDU](#) \* > [CreateCStoreRSPPDU](#) (const [DataSet](#) \*inDataSet, const [BasePDU](#) \*inPC)
- static std::vector< [BasePDU](#) \* > [CreateNActionPDU](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) \*inQuery)
- static std::vector< [BasePDU](#) \* > [CreateNCreatePDU](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) \*inQuery)
- static std::vector< [BasePDU](#) \* > [CreateNDeletePDU](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) \*inQuery)
- static std::vector< [BasePDU](#) \* > [CreateNEventReportPDU](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) \*inQuery)
- static std::vector< [BasePDU](#) \* > [CreateNGetPDU](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) \*inQuery)
- static std::vector< [BasePDU](#) \* > [CreateNSetPDU](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) \*inQuery)
- static [EEventID](#) [DetermineEventByPDU](#) (const [BasePDU](#) \*inPDU)
- static std::vector< [PresentationDataValue](#) > [GetPDVs](#) (const std::vector< [BasePDU](#) \* > &inDataPDUs)



### 10.231.1 Detailed Description

[PDUFactory](#) basically, given an initial byte, construct the.

appropriate PDU. This way, the event loop doesn't have to know about all the different PDU types.

### 10.231.2 Member Function Documentation

#### 10.231.2.1 ConstructAbortPDU()

```
static BasePDU * gdcm::network::PDUFactory::ConstructAbortPDU ( ) [static]
```

#### 10.231.2.2 ConstructPDU()

```
static BasePDU * gdcm::network::PDUFactory::ConstructPDU (
    uint8_t itemtype ) [static]
```

#### 10.231.2.3 ConstructReleasePDU()

```
static BasePDU * gdcm::network::PDUFactory::ConstructReleasePDU ( ) [static]
```

#### 10.231.2.4 CreateCEchoPDU()

```
static std::vector< BasePDU * > gdcm::network::PDUFactory::CreateCEchoPDU (
    const ULConnection & inConnection ) [static]
```

#### 10.231.2.5 CreateCFindPDU()

```
static std::vector< BasePDU * > gdcm::network::PDUFactory::CreateCFindPDU (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery ) [static]
```

#### 10.231.2.6 CreateCMovePDU()

```
static std::vector< BasePDU * > gdcm::network::PDUFactory::CreateCMovePDU (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery ) [static]
```

#### 10.231.2.7 CreateCStoreRQPDU()

```
static std::vector< BasePDU * > gdcm::network::PDUFactory::CreateCStoreRQPDU (
    const ULConnection & inConnection,
    const File & file,
    bool writeDataSet = true ) [static]
```

#### 10.231.2.8 CreateCStoreRSPPDU()

```
static std::vector< BasePDU * > gdcm::network::PDUFactory::CreateCStoreRSPPDU (
    const DataSet * inDataSet,
    const BasePDU * inPC ) [static]
```

#### 10.231.2.9 CreateNActionPDU()

```
static std::vector< BasePDU * > gdcm::network::PDUFactory::CreateNActionPDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

#### 10.231.2.10 CreateNCreatePDU()

```
static std::vector< BasePDU * > gdcm::network::PDUFactory::CreateNCreatePDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

#### 10.231.2.11 CreateNDeletePDU()

```
static std::vector< BasePDU * > gdcm::network::PDUFactory::CreateNDeletePDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

#### 10.231.2.12 CreateNEventReportPDU()

```
static std::vector< BasePDU * > gdcm::network::PDUFactory::CreateNEventReportPDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

#### 10.231.2.13 CreateNGetPDU()

```
static std::vector< BasePDU * > gdcm::network::PDUFactory::CreateNGetPDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

**10.231.2.14 CreateNSetPDU()**

```
static std::vector< BasePDU * > gdcm::network::PDUFactory::CreateNSetPDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

**10.231.2.15 DetermineEventByPDU()**

```
static EEventID gdcm::network::PDUFactory::DetermineEventByPDU (
    const BasePDU * inPDU ) [static]
```

**10.231.2.16 GetPDVs()**

```
static std::vector< PresentationDataValue > gdcm::network::PDUFactory::GetPDVs (
    const std::vector< BasePDU * > & inDataPDUs ) [static]
```

The documentation for this class was generated from the following file:

- [gdcmPDUFactory.h](#)

**10.232 gdcm::PersonName Class Reference**

[PersonName](#) class.

```
#include <gdcmPersonName.h>
```

**Public Member Functions**

- unsigned int [GetMaxLength](#) () const
- unsigned int [GetNumberOfComponents](#) () const
- void [Print](#) (std::ostream &os) const
- void [SetBlob](#) (const std::vector< char > &v)
- void [SetComponents](#) (const char \*comp1="", const char \*comp2="", const char \*comp3="", const char \*comp4="", const char \*comp5="")
- void [SetComponents](#) (const char \*components[])

**Public Attributes**

- char [Component](#) [[MaxNumberOfComponents](#)][[MaxLength](#)+1]

## Static Public Attributes

- static const unsigned int [MaxLength](#) = 64
- static const unsigned int [MaxNumberOfComponents](#) = 5
- static const char [Padding](#) = ''
- static const char [Separator](#) = '^'

## 10.232.1 Detailed Description

[PersonName](#) class.

## 10.232.2 Member Function Documentation

### 10.232.2.1 GetMaxLength()

```
unsigned int gdcm::PersonName::GetMaxLength ( ) const [inline]
```

### 10.232.2.2 GetNumberOfComponents()

```
unsigned int gdcm::PersonName::GetNumberOfComponents ( ) const [inline]
```

### 10.232.2.3 Print()

```
void gdcm::PersonName::Print (
    std::ostream & os ) const [inline]
```

### 10.232.2.4 SetBlob()

```
void gdcm::PersonName::SetBlob (
    const std::vector< char > & v ) [inline]
```

### 10.232.2.5 SetComponents() [1/2]

```
void gdcm::PersonName::SetComponents (
    const char * comp1 = "",
    const char * comp2 = "",
    const char * comp3 = "",
    const char * comp4 = "",
    const char * comp5 = "" ) [inline]
```

### 10.232.2.6 SetComponents() [2/2]

```
void gdcM::PersonName::SetComponents (
    const char * components[] ) [inline]
```

## 10.232.3 Member Data Documentation

### 10.232.3.1 Component

```
char gdcM::PersonName::Component [MaxNumberOfComponents] [MaxLength+1]
```

### 10.232.3.2 MaxLength

```
const unsigned int gdcM::PersonName::MaxLength = 64 [static]
```

### 10.232.3.3 MaxNumberOfComponents

```
const unsigned int gdcM::PersonName::MaxNumberOfComponents = 5 [static]
```

### 10.232.3.4 Padding

```
const char gdcM::PersonName::Padding = ' ' [static]
```

### 10.232.3.5 Separator

```
const char gdcM::PersonName::Separator = '^' [static]
```

The documentation for this class was generated from the following file:

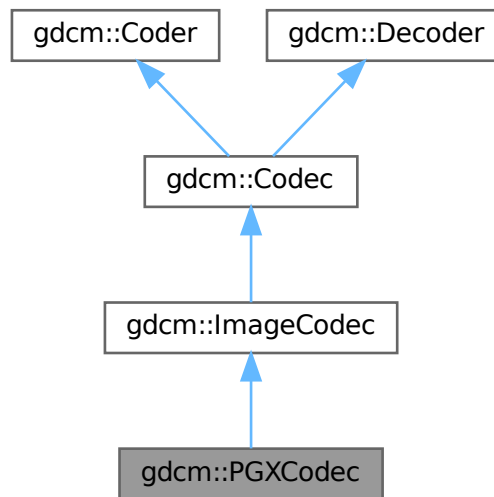
- [gdcMPersonName.h](#)

## 10.233 gdcm::PGXCodec Class Reference

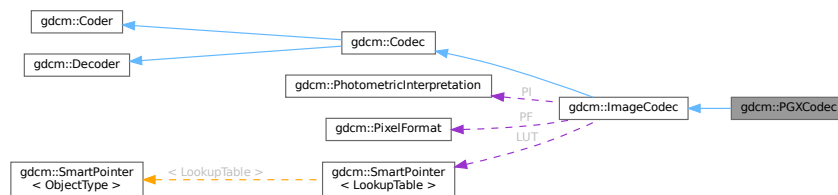
Class to do PGX.

```
#include <gdcmPGXCodec.h>
```

Inheritance diagram for gdcm::PGXCodec:



Collaboration diagram for gdcm::PGXCodec:



### Public Member Functions

- [PGXCodec](#) ()
- [~PGXCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const override  
*Return whether this coder support this transfer syntax (can code it)*

- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const override  
*Return whether this decoder support this transfer syntax (can decode it)*
- [ImageCodec](#) \* [Clone](#) () const override
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override
- bool [Read](#) (const char \*filename, [DataElement](#) &out) const
- bool [Write](#) (const char \*filename, const [DataElement](#) &out) const

### Public Member Functions inherited from [gdcm::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CleanupUnusedBits](#) (char \*data, size\_t datalen)
- bool [Decode](#) ([DataElement](#) const &is\_, [DataElement](#) &os) override  
*Decode.*
- const unsigned int \* [GetDimensions](#) () const
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- virtual void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)

### Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default
- virtual bool [Code](#) ([DataElement](#) const &in\_, [DataElement](#) &out\_)  
*Code.*

### Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

## Additional Inherited Members

### Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

### Protected Member Functions inherited from [gdcm::ImageCodec](#)

- virtual bool [AppendFrameEncode](#) (std::ostream &out, const char \*data, size\_t datalen)
- virtual bool [AppendRowEncode](#) (std::ostream &out, const char \*data, size\_t datalen)
- bool [DecodeByStreams](#) (std::istream &is\_, std::ostream &os) override
- bool [DoByteSwap](#) (std::istream &is\_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is\_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is\_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is\_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is\_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is\_, std::ostream &os)
- virtual bool [IsFrameEncoder](#) ()
- virtual bool [IsRowEncoder](#) ()
- virtual bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)
- virtual bool [StartEncode](#) (std::ostream &os)
- virtual bool [StopEncode](#) (std::ostream &os)

### Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char \*bv, unsigned long len, std::ostream &os)

### Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) [LUT](#)
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) [PF](#)
- [PhotometricInterpretation](#) [PI](#)
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

## 10.233.1 Detailed Description

Class to do PGX.

See PGX as used in JPEG 2000 implementation and reference images



## 10.233.2 Constructor & Destructor Documentation

### 10.233.2.1 PGXCodec()

```
gdcm::PGXCodec::PGXCodec ( )
```

### 10.233.2.2 ~PGXCodec()

```
gdcm::PGXCodec::~~PGXCodec ( ) [override]
```

## 10.233.3 Member Function Documentation

### 10.233.3.1 CanCode()

```
bool gdcm::PGXCodec::CanCode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.233.3.2 CanDecode()

```
bool gdcm::PGXCodec::CanDecode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.233.3.3 Clone()

```
ImageCodec * gdcm::PGXCodec::Clone ( ) const [override], [virtual]
```

Implements [gdcm::ImageCodec](#).

### 10.233.3.4 GetHeaderInfo()

```
bool gdcm::PGXCodec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.233.3.5 Read()

```
bool gdcM::PGXCodec::Read (
    const char * filename,
    DataElement & out ) const
```

### 10.233.3.6 Write()

```
bool gdcM::PGXCodec::Write (
    const char * filename,
    const DataElement & out ) const
```

The documentation for this class was generated from the following file:

- [gdcM\\_PGXCodec.h](#)

## 10.234 gdcM::PhotometricInterpretation Class Reference

Class to represent an [PhotometricInterpretation](#).

```
#include <gdcMPhotometricInterpretation.h>
```

### Public Types

- enum [PType](#) {  
    [UNKNOWN](#) = 0 ,  
    [MONOCHROME1](#) ,  
    [MONOCHROME2](#) ,  
    [PALETTE\\_COLOR](#) ,  
    [RGB](#) ,  
    [HSV](#) ,  
    [ARGB](#) ,  
    [CMYK](#) ,  
    [YBR\\_FULL](#) ,  
    [YBR\\_FULL\\_422](#) ,  
    [YBR\\_PARTIAL\\_422](#) ,  
    [YBR\\_PARTIAL\\_420](#) ,  
    [YBR\\_ICT](#) ,  
    [YBR\\_RCT](#) ,  
    [PI\\_END](#) }

## Public Member Functions

- [PhotometricInterpretation](#) ([PIType](#) pi=[UNKNOWN](#))
- unsigned short [GetSamplesPerPixel](#) () const  
*return the value for Sample Per Pixel associated with a particular Photometric Interpretation*
- const char \* [GetString](#) () const
- [PIType](#) [GetType](#) () const
- bool [IsLossless](#) () const
- bool [IsLossy](#) () const
- bool [IsSameColorSpace](#) ([PhotometricInterpretation](#) const &pi) const
- [operator PIType](#) () const

## Static Public Member Functions

- static const char \* [GetPIString](#) ([PIType](#) pi)
- static [PIType](#) [GetPIType](#) (const char \*pi)
- static bool [IsRetired](#) ([PIType](#) pi)

## Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [PhotometricInterpretation](#) &pi)

## 10.234.1 Detailed Description

Class to represent an [PhotometricInterpretation](#).

### Examples

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImage.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [ExtractImageRegion.cs](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [HelloVizWorld.cxx](#), [MpegVideoInfo.cs](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

## 10.234.2 Member Enumeration Documentation

### 10.234.2.1 PIType

```
enum gdcm::PhotometricInterpretation::PIType
```

#### Enumerator

UNKNOWN	
MONOCHROME1	
MONOCHROME2	
PALETTE_COLOR	
RGB	
HSV	

## Enumerator

ARGB	
CMYK	
YBR_FULL	
YBR_FULL_422	
YBR_PARTIAL_422	
YBR_PARTIAL_420	
YBR_ICT	
YBR_RCT	
PI_END	

## Examples

[DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), and [MpegVideoInfo.cs](#).

### 10.234.3 Constructor & Destructor Documentation

#### 10.234.3.1 PhotometricInterpretation()

```
gdcmm::PhotometricInterpretation::PhotometricInterpretation (
    PType pi = UNKNOWN ) [inline]
```

### 10.234.4 Member Function Documentation

#### 10.234.4.1 GetPIString()

```
static const char * gdcmm::PhotometricInterpretation::GetPIString (
    PType pi ) [static]
```

#### 10.234.4.2 GetPType()

```
static PType gdcmm::PhotometricInterpretation::GetPType (
    const char * pi ) [static]
```

#### 10.234.4.3 GetSamplesPerPixel()

```
unsigned short gdcmm::PhotometricInterpretation::GetSamplesPerPixel ( ) const
```

return the value for Sample Per Pixel associated with a particular Photometric Interpretation

References [gdcmm::operator<<\(\)](#).

#### 10.234.4.4 GetString()

```
const char * gdcm::PhotometricInterpretation::GetString ( ) const
```

#### 10.234.4.5 GetType()

```
PIType gdcm::PhotometricInterpretation::GetType ( ) const [inline]
```

#### 10.234.4.6 IsLossless()

```
bool gdcm::PhotometricInterpretation::IsLossless ( ) const
```

#### 10.234.4.7 IsLossy()

```
bool gdcm::PhotometricInterpretation::IsLossy ( ) const
```

#### 10.234.4.8 IsRetired()

```
static bool gdcm::PhotometricInterpretation::IsRetired (
    PIType pi ) [static]
```

#### 10.234.4.9 IsSameColorSpace()

```
bool gdcm::PhotometricInterpretation::IsSameColorSpace (
    PhotometricInterpretation const & pi ) const
```

#### 10.234.4.10 operator PIType()

```
gdcm::PhotometricInterpretation::operator PIType ( ) const [inline]
```

### 10.234.5 Friends And Related Symbol Documentation

#### 10.234.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & os,
    const PhotometricInterpretation & pi ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmPhotometricInterpretation.h](#)

## 10.235 gdcm::PixelFormat Class Reference

[PixelFormat](#).

```
#include <gdcmPixelFormat.h>
```

### Public Types

- enum [ScalarType](#) {  
[UINT8](#) ,  
[INT8](#) ,  
[UINT12](#) ,  
[INT12](#) ,  
[UINT16](#) ,  
[INT16](#) ,  
[UINT32](#) ,  
[INT32](#) ,  
[UINT64](#) ,  
[INT64](#) ,  
[FLOAT16](#) ,  
[FLOAT32](#) ,  
[FLOAT64](#) ,  
[SINGLEBIT](#) ,  
[UNKNOWN](#) }

### Public Member Functions

- [PixelFormat](#) ()
- [PixelFormat](#) ([ScalarType](#) st)
- [PixelFormat](#) (unsigned short samplesperpixel, unsigned short bitsallocated=8, unsigned short bitsstored=8, unsigned short highbit=7, unsigned short pixelrepresentation=0)
- unsigned short [GetBitsAllocated](#) () const  
*BitsAllocated see [Tag](#) (0028,0100) US Bits Allocated.*
- unsigned short [GetBitsStored](#) () const  
*BitsStored see [Tag](#) (0028,0101) US Bits Stored.*
- unsigned short [GetHighBit](#) () const  
*HighBit see [Tag](#) (0028,0102) US High Bit.*
- int64\_t [GetMax](#) () const  
*return the max possible of the pixel*
- int64\_t [GetMin](#) () const  
*return the min possible of the pixel*
- unsigned short [GetPixelRepresentation](#) () const  
*PixelRepresentation: 0 or 1, see [Tag](#) (0028,0103) US Pixel Representation.*
- uint8\_t [GetPixelSize](#) () const
- unsigned short [GetSamplesPerPixel](#) () const
- [ScalarType](#) [GetScalarType](#) () const  
*ScalarType does not take into account the sample per pixel.*
- const char \* [GetScalarTypeAsString](#) () const

- bool [IsCompatible](#) (const [TransferSyntax](#) &ts) const
- bool [IsValid](#) () const  
*return IsValid*
- [operator ScalarType](#) () const
- bool [operator!=](#) (const [PixelFormat](#) &pf) const
- bool [operator!=](#) ([ScalarType](#) st) const
- bool [operator==](#) (const [PixelFormat](#) &pf) const
- bool [operator==](#) ([ScalarType](#) st) const
- void [Print](#) (std::ostream &os) const  
*Print.*
- void [SetBitsAllocated](#) (unsigned short ba)
- void [SetBitsStored](#) (unsigned short bs)
- void [SetHighBit](#) (unsigned short hb)
- void [SetPixelRepresentation](#) (unsigned short pr)
- void [SetSamplesPerPixel](#) (unsigned short spp)
- void [SetScalarType](#) ([ScalarType](#) st)

### Protected Member Functions

- bool [Validate](#) ()  
*When image with 24/24/23 was read, need to validate.*

### Friends

- class [Bitmap](#)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [PixelFormat](#) &pf)

## 10.235.1 Detailed Description

### [PixelFormat](#).

By default the Pixel [Type](#) will be instantiated with the following parameters:

- SamplesPerPixel : 1
- BitsAllocated : 8
- BitsStored : 8
- HighBit : 7
- PixelRepresentation : 0

Fundamentally [PixelFormat](#) is very close to what DICOM allows. It will be very hard to extend this class for the upcoming DICOM standard where Floating 32 and 64bits will be allowed.

It is also very hard for this class to fully support 64bits integer type (see GetMin / GetMax signature restricted to 64bits signed).

### Examples

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImage.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), [ExtractOneFrame.cs](#), [FixJAIBugJPEGLS.cxx](#), [GetArray.cs](#), [GetJPEGSamplePrecision.cxx](#), [MpegVideoInfo.cs](#), [RescaleImage.cs](#), [TemplateEmptyImage.cxx](#), [csa2img.cxx](#), [iU22tomultisc.cxx](#), and [threadgdcm.cxx](#).

## 10.235.2 Member Enumeration Documentation

### 10.235.2.1 ScalarType

enum `gdcm::PixelFormat::ScalarType`

Enumerator

UINT8	
INT8	
UINT12	
INT12	
UINT16	
INT16	
UINT32	
INT32	
UINT64	
INT64	
FLOAT16	
FLOAT32	
FLOAT64	
SINGLEBIT	
UNKNOWN	

Examples

[GetArray.cs](#).

## 10.235.3 Constructor & Destructor Documentation

### 10.235.3.1 PixelFormat() [1/3]

```
gdcm::PixelFormat::PixelFormat ( ) [inline]
```

### 10.235.3.2 PixelFormat() [2/3]

```
gdcm::PixelFormat::PixelFormat (
    unsigned short samplesperpixel,
    unsigned short bitsallocated = 8,
    unsigned short bitsstored = 8,
    unsigned short highbit = 7,
    unsigned short pixelrepresentation = 0 ) [inline], [explicit]
```

### 10.235.3.3 PixelFormat() [3/3]

```
gdcm::PixelFormat::PixelFormat (
    ScalarType st )
```



## 10.235.4 Member Function Documentation

### 10.235.4.1 GetBitsAllocated()

```
unsigned short gdcm::PixelFormat::GetBitsAllocated ( ) const [inline]
```

BitsAllocated see [Tag](#) (0028,0100) US Bits Allocated.

#### Examples

[GetJPEGSamplePrecision.cxx](#).

### 10.235.4.2 GetBitsStored()

```
unsigned short gdcm::PixelFormat::GetBitsStored ( ) const [inline]
```

BitsStored see [Tag](#) (0028,0101) US Bits Stored.

#### Examples

[GetJPEGSamplePrecision.cxx](#).

### 10.235.4.3 GetHighBit()

```
unsigned short gdcm::PixelFormat::GetHighBit ( ) const [inline]
```

HighBit see [Tag](#) (0028,0102) US High Bit.

### 10.235.4.4 GetMax()

```
int64_t gdcm::PixelFormat::GetMax ( ) const
```

return the max possible of the pixel

### 10.235.4.5 GetMin()

```
int64_t gdcm::PixelFormat::GetMin ( ) const
```

return the min possible of the pixel

#### 10.235.4.6 GetPixelRepresentation()

```
unsigned short gdcm::PixelFormat::GetPixelRepresentation ( ) const [inline]
```

PixelRepresentation: 0 or 1, see [Tag](#) (0028,0103) US Pixel Representation.

#### 10.235.4.7 GetPixelSize()

```
uint8_t gdcm::PixelFormat::GetPixelSize ( ) const
```

return the size of the pixel This is the number of words it would take to store one pixel

##### Warning

the return value takes into account the SamplesPerPixel  
in the rare case when BitsAllocated == 12, the function assume word padding and value returned will be identical  
as if BitsAllocated == 16

##### Examples

[ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), [ExtractOneFrame.cs](#), and [threadgdcm.cxx](#).

#### 10.235.4.8 GetSamplesPerPixel()

```
unsigned short gdcm::PixelFormat::GetSamplesPerPixel ( ) const
```

Samples Per Pixel see (0028,0002) US Samples Per Pixel DICOM - only allows 1, 3 and 4 as valid value. Other value are undefined behavior.

##### Examples

[threadgdcm.cxx](#).

#### 10.235.4.9 GetScalarType()

```
ScalarType gdcm::PixelFormat::GetScalarType ( ) const
```

ScalarType does not take into account the sample per pixel.

##### Examples

[GetArray.cs](#).

#### 10.235.4.10 GetScalarTypeAsString()

```
const char * gdcm::PixelFormat::GetScalarTypeAsString ( ) const
```

##### Examples

[GetArray.cs](#).

#### 10.235.4.11 IsCompatible()

```
bool gdcm::PixelFormat::IsCompatible (
    const TransferSyntax & ts ) const
```

#### 10.235.4.12 IsValid()

```
bool gdcm::PixelFormat::IsValid ( ) const

return IsValid
```

#### 10.235.4.13 operator ScalarType()

```
gdcm::PixelFormat::operator ScalarType ( ) const [inline]
```

#### 10.235.4.14 operator!=( ) [1/2]

```
bool gdcm::PixelFormat::operator!= (
    const PixelFormat & pf ) const [inline]
```

#### 10.235.4.15 operator!=( ) [2/2]

```
bool gdcm::PixelFormat::operator!= (
    ScalarType st ) const [inline]
```

#### 10.235.4.16 operator==( ) [1/2]

```
bool gdcm::PixelFormat::operator== (
    const PixelFormat & pf ) const [inline]
```

#### 10.235.4.17 operator==( ) [2/2]

```
bool gdcm::PixelFormat::operator== (
    ScalarType st ) const [inline]
```

**10.235.4.18 Print()**

```
void gdcM::PixelFormat::Print (
    std::ostream & os ) const
```

Print.

**10.235.4.19 SetBitsAllocated()**

```
void gdcM::PixelFormat::SetBitsAllocated (
    unsigned short ba ) [inline]
```

**10.235.4.20 SetBitsStored()**

```
void gdcM::PixelFormat::SetBitsStored (
    unsigned short bs ) [inline]
```

**10.235.4.21 SetHighBit()**

```
void gdcM::PixelFormat::SetHighBit (
    unsigned short hb ) [inline]
```

**10.235.4.22 SetPixelRepresentation()**

```
void gdcM::PixelFormat::SetPixelRepresentation (
    unsigned short pr ) [inline]
```

**Examples**

[TemplateEmptyImage.cxx](#).

**10.235.4.23 SetSamplesPerPixel()**

```
void gdcM::PixelFormat::SetSamplesPerPixel (
    unsigned short spp ) [inline]
```

**Examples**

[CreateARGBImage.cxx](#), and [CreateCMYKImage.cxx](#).

References [gdcMAssertMacro](#).

#### 10.235.4.24 SetScalarType()

```
void gdcm::PixelFormat::SetScalarType (
    ScalarType st )
```

Set [PixelFormat](#) based only on the ScalarType

##### Warning

: You need to call SetScalarType *before* SetSamplesPerPixel

#### 10.235.4.25 Validate()

```
bool gdcm::PixelFormat::Validate ( ) [protected]
```

When image with 24/24/23 was read, need to validate.

Referenced by [gdcm::Bitmap::SetPixelFormat\(\)](#).

### 10.235.5 Friends And Related Symbol Documentation

#### 10.235.5.1 Bitmap

```
friend class Bitmap [friend]
```

#### 10.235.5.2 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const PixelFormat & pf ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmPixelFormat.h](#)

Pixmap class.

Inheritance diagram for gdcm::Pixmap:

[illegible]

## Public Member Functions

- [Pixmap](#) ()
- [~Pixmap](#) () override
- bool [AreOverlaysInPixelData](#) () const override  
*returns if Overlays are stored in the unused bit of the pixel data:*
- [Curve](#) & [GetCurve](#) (size\_t i=0)  
*Curve: group 50xx.*
- const [Curve](#) & [GetCurve](#) (size\_t i=0) const
- [IconImage](#) & [GetIconImage](#) ()
- const [IconImage](#) & [GetIconImage](#) () const  
*Set/Get Icon Image.*
- size\_t [GetNumberOfCurves](#) () const
- size\_t [GetNumberOfOverlays](#) () const
- [Overlay](#) & [GetOverlay](#) (size\_t i=0)  
*Overlay: group 60xx.*
- const [Overlay](#) & [GetOverlay](#) (size\_t i=0) const
- void [Print](#) (std::ostream &) const override
- void [RemoveOverlay](#) (size\_t i)
- void [SetIconImage](#) ([IconImage](#) const &ii)
- void [SetNumberOfCurves](#) (size\_t n)
- void [SetNumberOfOverlays](#) (size\_t n)
- bool [UnusedBitsPresentInPixelData](#) () const override  
*returns if there are unused bits in the pixel data*

## Public Member Functions inherited from [gdcm::Bitmap](#)

- [Bitmap](#) ()
- [~Bitmap](#) () override
- void [Clear](#) ()
- bool [GetBuffer](#) (char \*buffer) const  
*Access the raw data.*
- unsigned long [GetBufferLength](#) () const
- unsigned int [GetColumns](#) () const
- [DataElement](#) & [GetDataElement](#) ()
- const [DataElement](#) & [GetDataElement](#) () const
- unsigned int [GetDimension](#) (unsigned int idx) const
- const unsigned int \* [GetDimensions](#) () const  
*Return the dimension of the pixel data, first dimension (x), then 2nd (y), then 3rd (z)...*
- [LookupTable](#) & [GetLUT](#) ()
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const  
*INTERNAL do not use.*
- unsigned int [GetNumberOfDimensions](#) () const  
*Return the number of dimension of the pixel data bytes; for example 2 for a 2D matrices of values.*
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const  
*return the photometric interpretation*
- [PixelFormat](#) & [GetPixelFormat](#) ()

- const [PixelFormat](#) & [GetPixelFormat](#) () const  
*Get/Set [PixelFormat](#).*
- unsigned int [GetPlanarConfiguration](#) () const  
*return the planar configuration*
- unsigned int [GetRows](#) () const
- const [TransferSyntax](#) & [GetTransferSyntax](#) () const
- bool [IsEmpty](#) () const
- bool [IsLossy](#) () const  
*Return whether or not the image was compressed using a lossy compressor or not.*
- bool [IsTransferSyntaxCompatible](#) ([TransferSyntax](#) const &ts) const
- void [SetColumns](#) (unsigned int col)
- void [SetDataElement](#) ([DataElement](#) const &de)
- void [SetDimension](#) (unsigned int idx, unsigned int dim)
- void [SetDimensions](#) (const unsigned int dims[3])
- void [SetLossyFlag](#) (bool f)  
*Specifically set that the image was compressed using a lossy compression mechanism.*
- void [SetLUT](#) ([LookupTable](#) const &lut)  
*Set/Get LUT.*
- void [SetNeedByteSwap](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)
- void [SetRows](#) (unsigned int rows)
- void [SetTransferSyntax](#) ([TransferSyntax](#) const &ts)  
*Transfer syntax.*

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

## Protected Attributes

- std::vector< [Curve](#) > [Curves](#)
- [SmartPointer](#)< [IconImage](#) > [Icon](#)
- std::vector< [Overlay](#) > [Overlays](#)



### Protected Attributes inherited from [gdcm::Bitmap](#)

- `std::vector< unsigned int >` [Dimensions](#)
- `bool` [LossyFlag](#)
- `LUTPtr` [LUT](#)
- `bool` [NeedByteSwap](#)
- `unsigned int` [NumberOfDimensions](#)
- `PixelFormat` [PF](#)
- `PhotometricInterpretation` [PI](#)
- `DataElement` [PixelData](#)
- `unsigned int` [PlanarConfiguration](#)
- `TransferSyntax` [TS](#)

### Additional Inherited Members

### Protected Types inherited from [gdcm::Bitmap](#)

- `typedef` [SmartPointer< LookupTable >](#) [LUTPtr](#)

### Protected Member Functions inherited from [gdcm::Bitmap](#)

- `bool` [ComputeLossyFlag](#) ()
- `bool` [GetBuffer2](#) (std::ostream &os) const
- `bool` [TryJPEG2000Codec](#) (char \*buffer, bool &lossyflag) const
- `bool` [TryJPEG2000Codec2](#) (std::ostream &os) const
- `bool` [TryJPEGCodec](#) (char \*buffer, bool &lossyflag) const
- `bool` [TryJPEGCodec2](#) (std::ostream &os) const
- `bool` [TryJPEGLSCCodec](#) (char \*buffer, bool &lossyflag) const
- `bool` [TryKAKADUCoec](#) (char \*buffer, bool &lossyflag) const
- `bool` [TryPVRGCodec](#) (char \*buffer, bool &lossyflag) const
- `bool` [TryRAWCodec](#) (char \*buffer, bool &lossyflag) const
- `bool` [TryRLECodec](#) (char \*buffer, bool &lossyflag) const

### Protected Member Functions inherited from [gdcm::Object](#)

- `void` [Register](#) ()
- `void` [UnRegister](#) ()

## 10.236.1 Detailed Description

[Pixmap](#) class.

A bitmap based image. Used as parent for both IconImage and the main Pixel Data [Image](#) It does not contains any World Space information (IPP, IOP)

See also

[PixmapReader](#)

Examples

[FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), and [StandardizeFiles.cs](#).

## 10.236.2 Constructor & Destructor Documentation

### 10.236.2.1 Pixmap()

```
gdcm::Pixmap::Pixmap ( )
```

### 10.236.2.2 ~Pixmap()

```
gdcm::Pixmap::~~Pixmap ( ) [override]
```

## 10.236.3 Member Function Documentation

### 10.236.3.1 AreOverlaysInPixelData()

```
bool gdcm::Pixmap::AreOverlaysInPixelData ( ) const [override], [virtual]
```

returns if Overlays are stored in the unused bit of the pixel data:

Reimplemented from [gdcm::Bitmap](#).

### 10.236.3.2 GetCurve() [1/2]

```
Curve & gdcm::Pixmap::GetCurve (
    size_t i = 0 ) [inline]
```

[Curve](#): group 50xx.

### 10.236.3.3 GetCurve() [2/2]

```
const Curve & gdcm::Pixmap::GetCurve (
    size_t i = 0 ) const [inline]
```

### 10.236.3.4 GetIconImage() [1/2]

```
IconImage & gdcm::Pixmap::GetIconImage ( ) [inline]
```

### 10.236.3.5 GetIconImage() [2/2]

```
const IconImage & gdcm::Pixmap::GetIconImage ( ) const [inline]
```

Set/Get Icon [Image](#).

### 10.236.3.6 GetNumberOfCurves()

```
size_t gdcm::Pixmap::GetNumberOfCurves ( ) const [inline]
```

### 10.236.3.7 GetNumberOfOverlays()

```
size_t gdcm::Pixmap::GetNumberOfOverlays ( ) const [inline]
```

### 10.236.3.8 GetOverlay() [1/2]

```
Overlay & gdcm::Pixmap::GetOverlay (
    size_t i = 0 ) [inline]
```

[Overlay](#): group 60xx.

### 10.236.3.9 GetOverlay() [2/2]

```
const Overlay & gdcm::Pixmap::GetOverlay (
    size_t i = 0 ) const [inline]
```

### 10.236.3.10 Print()

```
void gdcm::Pixmap::Print (
    std::ostream & ) const [override], [virtual]
```

Reimplemented from [gdcm::Bitmap](#).

### 10.236.3.11 RemoveOverlay()

```
void gdcm::Pixmap::RemoveOverlay (
    size_t i ) [inline]
```

### 10.236.3.12 SetIconImage()

```
void gdcm::Pixmap::SetIconImage (
    IconImage const & ii ) [inline]
```

### 10.236.3.13 SetNumberOfCurves()

```
void gdcm::Pixmap::SetNumberOfCurves (
    size_t n ) [inline]
```

#### 10.236.3.14 SetNumberOfOverlays()

```
void gdcM::Pixmap::SetNumberOfOverlays (
    size_t n ) [inline]
```

#### 10.236.3.15 UnusedBitsPresentInPixelData()

```
bool gdcM::Pixmap::UnusedBitsPresentInPixelData ( ) const [override], [virtual]
```

returns if there are unused bits in the pixel data

Reimplemented from [gdcM::Bitmap](#).

### 10.236.4 Member Data Documentation

#### 10.236.4.1 Curves

```
std::vector<Curve> gdcM::Pixmap::Curves [protected]
```

#### 10.236.4.2 Icon

```
SmartPointer<IconImage> gdcM::Pixmap::Icon [protected]
```

#### 10.236.4.3 Overlays

```
std::vector<Overlay> gdcM::Pixmap::Overlays [protected]
```

The documentation for this class was generated from the following file:

- [gdcMPixmap.h](#)

## 10.237 gdcm::PixmapReader Class Reference

[PixmapReader](#).

```
#include <gdcmPixmapReader.h>
```

Inheritance diagram for gdcm::PixmapReader:



Collaboration diagram for `gdcm::PixmapReader`:



### Public Member Functions

- `PixmapReader ()`
- `~PixmapReader ()` override
- `Pixmap & GetPixmap ()`
- `const Pixmap & GetPixmap () const`  
*Return the read image (need to call `Read()` first)*
- `bool Read ()` override

### Public Member Functions inherited from `gdcm::Reader`

- `Reader ()`
- `virtual ~Reader ()`
- `bool CanRead () const`
- `File & GetFile ()`  
*Set/Get File.*
- `const File & GetFile () const`  
*Set/Get File.*
- `size_t GetStreamCurrentPosition () const`

- bool [ReadSelectedPrivateTags](#) (std::set< [PrivateTag](#) > const &ptags, bool readvalues=true)  
*Will only read the specified selected private tags.*
- bool [ReadSelectedTags](#) (std::set< [Tag](#) > const &tags, bool readvalues=true)  
*Will only read the specified selected tags.*
- bool [ReadUpToTag](#) (const [Tag](#) &tag, std::set< [Tag](#) > const &skiptags=std::set< [Tag](#) >())
- void [SetFile](#) ([File](#) &file)  
*Set/Get [File](#).*
- void [SetFileName](#) (const char \*filename\_native)
- void [SetStream](#) (std::istream &input\_stream)  
*Set the open-ed stream directly.*

### Protected Member Functions

- virtual bool [ReadACRNEMAIImage](#) ()
- virtual bool [ReadImage](#) ([MediaStorage](#) const &ms)
- bool [ReadImageInternal](#) ([MediaStorage](#) const &ms, bool handlepixeldata=true)

### Protected Member Functions inherited from [gdcm::Reader](#)

- std::istream \* [GetStreamPtr](#) () const
- bool [ReadDataSet](#) ()
- bool [ReadMetaInformation](#) ()
- bool [ReadPreamble](#) ()

### Protected Attributes

- [SmartPointer](#)< [Pixmap](#) > [PixelData](#)

### Protected Attributes inherited from [gdcm::Reader](#)

- [SmartPointer](#)< [File](#) > [F](#)

## 10.237.1 Detailed Description

[PixmapReader](#).

#### Note

its role is to convert the DICOM [DataSet](#) into a [Pixmap](#) representation By default it is also loading the lookup table and overlay when found as they impact the rendering or the image

See PS 3.3-2008, [Table C.7-11b](#) IMAGE PIXEL MACRO ATTRIBUTES for the list of attribute that belong to what gdcm calls a '[Pixmap](#)'

#### Warning

the API [ReadUpToTag](#) and [ReadSelectedTag](#)

#### See also

[Pixmap](#)

#### Examples

[StandardizeFiles.cs](#).

## 10.237.2 Constructor & Destructor Documentation

### 10.237.2.1 PixmapReader()

```
gdcm::PixmapReader::PixmapReader ( )
```

### 10.237.2.2 ~PixmapReader()

```
gdcm::PixmapReader::~~PixmapReader ( ) [override]
```

## 10.237.3 Member Function Documentation

### 10.237.3.1 GetPixmap() [1/2]

```
Pixmap & gdcm::PixmapReader::GetPixmap ( )
```

### 10.237.3.2 GetPixmap() [2/2]

```
const Pixmap & gdcm::PixmapReader::GetPixmap ( ) const
```

Return the read image (need to call [Read\(\)](#) first)

#### Examples

[StandardizeFiles.cs](#).

### 10.237.3.3 Read()

```
bool gdcm::PixmapReader::Read ( ) [override], [virtual]
```

Read the DICOM image. There are two reason for failure:

1. The input filename is not DICOM
2. The input DICOM file does not contains an [Pixmap](#).

Reimplemented from [gdcm::Reader](#).

#### Examples

[StandardizeFiles.cs](#).



### 10.237.3.4 ReadACRNEMAIImage()

```
virtual bool gdcm::PixmapReader::ReadACRNEMAIImage ( ) [protected], [virtual]
```

Reimplemented in [gdcm::ImageReader](#).

### 10.237.3.5 ReadImage()

```
virtual bool gdcm::PixmapReader::ReadImage (
    MediaStorage const & ms ) [protected], [virtual]
```

Reimplemented in [gdcm::ImageReader](#).

### 10.237.3.6 ReadImageInternal()

```
bool gdcm::PixmapReader::ReadImageInternal (
    MediaStorage const & ms,
    bool handlepixeldata = true ) [protected]
```

## 10.237.4 Member Data Documentation

### 10.237.4.1 PixelData

```
SmartPointer<Pixmap> gdcm::PixmapReader::PixelData [protected]
```

The documentation for this class was generated from the following file:

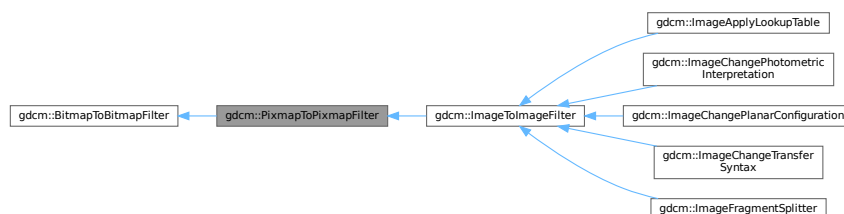
- [gdcmPixmapReader.h](#)

## 10.238 gdcm::PixmapToPixmapFilter Class Reference

[PixmapToPixmapFilter](#) class.

```
#include <gdcmPixmapToPixmapFilter.h>
```

Inheritance diagram for gdcm::PixmapToPixmapFilter:



Collaboration diagram for `gdcm::PixmapToPixmapFilter`:



### Public Member Functions

- `PixmapToPixmapFilter ()`
- `~PixmapToPixmapFilter ()=default`
- `Pixmap & GetInput ()`
- `const Pixmap & GetOutput () const`  
*Get Output image.*
- `const Pixmap & GetOutputAsPixmap () const`

### Public Member Functions inherited from `gdcm::BitmapToBitmapFilter`

- `BitmapToBitmapFilter ()`
- `~BitmapToBitmapFilter ()=default`
- `const Bitmap & GetOutput () const`  
*Get Output image.*
- `const Bitmap & GetOutputAsBitmap () const`
- `void SetInput (const Bitmap &image)`  
*Set input image.*

## Additional Inherited Members

## Protected Attributes inherited from [gdcm::BitmapToBitmapFilter](#)

- [SmartPointer](#)< [Bitmap](#) > Input
- [SmartPointer](#)< [Bitmap](#) > Output

### 10.238.1 Detailed Description

[PixmapToPixmapFilter](#) class.

Super class for all filter taking an image and producing an output image

#### Examples

[StandardizeFiles.cs](#).

### 10.238.2 Constructor & Destructor Documentation

#### 10.238.2.1 PixmapToPixmapFilter()

```
gdcm::PixmapToPixmapFilter::PixmapToPixmapFilter ( )
```

#### 10.238.2.2 ~PixmapToPixmapFilter()

```
gdcm::PixmapToPixmapFilter::~~PixmapToPixmapFilter ( ) [default]
```

### 10.238.3 Member Function Documentation

#### 10.238.3.1 GetInput()

```
Pixmap & gdcm::PixmapToPixmapFilter::GetInput ( )
```

#### 10.238.3.2 GetOutput()

```
const Pixmap & gdcm::PixmapToPixmapFilter::GetOutput ( ) const
```

Get Output image.

### 10.238.3.3 GetOutputAsPixmap()

```
const Pixmap & gdcM::PixmapToPixmapFilter::GetOutputAsPixmap ( ) const
```

The documentation for this class was generated from the following file:

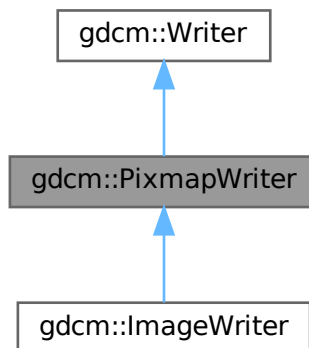
- [gdcMPixmapToPixmapFilter.h](#)

## 10.239 gdcM::PixmapWriter Class Reference

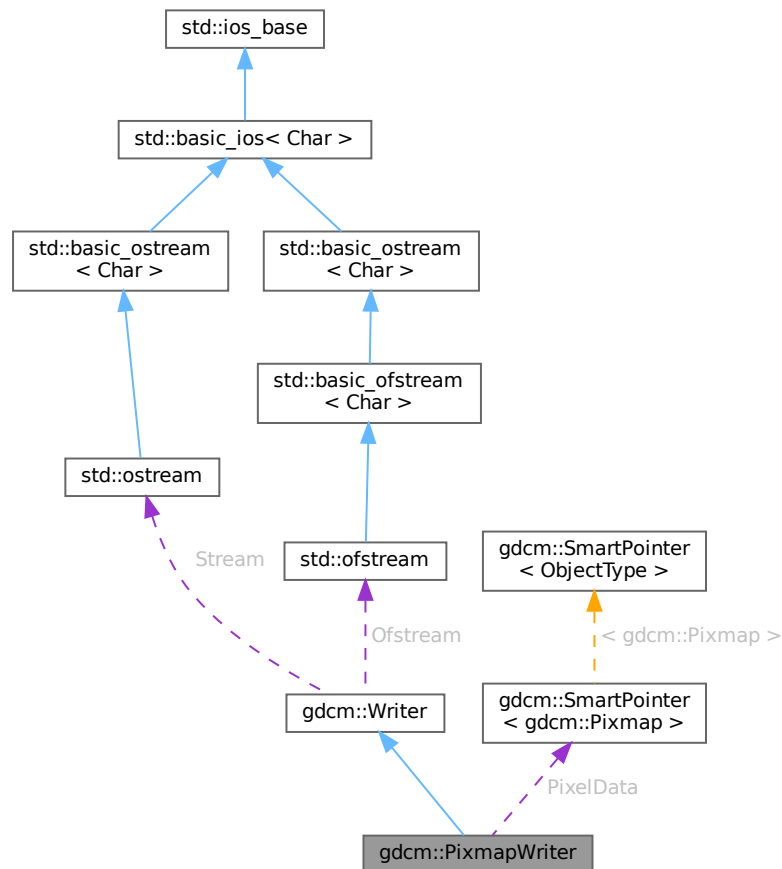
[PixmapWriter](#).

```
#include <gdcMPixmapWriter.h>
```

Inheritance diagram for gdcM::PixmapWriter:



Collaboration diagram for gdcm::PixmapWriter:



## Public Member Functions

- `PixmapWriter ()`
- `~PixmapWriter ()` override
- `virtual Pixmap & GetImage ()`
- `virtual const Pixmap & GetImage () const`
- `Pixmap & GetPixmap ()`
- `const Pixmap & GetPixmap () const`
- `virtual void SetImage (Pixmap const &img)`
- `void SetPixmap (Pixmap const &img)`
- `bool Write ()` override

*Write.*

## Public Member Functions inherited from [gdcm::Writer](#)

- [Writer](#) ()
- virtual [~Writer](#) ()
- void [CheckFileMetaInformationOff](#) ()
- void [CheckFileMetaInformationOn](#) ()
- [File](#) & [GetFile](#) ()
- void [SetCheckFileMetaInformation](#) (bool b)  
*Undocumented function, do not use (= leave default)*
- void [SetFile](#) (const [File](#) &f)  
*Set/Get the DICOM file ([DataSet](#) + Header)*
- void [SetFileName](#) (const char \*filename\_native)  
*Set the filename of DICOM file to write:*
- void [SetStream](#) (std::ostream &output\_stream)  
*Set user ostream buffer.*

## Protected Member Functions

- void [DolconImage](#) ([DataSet](#) &ds, [Pixmap](#) const &image)
- bool [PrepareWrite](#) ([MediaStorage](#) const &refms)

## Protected Member Functions inherited from [gdcm::Writer](#)

- bool [GetCheckFileMetaInformation](#) () const
- std::ostream \* [GetStreamPtr](#) () const
- void [SetWriteDataSetOnly](#) (bool b)

## Protected Attributes

- [SmartPointer](#)< [Pixmap](#) > [PixelData](#)

## Protected Attributes inherited from [gdcm::Writer](#)

- std::ofstream \* [Ofstream](#)
- std::ostream \* [Stream](#)

## 10.239.1 Detailed Description

[PixmapWriter](#).

This class will takes two inputs:

1. The DICOM [DataSet](#)
2. The [Image](#) input It will override any info from the [Image](#) over the [DataSet](#).

For instance when one read in a lossy compressed image and write out as unencapsulated (ie implicitly lossless) then some attribute are definitely needed to mark this dataset as Lossy (typically 0028,2114)

### Examples

[StandardizeFiles.cs](#).

## 10.239.2 Constructor & Destructor Documentation

### 10.239.2.1 PixmapWriter()

```
gdcm::PixmapWriter::PixmapWriter ( )
```

### 10.239.2.2 ~PixmapWriter()

```
gdcm::PixmapWriter::~~PixmapWriter ( ) [override]
```

## 10.239.3 Member Function Documentation

### 10.239.3.1 DoIconImage()

```
void gdcm::PixmapWriter::DoIconImage (
    DataSet & ds,
    Pixmap const & image ) [protected]
```

### 10.239.3.2 GetImage() [1/2]

```
virtual Pixmap & gdcm::PixmapWriter::GetImage ( ) [inline], [virtual]
```

Reimplemented in [gdcm::ImageWriter](#).

### 10.239.3.3 GetImage() [2/2]

```
virtual const Pixmap & gdcm::PixmapWriter::GetImage ( ) const [inline], [virtual]
```

Set/Get [Pixmap](#) to be written It will overwrite anything [Pixmap](#) infos found in [DataSet](#) (see parent class to see how to pass dataset)

Reimplemented in [gdcm::ImageWriter](#).

### 10.239.3.4 GetPixmap() [1/2]

```
Pixmap & gdcm::PixmapWriter::GetPixmap ( ) [inline]
```

### 10.239.3.5 GetPixmap() [2/2]

```
const Pixmap & gdcm::PixmapWriter::GetPixmap ( ) const [inline]
```

### 10.239.3.6 PrepareWrite()

```
bool gdcm::PixmapWriter::PrepareWrite (
    MediaStorage const & refms ) [protected]
```

### 10.239.3.7 SetImage()

```
virtual void gdcm::PixmapWriter::SetImage (
    Pixmap const & img ) [virtual]
```

#### Examples

[BasicImageAnonymizer.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [DecompressImage.cs](#), [GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [MergeTwoFiles.cxx](#), [MpegVideoInfo.cs](#), and [TemplateEmptyImage.cxx](#).

### 10.239.3.8 SetPixmap()

```
void gdcm::PixmapWriter::SetPixmap (
    Pixmap const & img )
```

#### Examples

[StandardizeFiles.cs](#).

### 10.239.3.9 Write()

```
bool gdcm::PixmapWriter::Write ( ) [override], [virtual]
```

Write.

Reimplemented from [gdcm::Writer](#).

#### Examples

[StandardizeFiles.cs](#).

## 10.239.4 Member Data Documentation

### 10.239.4.1 PixelData

```
SmartPointer<Pixmap> gdcm::PixmapWriter::PixelData [protected]
```

The documentation for this class was generated from the following file:

- [gdcmPixmapWriter.h](#)



## 10.240 gdcm::PNMCodec Class Reference

Class to do PNM.

```
#include <gdcmPNMCodec.h>
```

Inheritance diagram for gdcm::PNMCodec:



Collaboration diagram for gdcm::PNMCodec:



### Public Member Functions

- [PNMCodec](#) ()
- [~PNMCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const override  
*Return whether this coder support this transfer syntax (can code it)*

- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const override  
*Return whether this decoder support this transfer syntax (can decode it)*
- [ImageCodec](#) \* [Clone](#) () const override
- unsigned long [GetBufferLength](#) () const
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override
- bool [Read](#) (const char \*filename, [DataElement](#) &out) const
- void [SetBufferLength](#) (unsigned long l)
- bool [Write](#) (const char \*filename, const [DataElement](#) &out) const

### Public Member Functions inherited from [gdcm::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CleanupUnusedBits](#) (char \*data, size\_t datalen)
- bool [Decode](#) ([DataElement](#) const &is\_, [DataElement](#) &os) override  
*Decode.*
- const unsigned int \* [GetDimensions](#) () const
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- virtual void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)

### Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default
- virtual bool [Code](#) ([DataElement](#) const &in\_, [DataElement](#) &out\_)  
*Code.*

### Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

## Additional Inherited Members

## Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > LUTPtr

## Protected Member Functions inherited from [gdcm::ImageCodec](#)

- virtual bool [AppendFrameEncode](#) (std::ostream &out, const char \*data, size\_t datalen)
- virtual bool [AppendRowEncode](#) (std::ostream &out, const char \*data, size\_t datalen)
- bool [DecodeByStreams](#) (std::istream &is\_, std::ostream &os) override
- bool [DoByteSwap](#) (std::istream &is\_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is\_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is\_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is\_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is\_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is\_, std::ostream &os)
- virtual bool [IsFrameEncoder](#) ()
- virtual bool [IsRowEncoder](#) ()
- virtual bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)
- virtual bool [StartEncode](#) (std::ostream &os)
- virtual bool [StopEncode](#) (std::ostream &os)

## Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char \*bv, unsigned long len, std::ostream &os)

## Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) LUT
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) PF
- [PhotometricInterpretation](#) PI
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

### 10.240.1 Detailed Description

Class to do PNM.

PNM is the Portable anymap file format. The main web page can be found at: <http://netpbm.sourceforge.net/>

#### Note

Only support P5 & P6 PNM file (binary grayscale and binary rgb)

#### Examples

[ExtractIconFromFile.cxx](#).

### 10.240.2 Constructor & Destructor Documentation

#### 10.240.2.1 PNMCodec()

```
gdcmm::PNMCodec::PNMCodec ( )
```

#### 10.240.2.2 ~PNMCodec()

```
gdcmm::PNMCodec::~~PNMCodec ( ) [override]
```

### 10.240.3 Member Function Documentation

#### 10.240.3.1 CanCode()

```
bool gdcmm::PNMCodec::CanCode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcmm::ImageCodec](#).

#### 10.240.3.2 CanDecode()

```
bool gdcmm::PNMCodec::CanDecode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcmm::ImageCodec](#).

### 10.240.3.3 Clone()

```
ImageCodec * gdcm::PNMCodec::Clone ( ) const [override], [virtual]
```

Implements [gdcm::ImageCodec](#).

### 10.240.3.4 GetBufferLength()

```
unsigned long gdcm::PNMCodec::GetBufferLength ( ) const [inline]
```

### 10.240.3.5 GetHeaderInfo()

```
bool gdcm::PNMCodec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.240.3.6 Read()

```
bool gdcm::PNMCodec::Read (
    const char * filename,
    DataElement & out ) const
```

### 10.240.3.7 SetBufferLength()

```
void gdcm::PNMCodec::SetBufferLength (
    unsigned long l ) [inline]
```

### 10.240.3.8 Write()

```
bool gdcm::PNMCodec::Write (
    const char * filename,
    const DataElement & out ) const
```

#### Examples

[ExtractIconFromFile.cxx](#).

The documentation for this class was generated from the following file:

- [gdcmPNMCodec.h](#)

## 10.241 gdcm::Preamble Class Reference

DICOM [Preamble](#) (Part 10)

```
#include <gdcmPreamble.h>
```

### Public Member Functions

- [Preamble](#) ()
- [Preamble](#) ([Preamble](#) const &)
- [~Preamble](#) ()
- void [Clear](#) ()  
*Clear.*
- void [Create](#) ()
- const char \* [GetInternal](#) () const  
*Get internal pointer to preamble.*
- [VL GetLength](#) () const  
*Return size of [Preamble](#).*
- bool [IsEmpty](#) () const  
*Check if [Preamble](#) is empty.*
- [Preamble](#) & [operator=](#) ([Preamble](#) const &)
- void [Print](#) (std::ostream &os) const  
*Print [Preamble](#).*
- std::istream & [Read](#) (std::istream &is)  
*Read [Preamble](#).*
- void [Remove](#) ()
- void [Valid](#) ()  
*Set [Preamble](#) to the default one.*
- std::ostream const & [Write](#) (std::ostream &os) const  
*Write [Preamble](#).*

### Protected Member Functions

- bool [IsValid](#) () const

### Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [Preamble](#) &\_val)

### 10.241.1 Detailed Description

DICOM [Preamble](#) (Part 10)

## 10.241.2 Constructor & Destructor Documentation

### 10.241.2.1 Preamble() [1/2]

```
gdcm::Preamble::Preamble ( )
```

### 10.241.2.2 ~Preamble()

```
gdcm::Preamble::~~Preamble ( )
```

References [gdcm::operator<<\(\)](#).

### 10.241.2.3 Preamble() [2/2]

```
gdcm::Preamble::Preamble (
    Preamble const & ) [inline]
```

## 10.241.3 Member Function Documentation

### 10.241.3.1 Clear()

```
void gdcm::Preamble::Clear ( )
```

Clear.

### 10.241.3.2 Create()

```
void gdcm::Preamble::Create ( )
```

### 10.241.3.3 GetInternal()

```
const char * gdcm::Preamble::GetInternal ( ) const [inline]
```

Get internal pointer to preamble.

### 10.241.3.4 GetLength()

```
VL gdcm::Preamble::GetLength ( ) const [inline]
```

Return size of [Preamble](#).

#### 10.241.3.5 IsEmpty()

```
bool gdcM::Preamble::IsEmpty ( ) const [inline]
```

Check if [Preamble](#) is empty.

#### 10.241.3.6 IsValid()

```
bool gdcM::Preamble::IsValid ( ) const [inline], [protected]
```

#### 10.241.3.7 operator=()

```
Preamble & gdcM::Preamble::operator= (
    Preamble const & ) [inline]
```

#### 10.241.3.8 Print()

```
void gdcM::Preamble::Print (
    std::ostream & os ) const
```

Print [Preamble](#).

#### 10.241.3.9 Read()

```
std::istream & gdcM::Preamble::Read (
    std::istream & is )
```

Read [Preamble](#).

#### 10.241.3.10 Remove()

```
void gdcM::Preamble::Remove ( )
```

#### 10.241.3.11 Valid()

```
void gdcM::Preamble::Valid ( )
```

Set [Preamble](#) to the default one.



### 10.241.3.12 Write()

```
std::ostream const & gdcm::Preamble::Write (
    std::ostream & os ) const
```

Write [Preamble](#).

## 10.241.4 Friends And Related Symbol Documentation

### 10.241.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const Preamble & _val ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmPreamble.h](#)

## 10.242 gdcm::PresentationContext Class Reference

[PresentationContext](#).

```
#include <gdcmPresentationContext.h>
```

Collaboration diagram for gdcm::PresentationContext:



## Public Types

- typedef TransferSyntaxArrayType::size\_type [SizeType](#)
- typedef std::vector< std::string > [TransferSyntaxArrayType](#)

## Public Member Functions

- [PresentationContext](#) ()
- [PresentationContext](#) (UIDs::TSName asname, UIDs::TSName tsname=UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM)
- void [AddTransferSyntax](#) (const char \*tsstr)
- const char \* [GetAbstractSyntax](#) () const
- [SizeType](#) [GetNumberOfTransferSyntaxes](#) () const
- uint8\_t [GetPresentationContextID](#) () const
- const char \* [GetTransferSyntax](#) ([SizeType](#) i) const
- bool [operator==](#) (const [PresentationContext](#) &pc) const
- void [Print](#) (std::ostream &os) const
- void [SetAbstractSyntax](#) (const char \*absyn)
- void [SetPresentationContextID](#) (uint8\_t id)

## Protected Attributes

- std::string [AbstractSyntax](#)
- uint8\_t [ID](#)
- std::vector< std::string > [TransferSyntaxes](#)

## 10.242.1 Detailed Description

[PresentationContext](#).

See also

[PresentationContextAC](#) [PresentationContextRQ](#)

## 10.242.2 Member Typedef Documentation

### 10.242.2.1 SizeType

```
typedef TransferSyntaxArrayType::size_type gdcm::PresentationContext::SizeType
```

### 10.242.2.2 TransferSyntaxArrayType

```
typedef std::vector<std::string> gdcm::PresentationContext::TransferSyntaxArrayType
```

## 10.242.3 Constructor & Destructor Documentation

### 10.242.3.1 PresentationContext() [1/2]

```
gdcm::PresentationContext::PresentationContext ( )
```

### 10.242.3.2 PresentationContext() [2/2]

```
gdcm::PresentationContext::PresentationContext (
    UIDs::TSName asname,
    UIDs::TSName tsname = UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM )
```

Initialize Presentation Context with AbstractSyntax set to asname and with a single [TransferSyntax](#) set to tsname (default to Implicit [VR](#) LittleEndian when not specified ).

## 10.242.4 Member Function Documentation

### 10.242.4.1 AddTransferSyntax()

```
void gdcm::PresentationContext::AddTransferSyntax (
    const char * tsstr )
```

### 10.242.4.2 GetAbstractSyntax()

```
const char * gdcm::PresentationContext::GetAbstractSyntax ( ) const [inline]
```

### 10.242.4.3 GetNumberOfTransferSyntaxes()

```
SizeType gdcm::PresentationContext::GetNumberOfTransferSyntaxes ( ) const [inline]
```

### 10.242.4.4 GetPresentationContextID()

```
uint8_t gdcm::PresentationContext::GetPresentationContextID ( ) const
```

### 10.242.4.5 GetTransferSyntax()

```
const char * gdcm::PresentationContext::GetTransferSyntax (
    SizeType i ) const [inline]
```

#### 10.242.4.6 operator==()

```
bool gdcM::PresentationContext::operator== (
    const PresentationContext & pc ) const [inline]
```

References [AbstractSyntax](#), and [TransferSyntaxes](#).

#### 10.242.4.7 Print()

```
void gdcM::PresentationContext::Print (
    std::ostream & os ) const
```

#### 10.242.4.8 SetAbstractSyntax()

```
void gdcM::PresentationContext::SetAbstractSyntax (
    const char * absyn ) [inline]
```

#### 10.242.4.9 SetPresentationContextID()

```
void gdcM::PresentationContext::SetPresentationContextID (
    uint8_t id )
```

### 10.242.5 Member Data Documentation

#### 10.242.5.1 AbstractSyntax

```
std::string gdcM::PresentationContext::AbstractSyntax [protected]
```

Referenced by [operator==\(\)](#).

#### 10.242.5.2 ID

```
uint8_t gdcM::PresentationContext::ID [protected]
```

#### 10.242.5.3 TransferSyntaxes

```
std::vector<std::string> gdcM::PresentationContext::TransferSyntaxes [protected]
```

Referenced by [operator==\(\)](#).

The documentation for this class was generated from the following file:

- [gdcMPresentationContext.h](#)

## 10.243 gdcm::network::PresentationContextAC Class Reference

[PresentationContextAC](#).

```
#include <gdcmPresentationContextAC.h>
```

### Public Member Functions

- [PresentationContextAC](#) ()
- uint8\_t [GetPresentationContextID](#) () const
- uint8\_t [GetReason](#) () const
- [TransferSyntaxSub](#) const & [GetTransferSyntax](#) () const
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [SetPresentationContextID](#) (uint8\_t id)
- void [SetReason](#) (uint8\_t r)
- void [SetTransferSyntax](#) ([TransferSyntaxSub](#) const &ts)
- size\_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

### 10.243.1 Detailed Description

[PresentationContextAC](#).

[Table](#) 9-18 PRESENTATION CONTEXT ITEM FIELDS

See also

[PresentationContext](#)

### 10.243.2 Constructor & Destructor Documentation

#### 10.243.2.1 PresentationContextAC()

```
gdcm::network::PresentationContextAC::PresentationContextAC ( )
```

### 10.243.3 Member Function Documentation

#### 10.243.3.1 GetPresentationContextID()

```
uint8_t gdcm::network::PresentationContextAC::GetPresentationContextID ( ) const [inline]
```

#### 10.243.3.2 GetReason()

```
uint8_t gdcM::network::PresentationContextAC::GetReason ( ) const [inline]
```

#### 10.243.3.3 GetTransferSyntax()

```
TransferSyntaxSub const & gdcM::network::PresentationContextAC::GetTransferSyntax ( ) const [inline]
```

#### 10.243.3.4 Print()

```
void gdcM::network::PresentationContextAC::Print (
    std::ostream & os ) const
```

#### 10.243.3.5 Read()

```
std::istream & gdcM::network::PresentationContextAC::Read (
    std::istream & is )
```

#### 10.243.3.6 SetPresentationContextID()

```
void gdcM::network::PresentationContextAC::SetPresentationContextID (
    uint8_t id )
```

#### 10.243.3.7 SetReason()

```
void gdcM::network::PresentationContextAC::SetReason (
    uint8_t r ) [inline]
```

#### 10.243.3.8 SetTransferSyntax()

```
void gdcM::network::PresentationContextAC::SetTransferSyntax (
    TransferSyntaxSub const & ts )
```

#### 10.243.3.9 Size()

```
size_t gdcM::network::PresentationContextAC::Size ( ) const
```

### 10.243.3.10 Write()

```
const std::ostream & gdcm::network::PresentationContextAC::Write (
    std::ostream & os ) const
```

The documentation for this class was generated from the following file:

- [gdcmPresentationContextAC.h](#)

## 10.244 gdcm::PresentationContextGenerator Class Reference

[PresentationContextGenerator](#).

```
#include <gdcmPresentationContextGenerator.h>
```

### Public Types

- typedef std::vector< [PresentationContext](#) > [PresentationContextArrayType](#)
- typedef PresentationContextArrayType::size\_type [SizeType](#)

### Public Member Functions

- [PresentationContextGenerator](#) ()
- bool [AddFromFile](#) (const [File](#) &file)
- bool [GenerateFromFilenames](#) (const [Directory::FilenamesType](#) &files)
- bool [GenerateFromUID](#) ([UIDs::TSName](#) asname)  
*Generate the [PresentationContext](#) array from a UID (eg. [VerificationSOPClass](#))*
- [PresentationContextArrayType](#) const & [GetPresentationContexts](#) ()
- void [SetDefaultTransferSyntax](#) (const [TransferSyntax](#) &ts)  
*Not implemented for now. GDCM internally uses Implicit Little Endian.*
- void [SetMergeModeToAbstractSyntax](#) ()
- void [SetMergeModeToTransferSyntax](#) ()

### Protected Member Functions

- bool [AddPresentationContext](#) (const char \*absyn, const char \*ts)
- const char \* [GetDefaultTransferSyntax](#) () const

## 10.244.1 Detailed Description

### [PresentationContextGenerator](#).

This class is responsible for generating the proper [PresentationContext](#) that will be used in subsequent operation during a DICOM Query/Retrieve association. The step of the association is very sensible as special care need to be taken to explicitly define what instance are going to be send and how they are encoded.

For example a [PresentationContext](#) will express that negotiation requires that CT [Image](#) Storage are send using JPEG Lossless, while US [Image](#) Storage are sent using RLE Transfer Syntax.

Two very different API are exposed one which will always default to little endian transfer syntax see [GenerateFromUID\(\)](#) This API is used for C-ECHO, C-FIND and C-MOVE (SCU). Another API: [GenerateFromFilenames\(\)](#) is used for C-STORE (SCU) as it will loop over all filenames argument to detect the actual encoding. and therefore find the proper encoding to be used.

Two modes are available. The default mode (SetMergeModeToAbstractSyntax) append [PresentationContext](#) (one AbstractSyntax and one [TransferSyntax](#)), as long a they are different. Eg MR [Image](#) Storage/JPEG2000 and MR [Image](#) Storage/JPEGLossless would be considered different. the other mode SetMergeModeToTransferSyntax merge any new [TransferSyntax](#) to the already existing [PresentationContext](#) in order to re-use the same AbstractSyntax.

See also

[PresentationContext](#)

Examples

[CStoreQtProgress.cxx](#).

## 10.244.2 Member Typedef Documentation

### 10.244.2.1 [PresentationContextArrayType](#)

```
typedef std::vector<PresentationContext> gdcm::PresentationContextGenerator::PresentationContextArrayType
```

### 10.244.2.2 [SizeType](#)

```
typedef PresentationContextArrayType::size\_type gdcm::PresentationContextGenerator::SizeType
```

## 10.244.3 Constructor & Destructor Documentation

### 10.244.3.1 [PresentationContextGenerator\(\)](#)

```
gdcm::PresentationContextGenerator::PresentationContextGenerator ( )
```



## 10.244.4 Member Function Documentation

### 10.244.4.1 AddFromFile()

```
bool gdcm::PresentationContextGenerator::AddFromFile (
    const File & file )
```

Add a single [PresentationContext](#) from a single [File](#). Call multiple times when dealing with multiple files.

### 10.244.4.2 AddPresentationContext()

```
bool gdcm::PresentationContextGenerator::AddPresentationContext (
    const char * absyn,
    const char * ts ) [protected]
```

### 10.244.4.3 GenerateFromFileNames()

```
bool gdcm::PresentationContextGenerator::GenerateFromFileNames (
    const Directory::FileNamesType & files )
```

Generate the [PresentationContext](#) array from a File-Set. [File](#) specified needs to be valid DICOM files. Used for C-STORE operations

#### Examples

[CStoreQtProgress.cxx](#).

### 10.244.4.4 GenerateFromUID()

```
bool gdcm::PresentationContextGenerator::GenerateFromUID (
    UIDs::TSName asname )
```

Generate the [PresentationContext](#) array from a UID (eg. VerificationSOPClass)

### 10.244.4.5 GetDefaultTransferSyntax()

```
const char * gdcm::PresentationContextGenerator::GetDefaultTransferSyntax ( ) const [protected]
```

### 10.244.4.6 GetPresentationContexts()

```
PresentationContextArrayType const & gdcm::PresentationContextGenerator::GetPresentationContexts (
) [inline]
```

#### Examples

[CStoreQtProgress.cxx](#).

**10.244.4.7 SetDefaultTransferSyntax()**

```
void gdcmm::PresentationContextGenerator::SetDefaultTransferSyntax (
    const TransferSyntax & ts )
```

Not implemented for now. GDCM internally uses Implicit Little Endian.

**10.244.4.8 SetMergeModeToAbstractSyntax()**

```
void gdcmm::PresentationContextGenerator::SetMergeModeToAbstractSyntax ( )
```

**10.244.4.9 SetMergeModeToTransferSyntax()**

```
void gdcmm::PresentationContextGenerator::SetMergeModeToTransferSyntax ( )
```

The documentation for this class was generated from the following file:

- [gdcmmPresentationContextGenerator.h](#)

**10.245 gdcmm::network::PresentationContextRQ Class Reference**

[PresentationContextRQ](#).

```
#include <gdcmmPresentationContextRQ.h>
```

**Public Types**

- typedef std::vector< [TransferSyntaxSub](#) >::size\_type [SizeType](#)

**Public Member Functions**

- [PresentationContextRQ](#) ()
- [PresentationContextRQ](#) (const [PresentationContext](#) &pc)
- [PresentationContextRQ](#) (UIDs::TSName asname, UIDs::TSName tsname=UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM)
- void [AddTransferSyntax](#) ([TransferSyntaxSub](#) const &ts)
- [AbstractSyntax](#) & [GetAbstractSyntax](#) ()
- [AbstractSyntax](#) const & [GetAbstractSyntax](#) () const
- [SizeType](#) [GetNumberOfTransferSyntaxes](#) () const
- uint8\_t [GetPresentationContextID](#) () const
- [TransferSyntaxSub](#) & [GetTransferSyntax](#) ([SizeType](#) i)
- [TransferSyntaxSub](#) const & [GetTransferSyntax](#) ([SizeType](#) i) const
- std::vector< [TransferSyntaxSub](#) > const & [GetTransferSyntaxes](#) () const
- bool [operator==](#) (const [PresentationContextRQ](#) &pc) const
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [SetAbstractSyntax](#) ([AbstractSyntax](#) const &absyn)
- void [SetPresentationContextID](#) (uint8\_t id)
- size\_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

## 10.245.1 Detailed Description

[PresentationContextRQ](#).

[Table](#) 9-13 PRESENTATION CONTEXT ITEM FIELDS

See also

[PresentationContextAC](#)

## 10.245.2 Member Typedef Documentation

### 10.245.2.1 SizeType

```
typedef std::vector<TransferSyntaxSub>::size_type gdcm::network::PresentationContextRQ::SizeType
```

## 10.245.3 Constructor & Destructor Documentation

### 10.245.3.1 PresentationContextRQ() [1/3]

```
gdcm::network::PresentationContextRQ::PresentationContextRQ ( )
```

### 10.245.3.2 PresentationContextRQ() [2/3]

```
gdcm::network::PresentationContextRQ::PresentationContextRQ (
    UIDs::TSName asname,
    UIDs::TSName tname = UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM )
```

Initialize Presentation Context with [AbstractSyntax](#) set to asname and with a single [TransferSyntax](#) set to tname (default to Implicit [VR](#) LittleEndian when not specified ).

### 10.245.3.3 PresentationContextRQ() [3/3]

```
gdcm::network::PresentationContextRQ::PresentationContextRQ (
    const PresentationContext & pc )
```

## 10.245.4 Member Function Documentation

### 10.245.4.1 AddTransferSyntax()

```
void gdcm::network::PresentationContextRQ::AddTransferSyntax (
    TransferSyntaxSub const & ts )
```

**10.245.4.2 GetAbstractSyntax() [1/2]**

```
AbstractSyntax & gdc::network::PresentationContextRQ::GetAbstractSyntax ( ) [inline]
```

**10.245.4.3 GetAbstractSyntax() [2/2]**

```
AbstractSyntax const & gdc::network::PresentationContextRQ::GetAbstractSyntax ( ) const [inline]
```

**10.245.4.4 GetNumberOfTransferSyntaxes()**

```
SizeType gdc::network::PresentationContextRQ::GetNumberOfTransferSyntaxes ( ) const [inline]
```

**10.245.4.5 GetPresentationContextID()**

```
uint8_t gdc::network::PresentationContextRQ::GetPresentationContextID ( ) const
```

**10.245.4.6 GetTransferSyntax() [1/2]**

```
TransferSyntaxSub & gdc::network::PresentationContextRQ::GetTransferSyntax (
    SizeType i ) [inline]
```

**10.245.4.7 GetTransferSyntax() [2/2]**

```
TransferSyntaxSub const & gdc::network::PresentationContextRQ::GetTransferSyntax (
    SizeType i ) const [inline]
```

**10.245.4.8 GetTransferSyntaxes()**

```
std::vector< TransferSyntaxSub > const & gdc::network::PresentationContextRQ::GetTransfer←
Syntaxes ( ) const [inline]
```

**10.245.4.9 operator==( )**

```
bool gdc::network::PresentationContextRQ::operator== (
    const PresentationContextRQ & pc ) const [inline]
```

**10.245.4.10 Print()**

```
void gdc::network::PresentationContextRQ::Print (
    std::ostream & os ) const
```

#### 10.245.4.11 Read()

```
std::istream & gdcm::network::PresentationContextRQ::Read (
    std::istream & is )
```

#### 10.245.4.12 SetAbstractSyntax()

```
void gdcm::network::PresentationContextRQ::SetAbstractSyntax (
    AbstractSyntax const & absyn )
```

#### 10.245.4.13 SetPresentationContextID()

```
void gdcm::network::PresentationContextRQ::SetPresentationContextID (
    uint8_t id )
```

#### 10.245.4.14 Size()

```
size_t gdcm::network::PresentationContextRQ::Size ( ) const
```

#### 10.245.4.15 Write()

```
const std::ostream & gdcm::network::PresentationContextRQ::Write (
    std::ostream & os ) const
```

The documentation for this class was generated from the following file:

- [gdcmPresentationContextRQ.h](#)

## 10.246 gdcm::network::PresentationDataValue Class Reference

[PresentationDataValue](#).

```
#include <gdcmPresentationDataValue.h>
```

## Public Member Functions

- [PresentationDataValue](#) ()
- const std::string & [GetBlob](#) () const
- bool [GetIsCommand](#) () const
- bool [GetIsLastFragment](#) () const
- uint8\_t [GetMessageHeader](#) () const
- uint8\_t [GetPresentationContextID](#) () const
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- std::istream & [ReadInto](#) (std::istream &is, std::ostream &os)
- void [SetBlob](#) (const std::string &partialblob)
- void [SetCommand](#) (bool inCommand)
- void [SetDataSet](#) (const [DataSet](#) &ds)
- void [SetLastFragment](#) (bool inLast)
- void [SetMessageHeader](#) (uint8\_t messageheader)
- void [SetPresentationContextID](#) (uint8\_t id)
- size\_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

## Static Public Member Functions

- static [DataSet](#) [ConcatenatePDVBlobs](#) (const std::vector< [PresentationDataValue](#) > &inPDVs)
- static [DataSet](#) [ConcatenatePDVBlobsAsExplicit](#) (const std::vector< [PresentationDataValue](#) > &inPDVs)

## 10.246.1 Detailed Description

[PresentationDataValue](#).

[Table](#) 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS

## 10.246.2 Constructor & Destructor Documentation

### 10.246.2.1 PresentationDataValue()

```
gdcmm::network::PresentationDataValue::PresentationDataValue ( )
```

## 10.246.3 Member Function Documentation

### 10.246.3.1 ConcatenatePDVBlobs()

```
static DataSet gdcmm::network::PresentationDataValue::ConcatenatePDVBlobs (
    const std::vector< PresentationDataValue > & inPDVs ) [static]
```

#### Warning

[DataSet](#) will be read as Implicit Little Endian TS

### 10.246.3.2 ConcatenatePDVBlobsAsExplicit()

```
static DataSet gdcm::network::PresentationDataValue::ConcatenatePDVBlobsAsExplicit (
    const std::vector< PresentationDataValue > & inPDVs ) [static]
```

### 10.246.3.3 GetBlob()

```
const std::string & gdcm::network::PresentationDataValue::GetBlob ( ) const
```

### 10.246.3.4 GetIsCommand()

```
bool gdcm::network::PresentationDataValue::GetIsCommand ( ) const
```

### 10.246.3.5 GetIsLastFragment()

```
bool gdcm::network::PresentationDataValue::GetIsLastFragment ( ) const
```

### 10.246.3.6 GetMessageHeader()

```
uint8_t gdcm::network::PresentationDataValue::GetMessageHeader ( ) const [inline]
```

### 10.246.3.7 GetPresentationContextID()

```
uint8_t gdcm::network::PresentationDataValue::GetPresentationContextID ( ) const [inline]
```

### 10.246.3.8 Print()

```
void gdcm::network::PresentationDataValue::Print (
    std::ostream & os ) const
```

### 10.246.3.9 Read()

```
std::istream & gdcm::network::PresentationDataValue::Read (
    std::istream & is )
```

### 10.246.3.10 ReadInto()

```
std::istream & gdcm::network::PresentationDataValue::ReadInto (
    std::istream & is,
    std::ostream & os )
```

**10.246.3.11 SetBlob()**

```
void gdcm::network::PresentationDataValue::SetBlob (
    const std::string & partialblob )
```

**10.246.3.12 SetCommand()**

```
void gdcm::network::PresentationDataValue::SetCommand (
    bool inCommand )
```

**10.246.3.13 SetDataSet()**

```
void gdcm::network::PresentationDataValue::SetDataSet (
    const DataSet & ds )
```

Set DataSet. Write DataSet in implicit.

**Warning**

size of dataset should be below maxpdusize

**10.246.3.14 SetLastFragment()**

```
void gdcm::network::PresentationDataValue::SetLastFragment (
    bool inLast )
```

**10.246.3.15 SetMessageHeader()**

```
void gdcm::network::PresentationDataValue::SetMessageHeader (
    uint8_t messageheader ) [inline]
```

**10.246.3.16 SetPresentationContextID()**

```
void gdcm::network::PresentationDataValue::SetPresentationContextID (
    uint8_t id ) [inline]
```

**10.246.3.17 Size()**

```
size_t gdcm::network::PresentationDataValue::Size ( ) const
```



### 10.246.3.18 Write()

```
const std::ostream & gdcm::network::PresentationDataValue::Write (
    std::ostream & os ) const
```

The documentation for this class was generated from the following file:

- [gdcmPresentationDataValue.h](#)

## 10.247 gdcm::Printer Class Reference

[Printer](#) class.

```
#include <gdcmPrinter.h>
```

Inheritance diagram for gdcm::Printer:



Collaboration diagram for gdcm::Printer:



**Public Types**

- enum [PrintStyles](#) {  
[VERBOSE\\_STYLE](#) = 0 ,  
[CONDENSED\\_STYLE](#) ,  
[XML](#) ,  
[CXX](#) }

**Public Member Functions**

- [Printer](#) ()
- [~Printer](#) ()=default
- [PrintStyles](#) [GetPrintStyle](#) () const  
*Get PrintStyle value.*
- void [Print](#) (std::ostream &os)  
*Print.*
- void [PrintDataSet](#) (const [DataSet](#) &ds, std::ostream &os, const std::string &s="")  
*Print an individual dataset.*
- void [SetColor](#) (bool c)  
*Set color mode or not.*
- void [SetFile](#) ([File](#) const &f)  
*Set file.*
- void [SetStyle](#) ([PrintStyles](#) ps)  
*Set PrintStyle value.*

**Protected Member Functions**

- [VR](#) [PrintDataElement](#) (std::ostringstream &os, const [Dicts](#) &dicts, const [DataSet](#) &ds, const [DataElement](#) &de, std::ostream &out, std::string const &indent)
- void [PrintSQ](#) (const [SequenceOfItems](#) \*sqi, std::ostream &os, std::string const &indent)

**Protected Attributes**

- const [File](#) \* [F](#)
- [VL](#) [MaxPrintLength](#)
- [PrintStyles](#) [PrintStyle](#)

**10.247.1 Detailed Description**

[Printer](#) class.

**Examples**

[DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), and [DumpToshibaDTI2.cxx](#).

**10.247.2 Member Enumeration Documentation****10.247.2.1 PrintStyles**

```
enum gdcm::Printer::PrintStyles
```

## Enumerator

VERBOSE_STYLE	
CONDENSED_STYLE	
XML	
CXX	

## 10.247.3 Constructor & Destructor Documentation

### 10.247.3.1 Printer()

```
gdcm::Printer::Printer ( )
```

### 10.247.3.2 ~Printer()

```
gdcm::Printer::~~Printer ( ) [default]
```

## 10.247.4 Member Function Documentation

### 10.247.4.1 GetPrintStyle()

```
PrintStyle gdcm::Printer::GetPrintStyle ( ) const [inline]
```

Get PrintStyle value.

### 10.247.4.2 Print()

```
void gdcm::Printer::Print (
    std::ostream & os )
```

Print.

#### Examples

[DumpSiemensBase64.cxx](#).

### 10.247.4.3 PrintDataElement()

```
VR gdcm::Printer::PrintDataElement (
    std::ostringstream & os,
    const Dicts & dicts,
    const DataSet & ds,
    const DataElement & de,
    std::ostream & out,
    std::string const & indent ) [protected]
```

#### 10.247.4.4 PrintDataSet()

```
void gdcM::Printer::PrintDataSet (
    const DataSet & ds,
    std::ostream & os,
    const std::string & s = "" )
```

Print an individual dataset.

#### 10.247.4.5 PrintSQ()

```
void gdcM::Printer::PrintSQ (
    const SequenceOfItems * sqi,
    std::ostream & os,
    std::string const & indent ) [protected]
```

#### 10.247.4.6 SetColor()

```
void gdcM::Printer::SetColor (
    bool c )
```

Set color mode or not.

#### 10.247.4.7 SetFile()

```
void gdcM::Printer::SetFile (
    File const & f ) [inline]
```

Set file.

#### Examples

[DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), and [DumpToshibaDTI2.cxx](#).

#### 10.247.4.8 SetStyle()

```
void gdcM::Printer::SetStyle (
    PrintStyles ps ) [inline]
```

Set PrintStyle value.

### 10.247.5 Member Data Documentation

#### 10.247.5.1 F

```
const File* gdcM::Printer::F [protected]
```

### 10.247.5.2 MaxPrintLength

VL gdcm::Printer::MaxPrintLength [protected]

### 10.247.5.3 PrintStyle

PrintStyles gdcm::Printer::PrintStyle [protected]

The documentation for this class was generated from the following file:

- [gdcmPrinter.h](#)

## 10.248 gdcm::PrivateDict Class Reference

Private [Dict](#).

```
#include <gdcmDict.h>
```

### Public Member Functions

- [PrivateDict](#) ()=default
- [~PrivateDict](#) ()=default
- void [AddDictEntry](#) (const [PrivateTag](#) &tag, const [DictEntry](#) &de)
- bool [FindDictEntry](#) (const [PrivateTag](#) &tag) const
- const [DictEntry](#) & [GetDictEntry](#) (const [PrivateTag](#) &tag) const
- bool [IsEmpty](#) () const
- void [PrintXML](#) () const
- bool [RemoveDictEntry](#) (const [PrivateTag](#) &tag)

### Protected Member Functions

- void [LoadDefault](#) ()

### Friends

- class [Dicts](#)
- std::ostream & [operator<<](#) (std::ostream &os, const [PrivateDict](#) &val)

## 10.248.1 Detailed Description

Private [Dict](#).

## 10.248.2 Constructor & Destructor Documentation

### 10.248.2.1 PrivateDict()

```
gdcm::PrivateDict::PrivateDict ( ) [default]
```

### 10.248.2.2 ~PrivateDict()

```
gdcm::PrivateDict::~~PrivateDict ( ) [default]
```

## 10.248.3 Member Function Documentation

### 10.248.3.1 AddDictEntry()

```
void gdcm::PrivateDict::AddDictEntry (
    const PrivateTag & tag,
    const DictEntry & de ) [inline]
```

References [gdcm::DictEntry::GetVM\(\)](#), [gdcm::DictEntry::GetVR\(\)](#), [gdcm::DictEntry::SetVM\(\)](#), and [gdcm::DictEntry::SetVR\(\)](#).

### 10.248.3.2 FindDictEntry()

```
bool gdcm::PrivateDict::FindDictEntry (
    const PrivateTag & tag ) const [inline]
```

### 10.248.3.3 GetDictEntry()

```
const DictEntry & gdcm::PrivateDict::GetDictEntry (
    const PrivateTag & tag ) const [inline]
```

### 10.248.3.4 IsEmpty()

```
bool gdcm::PrivateDict::IsEmpty ( ) const [inline]
```

### 10.248.3.5 LoadDefault()

```
void gdcm::PrivateDict::LoadDefault ( ) [protected]
```

### 10.248.3.6 PrintXML()

```
void gdcm::PrivateDict::PrintXML ( ) const [inline]
```

References [gdcm::Tag::GetElement\(\)](#), [gdcm::Tag::GetGroup\(\)](#), [gdcm::DictEntry::GetName\(\)](#), [gdcm::PrivateTag::GetOwner\(\)](#), [gdcm::DictEntry::GetVM\(\)](#), and [gdcm::DictEntry::GetVR\(\)](#).

### 10.248.3.7 RemoveDictEntry()

```
bool gdcm::PrivateDict::RemoveDictEntry (
    const PrivateTag & tag ) [inline]
```

Remove entry 'tag'. Return true on success (element was found and remove). return false if element was not found.

## 10.248.4 Friends And Related Symbol Documentation

### 10.248.4.1 Dicts

```
friend class Dicts [friend]
```

### 10.248.4.2 operator<<

```
std::ostream & operator<< (
    std::ostream & os,
    const PrivateDict & val ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmDict.h](#)

## 10.249 gdcm::PrivateTag Class Reference

Class to represent a Private DICOM Data [Element](#) ([Attribute](#)) [Tag](#) (Group, [Element](#), Owner)

```
#include <gdcmPrivateTag.h>
```

Inheritance diagram for `gdcm::PrivateTag`:



Collaboration diagram for `gdcm::PrivateTag`:



## Public Member Functions

- `PrivateTag` (`Tag` const &t, const char \*owner="")
- `PrivateTag` (uint16\_t group=0, uint16\_t element=0, const char \*owner="")
- `DataElement GetAsDataElement` () const
- const char \* `GetOwner` () const
- bool `operator!=` (const `PrivateTag` &\_val) const
- bool `operator!=` (const `Tag` &\_val) const
- bool `operator<` (const `PrivateTag` &\_val) const
- `PrivateTag` & `operator=` (const `PrivateTag` &\_val)
- bool `operator==` (const `PrivateTag` &\_val) const
- bool `operator==` (const `Tag` &\_val) const
- bool `ReadFromCommaSeparatedString` (const char \*str)
- void `SetOwner` (const char \*owner)



## Public Member Functions inherited from `gdcm::Tag`

- `Tag` (const `Tag` &\_val)
- `Tag` (uint16\_t group, uint16\_t element)  
*Constructor with 2\*uint16\_t.*
- `Tag` (uint32\_t tag=0)  
*Constructor with 1\*uint32\_t Prefer the ctor that takes two uint16\_t.*
- uint16\_t `GetElement` () const  
*Returns the 'Element number' of the given Tag.*
- uint32\_t `GetElementTag` () const  
*Returns the full tag value of the given Tag.*
- uint16\_t `GetGroup` () const  
*Returns the 'Group number' of the given Tag.*
- uint32\_t `GetLength` () const  
*return the length of tag (read: size on disk)*
- `Tag` `GetPrivateCreator` () const  
*Return the Private Creator Data Element tag of a private data element.*
- bool `IsGroupLength` () const  
*return whether the tag correspond to a group length tag:*
- bool `IsGroupXX` (const `Tag` &t) const  
*e.g 6002,3000 belong to groupXX: 6000,3000*
- bool `IsIllegal` () const  
*return if the tag is considered to be an illegal tag*
- bool `IsPrivate` () const
- bool `IsPrivateCreator` () const
- bool `IsPublic` () const
- bool `operator!=` (const `Tag` &\_val) const
- bool `operator<` (const `Tag` &\_val) const
- bool `operator<=` (const `Tag` &t2) const
- `Tag` & `operator=` (const `Tag` &\_val)
- bool `operator==` (const `Tag` &\_val) const
- uint16\_t & `operator[]` (const unsigned int &\_id)  
*Returns the Group or Element of the given Tag, depending on id (0/1)*
- const uint16\_t & `operator[]` (const unsigned int &\_id) const  
*Returns the Group or Element of the given Tag, depending on id (0/1)*
- std::string `PrintAsContinuousString` () const
- std::string `PrintAsContinuousUpperCaseString` () const  
*Same as PrintAsContinuousString, but hexadecimal [a-f] are printed using upper case.*
- std::string `PrintAsPipeSeparatedString` () const
- template<typename TSwap >  
std::istream & `Read` (std::istream &is)  
*Read a tag from binary representation.*
- bool `ReadFromCommaSeparatedString` (const char \*str)
- bool `ReadFromContinuousString` (const char \*str)
- bool `ReadFromPipeSeparatedString` (const char \*str)
- void `SetElement` (uint16\_t element)  
*Sets the 'Element number' of the given Tag.*
- void `SetElementTag` (uint16\_t group, uint16\_t element)

- Sets the 'Group number' & 'Element number' of the given [Tag](#).
- void [SetElementTag](#) (uint32\_t tag)  
Sets the full tag value of the given [Tag](#).
- void [SetGroup](#) (uint16\_t group)  
Sets the 'Group number' of the given [Tag](#).
- void [SetPrivateCreator](#) ([Tag](#) const &t)  
Set private creator:
- template<typename TSwap >  
const std::ostream & [Write](#) (std::ostream &os) const  
Write a tag in binary rep.

## Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [PrivateTag](#) &\_val)

## 10.249.1 Detailed Description

Class to represent a Private DICOM Data [Element](#) ([Attribute](#)) [Tag](#) (Group, [Element](#), Owner)

### Note

private tag have element value in: [0x10,0xff], for instance 0x0009,0x0000 is NOT a private tag

### Examples

[ChangePrivateTags.cxx](#), [Cleaner.cs](#), [DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [ELSCINT1WaveToText.cxx](#), [FileStreaming.cs](#), [GetSubSequenceData.cxx](#), [MrProtocol.cxx](#), [PublicDict.cxx](#), [ReadGEMSSDO.cxx](#), [csa2img.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

## 10.249.2 Constructor & Destructor Documentation

### 10.249.2.1 PrivateTag() [1/2]

```
gdcm::PrivateTag::PrivateTag (
    uint16_t group = 0,
    uint16_t element = 0,
    const char * owner = "" ) [inline]
```

### 10.249.2.2 PrivateTag() [2/2]

```
gdcm::PrivateTag::PrivateTag (
    Tag const & t,
    const char * owner = "" ) [inline]
```

References [gdcm::Tag::GetElement\(\)](#).

## 10.249.3 Member Function Documentation

### 10.249.3.1 GetAsDataElement()

```
DataElement gdcm::PrivateTag::GetAsDataElement ( ) const
```

### 10.249.3.2 GetOwner()

```
const char * gdcm::PrivateTag::GetOwner ( ) const [inline]
```

#### Examples

[PublicDict.cxx](#).

Referenced by [gdcm::PrivateDict::PrintXML\(\)](#).

### 10.249.3.3 operator"!="() [1/2]

```
bool gdcm::PrivateTag::operator!= (
    const PrivateTag & _val ) const [inline]
```

References [gdcm::Tag::GetElementTag\(\)](#).

### 10.249.3.4 operator"!="() [2/2]

```
bool gdcm::PrivateTag::operator!= (
    const Tag & _val ) const [inline]
```

References [gdcm::Tag::GetElementTag\(\)](#).

### 10.249.3.5 operator<()

```
bool gdcm::PrivateTag::operator< (
    const PrivateTag & _val ) const
```

### 10.249.3.6 operator=()

```
PrivateTag & gdcm::PrivateTag::operator= (
    const PrivateTag & _val ) [inline]
```

References [gdcm::Tag::GetElementTag\(\)](#).

### 10.249.3.7 operator==( ) [1/2]

```
bool gdcM::PrivateTag::operator==(
    const PrivateTag & _val ) const [inline]
```

References [gdcM::Tag::GetElementTag\(\)](#).

### 10.249.3.8 operator==( ) [2/2]

```
bool gdcM::PrivateTag::operator==(
    const Tag & _val ) const [inline]
```

References [gdcM::Tag::GetElementTag\(\)](#).

### 10.249.3.9 ReadFromCommaSeparatedString()

```
bool gdcM::PrivateTag::ReadFromCommaSeparatedString (
    const char * str )
```

Read [PrivateTag](#) from a string. [Element](#) number will be truncated to 8bits. Eg: "1234,5678,GDCM" is private tag: (1234,78,"GDCM")

### 10.249.3.10 SetOwner()

```
void gdcM::PrivateTag::SetOwner (
    const char * owner ) [inline]
```

## 10.249.4 Friends And Related Symbol Documentation

### 10.249.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const PrivateTag & _val ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcMPrivateTag.h](#)

## 10.250 gdcm::ProgressEvent Class Reference

[ProgressEvent](#).

```
#include <gdcmProgressEvent.h>
```

Inheritance diagram for gdcm::ProgressEvent:



Collaboration diagram for gdcm::ProgressEvent:



### Public Types

- typedef [ProgressEvent](#) Self
- typedef [AnyEvent](#) Superclass

## Public Member Functions

- [ProgressEvent](#) (const [Self](#) &s)
- [ProgressEvent](#) (double p=0)
- [~ProgressEvent](#) () override=default
- bool [CheckEvent](#) (const [::gdcmm::Event](#) \*e) const override
- const char \* [GetEventName](#) () const override
- double [GetProgress](#) () const
- [::gdcmm::Event](#) \* [MakeObject](#) () const override
- void [operator=](#) (const [Self](#) &)=delete
- void [SetProgress](#) (double p)

## Public Member Functions inherited from [gdcmm::Event](#)

- [Event](#) ()
- [Event](#) (const [Event](#) &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) \*) const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

### 10.250.1 Detailed Description

[ProgressEvent](#).

Special type of event triggered during

See also

[AnyEvent](#)

Examples

[BasicAnonymizer.cs](#), [Cleaner.cs](#), and [ClinicalTrialIdentificationWorkflow.cs](#).

### 10.250.2 Member Typedef Documentation

#### 10.250.2.1 Self

```
typedef ProgressEvent gdcmm::ProgressEvent::Self
```

#### 10.250.2.2 Superclass

```
typedef AnyEvent gdcmm::ProgressEvent::Superclass
```

## 10.250.3 Constructor & Destructor Documentation

### 10.250.3.1 ProgressEvent() [1/2]

```
gdcm::ProgressEvent::ProgressEvent (
    double p = 0 ) [inline]
```

### 10.250.3.2 ~ProgressEvent()

```
gdcm::ProgressEvent::~~ProgressEvent ( ) [override], [default]
```

### 10.250.3.3 ProgressEvent() [2/2]

```
gdcm::ProgressEvent::ProgressEvent (
    const Self & s ) [inline]
```

## 10.250.4 Member Function Documentation

### 10.250.4.1 CheckEvent()

```
bool gdcm::ProgressEvent::CheckEvent (
    const ::gdcm::Event * e ) const [inline], [override]
```

### 10.250.4.2 GetEventName()

```
const char * gdcm::ProgressEvent::GetEventName ( ) const [inline], [override], [virtual]
```

Return the StringName associated with the event.

Implements [gdcm::Event](#).

### 10.250.4.3 GetProgress()

```
double gdcm::ProgressEvent::GetProgress ( ) const [inline]
```

#### Examples

[BasicAnonymizer.cs](#), [Cleaner.cs](#), and [ClinicalTrialIdentificationWorkflow.cs](#).

#### 10.250.4.4 MakeObject()

```
::gdcM::Event * gdcM::ProgressEvent::MakeObject ( ) const [inline], [override], [virtual]
```

Create an [Event](#) of this type This method work as a Factory for creating events of each particular type.

Implements [gdcM::Event](#).

#### 10.250.4.5 operator=()

```
void gdcM::ProgressEvent::operator= (
    const Self & ) [delete]
```

#### 10.250.4.6 SetProgress()

```
void gdcM::ProgressEvent::SetProgress (
    double p ) [inline]
```

The documentation for this class was generated from the following file:

- [gdcMProgressEvent.h](#)

## 10.251 gdcM::PVRGCodec Class Reference

[PVRGCodec](#).

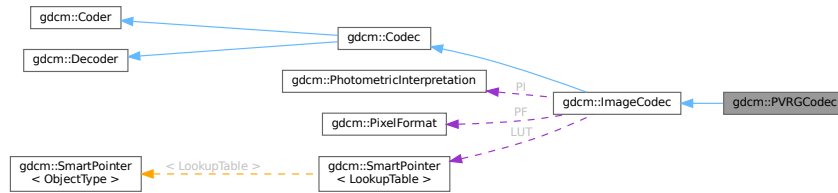
```
#include <gdcMPVRGCodec.h>
```

Inheritance diagram for gdcM::PVRGCodec:





Collaboration diagram for gdcm::PVRGCodec:



## Public Member Functions

- [PVRGCodec](#) ()
- [~PVRGCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const override  
*Return whether this coder support this transfer syntax (can code it)*
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const override  
*Return whether this decoder support this transfer syntax (can decode it)*
- [ImageCodec](#) \* [Clone](#) () const override
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out) override  
*Code.*
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override  
*Decode.*
- void [SetLossyFlag](#) (bool l)

## Public Member Functions inherited from [gdcm::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CleanupUnusedBits](#) (char \*data, size\_t datalen)
- const unsigned int \* [GetDimensions](#) () const
- virtual bool [GetHeaderInfo](#) (std::istream &is\_, [TransferSyntax](#) &ts)
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- virtual void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)

## Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default

## Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

## Additional Inherited Members

## Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > LUTPtr

## Protected Member Functions inherited from [gdcm::ImageCodec](#)

- virtual bool [AppendFrameEncode](#) (std::ostream &out, const char \*data, size\_t datalen)
- virtual bool [AppendRowEncode](#) (std::ostream &out, const char \*data, size\_t datalen)
- bool [DecodeByStreams](#) (std::istream &is\_, std::ostream &os) override
- bool [DoByteSwap](#) (std::istream &is\_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is\_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is\_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is\_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is\_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is\_, std::ostream &os)
- virtual bool [IsFrameEncoder](#) ()
- virtual bool [IsRowEncoder](#) ()
- virtual bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)
- virtual bool [StartEncode](#) (std::ostream &os)
- virtual bool [StopEncode](#) (std::ostream &os)

## Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char \*bv, unsigned long len, std::ostream &os)

## Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) LUT
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) PF
- [PhotometricInterpretation](#) PI
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

## 10.251.1 Detailed Description

[PVRGCodec](#).

### Note

pvrp is a broken implementation of the JPEG standard. It is known to have a bug in the 16bits lossless implementation of the standard.

In an ideal world, you should not need this codec at all. But to support some broken file such as:

PHILIPS\_Gyrosan-12-Jpeg\_Extended\_Process\_2\_4.dcm

we have to...

## 10.251.2 Constructor & Destructor Documentation

### 10.251.2.1 PVRGCodec()

```
gdcm::PVRGCodec::PVRGCodec ( )
```

### 10.251.2.2 ~PVRGCodec()

```
gdcm::PVRGCodec::~~PVRGCodec ( ) [override]
```

## 10.251.3 Member Function Documentation

### 10.251.3.1 CanCode()

```
bool gdcm::PVRGCodec::CanCode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.251.3.2 CanDecode()

```
bool gdcm::PVRGCodec::CanDecode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.251.3.3 Clone()

```
ImageCodec * gdcm::PVRGCodec::Clone ( ) const [override], [virtual]
```

Implements [gdcm::ImageCodec](#).

### 10.251.3.4 Code()

```
bool gdcm::PVRGCodec::Code (
    DataElement const & in_,
    DataElement & out_ ) [override], [virtual]
```

Code.

Reimplemented from [gdcm::Coder](#).

### 10.251.3.5 Decode()

```
bool gdcm::PVRGCodec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcm::ImageCodec](#).

### 10.251.3.6 SetLossyFlag()

```
void gdcm::PVRGCodec::SetLossyFlag (
    bool l )
```

The documentation for this class was generated from the following file:

- [gdcmPVRGCodec.h](#)

## 10.252 gdcm::PythonFilter Class Reference

[PythonFilter](#) [PythonFilter](#) is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a [DataElement](#) into a string, typically this is a nice feature to have for wrapped language.

```
#include <gdcmPythonFilter.h>
```

## Public Member Functions

- [PythonFilter](#) ()
- [~PythonFilter](#) ()
- [File](#) & [GetFile](#) ()
- const [File](#) & [GetFile](#) () const
- void [SetDicts](#) (const [Dicts](#) &dicts)
- void [SetFile](#) (const [File](#) &f)
- PyObject \* [ToPyObject](#) (const [Tag](#) &t) const
- void [UseDictAlways](#) (bool)

### 10.252.1 Detailed Description

[PythonFilter](#) [PythonFilter](#) is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a [DataElement](#) into a string, typically this is a nice feature to have for wrapped language.

### 10.252.2 Constructor & Destructor Documentation

#### 10.252.2.1 PythonFilter()

```
gdcm::PythonFilter::PythonFilter ( )
```

#### 10.252.2.2 ~PythonFilter()

```
gdcm::PythonFilter::~~PythonFilter ( )
```

### 10.252.3 Member Function Documentation

#### 10.252.3.1 GetFile() [1/2]

```
File & gdcm::PythonFilter::GetFile ( )
```

#### 10.252.3.2 GetFile() [2/2]

```
const File & gdcm::PythonFilter::GetFile ( ) const
```

#### 10.252.3.3 SetDicts()

```
void gdcm::PythonFilter::SetDicts (
    const Dicts & dicts )
```

#### 10.252.3.4 SetFile()

```
void gdcM::PythonFilter::SetFile (
    const File & f )
```

#### 10.252.3.5 ToPyObject()

```
PyObject * gdcM::PythonFilter::ToPyObject (
    const Tag & t ) const
```

#### 10.252.3.6 UseDictAlways()

```
void gdcM::PythonFilter::UseDictAlways (
    bool ) [inline]
```

The documentation for this class was generated from the following file:

- [gdcMPythonFilter.h](#)

### 10.253 gdcM::QueryBase Class Reference

[QueryBase.](#)

```
#include <gdcMQueryBase.h>
```

Inheritance diagram for gdcM::QueryBase:



#### Public Member Functions

- virtual [~QueryBase](#) ()=default
- std::vector< [Tag](#) > [GetAllRequiredTags](#) (const [ERootType](#) &inRootType) const
- std::vector< [Tag](#) > [GetAllTags](#) (const [ERootType](#) &inRootType) const
- virtual std::vector< [Tag](#) > [GetHierarchicalSearchTags](#) (const [ERootType](#) &inRootType) const =0  
*Return all Unique Key for a particular Query Root type (from the same level and above).*
- virtual const char \* [GetName](#) () const =0
- virtual std::vector< [Tag](#) > [GetOptionalTags](#) (const [ERootType](#) &inRootType) const =0
- virtual [DataElement](#) [GetQueryLevel](#) () const =0
- virtual std::vector< [Tag](#) > [GetRequiredTags](#) (const [ERootType](#) &inRootType) const =0
- virtual std::vector< [Tag](#) > [GetUniqueTags](#) (const [ERootType](#) &inRootType) const =0

## 10.253.1 Detailed Description

[QueryBase](#).

contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE

There are four levels of C-FIND and C-MOVE query:

- [Patient](#)
- [Study](#)
- [Series](#)
- [Image](#)

Each one has its own required and optional tags. This class provides an interface for getting those tags. This is an interface class.

See 3.4 C 6.1 and 3.4 C 6.2 for the patient and study root query types. These sections define the tags allowed by a particular query. The caller must pass in which root type they want, patient or study. A third root type, Modality Worklist Query, isn't yet supported.

This class (or rather it's derived classes) will be held in the RootQuery types. These query types actually make the dataset, and will use this dataset to list the required, unique, and optional tags for each type of query. This design is somewhat overly complicated, but is kept so that if we ever wanted to try to guess the query type from the given tags, we could do so.

## 10.253.2 Constructor & Destructor Documentation

### 10.253.2.1 ~QueryBase()

```
virtual gdcmm::QueryBase::~QueryBase ( ) [virtual], [default]
```

## 10.253.3 Member Function Documentation

### 10.253.3.1 GetAllRequiredTags()

```
std::vector< Tag > gdcmm::QueryBase::GetAllRequiredTags (
    const ERootType & inRootType ) const
```

In order to validate a query dataset we need to check that there exists at least one required (or unique) key

### 10.253.3.2 GetAllTags()

```
std::vector< Tag > gdcmm::QueryBase::GetAllTags (
    const ERootType & inRootType ) const
```

In order to validate a query dataset, just check for the presence of a tag, not it's requirement level in the spec

### 10.253.3.3 GetHierachicalSearchTags()

```
virtual std::vector< Tag > gdcM::QueryBase::GetHierachicalSearchTags (
    const ERootType & inRootType ) const [pure virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implemented in [gdcM::QueryImage](#), [gdcM::QueryPatient](#), [gdcM::QuerySeries](#), and [gdcM::QueryStudy](#).

### 10.253.3.4 GetName()

```
virtual const char * gdcM::QueryBase::GetName ( ) const [pure virtual]
```

Implemented in [gdcM::QueryImage](#), [gdcM::QueryPatient](#), [gdcM::QuerySeries](#), and [gdcM::QueryStudy](#).

### 10.253.3.5 GetOptionalTags()

```
virtual std::vector< Tag > gdcM::QueryBase::GetOptionalTags (
    const ERootType & inRootType ) const [pure virtual]
```

Implemented in [gdcM::QueryImage](#), [gdcM::QueryPatient](#), [gdcM::QuerySeries](#), and [gdcM::QueryStudy](#).

### 10.253.3.6 GetQueryLevel()

```
virtual DataElement gdcM::QueryBase::GetQueryLevel ( ) const [pure virtual]
```

Implemented in [gdcM::QueryImage](#), [gdcM::QueryPatient](#), [gdcM::QuerySeries](#), and [gdcM::QueryStudy](#).

### 10.253.3.7 GetRequiredTags()

```
virtual std::vector< Tag > gdcM::QueryBase::GetRequiredTags (
    const ERootType & inRootType ) const [pure virtual]
```

Implemented in [gdcM::QueryImage](#), [gdcM::QueryPatient](#), [gdcM::QuerySeries](#), and [gdcM::QueryStudy](#).

### 10.253.3.8 GetUniqueTags()

```
virtual std::vector< Tag > gdcM::QueryBase::GetUniqueTags (
    const ERootType & inRootType ) const [pure virtual]
```

Implemented in [gdcM::QueryImage](#), [gdcM::QueryPatient](#), [gdcM::QuerySeries](#), and [gdcM::QueryStudy](#).

The documentation for this class was generated from the following file:

- [gdcMQueryBase.h](#)



## 10.254 gdcm::QueryFactory Class Reference

QueryFactory.h.

```
#include <gdcmQueryFactory.h>
```

### Static Public Member Functions

- static [ECharSet](#) [GetCharacterFromCurrentLocale](#) ()
- static void [ListCharSets](#) (std::ostream &os)  
*List all possible CharSet.*
- static [DataElement](#) [ProduceCharacterSetDataElement](#) (const std::vector< [ECharSet](#) > &inCharSetType)
- static [BaseQuery](#) \* [ProduceQuery](#) (const std::string &sopInstanceUID, [ENQueryType](#) inQueryType)
- static [BaseRootQuery](#) \* [ProduceQuery](#) ([ERootType](#) inRootType, [EQueryType](#) inQueryType, [EQueryLevel](#) inQueryLevel)

### 10.254.1 Detailed Description

QueryFactory.h.

contains: a class to produce a query based off of user-entered information

Essentially, this class is used to construct a query based off of user input (typically from the command line; if in code directly, the query itself could just be instantiated)

In theory, could also be used as the interface to validate incoming datasets as belonging to a particular query style

### 10.254.2 Member Function Documentation

#### 10.254.2.1 GetCharacterFromCurrentLocale()

```
static ECharSet gdcm::QueryFactory::GetCharacterFromCurrentLocale ( ) [static]
```

This function will return the corresponding [ECharSet](#) associated with the current locale of the running system (based on the value of `locale()` ).

#### 10.254.2.2 ListCharSets()

```
static void gdcm::QueryFactory::ListCharSets (
    std::ostream & os ) [static]
```

List all possible CharSet.

### 10.254.2.3 ProduceCharacterSetDataElement()

```
static DataElement gdcm::QueryFactory::ProduceCharacterSetDataElement (
    const std::vector< ECharSet > & inCharSetType ) [static]
```

This function will produce the appropriate dataelement given a list of charsets. The first charset will be used directly, while the second and subsequent will be prepended with "ISO2022 ". Redundant character sets are not permitted, so if they are encountered, they will just be skipped. if UTF8 or GB18030 is used, no subsequent character sets will be used if the vector passed in is empty, then the dataelement that's passed out will be empty and Latin1 is the presumed encoding

### 10.254.2.4 ProduceQuery() [1/2]

```
static BaseQuery * gdcm::QueryFactory::ProduceQuery (
    const std::string & sopInstanceUID,
    ENQueryType inQueryType ) [static]
```

### 10.254.2.5 ProduceQuery() [2/2]

```
static BaseRootQuery * gdcm::QueryFactory::ProduceQuery (
    ERootType inRootType,
    EQueryType inQueryType,
    EQueryLevel inQueryLevel ) [static]
```

this function will produce a query (basically, a wrapper to a dataset that can validate whether or not the query is a valid cfind/cmove query) and the level of the query (patient, study, series, image). If the user provides an invalid instantiation (ie, study root type, query level of patient), then the result is NULL.

The documentation for this class was generated from the following file:

- [gdcmQueryFactory.h](#)

## 10.255 gdcm::QueryImage Class Reference

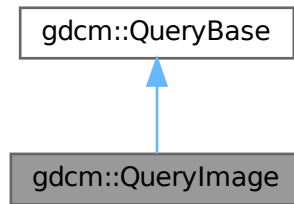
[QueryImage](#).

```
#include <gdcmQueryImage.h>
```

Inheritance diagram for gdcm::QueryImage:



Collaboration diagram for gdcm::QueryImage:



### Public Member Functions

- `std::vector< Tag > GetHierachicalSearchTags` (const `ERootType` &`inRootType`) const override  
*Return all Unique Key for a particular Query Root type (from the same level and above).*
- `const char * GetName` () const override
- `std::vector< Tag > GetOptionalTags` (const `ERootType` &`inRootType`) const override
- `DataElement GetQueryLevel` () const override
- `std::vector< Tag > GetRequiredTags` (const `ERootType` &`inRootType`) const override
- `std::vector< Tag > GetUniqueTags` (const `ERootType` &`inRootType`) const override

### Public Member Functions inherited from `gdcm::QueryBase`

- `virtual ~QueryBase` ()=default
- `std::vector< Tag > GetAllRequiredTags` (const `ERootType` &`inRootType`) const
- `std::vector< Tag > GetAllTags` (const `ERootType` &`inRootType`) const

## 10.255.1 Detailed Description

`QueryImage`.

contains: class to construct an image-based query for C-FIND and C-MOVE

## 10.255.2 Member Function Documentation

### 10.255.2.1 `GetHierachicalSearchTags()`

```
std::vector< Tag > gdcm::QueryImage::GetHierachicalSearchTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements `gdcm::QueryBase`.

#### 10.255.2.2 GetName()

```
const char * gdcM::QueryImage::GetName ( ) const [override], [virtual]
```

Implements [gdcM::QueryBase](#).

#### 10.255.2.3 GetOptionalTags()

```
std::vector< Tag > gdcM::QueryImage::GetOptionalTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Implements [gdcM::QueryBase](#).

#### 10.255.2.4 GetQueryLevel()

```
DataElement gdcM::QueryImage::GetQueryLevel ( ) const [override], [virtual]
```

Implements [gdcM::QueryBase](#).

#### 10.255.2.5 GetRequiredTags()

```
std::vector< Tag > gdcM::QueryImage::GetRequiredTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Implements [gdcM::QueryBase](#).

#### 10.255.2.6 GetUniqueTags()

```
std::vector< Tag > gdcM::QueryImage::GetUniqueTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Implements [gdcM::QueryBase](#).

The documentation for this class was generated from the following file:

- [gdcMQueryImage.h](#)

## 10.256 gdcm::QueryPatient Class Reference

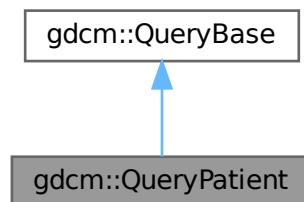
[QueryPatient](#).

```
#include <gdcmQueryPatient.h>
```

Inheritance diagram for gdcm::QueryPatient:



Collaboration diagram for gdcm::QueryPatient:



### Public Member Functions

- `std::vector< Tag > GetHierachicalSearchTags` (const `ERootType` &inRootType) const override  
*Return all Unique Key for a particular Query Root type (from the same level and above).*
- `const char * GetName` () const override
- `std::vector< Tag > GetOptionalTags` (const `ERootType` &inRootType) const override
- `DataElement GetQueryLevel` () const override
- `std::vector< Tag > GetRequiredTags` (const `ERootType` &inRootType) const override
- `std::vector< Tag > GetUniqueTags` (const `ERootType` &inRootType) const override

## Public Member Functions inherited from [gdcm::QueryBase](#)

- virtual [~QueryBase](#) ()=default
- std::vector< [Tag](#) > [GetAllRequiredTags](#) (const [ERootType](#) &inRootType) const
- std::vector< [Tag](#) > [GetAllTags](#) (const [ERootType](#) &inRootType) const

### 10.256.1 Detailed Description

[QueryPatient](#).

contains: class to construct a patient-based query for c-find and c-move

### 10.256.2 Member Function Documentation

#### 10.256.2.1 GetHierachicalSearchTags()

```
std::vector< Tag > gdcm::QueryPatient::GetHierachicalSearchTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements [gdcm::QueryBase](#).

#### 10.256.2.2 GetName()

```
const char * gdcm::QueryPatient::GetName ( ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

#### 10.256.2.3 GetOptionalTags()

```
std::vector< Tag > gdcm::QueryPatient::GetOptionalTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

#### 10.256.2.4 GetQueryLevel()

```
DataElement gdcm::QueryPatient::GetQueryLevel ( ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

### 10.256.2.5 GetRequiredTags()

```
std::vector< Tag > gdcm::QueryPatient::GetRequiredTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

### 10.256.2.6 GetUniqueTags()

```
std::vector< Tag > gdcm::QueryPatient::GetUniqueTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

The documentation for this class was generated from the following file:

- [gdcmQueryPatient.h](#)

## 10.257 gdcm::QuerySeries Class Reference

[QuerySeries](#).

```
#include <gdcmQuerySeries.h>
```

Inheritance diagram for `gdcm::QuerySeries`:



Collaboration diagram for gdcm::QuerySeries:



### Public Member Functions

- `std::vector< Tag > GetHierachicalSearchTags` (const `ERootType` &`inRootType`) const override  
*Return all Unique Key for a particular Query Root type (from the same level and above).*
- `const char * GetName` () const override
- `std::vector< Tag > GetOptionalTags` (const `ERootType` &`inRootType`) const override
- `DataElement GetQueryLevel` () const override
- `std::vector< Tag > GetRequiredTags` (const `ERootType` &`inRootType`) const override
- `std::vector< Tag > GetUniqueTags` (const `ERootType` &`inRootType`) const override

### Public Member Functions inherited from `gdcm::QueryBase`

- `virtual ~QueryBase` ()=default
- `std::vector< Tag > GetAllRequiredTags` (const `ERootType` &`inRootType`) const
- `std::vector< Tag > GetAllTags` (const `ERootType` &`inRootType`) const

## 10.257.1 Detailed Description

`QuerySeries`.

contains: class to construct a series-based query for c-find and c-move

## 10.257.2 Member Function Documentation

### 10.257.2.1 `GetHierachicalSearchTags()`

```
std::vector< Tag > gdcm::QuerySeries::GetHierachicalSearchTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements `gdcm::QueryBase`.



### 10.257.2.2 GetName()

```
const char * gdcm::QuerySeries::GetName ( ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

### 10.257.2.3 GetOptionalTags()

```
std::vector< Tag > gdcm::QuerySeries::GetOptionalTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

### 10.257.2.4 GetQueryLevel()

```
DataElement gdcm::QuerySeries::GetQueryLevel ( ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

### 10.257.2.5 GetRequiredTags()

```
std::vector< Tag > gdcm::QuerySeries::GetRequiredTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

### 10.257.2.6 GetUniqueTags()

```
std::vector< Tag > gdcm::QuerySeries::GetUniqueTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

The documentation for this class was generated from the following file:

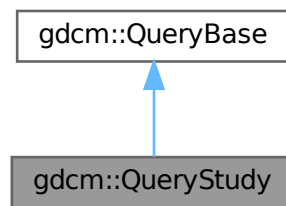
- [gdcmQuerySeries.h](#)

## 10.258 gdcm::QueryStudy Class Reference

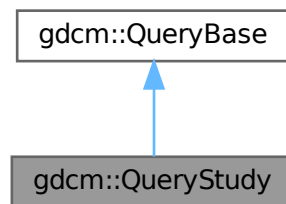
QueryStudy.h.

```
#include <gdcmQueryStudy.h>
```

Inheritance diagram for gdcm::QueryStudy:



Collaboration diagram for gdcm::QueryStudy:



### Public Member Functions

- `std::vector< Tag > GetHierachicalSearchTags` (const `ERootType` &inRootType) const override  
*Return all Unique Key for a particular Query Root type (from the same level and above).*
- `const char * GetName` () const override
- `std::vector< Tag > GetOptionalTags` (const `ERootType` &inRootType) const override
- `DataElement GetQueryLevel` () const override
- `std::vector< Tag > GetRequiredTags` (const `ERootType` &inRootType) const override
- `std::vector< Tag > GetUniqueTags` (const `ERootType` &inRootType) const override

## Public Member Functions inherited from [gdcm::QueryBase](#)

- virtual [~QueryBase](#) ()=default
- std::vector< [Tag](#) > [GetAllRequiredTags](#) (const [ERootType](#) &inRootType) const
- std::vector< [Tag](#) > [GetAllTags](#) (const [ERootType](#) &inRootType) const

### 10.258.1 Detailed Description

QueryStudy.h.

contains: class to construct a study-based query for C-FIND and C-MOVE

### 10.258.2 Member Function Documentation

#### 10.258.2.1 GetHierarchicalSearchTags()

```
std::vector< Tag > gdcm::QueryStudy::GetHierarchicalSearchTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements [gdcm::QueryBase](#).

#### 10.258.2.2 GetName()

```
const char * gdcm::QueryStudy::GetName ( ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

#### 10.258.2.3 GetOptionalTags()

```
std::vector< Tag > gdcm::QueryStudy::GetOptionalTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

#### 10.258.2.4 GetQueryLevel()

```
DataElement gdcm::QueryStudy::GetQueryLevel ( ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

### 10.258.2.5 GetRequiredTags()

```
std::vector< Tag > gdcM::QueryStudy::GetRequiredTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Implements [gdcM::QueryBase](#).

### 10.258.2.6 GetUniqueTags()

```
std::vector< Tag > gdcM::QueryStudy::GetUniqueTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Implements [gdcM::QueryBase](#).

The documentation for this class was generated from the following file:

- [gdcMQueryStudy.h](#)

## 10.259 gdcM::RAWCodec Class Reference

[RAWCodec](#) class.

```
#include <gdcMRAWCodec.h>
```

Inheritance diagram for [gdcM::RAWCodec](#):



Collaboration diagram for gdcm::RAWCodec:



## Public Member Functions

- [RAWCodec](#) ()
- [~RAWCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const override  
*Return whether this coder support this transfer syntax (can code it)*
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const override  
*Return whether this decoder support this transfer syntax (can decode it)*
- [ImageCodec](#) \* [Clone](#) () const override
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out) override  
*Code.*
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override  
*Decode.*
- bool [DecodeBytes](#) (const char \*inBytes, size\_t inBufferLength, char \*outBytes, size\_t inOutBufferLength)
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override

## Public Member Functions inherited from [gdcm::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CleanupUnusedBits](#) (char \*data, size\_t datalen)
- const unsigned int \* [GetDimensions](#) () const
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- virtual void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)

### Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default

### Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

### Protected Member Functions

- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os) override

### Protected Member Functions inherited from [gdcm::ImageCodec](#)

- virtual bool [AppendFrameEncode](#) (std::ostream &out, const char \*data, size\_t datalen)
- virtual bool [AppendRowEncode](#) (std::ostream &out, const char \*data, size\_t datalen)
- bool [DoByteSwap](#) (std::istream &is\_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is\_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is\_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is\_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is\_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is\_, std::ostream &os)
- virtual bool [IsFrameEncoder](#) ()
- virtual bool [IsRowEncoder](#) ()
- virtual bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)
- virtual bool [StartEncode](#) (std::ostream &os)
- virtual bool [StopEncode](#) (std::ostream &os)

### Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char \*bv, unsigned long len, std::ostream &os)

### Additional Inherited Members

### Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

## Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) LUT
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) PF
- [PhotometricInterpretation](#) PI
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

### 10.259.1 Detailed Description

[RAWCodec](#) class.

### 10.259.2 Constructor & Destructor Documentation

#### 10.259.2.1 RAWCodec()

```
gdcm::RAWCodec::RAWCodec ( )
```

#### 10.259.2.2 ~RAWCodec()

```
gdcm::RAWCodec::~~RAWCodec ( ) [override]
```

### 10.259.3 Member Function Documentation

#### 10.259.3.1 CanCode()

```
bool gdcm::RAWCodec::CanCode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

#### 10.259.3.2 CanDecode()

```
bool gdcm::RAWCodec::CanDecode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.259.3.3 Clone()

```
ImageCodec * gdcM::RAWCodec::Clone ( ) const [override], [virtual]
```

Implements [gdcM::ImageCodec](#).

### 10.259.3.4 Code()

```
bool gdcM::RAWCodec::Code (
    DataElement const & in_,
    DataElement & out_ ) [override], [virtual]
```

Code.

Reimplemented from [gdcM::Coder](#).

### 10.259.3.5 Decode()

```
bool gdcM::RAWCodec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcM::ImageCodec](#).

### 10.259.3.6 DecodeByStreams()

```
bool gdcM::RAWCodec::DecodeByStreams (
    std::istream & is,
    std::ostream & os ) [override], [protected], [virtual]
```

Reimplemented from [gdcM::ImageCodec](#).

### 10.259.3.7 DecodeBytes()

```
bool gdcM::RAWCodec::DecodeBytes (
    const char * inBytes,
    size_t inBufferLength,
    char * outBytes,
    size_t inOutBufferLength )
```

Used by the ImageStreamReader– converts a read in buffer into one with the proper encodings.



### 10.259.3.8 GetHeaderInfo()

```
bool gdcm::RAWCodec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

The documentation for this class was generated from the following file:

- [gdcmRAWCodec.h](#)

## 10.260 gdcm::Reader Class Reference

[Reader](#) ala DOM (Document [Object](#) Model)

```
#include <gdcmReader.h>
```

Inheritance diagram for gdcm::Reader:



Collaboration diagram for gdcm::Reader:



## Public Member Functions

- [Reader](#) ()
- virtual [~Reader](#) ()
- bool [CanRead](#) () const
- [File](#) & [GetFile](#) ()  
*Set/Get File.*
- const [File](#) & [GetFile](#) () const  
*Set/Get File.*
- size\_t [GetStreamCurrentPosition](#) () const
- virtual bool [Read](#) ()  
*Main function to read a file.*
- bool [ReadSelectedPrivateTags](#) (std::set< [PrivateTag](#) > const &ptags, bool readvalues=true)  
*Will only read the specified selected private tags.*
- bool [ReadSelectedTags](#) (std::set< [Tag](#) > const &tags, bool readvalues=true)  
*Will only read the specified selected tags.*
- bool [ReadUpToTag](#) (const [Tag](#) &tag, std::set< [Tag](#) > const &skiptags=std::set< [Tag](#) >())
- void [SetFile](#) ([File](#) &file)  
*Set/Get File.*
- void [SetFileName](#) (const char \*filename\_native)
- void [SetStream](#) (std::istream &input\_stream)  
*Set the open-ed stream directly.*

**Protected Member Functions**

- `std::istream * GetStreamPtr () const`
- `bool ReadDataSet ()`
- `bool ReadMetaInformation ()`
- `bool ReadPreamble ()`

**Protected Attributes**

- `SmartPointer< File > F`

**Friends**

- class [StreamImageReader](#)

**10.260.1 Detailed Description**

[Reader](#) ala DOM (Document [Object](#) Model)

This class is a non-validating reader, it will only performs well- formedness check only, and to some extent catch known error (non well-formed document).

Detailed description here

A [DataSet](#) DOES NOT contains group 0x0002 (see [FileMetaInformation](#))

This is really a [DataSet](#) reader. This will not make sure the dataset conform to any [IOD](#) at all. This is a completely different step. The reasoning was that user could control the [IOD](#) there lib would handle and thus we would not be able to read a [DataSet](#) if the [IOD](#) was not found Instead we separate the reading from the validation.

**Note**

From GDCM1.x. Users will realize that one feature is missing from this DOM implementation. In GDCM 1.x user used to be able to control the size of the [Value](#) to be read. By default it was 0xffff. The main author of GDCM2 thought this was too dangerous and harmful and therefore this feature did not make it into GDCM2

**Warning**

GDCM will not produce warning for unordered (non-alphabetical order).

**See also**

[Writer](#) [FileMetaInformation](#) [DataSet](#) [File](#)

**Examples**

[BasicAnonymizer.cs](#), [ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [Cleaner.cs](#), [ClinicalTrialAnnotate.cxx](#), [ClinicalTrialIdentificationWorkflow.cs](#), [CreateFakeRTDOSE.cxx](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [FixBrokenJ2K.cxx](#), [FixOrientation.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [LargeVRDSEExplicit.cxx](#), [MakeTemplate.cxx](#), [ManipulateFile.cs](#), [NewSequence.cs](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [ReadUTF8QtDir.cxx](#), [ReformatFile.cs](#), [SimplePrint.cs](#), [SimplePrintPatientName.cs](#), [TestReader.cxx](#), [csa2img.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

## 10.260.2 Constructor & Destructor Documentation

### 10.260.2.1 Reader()

```
gdcmm::Reader::Reader ( )
```

### 10.260.2.2 ~Reader()

```
virtual gdcmm::Reader::~Reader ( ) [virtual]
```

## 10.260.3 Member Function Documentation

### 10.260.3.1 CanRead()

```
bool gdcmm::Reader::CanRead ( ) const
```

Test whether this is a DICOM file

#### Warning

need to call either SetFileName or SetStream first

#### Examples

[ReadUTF8QtDir.cxx](#).

### 10.260.3.2 GetFile() [1/2]

```
File & gdcmm::Reader::GetFile ( ) [inline]
```

Set/Get [File](#).

### 10.260.3.3 GetFile() [2/2]

```
const File & gdcmm::Reader::GetFile ( ) const [inline]
```

Set/Get [File](#).

#### Examples

[BasicAnonymizer.cs](#), [BasicImageAnonymizer.cs](#), [ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [Cleaner.cs](#), [ClinicalTrialAnnotate.cxx](#), [ClinicalTrialIdentificationWorkflow.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [CreateFakeRTDOSE.cxx](#), [DecompressImage.cs](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), [FixBrokenJ2K.cxx](#), [FixJAI BugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [ManipulateFile.cs](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [ReformatFile.cs](#), [SimplePrint.cs](#), [SimplePrintPatientName.cs](#), [StandardizeFiles.cs](#), [TestReader.cxx](#), [csa2img.cxx](#), [gdcmmrtionplan.cxx](#), [gdcmmrtplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

### 10.260.3.4 GetStreamCurrentPosition()

```
size_t gdcm::Reader::GetStreamCurrentPosition ( ) const
```

For wrapped language. return type is compatible with [System::FileSize](#) return type Use native `std::streampos` / `std::streamoff` directly from the stream from C++

#### Examples

[ExtractImageRegion.cs](#).

### 10.260.3.5 GetStreamPtr()

```
std::istream * gdcm::Reader::GetStreamPtr ( ) const [inline], [protected]
```

### 10.260.3.6 Read()

```
virtual bool gdcm::Reader::Read ( ) [virtual]
```

Main function to read a file.

Reimplemented in [gdcm::ImageReader](#), [gdcm::ImageRegionReader](#), [gdcm::PixmapReader](#), [gdcm::SegmentReader](#), and [gdcm::SurfaceReader](#).

#### Examples

[BasicAnonymizer.cs](#), [ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [Cleaner.cs](#), [ClinicalTrialAnnotate.cxx](#), [ClinicalTrialIdentificationWorkflow.cs](#), [CreateFakeRTDOSE.cxx](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [FixBrokenJ2K.cxx](#), [FixOrientation.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [ManipulateFile.cs](#), [NewSequence.cs](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [ReformatFile.cs](#), [SimplePrint.cs](#), [SimplePrintPatientName.cs](#), [TestReader.cxx](#), [csa2img.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

### 10.260.3.7 ReadDataSet()

```
bool gdcm::Reader::ReadDataSet ( ) [protected]
```

### 10.260.3.8 ReadMetaInformation()

```
bool gdcm::Reader::ReadMetaInformation ( ) [protected]
```

### 10.260.3.9 ReadPreamble()

```
bool gdcM::Reader::ReadPreamble ( ) [protected]
```

### 10.260.3.10 ReadSelectedPrivateTags()

```
bool gdcM::Reader::ReadSelectedPrivateTags (
    std::set< PrivateTag > const & ptags,
    bool readvalues = true )
```

Will only read the specified selected private tags.

### 10.260.3.11 ReadSelectedTags()

```
bool gdcM::Reader::ReadSelectedTags (
    std::set< Tag > const & tags,
    bool readvalues = true )
```

Will only read the specified selected tags.

### 10.260.3.12 ReadUpToTag()

```
bool gdcM::Reader::ReadUpToTag (
    const Tag & tag,
    std::set< Tag > const & skiptags = std::set< Tag >() )
```

Will read only up to [Tag](#)

#### Parameters

<i>tag</i>	and skipping any tag specified in
<i>skiptags</i>	

#### Examples

[DumpVisusChange.cxx](#).

### 10.260.3.13 SetFile()

```
void gdcM::Reader::SetFile (
    File & file ) [inline]
```

Set/Get [File](#).

### 10.260.3.14 SetFileName()

```
void gdcm::Reader::SetFileName (
    const char * filename_native )
```

Set the filename to open. This will create a `std::ifstream` internally See `SetStream` if you are dealing with different `std::istream` object

#### Examples

[BasicAnonymizer.cs](#), [BasicImageAnonymizer.cs](#), [ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CheckBigEndianBug.cxx](#), [Cleaner.cs](#), [ClinicalTrialAnnotate.cxx](#), [ClinicalTrialIdentificationWorkflow.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [ConvertToQImage.cxx](#), [CreateFakeRTDOSE.cxx](#), [DecompressImage.cs](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetArray.cs](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [ManipulateFile.cs](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [PrintLUT.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [ReadMultiTimesException.cxx](#), [ReadUTF8QtDir.cxx](#), [ReformatFile.cs](#), [RescaleImage.cs](#), [SimplePrint.cs](#), [SimplePrintPatientName.cs](#), [StandardizeFiles.cs](#), [TemplateEmptyImage.cxx](#), [TestReader.cxx](#), [csa2img.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), [rle2img.cxx](#), and [threadgdcm.cxx](#).

### 10.260.3.15 SetStream()

```
void gdcm::Reader::SetStream (
    std::istream & input_stream ) [inline]
```

Set the open-ed stream directly.

#### Examples

[ReadUTF8QtDir.cxx](#).

## 10.260.4 Friends And Related Symbol Documentation

### 10.260.4.1 StreamImageReader

```
friend class StreamImageReader [friend]
```

## 10.260.5 Member Data Documentation

### 10.260.5.1 F

```
SmartPointer<File> gdcm::Reader::F [protected]
```

The documentation for this class was generated from the following file:

- [gdcmReader.h](#)

## 10.261 gdcm::RealWorldValueMappingContent Struct Reference

```
#include <gdcmImageHelper.h>
```

Collaboration diagram for gdcm::RealWorldValueMappingContent:



### Public Attributes

- std::string [CodeMeaning](#)
- std::string [CodeValue](#)
- double [RealWorldValueIntercept](#)
- double [RealWorldValueSlope](#)

### 10.261.1 Member Data Documentation

#### 10.261.1.1 CodeMeaning

```
std::string gdcm::RealWorldValueMappingContent::CodeMeaning
```

#### 10.261.1.2 CodeValue

```
std::string gdcm::RealWorldValueMappingContent::CodeValue
```



### 10.261.1.3 RealWorldValueIntercept

```
double gdcm::RealWorldValueMappingContent::RealWorldValueIntercept
```

### 10.261.1.4 RealWorldValueSlope

```
double gdcm::RealWorldValueMappingContent::RealWorldValueSlope
```

The documentation for this struct was generated from the following file:

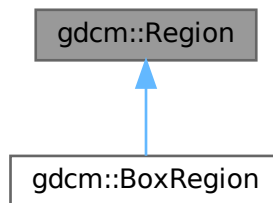
- [gdcmImageHelper.h](#)

## 10.262 gdcm::Region Class Reference

Class for manipulation region.

```
#include <gdcmRegion.h>
```

Inheritance diagram for gdcm::Region:



### Public Member Functions

- [Region](#) ()
- virtual [~Region](#) ()
- virtual size\_t [Area](#) () const =0  
*compute the area*
- virtual [Region](#) \* [Clone](#) () const =0
- virtual [BoxRegion](#) [ComputeBoundingBox](#) ()=0  
*Return the Axis-Aligned minimum bounding box for all regions.*
- virtual bool [Empty](#) () const =0  
*return whether this domain is empty:*
- virtual bool [IsValid](#) () const =0  
*return whether this is valid domain*
- virtual void [Print](#) (std::ostream &os=std::cout) const  
*Print.*

### 10.262.1 Detailed Description

Class for manipulation region.

### 10.262.2 Constructor & Destructor Documentation

#### 10.262.2.1 Region()

```
gdcm::Region::Region ( )
```

#### 10.262.2.2 ~Region()

```
virtual gdcm::Region::~~Region ( ) [virtual]
```

### 10.262.3 Member Function Documentation

#### 10.262.3.1 Area()

```
virtual size_t gdcm::Region::Area ( ) const [pure virtual]
```

compute the area

Implemented in [gdcm::BoxRegion](#).

#### 10.262.3.2 Clone()

```
virtual Region * gdcm::Region::Clone ( ) const [pure virtual]
```

Implemented in [gdcm::BoxRegion](#).

#### 10.262.3.3 ComputeBoundingBox()

```
virtual BoxRegion gdcm::Region::ComputeBoundingBox ( ) [pure virtual]
```

Return the Axis-Aligned minimum bounding box for all regions.

Implemented in [gdcm::BoxRegion](#).

#### 10.262.3.4 Empty()

```
virtual bool gdcm::Region::Empty ( ) const [pure virtual]
```

return whether this domain is empty:

Implemented in [gdcm::BoxRegion](#).

#### 10.262.3.5 IsValid()

```
virtual bool gdcm::Region::IsValid ( ) const [pure virtual]
```

return whether this is valid domain

Implemented in [gdcm::BoxRegion](#).

#### 10.262.3.6 Print()

```
virtual void gdcm::Region::Print (
    std::ostream & os = std::cout ) const [virtual]
```

Print.

Reimplemented in [gdcm::BoxRegion](#).

Referenced by [gdcm::operator<<\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmRegion.h](#)

## 10.263 gdcm::Rescaler Class Reference

Rescale class.

```
#include <gdcmRescaler.h>
```

## Public Member Functions

- [Rescaler](#) ()
- [~Rescaler](#) ()=default
- [PixelFormat::ScalarType ComputeInterceptSlopePixelFormat](#) ()
- [PixelFormat ComputePixelFormatFromMinMax](#) ()
- double [GetIntercept](#) () const
- double [GetSlope](#) () const
- bool [InverseRescale](#) (char \*out, const char \*in, size\_t n)  
*Inverse transform.*
- bool [Rescale](#) (char \*out, const char \*in, size\_t n)  
*Direct transform.*
- void [SetIntercept](#) (double i)  
*Set Intercept: used for both direct&inverse transformation.*
- void [SetMinMaxForPixelFormat](#) (double min, double max)
- void [SetPixelFormat](#) ([PixelFormat](#) const &pf)  
*Set Pixel Format of input data.*
- void [SetSlope](#) (double s)  
*Set Slope: user for both direct&inverse transformation.*
- void [SetTargetPixelFormat](#) ([PixelFormat](#) const &targetst)
- void [SetUseTargetPixelFormat](#) (bool b)  
*Override default behavior of Rescale.*

## Protected Member Functions

- template<typename TIn >  
void [InverseRescaleFunctionIntoBestFit](#) (char \*out, const TIn \*in, size\_t n)
- template<typename TIn >  
void [RescaleFunctionIntoBestFit](#) (char \*out, const TIn \*in, size\_t n)

### 10.263.1 Detailed Description

Rescale class.

This class is meant to apply the linear transform of Stored Pixel [Value](#) to Real World [Value](#). This is mostly found in CT or PET dataset, where the value are stored using one type, but need to be converted to another scale using a linear transform. There are basically two cases: In CT: the linear transform is generally integer based. E.g. the Stored Pixel [Type](#) is unsigned short 12bits, but to get Hounsfield unit, one need to apply the linear transform:

$$RWV = 1. * SV - 1024$$

So the best scalar to store the Real World [Value](#) will be 16 bits signed type.

In PET: the linear transform is generally floating point based. Since the dynamic range can be quite high, the Rescale Slope / Rescale Intercept can be changing throughout the [Series](#). So it is important to read all linear transform and deduce the best Pixel [Type](#) only at the end (when all the images to be read have been parsed).

### Warning

Internally any time a floating point value is found either in the Rescale Slope or the Rescale Intercept it is assumed that the best matching output pixel type is FLOAT64 (in previous implementation it was FLOAT32). Because [VR:DS](#) is closer to a 64bits floating point type FLOAT64 is thus a best matching pixel type for the floating point transformation.

Example: Let say input is FLOAT64, and we want UINT16 as output, we would do:

```
Rescaler ir;
ir.SetIntercept( 0 );
ir.SetSlope( 5.6789 );
ir.SetPixelFormat( FLOAT64 );
ir.SetMinMaxForPixelType( ((PixelFormat)UINT16).GetMin(), ((PixelFormat)UINT16).GetMax() );
ir.InverseRescale(output,input,numberofbytes );
```

### Note

handle floating point transformation back and forth to integer properly (no loss)

### See also

[Unpacker12Bits](#)

### Examples

[RescaleImage.cs](#).

## 10.263.2 Constructor & Destructor Documentation

### 10.263.2.1 Rescaler()

```
gdcm::Rescaler::Rescaler ( ) [inline]
```

### 10.263.2.2 ~Rescaler()

```
gdcm::Rescaler::~~Rescaler ( ) [default]
```

## 10.263.3 Member Function Documentation

### 10.263.3.1 ComputeInterceptSlopePixelType()

```
PixelFormat::ScalarType gdcm::Rescaler::ComputeInterceptSlopePixelType ( )
```

Compute the Pixel Format of the output data Used for direct transformation

### Examples

[RescaleImage.cs](#).

### 10.263.3.2 ComputePixelTypeFromMinMax()

```
PixelFormat gdcm::Rescaler::ComputePixelTypeFromMinMax ( )
```

Compute the Pixel Format of the output data Used for inverse transformation

### 10.263.3.3 GetIntercept()

```
double gdcm::Rescaler::GetIntercept ( ) const [inline]
```

### 10.263.3.4 GetSlope()

```
double gdcm::Rescaler::GetSlope ( ) const [inline]
```

### 10.263.3.5 InverseRescale()

```
bool gdcm::Rescaler::InverseRescale (
    char * out,
    const char * in,
    size_t n )
```

Inverse transform.

### 10.263.3.6 InverseRescaleFunctionIntoBestFit()

```
template<typename TIn >
void gdcm::Rescaler::InverseRescaleFunctionIntoBestFit (
    char * out,
    const TIn * in,
    size_t n ) [protected]
```

### 10.263.3.7 Rescale()

```
bool gdcm::Rescaler::Rescale (
    char * out,
    const char * in,
    size_t n )
```

Direct transform.

#### Examples

[RescaleImage.cs](#).

### 10.263.3.8 RescaleFunctionIntoBestFit()

```
template<typename TIn >
void gdcm::Rescaler::RescaleFunctionIntoBestFit (
    char * out,
    const TIn * in,
    size_t n ) [protected]
```

### 10.263.3.9 SetIntercept()

```
void gdcm::Rescaler::SetIntercept (
    double i ) [inline]
```

Set Intercept: used for both direct&inverse transformation.

#### Examples

[RescaleImage.cs.](#)

### 10.263.3.10 SetMinMaxForPixelType()

```
void gdcm::Rescaler::SetMinMaxForPixelType (
    double min,
    double max )
```

Set target interval for output data. A best match will be computed (if possible) Used for inverse transformation

### 10.263.3.11 SetPixelFormat()

```
void gdcm::Rescaler::SetPixelFormat (
    PixelFormat const & pf ) [inline]
```

Set Pixel Format of input data.

#### Examples

[RescaleImage.cs.](#)

### 10.263.3.12 SetSlope()

```
void gdcm::Rescaler::SetSlope (
    double s ) [inline]
```

Set Slope: user for both direct&inverse transformation.

#### Examples

[RescaleImage.cs.](#)

### 10.263.3.13 SetTargetPixelFormat()

```
void gdcm::Rescaler::SetTargetPixelFormat (
    PixelFormat const & targetst )
```

By default (when `UseTargetPixelFormat` is false), a best matching Target Pixel [Type](#) is computed. However user can override this auto selection by switching `UseTargetPixelFormat:true` and also specifying the specific Target Pixel [Type](#)

### 10.263.3.14 SetUseTargetPixelFormat()

```
void gdcm::Rescaler::SetUseTargetPixelFormat (
    bool b )
```

Override default behavior of Rescale.

The documentation for this class was generated from the following file:

- [gdcmRescaler.h](#)

## 10.264 gdcm::RLECodec Class Reference

Class to do RLE.

```
#include <gdcmRLECodec.h>
```

Inheritance diagram for `gdcm::RLECodec`:





Collaboration diagram for gdcm::RLECodec:



## Public Member Functions

- [RLECodec](#) ()
- [~RLECodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const override  
*Return whether this coder support this transfer syntax (can code it)*
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const override  
*Return whether this decoder support this transfer syntax (can decode it)*
- [ImageCodec](#) \* [Clone](#) () const override
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out) override  
*Code.*
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override  
*Decode.*
- unsigned long [GetBufferLength](#) () const
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override
- void [SetBufferLength](#) (unsigned long l)
- void [SetLength](#) (unsigned long l)

## Public Member Functions inherited from [gdcm::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CleanupUnusedBits](#) (char \*data, size\_t datalen)
- const unsigned int \* [GetDimensions](#) () const
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)

- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- virtual void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)

### Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default

### Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

### Protected Member Functions

- bool [AppendFrameEncode](#) (std::ostream &out, const char \*data, size\_t datalen) override
- bool [AppendRowEncode](#) (std::ostream &out, const char \*data, size\_t datalen) override
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os) override
- bool [DecodeExtent](#) (char \*buffer, unsigned int XMin, unsigned int XMax, unsigned int YMin, unsigned int YMax, unsigned int ZMin, unsigned int ZMax, std::istream &is)
- bool [IsFrameEncoder](#) () override
- bool [IsRowEncoder](#) () override
- bool [StartEncode](#) (std::ostream &) override
- bool [StopEncode](#) (std::ostream &) override

### Protected Member Functions inherited from [gdcm::ImageCodec](#)

- bool [DoByteSwap](#) (std::istream &is\_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is\_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is\_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is\_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is\_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is\_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is\_, std::ostream &os)
- virtual bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)

### Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char \*bv, unsigned long len, std::ostream &os)

**Friends**

- class [ImageRegionReader](#)

**Additional Inherited Members****Protected Types inherited from [gdcm::ImageCodec](#)**

- typedef [SmartPointer](#)< [LookupTable](#) > LUTPtr

**Protected Attributes inherited from [gdcm::ImageCodec](#)**

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) LUT
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) PF
- [PhotometricInterpretation](#) PI
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

**10.264.1 Detailed Description**

Class to do RLE.

**Note**

ANSI X3.9 A.4.2 RLE Compression Annex G defines a RLE Compression Transfer Syntax. This transfer Syntax is identified by the UID value "1.2.840.10008.1.2.5". If the object allows multi-frame images in the pixel data field, then each frame shall be encoded separately. Each frame shall be encoded in one and only one [Fragment](#) (see PS 3.5.8.2).

**10.264.2 Constructor & Destructor Documentation****10.264.2.1 RLECodec()**

```
gdcm::RLECodec::RLECodec ( )
```

**10.264.2.2 ~RLECodec()**

```
gdcm::RLECodec::~~RLECodec ( ) [override]
```

### 10.264.3 Member Function Documentation

#### 10.264.3.1 AppendFrameEncode()

```
bool gdcm::RLECodec::AppendFrameEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### 10.264.3.2 AppendRowEncode()

```
bool gdcm::RLECodec::AppendRowEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### 10.264.3.3 CanCode()

```
bool gdcm::RLECodec::CanCode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

#### 10.264.3.4 CanDecode()

```
bool gdcm::RLECodec::CanDecode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

#### 10.264.3.5 Clone()

```
ImageCodec * gdcm::RLECodec::Clone ( ) const [override], [virtual]
```

Implements [gdcm::ImageCodec](#).

### 10.264.3.6 Code()

```
bool gdcm::RLECodec::Code (
    DataElement const & in_,
    DataElement & out_ ) [override], [virtual]
```

Code.

Reimplemented from [gdcm::Coder](#).

### 10.264.3.7 Decode()

```
bool gdcm::RLECodec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcm::ImageCodec](#).

### 10.264.3.8 DecodeByStreams()

```
bool gdcm::RLECodec::DecodeByStreams (
    std::istream & is,
    std::ostream & os ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.264.3.9 DecodeExtent()

```
bool gdcm::RLECodec::DecodeExtent (
    char * buffer,
    unsigned int XMin,
    unsigned int XMax,
    unsigned int YMin,
    unsigned int YMax,
    unsigned int ZMin,
    unsigned int ZMax,
    std::istream & is ) [protected]
```

### 10.264.3.10 GetBufferLength()

```
unsigned long gdcm::RLECodec::GetBufferLength ( ) const [inline]
```

#### 10.264.3.11 GetHeaderInfo()

```
bool gdcm::RLECodec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### 10.264.3.12 IsFrameEncoder()

```
bool gdcm::RLECodec::IsFrameEncoder ( ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### 10.264.3.13 IsRowEncoder()

```
bool gdcm::RLECodec::IsRowEncoder ( ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### 10.264.3.14 SetBufferLength()

```
void gdcm::RLECodec::SetBufferLength (
    unsigned long l ) [inline]
```

#### 10.264.3.15 SetLength()

```
void gdcm::RLECodec::SetLength (
    unsigned long l ) [inline]
```

#### 10.264.3.16 StartEncode()

```
bool gdcm::RLECodec::StartEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### 10.264.3.17 StopEncode()

```
bool gdcm::RLECodec::StopEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

## 10.264.4 Friends And Related Symbol Documentation

### 10.264.4.1 ImageRegionReader

```
friend class ImageRegionReader [friend]
```

The documentation for this class was generated from the following file:

- [gdcmRLECodec.h](#)

## 10.265 gdcm::network::RoleSelectionSub Class Reference

[RoleSelectionSub.](#)

```
#include <gdcmRoleSelectionSub.h>
```

### Public Member Functions

- [RoleSelectionSub](#) ()
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [SetTuple](#) (const char \*uid, uint8\_t scurole, uint8\_t scprole)
- size\_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

### 10.265.1 Detailed Description

[RoleSelectionSub.](#)

PS 3.7 [Table D.3-9](#) SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

### 10.265.2 Constructor & Destructor Documentation

#### 10.265.2.1 RoleSelectionSub()

```
gdcm::network::RoleSelectionSub::RoleSelectionSub ( )
```

### 10.265.3 Member Function Documentation

#### 10.265.3.1 Print()

```
void gdcm::network::RoleSelectionSub::Print (
    std::ostream & os ) const
```

### 10.265.3.2 Read()

```
std::istream & gdcmm::network::RoleSelectionSub::Read (
    std::istream & is )
```

### 10.265.3.3 SetTuple()

```
void gdcmm::network::RoleSelectionSub::SetTuple (
    const char * uid,
    uint8_t scurole,
    uint8_t scprole )
```

### 10.265.3.4 Size()

```
size_t gdcmm::network::RoleSelectionSub::Size ( ) const
```

### 10.265.3.5 Write()

```
const std::ostream & gdcmm::network::RoleSelectionSub::Write (
    std::ostream & os ) const
```

The documentation for this class was generated from the following file:

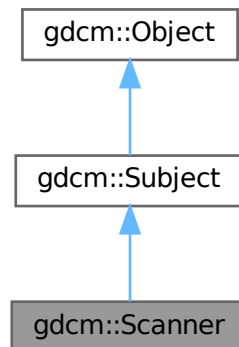
- [gdcmmRoleSelectionSub.h](#)

## 10.266 gdcmm::Scanner Class Reference

[Scanner](#).

```
#include <gdcmmScanner.h>
```

Inheritance diagram for gdcmm::Scanner:





Collaboration diagram for gdcmm::Scanner:



## Classes

- struct [Itstr](#)

## Public Types

- typedef MappingType::const\_iterator [ConstIterator](#)
- typedef std::map< const char \*, [TagToValue](#), [Itstr](#) > [MappingType](#)
- typedef std::map< [Tag](#), const char \* > [TagToValue](#)
- typedef TagToValue::value\_type [TagToValueValueType](#)
- typedef std::set< std::string > [ValuesType](#)

## Public Member Functions

- [Scanner](#) ()
- [~Scanner](#) () override
- void [AddPrivateTag](#) ([PrivateTag](#) const &t)
- void [AddSkipTag](#) ([Tag](#) const &t)  
*Add a tag that will need to be skipped. Those are root level skip tags.*
- void [AddTag](#) ([Tag](#) const &t)  
*Add a tag that will need to be read. Those are root level tags.*
- [ConstIterator](#) [Begin](#) () const
- void [ClearSkipTags](#) ()
- void [ClearTags](#) ()
- [ConstIterator](#) [End](#) () const
- [Directory::FilenameType](#) [GetAllFileNamesFromTagToValue](#) ([Tag](#) const &t, const char \*valueref) const
- const char \* [GetFilenameFromTagToValue](#) ([Tag](#) const &t, const char \*valueref) const
- [Directory::FilenameType](#) const & [GetFileNames](#) () const

- [Directory::FilenamesType GetKeys](#) () const
- [TagToValue](#) const & [GetMapping](#) (const char \*filename) const  
*Get the std::map mapping filenames to value for file 'filename'.*
- [TagToValue](#) const & [GetMappingFromTagToValue](#) ([Tag](#) const &t, const char \*value) const  
*See [GetFilenameFromTagToValue\(\)](#). This is simply [GetFilenameFromTagToValue](#) followed.*
- [MappingType](#) const & [GetMappings](#) () const  
*Mappings are the mapping from a particular tag to the map, mapping filename to value:*
- [Directory::FilenamesType GetOrderedValues](#) ([Tag](#) const &t) const
- const char \* [GetValue](#) (const char \*filename, [Tag](#) const &t) const
- [ValuesType](#) const & [GetValues](#) () const  
*Get all the values found (in lexicographic order)*
- [ValuesType GetValues](#) ([Tag](#) const &t) const  
*Get all the values found (in lexicographic order) associated with [Tag](#) 't'.*
- bool [IsKey](#) (const char \*filename) const
- void [Print](#) (std::ostream &os) const override  
*Print result.*
- void [PrintTable](#) (std::ostream &os) const
- bool [Scan](#) ([Directory::FilenamesType](#) const &filenames)  
*Start the scan !*

## Public Member Functions inherited from [gdcm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*) const
- [Command](#) \* [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

## Static Public Member Functions

- static [SmartPointer](#)< [Scanner](#) > [New](#) ()  
*for wrapped language: instantiate a reference counted object*

### Protected Member Functions

- void [ProcessPublicTag](#) ([StringFilter](#) &sf, const char \*filename)

### Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

### Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [Scanner](#) &s)

## 10.266.1 Detailed Description

### [Scanner](#).

This filter is meant for quickly browsing a [FileSet](#) (a set of files on disk). Special consideration are taken so as to read the minimum amount of information in each file in order to retrieve the user specified set of DICOM [Attribute](#).

This filter is dealing with both VRASCII and VRBINARY element, thanks to the help of [StringFilter](#)

### Warning

IMPORTANT In case of file where tags are not ordered (illegal as per DICOM specification), the output will be missing information

### Note

implementation details. All values are stored in a std::set of std::string. Then the address of the cstring underlying the std::string is used in the std::map.

This class implement the Subject/Observer pattern trigger the following events:

- [ProgressEvent](#)
- [StartEvent](#)
- [EndEvent](#)

### Examples

[DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).

## 10.266.2 Member Typedef Documentation

### 10.266.2.1 ConstIterator

```
typedef MappingType::const_iterator gdcm::Scanner::ConstIterator
```

### 10.266.2.2 MappingType

```
typedef std::map<const char *, TagToValue, ltstr> gdcm::Scanner::MappingType
```

### 10.266.2.3 TagToValue

```
typedef std::map<Tag, const char*> gdcm::Scanner::TagToValue
```

struct to map a filename to a value Implementation note: all std::map in this class will be using const char \* and not std::string since we are pointing to existing std::string (hold in a std::vector) this avoid an extra copy of the byte array. Tag are used as Tag class since sizeof(tag) <= sizeof(pointer)

### 10.266.2.4 TagToValueValueType

```
typedef TagToValue::value_type gdcm::Scanner::TagToValueValueType
```

### 10.266.2.5 ValuesType

```
typedef std::set< std::string > gdcm::Scanner::ValuesType
```

## 10.266.3 Constructor & Destructor Documentation

### 10.266.3.1 Scanner()

```
gdcm::Scanner::Scanner ( ) [inline]
```

### 10.266.3.2 ~Scanner()

```
gdcm::Scanner::~~Scanner ( ) [override]
```

## 10.266.4 Member Function Documentation

### 10.266.4.1 AddPrivateTag()

```
void gdcm::Scanner::AddPrivateTag (
    PrivateTag const & t )
```

#### 10.266.4.2 AddSkipTag()

```
void gdcm::Scanner::AddSkipTag (
    Tag const & t )
```

Add a tag that will need to be skipped. Those are root level skip tags.

#### 10.266.4.3 AddTag()

```
void gdcm::Scanner::AddTag (
    Tag const & t )
```

Add a tag that will need to be read. Those are root level tags.

#### Examples

[DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).

#### 10.266.4.4 Begin()

```
ConstIterator gdcm::Scanner::Begin ( ) const [inline]
```

#### 10.266.4.5 ClearSkipTags()

```
void gdcm::Scanner::ClearSkipTags ( )
```

#### 10.266.4.6 ClearTags()

```
void gdcm::Scanner::ClearTags ( )
```

#### 10.266.4.7 End()

```
ConstIterator gdcm::Scanner::End ( ) const [inline]
```

#### 10.266.4.8 GetAllFileNamesFromTagToValue()

```
Directory::FileNamesType gdcm::Scanner::GetAllFileNamesFromTagToValue (
    Tag const & t,
    const char * valuref ) const
```

Will loop over all files and return a vector of std::strings of filenames where value match the reference value 'valuref'

#### 10.266.4.9 GetFilenameFromTagToValue()

```
const char * gdcm::Scanner::GetFilenameFromTagToValue (
    Tag const & t,
    const char * valueref ) const
```

Will loop over all files and return the first file where value match the reference value 'valueref'

#### 10.266.4.10 GetFileNames()

```
Directory::FileNamesType const & gdcm::Scanner::GetFileNames ( ) const [inline]
```

#### 10.266.4.11 GetKeys()

```
Directory::FileNamesType gdcm::Scanner::GetKeys ( ) const
```

Return the list of filename that are key in the internal map, which means those filename were properly parsed

##### Examples

[VolumeSorter.cxx](#).

#### 10.266.4.12 GetMapping()

```
TagToValue const & gdcm::Scanner::GetMapping (
    const char * filename ) const
```

Get the std::map mapping filenames to value for file 'filename'.

##### Examples

[DumpToSQLITE3.cxx](#).

#### 10.266.4.13 GetMappingFromTagToValue()

```
TagToValue const & gdcm::Scanner::GetMappingFromTagToValue (
    Tag const & t,
    const char * value ) const
```

See [GetFilenameFromTagToValue\(\)](#). This is simply GetFilenameFromTagToValue followed.

#### 10.266.4.14 GetMappings()

```
MappingType const & gdcm::Scanner::GetMappings ( ) const [inline]
```

Mappings are the mapping from a particular tag to the map, mapping filename to value:

#### 10.266.4.15 GetOrderedValues()

```
Directory::FilenameType gdcm::Scanner::GetOrderedValues (
    Tag const & t ) const
```

Get all the values found (in a vector) associated with [Tag](#) 't' This function is identical to [GetValues](#), but is accessible from the wrapped layer (python, C#, java)

#### 10.266.4.16 GetValue()

```
const char * gdcm::Scanner::GetValue (
    const char * filename,
    Tag const & t ) const
```

Retrieve the value found for tag: t associated with file: filename This is meant for a single short call. If multiple calls (multiple tags) should be done, prefer the [GetMapping](#) function, and then reuse the [TagToValue](#) hash table.

#### Warning

[Tag](#) 't' should have been added via [AddTag\(\)](#) prior to the [Scan\(\)](#) call !

#### 10.266.4.17 GetValues() [1/2]

```
ValueType const & gdcm::Scanner::GetValues ( ) const [inline]
```

Get all the values found (in lexicographic order)

#### Examples

[SortImage.cxx](#), and [VolumeSorter.cxx](#).

#### 10.266.4.18 GetValues() [2/2]

```
ValueType gdcm::Scanner::GetValues (
    Tag const & t ) const
```

Get all the values found (in lexicographic order) associated with [Tag](#) 't'.

#### 10.266.4.19 IsKey()

```
bool gdcM::Scanner::IsKey (
    const char * filename ) const
```

Check if filename is a key in the Mapping table. returns true only if file can be found, which means the file was indeed a DICOM file that could be processed

##### Examples

[DumpToSQLITE3.cxx](#).

#### 10.266.4.20 New()

```
static SmartPointer< Scanner > gdcM::Scanner::New ( ) [inline], [static]
```

for wrapped language: instantiate a reference counted object

#### 10.266.4.21 Print()

```
void gdcM::Scanner::Print (
    std::ostream & os ) const [override], [virtual]
```

Print result.

Reimplemented from [gdcM::Object](#).

#### 10.266.4.22 PrintTable()

```
void gdcM::Scanner::PrintTable (
    std::ostream & os ) const
```

#### 10.266.4.23 ProcessPublicTag()

```
void gdcM::Scanner::ProcessPublicTag (
    StringFilter & sf,
    const char * filename ) [protected]
```

#### 10.266.4.24 Scan()

```
bool gdcM::Scanner::Scan (
    Directory::FileNamesType const & filenames )
```

Start the scan !

##### Examples

[DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).



## 10.266.5 Friends And Related Symbol Documentation

### 10.266.5.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & _os,  
    const Scanner & s ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmScanner.h](#)

## 10.267 gdcm::Scanner2 Class Reference

[Scanner2.](#)

```
#include <gdcmScanner2.h>
```

Inheritance diagram for gdcm::Scanner2:



Collaboration diagram for `gdcm::Scanner2`:



## Classes

- struct [Itstr](#)

## Public Types

- typedef `PrivateMappingType::const_iterator` [PrivateConstIterator](#)
- typedef `std::map< const char *, PrivateTagToValue, Itstr >` [PrivateMappingType](#)
- typedef `std::map< PrivateTag, const char * >` [PrivateTagToValue](#)
- typedef `PrivateTagToValue::value_type` [PrivateTagToValueValueType](#)
- typedef `PublicMappingType::const_iterator` [PublicConstIterator](#)
- typedef `std::map< const char *, PublicTagToValue, Itstr >` [PublicMappingType](#)
- typedef `std::map< Tag, const char * >` [PublicTagToValue](#)
- typedef `PublicTagToValue::value_type` [PublicTagToValueValueType](#)
- typedef `std::set< std::string >` [ValuesType](#)

## Public Member Functions

- [Scanner2](#) ()
- [~Scanner2](#) () override
- bool [AddPrivateTag](#) ([PrivateTag](#) const &pt)
- bool [AddPublicTag](#) ([Tag](#) const &t)  
*Add a tag that will need to be read. Those are root level tags.*
- bool [AddSkipTag](#) ([Tag](#) const &t)  
*Add a tag that will need to be skipped. Those are root level skip tags.*
- [PublicConstIterator](#) [Begin](#) () const
- void [ClearPrivateTags](#) ()
- void [ClearPublicTags](#) ()

- void [ClearSkipTags](#) ()
- [PublicConstIterator End](#) () const
- [Directory::FilenamesType GetAllFilenamesFromPrivateTagToValue](#) ([PrivateTag](#) const &pt, const char \*valueref) const
- [Directory::FilenamesType GetAllFilenamesFromPublicTagToValue](#) ([Tag](#) const &t, const char \*valueref) const
- const char \* [GetFilenameFromPrivateTagToValue](#) ([PrivateTag](#) const &pt, const char \*valueref) const
- const char \* [GetFilenameFromPublicTagToValue](#) ([Tag](#) const &t, const char \*valueref) const
- [Directory::FilenamesType](#) const & [GetFilenames](#) () const
 

*Return the list of filenames.*
- [Directory::FilenamesType GetKeys](#) () const
- [PrivateTagToValue](#) const & [GetMappingFromPrivateTagToValue](#) ([PrivateTag](#) const &pt, const char \*value) const
- [PublicTagToValue](#) const & [GetMappingFromPublicTagToValue](#) ([Tag](#) const &t, const char \*value) const
 

*See [GetFilenameFromTagToValue](#)(). This is simply [GetFilenameFromTagToValue](#) followed.*
- [PrivateTagToValue](#) const & [GetPrivateMapping](#) (const char \*filename) const
- [PrivateMappingType](#) const & [GetPrivateMappings](#) () const
- [Directory::FilenamesType GetPrivateOrderedValues](#) ([PrivateTag](#) const &pt) const
- const char \* [GetPrivateValue](#) (const char \*filename, [PrivateTag](#) const &t) const
- [ValuesType GetPrivateValues](#) ([PrivateTag](#) const &pt) const
 

*Get all the values found (in lexicographic order) associated with [PrivateTag](#) 'pt'.*
- [PublicTagToValue](#) const & [GetPublicMapping](#) (const char \*filename) const
 

*Get the std::map mapping filenames to value for file 'filename'.*
- [PublicMappingType](#) const & [GetPublicMappings](#) () const
 

*Mappings are the mapping from a particular tag to the map, mapping filename to value:*
- [Directory::FilenamesType GetPublicOrderedValues](#) ([Tag](#) const &t) const
- const char \* [GetPublicValue](#) (const char \*filename, [Tag](#) const &t) const
- [ValuesType GetPublicValues](#) ([Tag](#) const &t) const
 

*Get all the values found (in lexicographic order) associated with [Tag](#) 't'.*
- [ValuesType](#) const & [GetValues](#) () const
 

*Get all the values found (in lexicographic order)*
- bool [IsKey](#) (const char \*filename) const
- void [Print](#) (std::ostream &os) const override
 

*Print result.*
- void [PrintTable](#) (std::ostream &os, bool header=false) const
 

*Print result as CSV table.*
- [PrivateConstIterator PrivateBegin](#) () const
- [PrivateConstIterator PrivateEnd](#) () const
- bool [Scan](#) ([Directory::FilenamesType](#) const &filenames)
 

*Start the scan !*

## Public Member Functions inherited from [gdcmm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*) const
- [Command](#) \* [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

## Static Public Member Functions

- static [SmartPointer](#)< [Scanner2](#) > [New](#) ()  
*for wrapped language: instantiate a reference counted object*

## Protected Member Functions

- void [ProcessPrivateTag](#) ([StringFilter](#) &sf, const char \*filename)
- void [ProcessPublicTag](#) ([StringFilter](#) &sf, const char \*filename)

## Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

## Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [Scanner2](#) &s)

## 10.267.1 Detailed Description

### [Scanner2](#).

This filter is meant for quickly browsing a [FileSet](#) (a set of files on disk). Special consideration are taken so as to read the minimum amount of information in each file in order to retrieve the user specified set of DICOM [Attribute](#).

This filter is dealing with both VRASCII and VRBINARY element, thanks to the help of [StringFilter](#)

### Warning

IMPORTANT In case of file where tags are not ordered (illegal as per DICOM specification), the output will be missing information

### Note

implementation details. All values are stored in a std::set of std::string. Then the address of the cstring underlying the std::string is used in the std::map.

This class implement the Subject/Observer pattern trigger the following events:

- [ProgressEvent](#)
- [StartEvent](#)
- [EndEvent](#)

## 10.267.2 Member Typedef Documentation

### 10.267.2.1 PrivateConstIterator

```
typedef PrivateMappingType::const_iterator gdcm::Scanner2::PrivateConstIterator
```

### 10.267.2.2 PrivateMappingType

```
typedef std::map<const char *,PrivateTagToValue, ltstr> gdcm::Scanner2::PrivateMappingType
```

### 10.267.2.3 PrivateTagToValue

```
typedef std::map<PrivateTag, const char*> gdcm::Scanner2::PrivateTagToValue
```

### 10.267.2.4 PrivateTagToValueValueType

```
typedef PrivateTagToValue::value_type gdcm::Scanner2::PrivateTagToValueValueType
```

### 10.267.2.5 PublicConstIterator

```
typedef PublicMappingType::const_iterator gdcm::Scanner2::PublicConstIterator
```

### 10.267.2.6 PublicMappingType

```
typedef std::map<const char *,PublicTagToValue, ltstr> gdcm::Scanner2::PublicMappingType
```

### 10.267.2.7 PublicTagToValue

```
typedef std::map<Tag, const char*> gdcm::Scanner2::PublicTagToValue
```

struct to map a filename to a value Implementation note: all std::map in this class will be using const char \* and not std::string since we are pointing to existing std::string (held in a std::vector) this avoid an extra copy of the byte array. [Tag](#) are used as [Tag](#) class since sizeof(tag) <= sizeof(pointer)

### 10.267.2.8 PublicTagToValueValueType

```
typedef PublicTagToValue::value_type gdcm::Scanner2::PublicTagToValueValueType
```

### 10.267.2.9 ValueType

```
typedef std::set< std::string > gdcM::Scanner2::ValueType
```

## 10.267.3 Constructor & Destructor Documentation

### 10.267.3.1 Scanner2()

```
gdcM::Scanner2::Scanner2 ( ) [inline]
```

### 10.267.3.2 ~Scanner2()

```
gdcM::Scanner2::~~Scanner2 ( ) [override]
```

## 10.267.4 Member Function Documentation

### 10.267.4.1 AddPrivateTag()

```
bool gdcM::Scanner2::AddPrivateTag (
    PrivateTag const & pt )
```

### 10.267.4.2 AddPublicTag()

```
bool gdcM::Scanner2::AddPublicTag (
    Tag const & t )
```

Add a tag that will need to be read. Those are root level tags.

### 10.267.4.3 AddSkipTag()

```
bool gdcM::Scanner2::AddSkipTag (
    Tag const & t )
```

Add a tag that will need to be skipped. Those are root level skip tags.

### 10.267.4.4 Begin()

```
PublicConstIterator gdcM::Scanner2::Begin ( ) const [inline]
```

#### 10.267.4.5 ClearPrivateTags()

```
void gdcm::Scanner2::ClearPrivateTags ( )
```

#### 10.267.4.6 ClearPublicTags()

```
void gdcm::Scanner2::ClearPublicTags ( )
```

#### 10.267.4.7 ClearSkipTags()

```
void gdcm::Scanner2::ClearSkipTags ( )
```

#### 10.267.4.8 End()

```
PublicConstIterator gdcm::Scanner2::End ( ) const [inline]
```

#### 10.267.4.9 GetAllFileNamesFromPrivateTagToValue()

```
Directory::FileNamesType gdcm::Scanner2::GetAllFileNamesFromPrivateTagToValue (
    PrivateTag const & pt,
    const char * valueref ) const
```

#### 10.267.4.10 GetAllFileNamesFromPublicTagToValue()

```
Directory::FileNamesType gdcm::Scanner2::GetAllFileNamesFromPublicTagToValue (
    Tag const & t,
    const char * valueref ) const
```

Will loop over all files and return a vector of std::strings of filenames where value match the reference value 'valueref'

#### 10.267.4.11 GetFilenameFromPrivateTagToValue()

```
const char * gdcm::Scanner2::GetFilenameFromPrivateTagToValue (
    PrivateTag const & pt,
    const char * valueref ) const
```

#### 10.267.4.12 GetFilenameFromPublicTagToValue()

```
const char * gdcm::Scanner2::GetFilenameFromPublicTagToValue (
    Tag const & t,
    const char * valueref ) const
```

Will loop over all files and return the first file where value match the reference value 'valueref'

#### 10.267.4.13 GetFilenames()

```
Directory::FilenamesType const & gdcm::Scanner2::GetFilenames ( ) const [inline]
```

Return the list of filenames.

#### 10.267.4.14 GetKeys()

```
Directory::FilenamesType gdcm::Scanner2::GetKeys ( ) const
```

Return the list of filename that are key in the internal map, which means those filename were properly parsed

#### 10.267.4.15 GetMappingFromPrivateTagToValue()

```
PrivateTagToValue const & gdcm::Scanner2::GetMappingFromPrivateTagToValue (
    PrivateTag const & pt,
    const char * value ) const
```

#### 10.267.4.16 GetMappingFromPublicTagToValue()

```
PublicTagToValue const & gdcm::Scanner2::GetMappingFromPublicTagToValue (
    Tag const & t,
    const char * value ) const
```

See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed.

#### 10.267.4.17 GetPrivateMapping()

```
PrivateTagToValue const & gdcm::Scanner2::GetPrivateMapping (
    const char * filename ) const
```

#### 10.267.4.18 GetPrivateMappings()

```
PrivateMappingType const & gdcm::Scanner2::GetPrivateMappings ( ) const [inline]
```

#### 10.267.4.19 GetPrivateOrderedValues()

```
Directory::FilenamesType gdcm::Scanner2::GetPrivateOrderedValues (
    PrivateTag const & pt ) const
```



#### 10.267.4.20 GetPrivateValue()

```
const char * gdcm::Scanner2::GetPrivateValue (
    const char * filename,
    PrivateTag const & t ) const
```

#### 10.267.4.21 GetPrivateValues()

```
ValueType gdcm::Scanner2::GetPrivateValues (
    PrivateTag const & pt ) const
```

Get all the values found (in lexicographic order) associated with [PrivateTag](#) 'pt'.

#### 10.267.4.22 GetPublicMapping()

```
PublicTagToValue const & gdcm::Scanner2::GetPublicMapping (
    const char * filename ) const
```

Get the std::map mapping filenames to value for file 'filename'.

#### 10.267.4.23 GetPublicMappings()

```
PublicMappingType const & gdcm::Scanner2::GetPublicMappings ( ) const [inline]
```

Mappings are the mapping from a particular tag to the map, mapping filename to value:

#### 10.267.4.24 GetPublicOrderedValues()

```
Directory::FileNamesType gdcm::Scanner2::GetPublicOrderedValues (
    Tag const & t ) const
```

Get all the values found (in a vector) associated with [Tag](#) 't' This function is identical to `GetValues`, but is accessible from the wrapped layer (python, C#, java)

#### 10.267.4.25 GetPublicValue()

```
const char * gdcm::Scanner2::GetPublicValue (
    const char * filename,
    Tag const & t ) const
```

Retrieve the value found for tag: t associated with file: filename This is meant for a single short call. If multiple calls (multiple tags) should be done, prefer the `GetMapping` function, and then reuse the `TagToValue` hash table.

#### Warning

[Tag](#) 't' should have been added via `AddTag()` prior to the `Scan()` call !

**10.267.4.26 GetPublicValues()**

```
ValueType gdcM::Scanner2::GetPublicValues (
    Tag const & t ) const
```

Get all the values found (in lexicographic order) associated with Tag 't'.

**10.267.4.27 GetValues()**

```
ValueType const & gdcM::Scanner2::GetValues ( ) const [inline]
```

Get all the values found (in lexicographic order)

**10.267.4.28 IsKey()**

```
bool gdcM::Scanner2::IsKey (
    const char * filename ) const
```

Check if filename is a key in the Mapping table. returns true only if file can be found, which means the file was indeed a DICOM file that could be processed

**10.267.4.29 New()**

```
static SmartPointer< Scanner2 > gdcM::Scanner2::New ( ) [inline], [static]
```

for wrapped language: instantiate a reference counted object

**10.267.4.30 Print()**

```
void gdcM::Scanner2::Print (
    std::ostream & os ) const [override], [virtual]
```

Print result.

Reimplemented from [gdcM::Object](#).

**10.267.4.31 PrintTable()**

```
void gdcM::Scanner2::PrintTable (
    std::ostream & os,
    bool header = false ) const
```

Print result as CSV table.

#### 10.267.4.32 PrivateBegin()

```
PrivateConstIterator gdcm::Scanner2::PrivateBegin ( ) const [inline]
```

#### 10.267.4.33 PrivateEnd()

```
PrivateConstIterator gdcm::Scanner2::PrivateEnd ( ) const [inline]
```

#### 10.267.4.34 ProcessPrivateTag()

```
void gdcm::Scanner2::ProcessPrivateTag (
    StringFilter & sf,
    const char * filename ) [protected]
```

#### 10.267.4.35 ProcessPublicTag()

```
void gdcm::Scanner2::ProcessPublicTag (
    StringFilter & sf,
    const char * filename ) [protected]
```

#### 10.267.4.36 Scan()

```
bool gdcm::Scanner2::Scan (
    Directory::FileNamesType const & filenames )
```

Start the scan !

### 10.267.5 Friends And Related Symbol Documentation

#### 10.267.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const Scanner2 & s ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmScanner2.h](#)

## 10.268 gdcm::Segment Class Reference

This class defines a segment.

```
#include <gdcmSegment.h>
```

Inheritance diagram for gdcm::Segment:



Collaboration diagram for gdcm::Segment:



### Public Types

- enum `ALGOType` {  
`AUTOMATIC = 0` ,  
`SEMIAUTOMATIC` ,  
`MANUAL` ,  
`ALGOType_END` }
- typedef `std::vector< SegmentHelper::BasicCodedEntry >` `BasicCodedEntryVector`
- typedef `std::vector< SmartPointer< Surface > >` `SurfaceVector`

**Public Member Functions**

- [Segment](#) ()
- [~Segment](#) () override
- void [AddSurface](#) ([SmartPointer](#)< [Surface](#) > surface)
- [SegmentHelper::BasicCodedEntry](#) & [GetAnatomicRegion](#) ()
- [SegmentHelper::BasicCodedEntry](#) const & [GetAnatomicRegion](#) () const
- [BasicCodedEntryVector](#) & [GetAnatomicRegionModifiers](#) ()
- [BasicCodedEntryVector](#) const & [GetAnatomicRegionModifiers](#) () const
- [SegmentHelper::BasicCodedEntry](#) & [GetPropertyCategory](#) ()
- [SegmentHelper::BasicCodedEntry](#) const & [GetPropertyCategory](#) () const
- [SegmentHelper::BasicCodedEntry](#) & [GetPropertyType](#) ()
- [SegmentHelper::BasicCodedEntry](#) const & [GetPropertyType](#) () const
- [BasicCodedEntryVector](#) & [GetPropertyTypeModifiers](#) ()
- [BasicCodedEntryVector](#) const & [GetPropertyTypeModifiers](#) () const
- const char \* [GetSegmentAlgorithmName](#) () const
- [ALGOType](#) [GetSegmentAlgorithmType](#) () const
- const char \* [GetSegmentDescription](#) () const
- const char \* [GetSegmentLabel](#) () const
- unsigned short [GetSegmentNumber](#) () const
- [SmartPointer](#)< [Surface](#) > [GetSurface](#) (const unsigned int idx=0) const
- unsigned long [GetSurfaceCount](#) ()
- [SurfaceVector](#) & [GetSurfaces](#) ()
- [SurfaceVector](#) const & [GetSurfaces](#) () const
- void [SetAnatomicRegion](#) ([SegmentHelper::BasicCodedEntry](#) const &BSE)
- void [SetAnatomicRegionModifiers](#) ([BasicCodedEntryVector](#) const &BSEV)
- void [SetPropertyCategory](#) ([SegmentHelper::BasicCodedEntry](#) const &BSE)
- void [SetPropertyType](#) ([SegmentHelper::BasicCodedEntry](#) const &BSE)
- void [SetPropertyTypeModifiers](#) ([BasicCodedEntryVector](#) const &BSEV)
- void [SetSegmentAlgorithmName](#) (const char \*name)
- void [SetSegmentAlgorithmType](#) ([ALGOType](#) type)
- void [SetSegmentAlgorithmType](#) (const char \*typeStr)
- void [SetSegmentDescription](#) (const char \*description)
- void [SetSegmentLabel](#) (const char \*label)
- void [SetSegmentNumber](#) (const unsigned short num)
- void [SetSurfaceCount](#) (const unsigned long nb)

**Public Member Functions inherited from [gdcm::Object](#)**

- [Object](#) ()
- [Object](#) (const [Object](#) &)
- *Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

**Static Public Member Functions**

- static [ALGOType](#) [GetALGOType](#) (const char \*type)
- static const char \* [GetALGOTypeString](#) ([ALGOType](#) type)

**Protected Attributes**

- [SegmentHelper::BasicCodedEntry](#) [AnatomicRegion](#)
- [BasicCodedEntryVector](#) [AnatomicRegionModifiers](#)
- [SegmentHelper::BasicCodedEntry](#) [PropertyCategory](#)
- [SegmentHelper::BasicCodedEntry](#) [PropertyType](#)
- [BasicCodedEntryVector](#) [PropertyTypeModifiers](#)
- `std::string` [SegmentAlgorithmName](#)
- [ALGOType](#) [SegmentAlgorithmType](#)
- `std::string` [SegmentDescription](#)
- `std::string` [SegmentLabel](#)
- `unsigned short` [SegmentNumber](#)
- `unsigned long` [SurfaceCount](#)
- [SurfaceVector](#) [Surfaces](#)

**Additional Inherited Members****Protected Member Functions inherited from [gdcm::Object](#)**

- `void` [Register](#) ()
- `void` [UnRegister](#) ()

**10.268.1 Detailed Description**

This class defines a segment.

It mainly contains attributes of group 0x0062. In addition, it can be associated with surface.

See also

PS 3.3 C.8.20.2 and C.8.23

**10.268.2 Member Typedef Documentation****10.268.2.1 BasicCodedEntryVector**

```
typedef std::vector< SegmentHelper::BasicCodedEntry > gdcm::Segment::BasicCodedEntryVector
```

**10.268.2.2 SurfaceVector**

```
typedef std::vector< SmartPointer< Surface > > gdcm::Segment::SurfaceVector
```

**10.268.3 Member Enumeration Documentation****10.268.3.1 ALGOType**

```
enum gdcm::Segment::ALGOType
```

## Enumerator

AUTOMATIC	
SEMIAUTOMATIC	
MANUAL	
ALGOType_END	

## 10.268.4 Constructor & Destructor Documentation

### 10.268.4.1 Segment()

```
gdcm::Segment::Segment ( )
```

### 10.268.4.2 ~Segment()

```
gdcm::Segment::~~Segment ( ) [override]
```

## 10.268.5 Member Function Documentation

### 10.268.5.1 AddSurface()

```
void gdcm::Segment::AddSurface (
    SmartPointer< Surface > surface )
```

### 10.268.5.2 GetALGOType()

```
static ALGOType gdcm::Segment::GetALGOType (
    const char * type ) [static]
```

### 10.268.5.3 GetALGOTypeString()

```
static const char * gdcm::Segment::GetALGOTypeString (
    ALGOType type ) [static]
```

### 10.268.5.4 GetAnatomicRegion() [1/2]

```
SegmentHelper::BasicCodedEntry & gdcm::Segment::GetAnatomicRegion ( )
```

**10.268.5.5 GetAnatomicRegion() [2/2]**

```
SegmentHelper::BasicCodedEntry const & gdcM::Segment::GetAnatomicRegion ( ) const
```

**10.268.5.6 GetAnatomicRegionModifiers() [1/2]**

```
BasicCodedEntryVector & gdcM::Segment::GetAnatomicRegionModifiers ( )
```

**10.268.5.7 GetAnatomicRegionModifiers() [2/2]**

```
BasicCodedEntryVector const & gdcM::Segment::GetAnatomicRegionModifiers ( ) const
```

**10.268.5.8 GetPropertyCategory() [1/2]**

```
SegmentHelper::BasicCodedEntry & gdcM::Segment::GetPropertyCategory ( )
```

**10.268.5.9 GetPropertyCategory() [2/2]**

```
SegmentHelper::BasicCodedEntry const & gdcM::Segment::GetPropertyCategory ( ) const
```

**10.268.5.10 GetPropertyType() [1/2]**

```
SegmentHelper::BasicCodedEntry & gdcM::Segment::GetPropertyType ( )
```

**10.268.5.11 GetPropertyType() [2/2]**

```
SegmentHelper::BasicCodedEntry const & gdcM::Segment::GetPropertyType ( ) const
```

**10.268.5.12 GetPropertyTypeModifiers() [1/2]**

```
BasicCodedEntryVector & gdcM::Segment::GetPropertyTypeModifiers ( )
```

**10.268.5.13 GetPropertyTypeModifiers() [2/2]**

```
BasicCodedEntryVector const & gdcM::Segment::GetPropertyTypeModifiers ( ) const
```

**10.268.5.14 GetSegmentAlgorithmName()**

```
const char * gdcM::Segment::GetSegmentAlgorithmName ( ) const
```



#### 10.268.5.15 GetSegmentAlgorithmType()

```
ALGOType gdcm::Segment::GetSegmentAlgorithmType ( ) const
```

#### 10.268.5.16 GetSegmentDescription()

```
const char * gdcm::Segment::GetSegmentDescription ( ) const
```

#### 10.268.5.17 GetSegmentLabel()

```
const char * gdcm::Segment::GetSegmentLabel ( ) const
```

#### 10.268.5.18 GetSegmentNumber()

```
unsigned short gdcm::Segment::GetSegmentNumber ( ) const
```

#### 10.268.5.19 GetSurface()

```
SmartPointer< Surface > gdcm::Segment::GetSurface (
    const unsigned int idx = 0 ) const
```

#### 10.268.5.20 GetSurfaceCount()

```
unsigned long gdcm::Segment::GetSurfaceCount ( )
```

#### 10.268.5.21 GetSurfaces() [1/2]

```
SurfaceVector & gdcm::Segment::GetSurfaces ( )
```

#### 10.268.5.22 GetSurfaces() [2/2]

```
SurfaceVector const & gdcm::Segment::GetSurfaces ( ) const
```

#### 10.268.5.23 SetAnatomicRegion()

```
void gdcm::Segment::SetAnatomicRegion (
    SegmentHelper::BasicCodedEntry const & BSE )
```

**10.268.5.24 SetAnatomicRegionModifiers()**

```
void gdcM::Segment::SetAnatomicRegionModifiers (
    BasicCodedEntryVector const & BSEV )
```

**10.268.5.25 SetPropertyCategory()**

```
void gdcM::Segment::SetPropertyCategory (
    SegmentHelper::BasicCodedEntry const & BSE )
```

**10.268.5.26 SetPropertyType()**

```
void gdcM::Segment::SetPropertyType (
    SegmentHelper::BasicCodedEntry const & BSE )
```

**10.268.5.27 SetPropertyTypeModifiers()**

```
void gdcM::Segment::SetPropertyTypeModifiers (
    BasicCodedEntryVector const & BSEV )
```

**10.268.5.28 SetSegmentAlgorithmName()**

```
void gdcM::Segment::SetSegmentAlgorithmName (
    const char * name )
```

**10.268.5.29 SetSegmentAlgorithmType() [1/2]**

```
void gdcM::Segment::SetSegmentAlgorithmType (
    ALGOType type )
```

**10.268.5.30 SetSegmentAlgorithmType() [2/2]**

```
void gdcM::Segment::SetSegmentAlgorithmType (
    const char * typeStr )
```

**10.268.5.31 SetSegmentDescription()**

```
void gdcM::Segment::SetSegmentDescription (
    const char * description )
```

### 10.268.5.32 SetSegmentLabel()

```
void gdcm::Segment::SetSegmentLabel (
    const char * label )
```

### 10.268.5.33 SetSegmentNumber()

```
void gdcm::Segment::SetSegmentNumber (
    const unsigned short num )
```

### 10.268.5.34 SetSurfaceCount()

```
void gdcm::Segment::SetSurfaceCount (
    const unsigned long nb )
```

## 10.268.6 Member Data Documentation

### 10.268.6.1 AnatomicRegion

```
SegmentHelper::BasicCodedEntry gdcm::Segment::AnatomicRegion [protected]
```

### 10.268.6.2 AnatomicRegionModifiers

```
BasicCodedEntryVector gdcm::Segment::AnatomicRegionModifiers [protected]
```

### 10.268.6.3 PropertyCategory

```
SegmentHelper::BasicCodedEntry gdcm::Segment::PropertyCategory [protected]
```

### 10.268.6.4 PropertyType

```
SegmentHelper::BasicCodedEntry gdcm::Segment::PropertyType [protected]
```

### 10.268.6.5 PropertyTypeModifiers

```
BasicCodedEntryVector gdcm::Segment::PropertyTypeModifiers [protected]
```

### 10.268.6.6 SegmentAlgorithmName

```
std::string gdcm::Segment::SegmentAlgorithmName [protected]
```

#### 10.268.6.7 SegmentAlgorithmType

`ALGOType` `gdcm::Segment::SegmentAlgorithmType` [protected]

#### 10.268.6.8 SegmentDescription

`std::string` `gdcm::Segment::SegmentDescription` [protected]

#### 10.268.6.9 SegmentLabel

`std::string` `gdcm::Segment::SegmentLabel` [protected]

#### 10.268.6.10 SegmentNumber

`unsigned short` `gdcm::Segment::SegmentNumber` [protected]

#### 10.268.6.11 SurfaceCount

`unsigned long` `gdcm::Segment::SurfaceCount` [protected]

#### 10.268.6.12 Surfaces

`SurfaceVector` `gdcm::Segment::Surfaces` [protected]

The documentation for this class was generated from the following file:

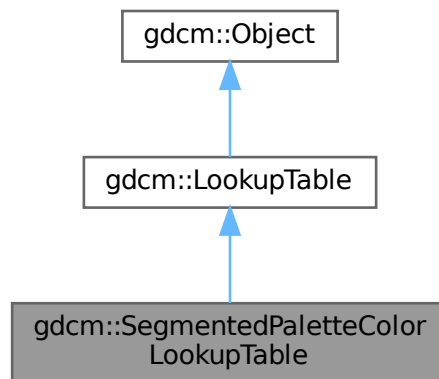
- [gdcmSegment.h](#)

## 10.269 gdcm::SegmentedPaletteColorLookupTable Class Reference

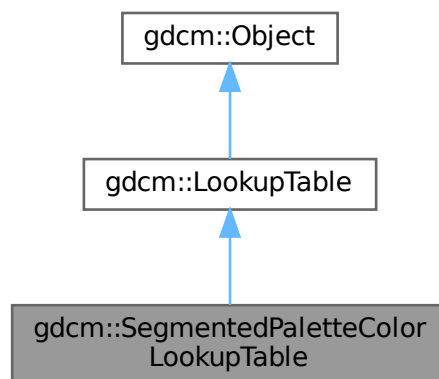
[SegmentedPaletteColorLookupTable](#) class.

```
#include <gdcmSegmentedPaletteColorLookupTable.h>
```

Inheritance diagram for gdcm::SegmentedPaletteColorLookupTable:



Collaboration diagram for gdcm::SegmentedPaletteColorLookupTable:



## Public Member Functions

- [SegmentedPaletteColorLookupTable](#) ()
- [~SegmentedPaletteColorLookupTable](#) () override
- void [Print](#) (std::ostream &) const override
- void [SetLUT](#) ([LookupTableType](#) type, const unsigned char \*array, unsigned int length) override

*Initialize a [SegmentedPaletteColorLookupTable](#).*

## Public Member Functions inherited from [gdcm::LookupTable](#)

- [LookupTable](#) ()
- [LookupTable](#) ([LookupTable](#) const &lut)
- [~LookupTable](#) () override
- void [Allocate](#) (unsigned short bitsample=8)  
*Allocate the LUT.*
- void [Clear](#) ()  
*Clear the LUT.*
- bool [Decode](#) (char \*outputbuffer, size\_t outlen, const char \*inputbuffer, size\_t inlen) const
- void [Decode](#) (std::istream &is, std::ostream &os) const  
*Decode the LUT.*
- bool [Decode8](#) (char \*outputbuffer, size\_t outlen, const char \*inputbuffer, size\_t inlen) const  
*Decode into RGB 8 bits space.*
- unsigned short [GetBitSample](#) () const  
*return the bit sample*
- bool [GetBufferAsRGBA](#) (unsigned char \*rgba) const  
*return the LUT as RGBA buffer*
- void [GetLUT](#) ([LookupTableType](#) type, unsigned char \*array, unsigned int &length) const
- void [GetLUTDescriptor](#) ([LookupTableType](#) type, unsigned short &length, unsigned short &subscript, unsigned short &bitsize) const
- unsigned int [GetLUTLength](#) ([LookupTableType](#) type) const
- const unsigned char \* [GetPointer](#) () const  
*return a raw pointer to the LUT*
- void [InitializeBlueLUT](#) (unsigned short length, unsigned short subscript, unsigned short bitsize)
- bool [Initialized](#) () const  
*return whether the LUT has been initialized*
- void [InitializeGreenLUT](#) (unsigned short length, unsigned short subscript, unsigned short bitsize)
- void [InitializeLUT](#) ([LookupTableType](#) type, unsigned short length, unsigned short subscript, unsigned short bitsize)  
*Generic interface:*
- void [InitializeRedLUT](#) (unsigned short length, unsigned short subscript, unsigned short bitsize)  
*RED / GREEN / BLUE specific:*
- bool [IsRGB8](#) () const  
*Return whether 16 bits LUT is in RGB 8 bits space.*
- void [SetBlueLUT](#) (const unsigned char \*blue, unsigned int length)
- void [SetGreenLUT](#) (const unsigned char \*green, unsigned int length)
- void [SetRedLUT](#) (const unsigned char \*red, unsigned int length)
- bool [WriteBufferAsRGBA](#) (const unsigned char \*rgba)  
*Write the LUT as RGBA.*

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

## Additional Inherited Members

## Public Types inherited from [gdcm::LookupTable](#)

- enum [LookupTableType](#) {  
    [RED](#) = 0 ,  
    [GREEN](#) ,  
    [BLUE](#) ,  
    [GRAY](#) ,  
    [UNKNOWN](#) }

## Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

## Protected Attributes inherited from [gdcm::LookupTable](#)

- unsigned short [BitSample](#)
- bool [IncompleteLUT](#):1
- [LookupTableInternal](#) \* [Internal](#)

### 10.269.1 Detailed Description

[SegmentedPaletteColorLookupTable](#) class.

### 10.269.2 Constructor & Destructor Documentation

#### 10.269.2.1 [SegmentedPaletteColorLookupTable](#)()

```
gdcm::SegmentedPaletteColorLookupTable::SegmentedPaletteColorLookupTable ( )
```

#### 10.269.2.2 [~SegmentedPaletteColorLookupTable](#)()

```
gdcm::SegmentedPaletteColorLookupTable::~~SegmentedPaletteColorLookupTable ( ) [override]
```

### 10.269.3 Member Function Documentation

#### 10.269.3.1 Print()

```
void gdcm::SegmentedPaletteColorLookupTable::Print (  
    std::ostream & ) const [inline], [override], [virtual]
```

Reimplemented from [gdcm::LookupTable](#).

#### 10.269.3.2 SetLUT()

```
void gdcm::SegmentedPaletteColorLookupTable::SetLUT (  
    LookupTableType type,  
    const unsigned char * array,  
    unsigned int length ) [override], [virtual]
```

Initialize a [SegmentedPaletteColorLookupTable](#).

Reimplemented from [gdcm::LookupTable](#).

The documentation for this class was generated from the following file:

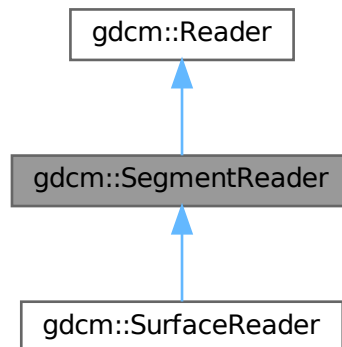
- [gdcmSegmentedPaletteColorLookupTable.h](#)

## 10.270 gdcm::SegmentReader Class Reference

This class defines a segment reader.

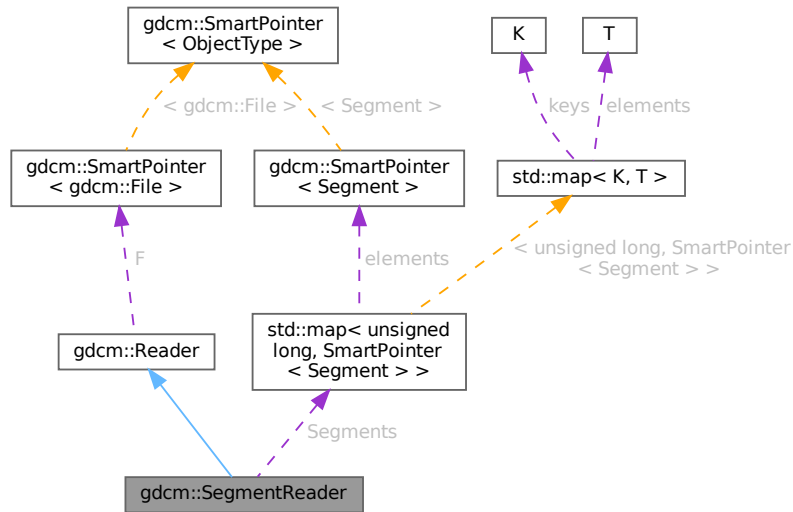
```
#include <gdcmSegmentReader.h>
```

Inheritance diagram for gdcm::SegmentReader:





Collaboration diagram for gdcm::SegmentReader:



## Public Types

- typedef std::vector< [SmartPointer< Segment >](#) > [SegmentVector](#)

## Public Member Functions

- [SegmentReader](#) ()
- [~SegmentReader](#) () override
- [SegmentVector](#) [GetSegments](#) ()
- [SegmentVector](#) [GetSegments](#) () const
- bool [Read](#) () override

*Read.*

## Public Member Functions inherited from [gdcm::Reader](#)

- [Reader](#) ()
- virtual [~Reader](#) ()
- bool [CanRead](#) () const
- [File](#) & [GetFile](#) ()  
*Set/Get File.*
- const [File](#) & [GetFile](#) () const  
*Set/Get File.*
- size\_t [GetStreamCurrentPosition](#) () const
- bool [ReadSelectedPrivateTags](#) (std::set< [PrivateTag](#) > const &ptags, bool readvalues=true)

*Will only read the specified selected private tags.*

- bool [ReadSelectedTags](#) (std::set< [Tag](#) > const &tags, bool readvalues=true)

*Will only read the specified selected tags.*

- bool [ReadUpToTag](#) (const [Tag](#) &tag, std::set< [Tag](#) > const &skiptags=std::set< [Tag](#) >())
- void [SetFile](#) ([File](#) &file)

*Set/Get [File](#).*

- void [SetFileName](#) (const char \*filename\_native)
- void [SetStream](#) (std::istream &input\_stream)

*Set the open-ed stream directly.*

## Protected Types

- typedef std::map< unsigned long, [SmartPointer](#)< [Segment](#) > > [SegmentMap](#)

## Protected Member Functions

- bool [ReadSegment](#) (const [Item](#) &segmentItem, const unsigned int idx)
- bool [ReadSegments](#) ()

## Protected Member Functions inherited from [gdcm::Reader](#)

- std::istream \* [GetStreamPtr](#) () const
- bool [ReadDataSet](#) ()
- bool [ReadMetaInformation](#) ()
- bool [ReadPreamble](#) ()

## Protected Attributes

- [SegmentMap](#) [Segments](#)

## Protected Attributes inherited from [gdcm::Reader](#)

- [SmartPointer](#)< [File](#) > [F](#)

## 10.270.1 Detailed Description

This class defines a segment reader.

It reads attributes of group 0x0062.

See also

PS 3.3 C.8.20.2 and C.8.23

## 10.270.2 Member Typedef Documentation

### 10.270.2.1 SegmentMap

```
typedef std::map< unsigned long, SmartPointer< Segment > > gdcm::SegmentReader::SegmentMap [protected]
```

### 10.270.2.2 SegmentVector

```
typedef std::vector< SmartPointer< Segment > > gdcm::SegmentReader::SegmentVector
```

## 10.270.3 Constructor & Destructor Documentation

### 10.270.3.1 SegmentReader()

```
gdcm::SegmentReader::SegmentReader ( )
```

### 10.270.3.2 ~SegmentReader()

```
gdcm::SegmentReader::~~SegmentReader ( ) [override]
```

## 10.270.4 Member Function Documentation

### 10.270.4.1 GetSegments() [1/2]

```
SegmentVector gdcm::SegmentReader::GetSegments ( )
```

### 10.270.4.2 GetSegments() [2/2]

```
SegmentVector gdcm::SegmentReader::GetSegments ( ) const
```

### 10.270.4.3 Read()

```
bool gdcm::SegmentReader::Read ( ) [override], [virtual]
```

Read.

Reimplemented from [gdcm::Reader](#).

Reimplemented in [gdcm::SurfaceReader](#).

#### 10.270.4.4 ReadSegment()

```
bool gdcm::SegmentReader::ReadSegment (
    const Item & segmentItem,
    const unsigned int idx ) [protected]
```

#### 10.270.4.5 ReadSegments()

```
bool gdcm::SegmentReader::ReadSegments ( ) [protected]
```

### 10.270.5 Member Data Documentation

#### 10.270.5.1 Segments

```
SegmentMap gdcm::SegmentReader::Segments [protected]
```

The documentation for this class was generated from the following file:

- [gdcmSegmentReader.h](#)

## 10.271 gdcm::SegmentWriter Class Reference

This class defines a segment writer.

```
#include <gdcmSegmentWriter.h>
```

Inheritance diagram for gdcm::SegmentWriter:



Collaboration diagram for gdcm::SegmentWriter:



## Public Types

- typedef std::vector< [SmartPointer< Segment >](#) > [SegmentVector](#)

## Public Member Functions

- [SegmentWriter](#) ()
- [~SegmentWriter](#) () override
- void [AddSegment](#) ([SmartPointer< Segment >](#) segment)
- unsigned int [GetNumberOfSegments](#) () const
- [SmartPointer< Segment >](#) [GetSegment](#) (const unsigned int idx=0) const
- [SegmentVector](#) & [GetSegments](#) ()
- const [SegmentVector](#) & [GetSegments](#) () const
- void [SetNumberOfSegments](#) (const unsigned int size)
- void [SetSegments](#) ([SegmentVector](#) &segments)
- bool [Write](#) () override

*Write.*

## Public Member Functions inherited from [gdcm::Writer](#)

- [Writer](#) ()
- virtual [~Writer](#) ()
- void [CheckFileMetaInformationOff](#) ()
- void [CheckFileMetaInformationOn](#) ()
- [File](#) & [GetFile](#) ()
- void [SetCheckFileMetaInformation](#) (bool b)  
*Undocumented function, do not use (= leave default)*
- void [SetFile](#) (const [File](#) &f)  
*Set/Get the DICOM file (DataSet + Header)*
- void [SetFileName](#) (const char \*filename\_native)  
*Set the filename of DICOM file to write:*
- void [SetStream](#) (std::ostream &output\_stream)  
*Set user ostream buffer.*

## Protected Member Functions

- bool [PrepareWrite](#) ()

## Protected Member Functions inherited from [gdcm::Writer](#)

- bool [GetCheckFileMetaInformation](#) () const
- std::ostream \* [GetStreamPtr](#) () const
- void [SetWriteDataSetOnly](#) (bool b)

## Protected Attributes

- [SegmentVector](#) [Segments](#)

## Protected Attributes inherited from [gdcm::Writer](#)

- std::ofstream \* [Ofstream](#)
- std::ostream \* [Stream](#)

### 10.271.1 Detailed Description

This class defines a segment writer.

It writes attributes of group 0x0062.

See also

PS 3.3 C.8.20.2 and C.8.23

### 10.271.2 Member Typedef Documentation

#### 10.271.2.1 SegmentVector

```
typedef std::vector< SmartPointer< Segment > > gdcm::SegmentWriter::SegmentVector
```

### 10.271.3 Constructor & Destructor Documentation

#### 10.271.3.1 SegmentWriter()

```
gdcm::SegmentWriter::SegmentWriter ( )
```

#### 10.271.3.2 ~SegmentWriter()

```
gdcm::SegmentWriter::~~SegmentWriter ( ) [override]
```

## 10.271.4 Member Function Documentation

### 10.271.4.1 AddSegment()

```
void gdcm::SegmentWriter::AddSegment (
    SmartPointer< Segment > segment )
```

### 10.271.4.2 GetNumberOfSegments()

```
unsigned int gdcm::SegmentWriter::GetNumberOfSegments ( ) const
```

### 10.271.4.3 GetSegment()

```
SmartPointer< Segment > gdcm::SegmentWriter::GetSegment (
    const unsigned int idx = 0 ) const
```

### 10.271.4.4 GetSegments() [1/2]

```
SegmentVector & gdcm::SegmentWriter::GetSegments ( )
```

### 10.271.4.5 GetSegments() [2/2]

```
const SegmentVector & gdcm::SegmentWriter::GetSegments ( ) const
```

### 10.271.4.6 PrepareWrite()

```
bool gdcm::SegmentWriter::PrepareWrite ( ) [protected]
```

### 10.271.4.7 SetNumberOfSegments()

```
void gdcm::SegmentWriter::SetNumberOfSegments (
    const unsigned int size )
```

### 10.271.4.8 SetSegments()

```
void gdcm::SegmentWriter::SetSegments (
    SegmentVector & segments )
```

#### 10.271.4.9 Write()

```
bool gdcM::SegmentWriter::Write ( ) [override], [virtual]
```

Write.

Reimplemented from [gdcM::Writer](#).

Reimplemented in [gdcM::SurfaceWriter](#).

### 10.271.5 Member Data Documentation

#### 10.271.5.1 Segments

```
SegmentVector gdcM::SegmentWriter::Segments [protected]
```

The documentation for this class was generated from the following file:

- [gdcMSegmentWriter.h](#)

## 10.272 gdcM::SequenceOfFragments Class Reference

Class to represent a Sequence Of Fragments.

```
#include <gdcMSequenceOfFragments.h>
```

Inheritance diagram for gdcM::SequenceOfFragments:





Collaboration diagram for gdcm::SequenceOfFragments:



## Public Types

- typedef `FragmentVector::const_iterator` [ConstIterator](#)
- typedef `std::vector< Fragment >` [FragmentVector](#)
- typedef `FragmentVector::iterator` [Iterator](#)
- typedef `FragmentVector::size_type` [SizeType](#)

## Public Member Functions

- [SequenceOfFragments](#) ()  
*constructor (UndefinedLength by default)*
- void [AddFragment](#) ([Fragment](#) const &item)  
*Appends a [Fragment](#) to the already added ones.*
- [Iterator](#) [Begin](#) ()
- [ConstIterator](#) [Begin](#) () const
- void [Clear](#) () override  
*Clear.*
- unsigned long [ComputeByteLength](#) () const
- [VL](#) [ComputeLength](#) () const
- [Iterator](#) [End](#) ()
- [ConstIterator](#) [End](#) () const
- bool [GetBuffer](#) (char \*buffer, unsigned long length) const
- bool [GetFragBuffer](#) (unsigned int fragNb, char \*buffer, unsigned long &length) const
- const [Fragment](#) & [GetFragment](#) ([SizeType](#) num) const
- [VL](#) [GetLength](#) () const override  
*Returns the SQ length, as read from disk.*
- [SizeType](#) [GetNumberOfFragments](#) () const
- [BasicOffsetTable](#) & [GetTable](#) ()

- const [BasicOffsetTable](#) & [GetTable](#) () const
- bool [operator==](#) (const [Value](#) &val) const override
- void [Print](#) (std::ostream &os) const override
- template<typename TSwap >  
std::istream & [Read](#) (std::istream &is, bool readvalues=true)
- template<typename TSwap >  
std::istream & [ReadPreValue](#) (std::istream &is)
- template<typename TSwap >  
std::istream & [ReadValue](#) (std::istream &is, bool)
- void [SetLength](#) (VL length) override  
*Sets the actual SQ length.*
- template<typename TSwap >  
std::ostream const & [Write](#) (std::ostream &os) const
- bool [WriteBuffer](#) (std::ostream &os) const

### Public Member Functions inherited from [gdcm::Value](#)

- [Value](#) ()=default
- [~Value](#) () override=default

### Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

### Static Public Member Functions

- static [SmartPointer](#)< [SequenceOfFragments](#) > [New](#) ()

### Additional Inherited Members

### Protected Member Functions inherited from [gdcm::Value](#)

- virtual void [SetLengthOnly](#) (VL l)

### Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

## 10.272.1 Detailed Description

Class to represent a Sequence Of Fragments.

**Todo** I do not enforce that Sequence of Fragments ends with a SQ end del

### Examples

[DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetJPEGSamplePrecision.cxx](#), and [MpegVideoInfo.cs](#).

## 10.272.2 Member Typedef Documentation

### 10.272.2.1 ConstIterator

```
typedef FragmentVector::const_iterator gdcm::SequenceOfFragments::ConstIterator
```

### 10.272.2.2 FragmentVector

```
typedef std::vector<Fragment> gdcm::SequenceOfFragments::FragmentVector
```

### 10.272.2.3 Iterator

```
typedef FragmentVector::iterator gdcm::SequenceOfFragments::Iterator
```

### 10.272.2.4 SizeType

```
typedef FragmentVector::size_type gdcm::SequenceOfFragments::SizeType
```

## 10.272.3 Constructor & Destructor Documentation

### 10.272.3.1 SequenceOfFragments()

```
gdcm::SequenceOfFragments::SequenceOfFragments ( ) [inline]
```

constructor (UndefinedLength by default)

## 10.272.4 Member Function Documentation

### 10.272.4.1 AddFragment()

```
void gdc::SequenceOfFragments::AddFragment (
    Fragment const & item )
```

Appends a [Fragment](#) to the already added ones.

### 10.272.4.2 Begin() [1/2]

```
Iterator gdc::SequenceOfFragments::Begin ( ) [inline]
```

### 10.272.4.3 Begin() [2/2]

```
ConstIterator gdc::SequenceOfFragments::Begin ( ) const [inline]
```

### 10.272.4.4 Clear()

```
void gdc::SequenceOfFragments::Clear ( ) [override], [virtual]
```

Clear.

Implements [gdc::Value](#).

### 10.272.4.5 ComputeByteLength()

```
unsigned long gdc::SequenceOfFragments::ComputeByteLength ( ) const
```

### 10.272.4.6 ComputeLength()

```
VL gdc::SequenceOfFragments::ComputeLength ( ) const
```

### 10.272.4.7 End() [1/2]

```
Iterator gdc::SequenceOfFragments::End ( ) [inline]
```

### 10.272.4.8 End() [2/2]

```
ConstIterator gdc::SequenceOfFragments::End ( ) const [inline]
```

#### 10.272.4.9 GetBuffer()

```
bool gdcm::SequenceOfFragments::GetBuffer (
    char * buffer,
    unsigned long length ) const
```

#### 10.272.4.10 GetFragBuffer()

```
bool gdcm::SequenceOfFragments::GetFragBuffer (
    unsigned int fragNb,
    char * buffer,
    unsigned long & length ) const
```

#### 10.272.4.11 GetFragment()

```
const Fragment & gdcm::SequenceOfFragments::GetFragment (
    SizeType num ) const
```

#### Examples

[DecompressImage.cs](#), [FixBrokenJ2K.cxx](#), and [FixJAIBugJPEGLS.cxx](#).

#### 10.272.4.12 GetLength()

```
VL gdcm::SequenceOfFragments::GetLength ( ) const [inline], [override], [virtual]
```

Returns the SQ length, as read from disk.

Implements [gdcm::Value](#).

#### 10.272.4.13 GetNumberOfFragments()

```
SizeType gdcm::SequenceOfFragments::GetNumberOfFragments ( ) const
```

#### Examples

[FixJAIBugJPEGLS.cxx](#).

#### 10.272.4.14 GetTable() [1/2]

```
BasicOffsetTable & gdcm::SequenceOfFragments::GetTable ( ) [inline]
```

#### 10.272.4.15 GetTable() [2/2]

```
const BasicOffsetTable & gdcM::SequenceOfFragments::GetTable ( ) const [inline]
```

#### 10.272.4.16 New()

```
static SmartPointer< SequenceOfFragments > gdcM::SequenceOfFragments::New ( ) [inline], [static]
```

#### Examples

[DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), and [MpegVideoInfo.cs](#).

#### 10.272.4.17 operator==( )

```
bool gdcM::SequenceOfFragments::operator== (
    const Value & val ) const [inline], [override], [virtual]
```

Implements [gdcM::Value](#).

#### 10.272.4.18 Print()

```
void gdcM::SequenceOfFragments::Print (
    std::ostream & os ) const [inline], [override], [virtual]
```

Reimplemented from [gdcM::Object](#).

#### 10.272.4.19 Read()

```
template<typename TSwap >
std::istream & gdcM::SequenceOfFragments::Read (
    std::istream & is,
    bool readvalues = true ) [inline]
```

#### 10.272.4.20 ReadPreValue()

```
template<typename TSwap >
std::istream & gdcM::SequenceOfFragments::ReadPreValue (
    std::istream & is ) [inline]
```

References [gdcMDebugMacro](#).

#### 10.272.4.21 ReadValue()

```
template<typename TSwap >
std::istream & gdcm::SequenceOfFragments::ReadValue (
    std::istream & is,
    bool ) [inline]
```

References [gdcmAssertAlwaysMacro](#), [gdcmDebugMacro](#), [gdcmWarningMacro](#), [gdcm::Tag::GetElement\(\)](#), [gdcm::Tag::GetGroup\(\)](#), [gdcm::ByteValue::GetLength\(\)](#), [gdcm::ByteValue::GetPointer\(\)](#), [gdcm::DataElement::GetTag\(\)](#), [gdcm::DataElement::GetVL\(\)](#), [gdcm::Fragment::Read\(\)](#), [gdcm::Fragment::ReadBacktrack\(\)](#), and [gdcm::Exception::what\(\)](#).

#### 10.272.4.22 SetLength()

```
void gdcm::SequenceOfFragments::SetLength (
    VL length ) [inline], [override], [virtual]
```

Sets the actual SQ length.

Implements [gdcm::Value](#).

#### 10.272.4.23 Write()

```
template<typename TSwap >
std::ostream const & gdcm::SequenceOfFragments::Write (
    std::ostream & os ) const [inline]
```

References [gdcm::Tag::Write\(\)](#), and [gdcm::VL::Write\(\)](#).

#### 10.272.4.24 WriteBuffer()

```
bool gdcm::SequenceOfFragments::WriteBuffer (
    std::ostream & os ) const
```

#### Examples

[GetJPEGSamplePrecision.cxx](#).

The documentation for this class was generated from the following file:

- [gdcmSequenceOfFragments.h](#)

## 10.273 gdcm::SequenceOfItems Class Reference

Class to represent a Sequence Of Items.

```
#include <gdcmSequenceOfItems.h>
```

Inheritance diagram for gdcm::SequenceOfItems:



Collaboration diagram for gdcm::SequenceOfItems:





## Public Types

- typedef ItemVector::const\_iterator [ConstIterator](#)
- typedef std::vector< [Item](#) > [ItemVector](#)
- typedef ItemVector::iterator [Iterator](#)
- typedef ItemVector::size\_type [SizeType](#)

## Public Member Functions

- [SequenceOfItems](#) ()  
*constructor (UndefinedLength by default)*
- void [AddItem](#) ([Item](#) const &item)  
*Appends an [Item](#) to the already added ones.*
- [Item](#) & [AddNewUndefinedLengthItem](#) ()  
*Appends an [Item](#) to the already added ones.*
- [Iterator](#) [Begin](#) ()
- [ConstIterator](#) [Begin](#) () const
- void [Clear](#) () override  
*remove all items within the sequence*
- template<typename TDE >  
[VL ComputeLength](#) () const
- [Iterator](#) [End](#) ()
- [ConstIterator](#) [End](#) () const
- bool [FindDataElement](#) (const [Tag](#) &t) const
- [Item](#) & [GetItem](#) ([SizeType](#) position)
- const [Item](#) & [GetItem](#) ([SizeType](#) position) const
- [VL GetLength](#) () const override  
*Returns the SQ length, as read from disk.*
- [SizeType](#) [GetNumberOfItems](#) () const
- bool [IsEmpty](#) () const
- bool [IsUndefinedLength](#) () const  
*return if [Value](#) Length if of undefined length*
- [SequenceOfItems](#) & [operator=](#) (const [SequenceOfItems](#) &val)
- bool [operator==](#) (const [Value](#) &val) const override
- void [Print](#) (std::ostream &os) const override
- template<typename TDE , typename TSwap >  
std::istream & [Read](#) (std::istream &is, bool readvalues=true)
- bool [RemoveItemByIndex](#) (const [SizeType](#) index)
- void [SetLength](#) ([VL](#) length) override  
*Sets the actual SQ length.*
- void [SetLengthToUndefined](#) ()  
*Properly set the Sequence of [Item](#) to be undefined length.*
- void [SetNumberOfItems](#) ([SizeType](#) n)
- template<typename TDE , typename TSwap >  
std::ostream const & [Write](#) (std::ostream &os) const

## Public Member Functions inherited from [gdcm::Value](#)

- [Value](#) ()=default
- [~Value](#) () override=default

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

## Static Public Member Functions

- static [SmartPointer](#)< [SequenceOfItems](#) > [New](#) ()

## Public Attributes

- [ItemVector](#) [Items](#)  
*Vector of Sequence Items.*
- [VL](#) [SequenceLengthField](#)  
*Total length of the Sequence (or 0xffffffff) if undefined.*

## Additional Inherited Members

## Protected Member Functions inherited from [gdcm::Value](#)

- virtual void [SetLengthOnly](#) ([VL](#) l)

## Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

## 10.273.1 Detailed Description

Class to represent a Sequence Of Items.

(value representation : SQ)

- a [Value](#) Representation for Data Elements that contains a sequence of Data Sets.
- Sequence of [Item](#) allows for Nested Data Sets

See PS 3.5, 7.4.6 Data [Element Type](#) Within a Sequence

### Note

SEQUENCE OF ITEMS (VALUE REPRESENTATION SQ) A [Value](#) Representation for Data Elements that contain a sequence of Data Sets. Sequence of Items allows for Nested Data Sets.

### Examples

[DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GenAllIVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), and [ReadExplicitLengthSQIVR.cxx](#).

## 10.273.2 Member Typedef Documentation

### 10.273.2.1 ConstIterator

```
typedef ItemVector::const_iterator gdcm::SequenceOfItems::ConstIterator
```

### 10.273.2.2 ItemVector

```
typedef std::vector< Item > gdcm::SequenceOfItems::ItemVector
```

### 10.273.2.3 Iterator

```
typedef ItemVector::iterator gdcm::SequenceOfItems::Iterator
```

### 10.273.2.4 SizeType

```
typedef ItemVector::size_type gdcm::SequenceOfItems::SizeType
```

## 10.273.3 Constructor & Destructor Documentation

### 10.273.3.1 SequenceOfItems()

```
gdcm::SequenceOfItems::SequenceOfItems ( ) [inline]
```

constructor (UndefinedLength by default)

## 10.273.4 Member Function Documentation

### 10.273.4.1 AddItem()

```
void gdcm::SequenceOfItems::AddItem (
    Item const & item )
```

Appends an [Item](#) to the already added ones.

#### Examples

[Extracting\\_All\\_Resolution.cxx](#).

#### 10.273.4.2 AddNewUndefinedLengthItem()

```
Item & gdcM::SequenceOfItems::AddNewUndefinedLengthItem ( )
```

Appends an [Item](#) to the already added ones.

#### 10.273.4.3 Begin() [1/2]

```
Iterator gdcM::SequenceOfItems::Begin ( ) [inline]
```

#### 10.273.4.4 Begin() [2/2]

```
ConstIterator gdcM::SequenceOfItems::Begin ( ) const [inline]
```

#### 10.273.4.5 Clear()

```
void gdcM::SequenceOfItems::Clear ( ) [override], [virtual]
```

remove all items within the sequence

Implements [gdcM::Value](#).

#### 10.273.4.6 ComputeLength()

```
template<typename TDE >  
VL gdcM::SequenceOfItems::ComputeLength ( ) const
```

#### 10.273.4.7 End() [1/2]

```
Iterator gdcM::SequenceOfItems::End ( ) [inline]
```

#### 10.273.4.8 End() [2/2]

```
ConstIterator gdcM::SequenceOfItems::End ( ) const [inline]
```

#### 10.273.4.9 FindDataElement()

```
bool gdcM::SequenceOfItems::FindDataElement (   
    const Tag & t ) const
```

#### 10.273.4.10 GetItem() [1/2]

```
Item & gdcm::SequenceOfItems::GetItem (
    SizeType position )
```

#### 10.273.4.11 GetItem() [2/2]

```
const Item & gdcm::SequenceOfItems::GetItem (
    SizeType position ) const
```

#### Examples

[DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [ExtractEncryptedContent.cxx](#), and [GetSequenceUltrasound.cxx](#).

#### 10.273.4.12 GetLength()

```
VL gdcm::SequenceOfItems::GetLength ( ) const [inline], [override], [virtual]
```

Returns the SQ length, as read from disk.

Implements [gdcm::Value](#).

#### 10.273.4.13 GetNumberOfItems()

```
SizeType gdcm::SequenceOfItems::GetNumberOfItems ( ) const [inline]
```

#### Examples

[DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [ExtractEncryptedContent.cxx](#), and [GetSequenceUltrasound.cxx](#).

#### 10.273.4.14 IsEmpty()

```
bool gdcm::SequenceOfItems::IsEmpty ( ) const [inline]
```

#### 10.273.4.15 IsUndefinedLength()

```
bool gdcm::SequenceOfItems::IsUndefinedLength ( ) const [inline]
```

return if [Value](#) Length if of undefined length

#### 10.273.4.16 New()

```
static SmartPointer< SequenceOfItems > gdcM::SequenceOfItems::New ( ) [inline], [static]
```

##### Examples

[NewSequence.cs](#).

#### 10.273.4.17 operator=()

```
SequenceOfItems & gdcM::SequenceOfItems::operator= (
    const SequenceOfItems & val ) [inline]
```

References [Items](#), and [SequenceLengthField](#).

#### 10.273.4.18 operator==( )

```
bool gdcM::SequenceOfItems::operator== (
    const Value & val ) const [inline], [override], [virtual]
```

Implements [gdcM::Value](#).

References [Items](#), and [SequenceLengthField](#).

#### 10.273.4.19 Print()

```
void gdcM::SequenceOfItems::Print (
    std::ostream & os ) const [inline], [override], [virtual]
```

Reimplemented from [gdcM::Object](#).

#### 10.273.4.20 Read()

```
template<typename TDE , typename TSwap >
std::istream & gdcM::SequenceOfItems::Read (
    std::istream & is,
    bool readvalues = true ) [inline]
```

References [gdcM::Item::Clear\(\)](#), [gdcMDebugMacro](#), [gdcMWarningMacro](#), [gdcM::Exception::GetDescription\(\)](#), [gdcM::Item::GetNestedDataSet\(\)](#), [gdcM::DataElement::GetTag\(\)](#), [gdcM::DataElement::GetVL\(\)](#), [gdcM::Item::Read\(\)](#), and [gdcM::DataSet::Size\(\)](#).

#### 10.273.4.21 RemoveItemByIndex()

```
bool gdcm::SequenceOfItems::RemoveItemByIndex (
    const SizeType index )
```

Remove an [Item](#) as specified by its index, if index > size, false is returned Index starts at 1 not 0

#### 10.273.4.22 SetLength()

```
void gdcm::SequenceOfItems::SetLength (
    VL length ) [inline], [override], [virtual]
```

Sets the actual SQ length.

Implements [gdcm::Value](#).

#### 10.273.4.23 SetLengthToUndefined()

```
void gdcm::SequenceOfItems::SetLengthToUndefined ( )
```

Properly set the Sequence of [Item](#) to be undefined length.

#### 10.273.4.24 SetNumberOfItems()

```
void gdcm::SequenceOfItems::SetNumberOfItems (
    SizeType n ) [inline]
```

#### 10.273.4.25 Write()

```
template<typename TDE , typename TSwap >
std::ostream const & gdcm::SequenceOfItems::Write (
    std::ostream & os ) const [inline]
```

References [gdcm::Tag::Write\(\)](#), and [gdcm::VL::Write\(\)](#).

### 10.273.5 Member Data Documentation

#### 10.273.5.1 Items

```
ItemVector gdcm::SequenceOfItems::Items
```

Vector of Sequence Items.

Referenced by [operator=\(\)](#), and [operator==\(\)](#).

### 10.273.5.2 SequenceLengthField

[VL](#) `gdcm::SequenceOfItems::SequenceLengthField`

Total length of the Sequence (or 0xffffffff if undefined).

Referenced by [operator=\(\)](#), and [operator==\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmSequenceOfItems.h](#)

## 10.274 gdcm::SerieHelper Class Reference

[SerieHelper](#) DO NOT USE this class, it is only a temporary solution for ITK migration from GDCM 1.x to GDCM 2.x It will disappear soon, you've been warned.

```
#include <gdcmSerieHelper.h>
```

Collaboration diagram for `gdcm::SerieHelper`:





## Public Member Functions

- [SerieHelper](#) ()
- [~SerieHelper](#) ()
- void [AddRestriction](#) (const std::string &tag)
- void [AddRestriction](#) (uint16\_t group, uint16\_t [elem](#), std::string const &[value](#), int [op](#))
- void [Clear](#) ()
- void [CreateDefaultUniqueSeriesIdentifier](#) ()
- std::string [CreateUniqueSeriesIdentifier](#) ([File](#) \*inFile)
- [FileList](#) \* [GetFirstSingleSerieUIDFileSet](#) ()
- [FileList](#) \* [GetNextSingleSerieUIDFileSet](#) ()
- void [OrderFileList](#) ([FileList](#) \*fileSet)
- void [SetDirectory](#) (std::string const &dir, bool recursive=false)
- void [SetLoadMode](#) (int)
- void [SetUseSeriesDetails](#) (bool useSeriesDetails)

## Protected Types

- using [Rule](#) = RuleStructure{ uint16\_t group
- typedef std::vector< [Rule](#) > [SerieRestrictions](#)
- typedef std::map< std::string, [FileList](#) \* > [SingleSerieUIDFileSetmap](#)

## Protected Member Functions

- bool [AddFile](#) ([FileWithName](#) &header)
- void [AddFileName](#) (std::string const &filename)
- void [AddRestriction](#) (const [Tag](#) &tag)
- bool [FileNameOrdering](#) ([FileList](#) \*fileList)
- bool [ImageNumberOrdering](#) ([FileList](#) \*fileList)
- bool [ImagePositionPatientOrdering](#) ([FileList](#) \*fileSet)
- bool [UserOrdering](#) ([FileList](#) \*fileSet)

## Protected Attributes

- uint16\_t [elem](#)
- [SingleSerieUIDFileSetmap::iterator](#) [ItFileSetHt](#)
- int [op](#)
- [SingleSerieUIDFileSetmap](#) [SingleSerieUIDFileSetHT](#)
- std::string [value](#)

## 10.274.1 Detailed Description

[SerieHelper](#) DO NOT USE this class, it is only a temporary solution for ITK migration from GDCM 1.x to GDCM 2.x It will disappear soon, you've been warned.

Instead see [ImageHelper](#) or [IPPSorter](#)

## 10.274.2 Member Typedef Documentation

### 10.274.2.1 Rule

```
using gdcm::SerieHelper::Rule = RuleStructure{ uint16_t group [protected]
```

### 10.274.2.2 SerieRestrictions

```
typedef std::vector<Rule> gdcm::SerieHelper::SerieRestrictions [protected]
```

### 10.274.2.3 SingleSerieUIDFileSetmap

```
typedef std::map<std::string, FileList *> gdcm::SerieHelper::SingleSerieUIDFileSetmap [protected]
```

## 10.274.3 Constructor & Destructor Documentation

### 10.274.3.1 SerieHelper()

```
gdcm::SerieHelper::SerieHelper ( )
```

### 10.274.3.2 ~SerieHelper()

```
gdcm::SerieHelper::~~SerieHelper ( )
```

## 10.274.4 Member Function Documentation

### 10.274.4.1 AddFile()

```
bool gdcm::SerieHelper::AddFile (
    FileWithName & header ) [protected]
```

### 10.274.4.2 AddFileName()

```
void gdcm::SerieHelper::AddFileName (
    std::string const & filename ) [protected]
```

### 10.274.4.3 AddRestriction() [1/3]

```
void gdcm::SerieHelper::AddRestriction (
    const std::string & tag )
```

#### 10.274.4.4 AddRestriction() [2/3]

```
void gdcm::SerieHelper::AddRestriction (
    const Tag & tag ) [protected]
```

#### 10.274.4.5 AddRestriction() [3/3]

```
void gdcm::SerieHelper::AddRestriction (
    uint16_t group,
    uint16_t elem,
    std::string const & value,
    int op )
```

#### 10.274.4.6 Clear()

```
void gdcm::SerieHelper::Clear ( )
```

#### 10.274.4.7 CreateDefaultUniqueSeriesIdentifier()

```
void gdcm::SerieHelper::CreateDefaultUniqueSeriesIdentifier ( )
```

#### 10.274.4.8 CreateUniqueSeriesIdentifier()

```
std::string gdcm::SerieHelper::CreateUniqueSeriesIdentifier (
    File * inFile )
```

#### 10.274.4.9 FileNameOrdering()

```
bool gdcm::SerieHelper::FileNameOrdering (
    FileList * fileList ) [protected]
```

#### 10.274.4.10 GetFirstSingleSerieUIDFileSet()

```
FileList * gdcm::SerieHelper::GetFirstSingleSerieUIDFileSet ( )
```

#### 10.274.4.11 GetNextSingleSerieUIDFileSet()

```
FileList * gdcm::SerieHelper::GetNextSingleSerieUIDFileSet ( )
```

#### 10.274.4.12 ImageNumberOrdering()

```
bool gdcm::SerieHelper::ImageNumberOrdering (
    FileList * fileList ) [protected]
```

#### 10.274.4.13 ImagePositionPatientOrdering()

```
bool gdcm::SerieHelper::ImagePositionPatientOrdering (
    FileList * fileSet ) [protected]
```

#### 10.274.4.14 OrderFileList()

```
void gdcm::SerieHelper::OrderFileList (
    FileList * fileSet )
```

#### 10.274.4.15 SetDirectory()

```
void gdcm::SerieHelper::SetDirectory (
    std::string const & dir,
    bool recursive = false )
```

#### 10.274.4.16 SetLoadMode()

```
void gdcm::SerieHelper::SetLoadMode (
    int ) [inline]
```

#### 10.274.4.17 SetUseSeriesDetails()

```
void gdcm::SerieHelper::SetUseSeriesDetails (
    bool useSeriesDetails )
```

#### 10.274.4.18 UserOrdering()

```
bool gdcm::SerieHelper::UserOrdering (
    FileList * fileSet ) [protected]
```

### 10.274.5 Member Data Documentation

#### 10.274.5.1 elem

```
uint16_t gdcm::SerieHelper::elem [protected]
```

### 10.274.5.2 ItFileSetHt

```
SingleSerieUIDFileSetmap::iterator gdcm::SerieHelper::ItFileSetHt [protected]
```

### 10.274.5.3 op

```
int gdcm::SerieHelper::op [protected]
```

### 10.274.5.4 SingleSerieUIDFileSetHT

```
SingleSerieUIDFileSetmap gdcm::SerieHelper::SingleSerieUIDFileSetHT [protected]
```

### 10.274.5.5 value

```
std::string gdcm::SerieHelper::value [protected]
```

The documentation for this class was generated from the following file:

- [gdcmSerieHelper.h](#)

## 10.275 gdcm::Series Class Reference

[Series.](#)

```
#include <gdcmSeries.h>
```

### Public Member Functions

- [Series](#) ()=default

### 10.275.1 Detailed Description

[Series.](#)

### 10.275.2 Constructor & Destructor Documentation

#### 10.275.2.1 Series()

```
gdcm::Series::Series ( ) [default]
```

The documentation for this class was generated from the following file:

- [gdcmSeries.h](#)

## 10.276 gdcm::network::ServiceClassApplicationInformation Class Reference

```
#include <gdcmServiceClassApplicationInformation.h>
```

### Public Member Functions

- [ServiceClassApplicationInformation](#) ()
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [SetTuple](#) (uint8\_t levelofsupport, uint8\_t levelofdigitalsig, uint8\_t elementcoercion)
- size\_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

### 10.276.1 Detailed Description

PS 3.4 [Table B.3-1](#) SERVICE-CLASS-APPLICATION-INFORMATION (A-ASSOCIATE-RQ)

### 10.276.2 Constructor & Destructor Documentation

#### 10.276.2.1 ServiceClassApplicationInformation()

```
gdcm::network::ServiceClassApplicationInformation::ServiceClassApplicationInformation ( )
```

### 10.276.3 Member Function Documentation

#### 10.276.3.1 Print()

```
void gdcm::network::ServiceClassApplicationInformation::Print (
    std::ostream & os ) const
```

#### 10.276.3.2 Read()

```
std::istream & gdcm::network::ServiceClassApplicationInformation::Read (
    std::istream & is )
```

#### 10.276.3.3 SetTuple()

```
void gdcm::network::ServiceClassApplicationInformation::SetTuple (
    uint8_t levelofsupport,
    uint8_t levelofdigitalsig,
    uint8_t elementcoercion )
```

#### 10.276.3.4 Size()

```
size_t gdcm::network::ServiceClassApplicationInformation::Size ( ) const
```

#### 10.276.3.5 Write()

```
const std::ostream & gdcm::network::ServiceClassApplicationInformation::Write (
    std::ostream & os ) const
```

The documentation for this class was generated from the following file:

- [gdcmServiceClassApplicationInformation.h](#)

## 10.277 gdcm::ServiceClassUser Class Reference

[ServiceClassUser](#).

```
#include <gdcmServiceClassUser.h>
```

Inheritance diagram for gdcm::ServiceClassUser:



Collaboration diagram for gdcm::ServiceClassUser:



## Public Member Functions

- [ServiceClassUser](#) ()
- [ServiceClassUser](#) (const [ServiceClassUser](#) &)=delete
- [~ServiceClassUser](#) () override
- const char \* [GetAETitle](#) () const
- const char \* [GetCalledAETitle](#) () const
- double [GetTimeout](#) () const
- bool [InitializeConnection](#) ()
- bool [IsPresentationContextAccepted](#) (const [PresentationContext](#) &pc) const  
*Return if the passed in presentation was accepted during association negotiation.*
- void [operator=](#) (const [ServiceClassUser](#) &)=delete
- bool [SendEcho](#) ()  
*C-ECHO.*
- bool [SendFind](#) (const [BaseRootQuery](#) \*query, std::vector< [DataSet](#) > &retDatasets)  
*C-FIND a query, return result are in retDatasets.*
- bool [SendMove](#) (const [BaseRootQuery](#) \*query, const char \*outputdir)  
*Execute a C-MOVE, based on query, return files are written in outputdir.*
- bool [SendMove](#) (const [BaseRootQuery](#) \*query, std::vector< [DataSet](#) > &retDatasets)  
*Execute a C-MOVE, based on query, returned dataset are Implicit.*
- bool [SendMove](#) (const [BaseRootQuery](#) \*query, std::vector< [File](#) > &retFile)  
*Execute a C-MOVE, based on query, returned Files are stored in vector.*
- bool [SendStore](#) (const char \*filename)  
*Execute a C-STORE on file on disk, named filename.*
- bool [SendStore](#) ([DataSet](#) const &ds)  
*Execute a C-STORE on a DataSet, the transfer syntax used will be Implicit.*
- bool [SendStore](#) ([File](#) const &file)
- void [SetAETitle](#) (const char \*aetitle)



- set calling ae title*
  - void [SetCalledAETitle](#) (const char \*aetitle)
- set called ae title*
  - void [SetHostname](#) (const char \*hostname)
- Set the name of the called hostname (hostname or IP address)*
  - void [SetPort](#) (uint16\_t port)
- Set port of remote host (called application)*
  - void [SetPortSCP](#) (uint16\_t portscp)
- Set the port for any incoming C-STORE-SCP operation (typically in a return of C-MOVE)*
  - void [SetPresentationContexts](#) (std::vector< [PresentationContext](#) > const &pcs)
- Set the Presentation Context used for the Association.*
  - void [SetTimeout](#) (double t)
- set/get Timeout*
  - bool [StartAssociation](#) ()
- Start the association. Need to call SetPresentationContexts before.*
  - bool [StopAssociation](#) ()
- Stop the running association.*

### Public Member Functions inherited from [gdcm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*) const
- [Command](#) \* [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

### Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
- Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

### Static Public Member Functions

- static [SmartPointer](#)< [ServiceClassUser](#) > [New](#) ()
- for wrapped language: instantiate a reference counted object*

## Additional Inherited Members

## Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

### 10.277.1 Detailed Description

[ServiceClassUser](#).

#### Examples

[CStoreQtProgress.cxx](#).

### 10.277.2 Constructor & Destructor Documentation

#### 10.277.2.1 [ServiceClassUser](#)() [1/2]

```
gdcm::ServiceClassUser::ServiceClassUser ( )
```

Construct a SCU with default:

- hostname = localhost
- port = 104

#### 10.277.2.2 [~ServiceClassUser](#)()

```
gdcm::ServiceClassUser::~~ServiceClassUser ( ) [override]
```

#### 10.277.2.3 [ServiceClassUser](#)() [2/2]

```
gdcm::ServiceClassUser::ServiceClassUser (
    const ServiceClassUser & ) [delete]
```

### 10.277.3 Member Function Documentation

#### 10.277.3.1 [GetAETitle](#)()

```
const char * gdcm::ServiceClassUser::GetAETitle ( ) const
```

### 10.277.3.2 GetCalledAETitle()

```
const char * gdcm::ServiceClassUser::GetCalledAETitle ( ) const
```

### 10.277.3.3 GetTimeout()

```
double gdcm::ServiceClassUser::GetTimeout ( ) const
```

### 10.277.3.4 InitializeConnection()

```
bool gdcm::ServiceClassUser::InitializeConnection ( )
```

Will try to connect This will setup the actual timeout used during the whole connection time. Need to call SetTimeout first

#### Examples

[CStoreQtProgress.cxx](#).

### 10.277.3.5 IsPresentationContextAccepted()

```
bool gdcm::ServiceClassUser::IsPresentationContextAccepted (
    const PresentationContext & pc ) const
```

Return if the passed in presentation was accepted during association negotiation.

### 10.277.3.6 New()

```
static SmartPointer< ServiceClassUser > gdcm::ServiceClassUser::New ( ) [inline], [static]
```

for wrapped language: instantiate a reference counted object

### 10.277.3.7 operator=()

```
void gdcm::ServiceClassUser::operator= (
    const ServiceClassUser & ) [delete]
```

### 10.277.3.8 SendEcho()

```
bool gdcm::ServiceClassUser::SendEcho ( )
```

C-ECHO.

**10.277.3.9 SendFind()**

```
bool gdcmm::ServiceClassUser::SendFind (
    const BaseRootQuery * query,
    std::vector< DataSet > & retDatasets )
```

C-FIND a query, return result are in retDatasets.

**10.277.3.10 SendMove() [1/3]**

```
bool gdcmm::ServiceClassUser::SendMove (
    const BaseRootQuery * query,
    const char * outputdir )
```

Execute a C-MOVE, based on query, return files are written in outputdir.

**10.277.3.11 SendMove() [2/3]**

```
bool gdcmm::ServiceClassUser::SendMove (
    const BaseRootQuery * query,
    std::vector< DataSet > & retDatasets )
```

Execute a C-MOVE, based on query, returned dataset are Implicit.

**10.277.3.12 SendMove() [3/3]**

```
bool gdcmm::ServiceClassUser::SendMove (
    const BaseRootQuery * query,
    std::vector< File > & retFile )
```

Execute a C-MOVE, based on query, returned Files are stored in vector.

**10.277.3.13 SendStore() [1/3]**

```
bool gdcmm::ServiceClassUser::SendStore (
    const char * filename )
```

Execute a C-STORE on file on disk, named filename.

**Examples**

[CStoreQtProgress.cxx](#).

**10.277.3.14 SendStore()** [2/3]

```
bool gdcm::ServiceClassUser::SendStore (
    DataSet const & ds )
```

Execute a C-STORE on a [DataSet](#), the transfer syntax used will be Implicit.

**10.277.3.15 SendStore()** [3/3]

```
bool gdcm::ServiceClassUser::SendStore (
    File const & file )
```

Execute a C-STORE on a [File](#), the transfer syntax used for the query is based on the file.

**10.277.3.16 SetAETitle()**

```
void gdcm::ServiceClassUser::SetAETitle (
    const char * aetitle )
```

set calling ae title

**10.277.3.17 SetCalledAETitle()**

```
void gdcm::ServiceClassUser::SetCalledAETitle (
    const char * aetitle )
```

set called ae title

**Examples**

[CStoreQtProgress.cxx](#).

**10.277.3.18 SetHostname()**

```
void gdcm::ServiceClassUser::SetHostname (
    const char * hostname )
```

Set the name of the called hostname (hostname or IP address)

**Examples**

[CStoreQtProgress.cxx](#).

**10.277.3.19 SetPort()**

```
void gdcm::ServiceClassUser::SetPort (
    uint16_t port )
```

Set port of remote host (called application)

**Examples**

[CStoreQtProgress.cxx](#).

**10.277.3.20 SetPortSCP()**

```
void gdcm::ServiceClassUser::SetPortSCP (
    uint16_t portscp )
```

Set the port for any incoming C-STORE-SCP operation (typically in a return of C-MOVE)

**10.277.3.21 SetPresentationContexts()**

```
void gdcm::ServiceClassUser::SetPresentationContexts (
    std::vector< PresentationContext > const & pcs )
```

Set the Presentation Context used for the Association.

**Examples**

[CStoreQtProgress.cxx](#).

**10.277.3.22 SetTimeout()**

```
void gdcm::ServiceClassUser::SetTimeout (
    double t )
```

set/get Timeout

**Examples**

[CStoreQtProgress.cxx](#).

### 10.277.3.23 StartAssociation()

```
bool gdcm::ServiceClassUser::StartAssociation ( )
```

Start the association. Need to call SetPresentationContexts before.

#### Examples

[CStoreQtProgress.cxx](#).

### 10.277.3.24 StopAssociation()

```
bool gdcm::ServiceClassUser::StopAssociation ( )
```

Stop the running association.

#### Examples

[CStoreQtProgress.cxx](#).

The documentation for this class was generated from the following file:

- [gdcmServiceClassUser.h](#)

## 10.278 gdcm::SHA1 Class Reference

Class for [SHA1](#).

```
#include <gdcmSHA1.h>
```

### Public Member Functions

- [SHA1](#) ()
- [SHA1](#) (const [SHA1](#) &)=delete
- [~SHA1](#) ()
- void [operator=](#) (const [SHA1](#) &)=delete

### Static Public Member Functions

- static bool [Compute](#) (const char \*buffer, unsigned long buf\_len, char digest\_str[20 \*2+1])
- static bool [ComputeFile](#) (const char \*filename, char digest\_str[20 \*2+1])

## 10.278.1 Detailed Description

Class for [SHA1](#).

### Warning

this class is able to pick from one implementation:

1. the one from OpenSSL (when GDCM\_USE\_SYSTEM\_OPENSSL is turned ON)

In all other cases it will return an error

## 10.278.2 Constructor & Destructor Documentation

### 10.278.2.1 SHA1() [1/2]

```
gdcM::SHA1::SHA1 ( )
```

### 10.278.2.2 ~SHA1()

```
gdcM::SHA1::~~SHA1 ( )
```

### 10.278.2.3 SHA1() [2/2]

```
gdcM::SHA1::SHA1 (
    const SHA1 & ) [delete]
```

## 10.278.3 Member Function Documentation

### 10.278.3.1 Compute()

```
static bool gdcM::SHA1::Compute (
    const char * buffer,
    unsigned long buf_len,
    char digest_str[20 *2+1] ) [static]
```

### 10.278.3.2 ComputeFile()

```
static bool gdcM::SHA1::ComputeFile (
    const char * filename,
    char digest_str[20 *2+1] ) [static]
```



### 10.278.3.3 operator=()

```
void gdcm::SHA1::operator= (
    const SHA1 & ) [delete]
```

The documentation for this class was generated from the following file:

- [gdcmSHA1.h](#)

## 10.279 gdcm::SimpleMemberCommand< T > Class Template Reference

[Command](#) subclass that calls a pointer to a member function.

```
#include <gdcmCommand.h>
```

Inheritance diagram for gdcm::SimpleMemberCommand< T >:



Collaboration diagram for `gdcm::SimpleMemberCommand< T >`:



### Public Types

- typedef `SimpleMemberCommand Self`
- typedef `void(T::* TMemberFunctionPointer) ()`

### Public Member Functions

- `SimpleMemberCommand (const Self &)=delete`
- `void Execute (const Subject *, const Event &) override`
- `void Execute (Subject *, const Event &) override`
- `void operator= (const Self &)=delete`
- `void SetCallbackFunction (T *object, TMemberFunctionPointer memberFunction)`

### Public Member Functions inherited from `gdcm::Command`

- `Command (const Command &)=delete`
- `void operator= (const Command &)=delete`

## Public Member Functions inherited from [gdcmm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*) const
- [Command](#) \* [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

## Public Member Functions inherited from [gdcmm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

## Static Public Member Functions

- static [SmartPointer](#)< [SimpleMemberCommand](#) > [New](#) ()

## Protected Member Functions

- [SimpleMemberCommand](#) ()
- [~SimpleMemberCommand](#) () override=default

## Protected Member Functions inherited from [gdcmm::Command](#)

- [Command](#) ()
- [~Command](#) () override

## Protected Member Functions inherited from [gdcmm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

## Protected Attributes

- [TMemberFunctionPointer](#) [m\\_MemberFunction](#)
- T \* [m\\_This](#)

### 10.279.1 Detailed Description

```
template<typename T>
class gdcm::SimpleMemberCommand< T >
```

[Command](#) subclass that calls a pointer to a member function.

[SimpleMemberCommand](#) calls a pointer to a member function with no arguments.

### 10.279.2 Member Typedef Documentation

#### 10.279.2.1 Self

```
template<typename T >
typedef SimpleMemberCommand gdcm::SimpleMemberCommand< T >::Self
```

Standard class typedefs.

#### 10.279.2.2 TMemberFunctionPointer

```
template<typename T >
typedef void(T::* gdcm::SimpleMemberCommand< T >::TMemberFunctionPointer) ()
```

A method callback.

### 10.279.3 Constructor & Destructor Documentation

#### 10.279.3.1 SimpleMemberCommand() [1/2]

```
template<typename T >
gdcm::SimpleMemberCommand< T >::SimpleMemberCommand (
    const Self & ) [delete]
```

#### 10.279.3.2 SimpleMemberCommand() [2/2]

```
template<typename T >
gdcm::SimpleMemberCommand< T >::SimpleMemberCommand ( ) [inline], [protected]
```

Referenced by [gdcm::SimpleMemberCommand< T >::New\(\)](#).

#### 10.279.3.3 ~SimpleMemberCommand()

```
template<typename T >
gdcm::SimpleMemberCommand< T >::~~SimpleMemberCommand ( ) [override], [protected], [default]
```

## 10.279.4 Member Function Documentation

### 10.279.4.1 Execute() [1/2]

```
template<typename T >
void gdcmm::SimpleMemberCommand< T >::Execute (
    const Subject * caller,
    const Event & event ) [inline], [override], [virtual]
```

Abstract method that defines the action to be taken by the command. This variant is expected to be used when requests comes from a const [Object](#)

Implements [gdcmm::Command](#).

References [gdcmm::SimpleMemberCommand< T >::m\\_MemberFunction](#).

### 10.279.4.2 Execute() [2/2]

```
template<typename T >
void gdcmm::SimpleMemberCommand< T >::Execute (
    Subject * ,
    const Event & ) [inline], [override], [virtual]
```

Invoke the callback function.

Implements [gdcmm::Command](#).

References [gdcmm::SimpleMemberCommand< T >::m\\_MemberFunction](#).

### 10.279.4.3 New()

```
template<typename T >
static SmartPointer< SimpleMemberCommand > gdcmm::SimpleMemberCommand< T >::New ( ) [inline],
[static]
```

Run-time type information (and related methods). Method for creation through the object factory.

References [gdcmm::SimpleMemberCommand< T >::SimpleMemberCommand\(\)](#).

### 10.279.4.4 operator=()

```
template<typename T >
void gdcmm::SimpleMemberCommand< T >::operator= (
    const Self & ) [delete]
```

#### 10.279.4.5 SetCallbackFunction()

```
template<typename T >
void gdcm::SimpleMemberCommand< T >::SetCallbackFunction (
    T * object,
    TMemberFunctionPointer memberFunction ) [inline]
```

Specify the callback function.

References [gdcm::SimpleMemberCommand< T >::m\\_MemberFunction](#), and [gdcm::SimpleMemberCommand< T >::m\\_This](#).

### 10.279.5 Member Data Documentation

#### 10.279.5.1 m\_MemberFunction

```
template<typename T >
TMemberFunctionPointer gdcm::SimpleMemberCommand< T >::m_MemberFunction [protected]
```

Referenced by [gdcm::SimpleMemberCommand< T >::Execute\(\)](#), [gdcm::SimpleMemberCommand< T >::Execute\(\)](#), and [gdcm::SimpleMemberCommand< T >::SetCallbackFunction\(\)](#).

#### 10.279.5.2 m\_This

```
template<typename T >
T* gdcm::SimpleMemberCommand< T >::m_This [protected]
```

Referenced by [gdcm::SimpleMemberCommand< T >::SetCallbackFunction\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmCommand.h](#)

## 10.280 gdcm::SimpleSubjectWatcher Class Reference

[SimpleSubjectWatcher](#).

```
#include <gdcmSimpleSubjectWatcher.h>
```

### Public Member Functions

- [SimpleSubjectWatcher](#) (const [SimpleSubjectWatcher](#) &)=delete
- [SimpleSubjectWatcher](#) ([Subject](#) \*s, const char \*comment="")
- virtual [~SimpleSubjectWatcher](#) ()
- void [operator=](#) (const [SimpleSubjectWatcher](#) &)=delete

## Protected Member Functions

- virtual void [EndFilter](#) ()
- virtual void [ShowAbort](#) ()
- virtual void [ShowAnonymization](#) ([Subject](#) \*caller, const [Event](#) &evt)
- virtual void [ShowData](#) ([Subject](#) \*caller, const [Event](#) &evt)
- virtual void [ShowDataSet](#) ([Subject](#) \*caller, const [Event](#) &evt)
- virtual void [ShowFileName](#) ([Subject](#) \*caller, const [Event](#) &evt)
- virtual void [ShowIteration](#) ()
- virtual void [ShowProgress](#) ([Subject](#) \*caller, const [Event](#) &evt)
- virtual void [StartFilter](#) ()
- void [TestAbortOff](#) ()
- void [TestAbortOn](#) ()

### 10.280.1 Detailed Description

[SimpleSubjectWatcher](#).

This is a typical [Subject](#) Watcher class. It will observe all events.

#### Examples

[BasicAnonymizer.cs](#), [Cleaner.cs](#), [ClinicalTrialIdentificationWorkflow.cs](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [ScanDirectory.cs](#), and [SimpleScanner.cxx](#).

### 10.280.2 Constructor & Destructor Documentation

#### 10.280.2.1 SimpleSubjectWatcher() [1/2]

```
gdcm::SimpleSubjectWatcher::SimpleSubjectWatcher (
    Subject * s,
    const char * comment = "" )
```

#### 10.280.2.2 ~SimpleSubjectWatcher()

```
virtual gdcm::SimpleSubjectWatcher::~~SimpleSubjectWatcher ( ) [virtual]
```

#### 10.280.2.3 SimpleSubjectWatcher() [2/2]

```
gdcm::SimpleSubjectWatcher::SimpleSubjectWatcher (
    const SimpleSubjectWatcher & ) [delete]
```

### 10.280.3 Member Function Documentation

#### 10.280.3.1 EndFilter()

```
virtual void gdcM::SimpleSubjectWatcher::EndFilter ( ) [protected], [virtual]
```

#### 10.280.3.2 operator=()

```
void gdcM::SimpleSubjectWatcher::operator= (
    const SimpleSubjectWatcher & ) [delete]
```

#### 10.280.3.3 ShowAbort()

```
virtual void gdcM::SimpleSubjectWatcher::ShowAbort ( ) [protected], [virtual]
```

#### 10.280.3.4 ShowAnonymization()

```
virtual void gdcM::SimpleSubjectWatcher::ShowAnonymization (
    Subject * caller,
    const Event & evt ) [protected], [virtual]
```

#### 10.280.3.5 ShowData()

```
virtual void gdcM::SimpleSubjectWatcher::ShowData (
    Subject * caller,
    const Event & evt ) [protected], [virtual]
```

#### 10.280.3.6 ShowDataSet()

```
virtual void gdcM::SimpleSubjectWatcher::ShowDataSet (
    Subject * caller,
    const Event & evt ) [protected], [virtual]
```

#### 10.280.3.7 ShowFileName()

```
virtual void gdcM::SimpleSubjectWatcher::ShowFileName (
    Subject * caller,
    const Event & evt ) [protected], [virtual]
```

#### Examples

[SimpleScanner.cxx](#).



### 10.280.3.8 ShowIteration()

```
virtual void gdcm::SimpleSubjectWatcher::ShowIteration ( ) [protected], [virtual]
```

### 10.280.3.9 ShowProgress()

```
virtual void gdcm::SimpleSubjectWatcher::ShowProgress (
    Subject * caller,
    const Event & evt ) [protected], [virtual]
```

### 10.280.3.10 StartFilter()

```
virtual void gdcm::SimpleSubjectWatcher::StartFilter ( ) [protected], [virtual]
```

### 10.280.3.11 TestAbortOff()

```
void gdcm::SimpleSubjectWatcher::TestAbortOff ( ) [protected]
```

### 10.280.3.12 TestAbortOn()

```
void gdcm::SimpleSubjectWatcher::TestAbortOn ( ) [protected]
```

The documentation for this class was generated from the following file:

- [gdcmSimpleSubjectWatcher.h](#)

## 10.281 gdcm::MrProtocol::Slice Struct Reference

```
#include <gdcmMrProtocol.h>
```

Collaboration diagram for gdcm::MrProtocol::Slice:



**Public Attributes**

- [Vector3 Normal](#)
- [Vector3 Position](#)

**10.281.1 Member Data Documentation****10.281.1.1 Normal**

`Vector3` `gdcm::MrProtocol::Slice::Normal`

**10.281.1.2 Position**

`Vector3` `gdcm::MrProtocol::Slice::Position`

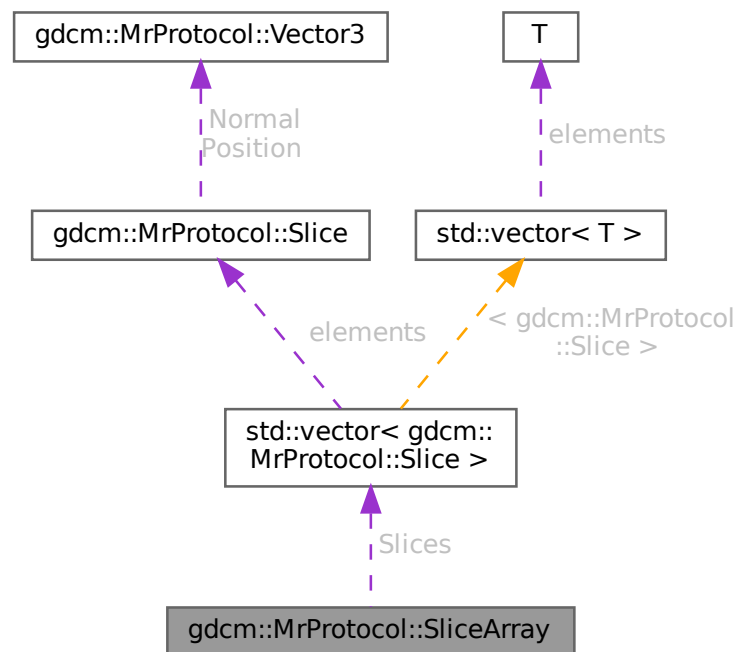
The documentation for this struct was generated from the following file:

- [gdcmMrProtocol.h](#)

**10.282 gdcm::MrProtocol::SliceArray Struct Reference**

```
#include <gdcmMrProtocol.h>
```

Collaboration diagram for `gdcm::MrProtocol::SliceArray`:



**Public Attributes**

- std::vector< [Slice](#) > [Slices](#)

**10.282.1 Member Data Documentation****10.282.1.1 Slices**

```
std::vector< Slice > gdcM::MrProtocol::SliceArray::Slices
```

The documentation for this struct was generated from the following file:

- [gdcMProtocol.h](#)

**10.283 gdcM::SmartPointer< ObjectType > Class Template Reference**

Class for Smart Pointer.

```
#include <gdcMSmartPointer.h>
```

Inheritance diagram for gdcM::SmartPointer< ObjectType >:



## Public Member Functions

- [SmartPointer](#) ()
- [SmartPointer](#) (const [SmartPointer](#)< ObjectType > &p)
- [SmartPointer](#) (ObjectType \*p)
- [SmartPointer](#) (ObjectType const &p)
- [~SmartPointer](#) ()
- ObjectType \* [GetPointer](#) () const  
*Explicit function to retrieve the pointer.*
- [operator ObjectType \\*](#) () const  
*Return pointer to object.*
- ObjectType & [operator\\*](#) () const
- ObjectType \* [operator->](#) () const  
*Overload operator ->*
- [SmartPointer](#) & [operator=](#) (ObjectType \*r)  
*Overload operator assignment.*
- [SmartPointer](#) & [operator=](#) (ObjectType const &r)
- [SmartPointer](#) & [operator=](#) ([SmartPointer](#) const &r)  
*Overload operator assignment.*

## 10.283.1 Detailed Description

```
template<class ObjectType>
class gdcm::SmartPointer< ObjectType >
```

Class for Smart Pointer.

Will only work for subclass of [gdcm::Object](#) See `tr1/shared_ptr` for a more general approach (not invasive) `#include <tr1/memory> { shared_ptr<Bla> b(new Bla); }`

### Note

Class partly based on post by Bill Hubauer: <http://groups.google.com/group/comp.lang.c++.msg/173ddc38a827a930>

### See also

<http://www.davethehat.com/articles/smarterp.htm>

and `itk::SmartPointer`

### Examples

[CStoreQtProgress.cxx](#), [ChangeSequenceUltrasound.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_FixBrokenJ2K.cxx](#), [GenAllVR.cxx](#), [GenFakelIdentifyFile.cxx](#), [GenFakelImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [SimpleScanner.cxx](#), [gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

## 10.283.2 Constructor & Destructor Documentation

### 10.283.2.1 SmartPointer() [1/4]

```
template<class ObjectType >
gdcmm::SmartPointer< ObjectType >::SmartPointer ( ) [inline]
```

### 10.283.2.2 SmartPointer() [2/4]

```
template<class ObjectType >
gdcmm::SmartPointer< ObjectType >::SmartPointer (
    const SmartPointer< ObjectType > & p ) [inline]
```

### 10.283.2.3 SmartPointer() [3/4]

```
template<class ObjectType >
gdcmm::SmartPointer< ObjectType >::SmartPointer (
    ObjectType * p ) [inline]
```

### 10.283.2.4 SmartPointer() [4/4]

```
template<class ObjectType >
gdcmm::SmartPointer< ObjectType >::SmartPointer (
    ObjectType const & p ) [inline]
```

### 10.283.2.5 ~SmartPointer()

```
template<class ObjectType >
gdcmm::SmartPointer< ObjectType >::~SmartPointer ( ) [inline]
```

## 10.283.3 Member Function Documentation

### 10.283.3.1 GetPointer()

```
template<class ObjectType >
ObjectType * gdcmm::SmartPointer< ObjectType >::GetPointer ( ) const [inline]
```

Explicit function to retrieve the pointer.

### 10.283.3.2 operator ObjectType \*()

```
template<class ObjectType >
gdcM::SmartPointer< ObjectType >::operator ObjectType * ( ) const [inline]
```

Return pointer to object.

### 10.283.3.3 operator\*()

```
template<class ObjectType >
ObjectType & gdcM::SmartPointer< ObjectType >::operator* ( ) const [inline]
```

### 10.283.3.4 operator->()

```
template<class ObjectType >
ObjectType * gdcM::SmartPointer< ObjectType >::operator-> ( ) const [inline]
```

Overload operator ->

### 10.283.3.5 operator=( ) [1/3]

```
template<class ObjectType >
SmartPointer & gdcM::SmartPointer< ObjectType >::operator= (
    ObjectType * r ) [inline]
```

Overload operator assignment.

### 10.283.3.6 operator=( ) [2/3]

```
template<class ObjectType >
SmartPointer & gdcM::SmartPointer< ObjectType >::operator= (
    ObjectType const & r ) [inline]
```

References [gdcM::SmartPointer< ObjectType >::operator=\( \)](#).

### 10.283.3.7 operator=( ) [3/3]

```
template<class ObjectType >
SmartPointer & gdcM::SmartPointer< ObjectType >::operator= (
    SmartPointer< ObjectType > const & r ) [inline]
```

Overload operator assignment.

References [gdcM::SmartPointer< ObjectType >::operator=\( \)](#).

Referenced by [gdcM::SmartPointer< ObjectType >::operator=\( \)](#), and [gdcM::SmartPointer< ObjectType >::operator=\( \)](#).

The documentation for this class was generated from the following files:

- [gdcMObject.h](#)
- [gdcMSmartPointer.h](#)

## 10.284 gdcm::network::SOPClassExtendedNegociationSub Class Reference

[SOPClassExtendedNegociationSub](#).

```
#include <gdcmSOPClassExtendedNegociationSub.h>
```

### Public Member Functions

- [SOPClassExtendedNegociationSub](#) ()
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [SetTuple](#) (const char \*uid, uint8\_t levelofsupport=3, uint8\_t levelofdignalsig=0, uint8\_t elementcoercion=2)
- size\_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

### 10.284.1 Detailed Description

[SOPClassExtendedNegociationSub](#).

PS 3.7 [Table](#) D.3-11 SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-↔ ASSOCIATE-AC)

### 10.284.2 Constructor & Destructor Documentation

#### 10.284.2.1 SOPClassExtendedNegociationSub()

```
gdcm::network::SOPClassExtendedNegociationSub::SOPClassExtendedNegociationSub ( )
```

### 10.284.3 Member Function Documentation

#### 10.284.3.1 Print()

```
void gdcm::network::SOPClassExtendedNegociationSub::Print (
    std::ostream & os ) const
```

#### 10.284.3.2 Read()

```
std::istream & gdcm::network::SOPClassExtendedNegociationSub::Read (
    std::istream & is )
```

### 10.284.3.3 SetTuple()

```
void gdcmm::network::SOPClassExtendedNegociationSub::SetTuple (
    const char * uid,
    uint8_t levelofsupport = 3,
    uint8_t levelofdigitalsig = 0,
    uint8_t elementcoercion = 2 )
```

### 10.284.3.4 Size()

```
size_t gdcmm::network::SOPClassExtendedNegociationSub::Size ( ) const
```

### 10.284.3.5 Write()

```
const std::ostream & gdcmm::network::SOPClassExtendedNegociationSub::Write (
    std::ostream & os ) const
```

The documentation for this class was generated from the following file:

- [gdcmmSOPClassExtendedNegociationSub.h](#)

## 10.285 gdcmm::SOPClassUIDToIOD Class Reference

Class convert a class SOP Class UID into [IOD](#).

```
#include <gdcmmSOPClassUIDToIOD.h>
```

### Public Types

- typedef const char \* [const](#)(SOPClassUIDToIODType)[2]

### Static Public Member Functions

- static [const](#) char \* [GetIOD](#) (UIDs [const](#) &uid)
- static [const](#) char \* [GetIODFromSOPClassUID](#) ([const](#) char \*sopclassuid)
- static unsigned int [GetNumberOfSOPClassToIOD](#) ()  
*Return the number of SOP Class UID listed internally.*
- static [const](#) char \* [GetSOPClassUIDFromIOD](#) ([const](#) char \*iod)
- static SOPClassUIDToIODType & [GetSOPClassUIDToIOD](#) (unsigned int i)
- static SOPClassUIDToIODType \* [GetSOPClassUIDToIODs](#) ()



## 10.285.1 Detailed Description

Class convert a class SOP Class UID into [IOD](#).

Reference PS 3.4 [Table B.5-1 STANDARD SOP CLASSES](#)

## 10.285.2 Member Typedef Documentation

### 10.285.2.1 const

```
typedef const char * gdcm::SOPClassUIDToIOD::const (SOPClassUIDToIODType) [2]
```

## 10.285.3 Member Function Documentation

### 10.285.3.1 GetIOD()

```
static const char * gdcm::SOPClassUIDToIOD::GetIOD (
    UIDs const & uid ) [static]
```

Return the associated [IOD](#) based on a SOP Class UID uid (there is a one-to-one mapping from SOP Class UID to matching [IOD](#))

#### Examples

[GenerateStandardSOPClasses.cxx](#).

### 10.285.3.2 GetIODFromSOPClassUID()

```
static const char * gdcm::SOPClassUIDToIOD::GetIODFromSOPClassUID (
    const char * sopclassuid ) [static]
```

### 10.285.3.3 GetNumberOfSOPClassToIOD()

```
static unsigned int gdcm::SOPClassUIDToIOD::GetNumberOfSOPClassToIOD ( ) [static]
```

Return the number of SOP Class UID listed internally.

### 10.285.3.4 GetSOPClassUIDFromIOD()

```
static const char * gdcm::SOPClassUIDToIOD::GetSOPClassUIDFromIOD (
    const char * iod ) [static]
```

### 10.285.3.5 GetSOPClassUIDToIOD()

```
static SOPClassUIDToIODType & gdcM::SOPClassUIDToIOD::GetSOPClassUIDToIOD (
    unsigned int i ) [static]
```

### 10.285.3.6 GetSOPClassUIDToIODs()

```
static SOPClassUIDToIODType * gdcM::SOPClassUIDToIOD::GetSOPClassUIDToIODs ( ) [static]
```

The documentation for this class was generated from the following file:

- [gdcMSOPClassUIDToIOD.h](#)

## 10.286 gdcM::Sorter Class Reference

[Sorter.](#)

```
#include <gdcMSorter.h>
```

Inheritance diagram for gdcM::Sorter:



Collaboration diagram for gdcM::Sorter:



## Public Types

- typedef bool(\* [SortFunction](#)) ([DataSet](#) const &, [DataSet](#) const &)  
*Set the sort function which compares one dataset to the other.*

## Public Member Functions

- [Sorter](#) ()
- virtual [~Sorter](#) ()
- bool [AddSelect](#) ([Tag](#) const &tag, const char \*value)  
*UNSUPPORTED FOR NOW.*
- const std::vector< std::string > & [GetFileNames](#) () const
- void [Print](#) (std::ostream &os) const  
*Print.*
- void [SetSortFunction](#) ([SortFunction](#) f)
- void [SetTagsToRead](#) (std::set< [Tag](#) > const &tags)
- virtual bool [Sort](#) (std::vector< std::string > const &filenames)  
*Typically the output of [Directory::GetFileNames\(\)](#)*
- virtual bool [StableSort](#) (std::vector< std::string > const &filenames)

## Protected Types

- typedef std::map< [Tag](#), std::string > [SelectionMap](#)

## Protected Attributes

- std::vector< std::string > [FileNames](#)
- std::map< [Tag](#), std::string > [Selection](#)
- [SortFunction](#) [SortFunc](#)
- std::set< [Tag](#) > [TagsToRead](#)

## Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [Sorter](#) &s)

## 10.286.1 Detailed Description

[Sorter](#).

General class to do sorting using a custom function You simply need to provide a function of type: [Sorter::SortFunction](#)

### Warning

implementation details. For now there is no cache mechanism. Which means that every time you call Sort, all files specified as input parameter are *read*

### See also

[Scanner](#)

### Examples

[SortImage.cxx](#), [SortImage2.cs](#), and [VolumeSorter.cxx](#).

## 10.286.2 Member Typedef Documentation

### 10.286.2.1 SelectionMap

```
typedef std::map<Tag, std::string> gdcM::Sorter::SelectionMap [protected]
```

### 10.286.2.2 SortFunction

```
typedef bool(* gdcM::Sorter::SortFunction) (DataSet const &, DataSet const &)
```

Set the sort function which compares one dataset to the other.

## 10.286.3 Constructor & Destructor Documentation

### 10.286.3.1 Sorter()

```
gdcM::Sorter::Sorter ( )
```

### 10.286.3.2 ~Sorter()

```
virtual gdcM::Sorter::~Sorter ( ) [virtual]
```

## 10.286.4 Member Function Documentation

### 10.286.4.1 AddSelect()

```
bool gdcM::Sorter::AddSelect (
    Tag const & tag,
    const char * value )
```

UNSUPPORTED FOR NOW.

### 10.286.4.2 GetFileNames()

```
const std::vector< std::string > & gdcM::Sorter::GetFileNames ( ) const [inline]
```

Return the list of filenames as sorted by the specific algorithm used. Empty by default (before [Sort\(\)](#) is called)

#### Examples

[Compute3DSpacing.cxx](#), [SortImage.cxx](#), [VolumeSorter.cxx](#), [gdcMorthoplanes.cxx](#), and [reslicesphere.cxx](#).

### 10.286.4.3 Print()

```
void gdcm::Sorter::Print (
    std::ostream & os ) const
```

Print.

#### Examples

[SortImage.cxx](#), [VolumeSorter.cxx](#), and [gdcmorthoplanes.cxx](#).

### 10.286.4.4 SetSortFunction()

```
void gdcm::Sorter::SetSortFunction (
    SortFunction f )
```

#### Examples

[SortImage.cxx](#), [SortImage2.cs](#), and [VolumeSorter.cxx](#).

### 10.286.4.5 SetTagsToRead()

```
void gdcm::Sorter::SetTagsToRead (
    std::set< Tag > const & tags )
```

Specify a set of tags to be read in during the sort procedure. By default this set is empty, in which case the entire image, including pixel data, is read in.

### 10.286.4.6 Sort()

```
virtual bool gdcm::Sorter::Sort (
    std::vector< std::string > const & filenames ) [virtual]
```

Typically the output of [Directory::GetFilenames\(\)](#)

Reimplemented in [gdcm::IPPSorter](#).

#### Examples

[SortImage.cxx](#).

#### 10.286.4.7 StableSort()

```
virtual bool gdcm::Sorter::StableSort (
    std::vector< std::string > const & filenames ) [virtual]
```

#### Examples

[SortImage.cxx](#), and [VolumeSorter.cxx](#).

### 10.286.5 Friends And Related Symbol Documentation

#### 10.286.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const Sorter & s ) [friend]
```

### 10.286.6 Member Data Documentation

#### 10.286.6.1 Filenames

```
std::vector<std::string> gdcm::Sorter::Filenames [protected]
```

#### 10.286.6.2 Selection

```
std::map<Tag,std::string> gdcm::Sorter::Selection [protected]
```

#### 10.286.6.3 SortFunc

```
SortFunction gdcm::Sorter::SortFunc [protected]
```

#### 10.286.6.4 TagsToRead

```
std::set<Tag> gdcm::Sorter::TagsToRead [protected]
```

The documentation for this class was generated from the following file:

- [gdcmSorter.h](#)

## 10.287 gdcm::Spacing Class Reference

Class for [Spacing](#).

```
#include <gdcmSpacing.h>
```

### Public Types

- enum [SpacingType](#) {  
    [DETECTOR](#) = 0 ,  
    [MAGNIFIED](#) ,  
    [CALIBRATED](#) ,  
    [UNKNOWN](#) }

### Public Member Functions

- [Spacing](#) ()
- [~Spacing](#) ()=default

### Static Public Member Functions

- static [Attribute](#)< 0x28, 0x34 > [ComputePixelAspectRatioFromPixelSpacing](#) (const [Attribute](#)< 0x28, 0x30 > &pixelspacing)

### 10.287.1 Detailed Description

Class for [Spacing](#).

It all began with a mail to WG6:

**Subject:** Imager Pixel [Spacing](#) vs Pixel [Spacing](#) **Body:** [Apologies for the duplicate post, namely to David Clunie & OFFIS team]

I have been trying to understand CP-586 in the following two cases:

On the one hand:

- DISCIMG/IMAGES/CRIMAGE taken from <http://dclunie.com/images/pixelspacingtestimages.zip>

And on the other hand:

- [http://gdcm.sourceforge.net/thingies/cr\\_pixelspacing.dcm](http://gdcm.sourceforge.net/thingies/cr_pixelspacing.dcm)

If I understand correctly the CP, one is required to use Pixel [Spacing](#) for measurement ('true size' print) instead of Imager Pixel [Spacing](#), since the two attributes are present and Pixel [Spacing](#) is different from Imager Pixel [Spacing](#).

If this is correct, then the test data DISCIMG/IMAGES/CRIMAGE is incorrect. If this is incorrect (ie. I need to use Imager Pixel [Spacing](#)), then the display of cr\_pixelspacing.dcm for measurement will be incorrect.

Could someone please let me know what am I missing here? I could not find any information in any header that would allow me to differentiate those.

Thank you for your time,

Ref: <http://lists.nema.org/scripts/lyris.pl?sub=488573&id=400720477>

See PS 3.3-2008, [Table C.7-11b](#) IMAGE PIXEL MACRO ATTRIBUTES

Ratio of the vertical size and horizontal size of the pixels in the image specified by a pair of integer values where the first value is the vertical pixel size, and the second value is the horizontal pixel size. Required if the aspect ratio values do not have a ratio of 1:1 and the physical pixel spacing is not specified by Pixel [Spacing](#) (0028,0030), or Imager Pixel [Spacing](#) (0018,1164) or Nominal Scanned Pixel [Spacing](#) (0018,2010), either for the entire [Image](#) or per-frame in a Functional Group [Macro](#). See C.7.6.3.1.7.

PS 3.3-2008 10.7.1.3 Pixel [Spacing Value](#) Order and Valid Values All pixel spacing related attributes shall have non-zero values, except when there is only a single row or column or pixel of data present, in which case the corresponding value may be zero.

Ref: [http://gdcm.sourceforge.net/wiki/index.php/Imager\\_Pixel\\_Spacing](http://gdcm.sourceforge.net/wiki/index.php/Imager_Pixel_Spacing)

## 10.287.2 Member Enumeration Documentation

### 10.287.2.1 SpacingType

```
enum gdcm::Spacing::SpacingType
```

Enumerator

DETECTOR	
MAGNIFIED	
CALIBRATED	
UNKNOWN	

## 10.287.3 Constructor & Destructor Documentation

### 10.287.3.1 Spacing()

```
gdcm::Spacing::Spacing ( )
```



### 10.287.3.2 ~Spacing()

```
gdcm::Spacing::~Spacing ( ) [default]
```

## 10.287.4 Member Function Documentation

### 10.287.4.1 ComputePixelAspectRatioFromPixelSpacing()

```
static Attribute< 0x28, 0x34 > gdcm::Spacing::ComputePixelAspectRatioFromPixelSpacing (
    const Attribute< 0x28, 0x30 > & pixelspacing ) [static]
```

The documentation for this class was generated from the following file:

- [gdcmSpacing.h](#)

## 10.288 gdcm::Spectroscopy Class Reference

[Spectroscopy](#) class.

```
#include <gdcmSpectroscopy.h>
```

### Public Member Functions

- [Spectroscopy](#) ()=default

### 10.288.1 Detailed Description

[Spectroscopy](#) class.

## 10.288.2 Constructor & Destructor Documentation

### 10.288.2.1 Spectroscopy()

```
gdcm::Spectroscopy::Spectroscopy ( ) [default]
```

The documentation for this class was generated from the following file:

- [gdcmSpectroscopy.h](#)

## 10.289 gdcm::SplitMosaicFilter Class Reference

[SplitMosaicFilter](#) class.

```
#include <gdcmSplitMosaicFilter.h>
```

### Public Member Functions

- [SplitMosaicFilter](#) ()
- [~SplitMosaicFilter](#) ()
- bool [ComputeMOSAICDimensions](#) (unsigned int dims[3])
- bool [ComputeMOSAICImagePositionPatient](#) (double pos[3], const double ipp[6], const double dircos[6], const double pixelspacing[3], const unsigned int image\_dims[3], const unsigned int mosaic\_dims[3], bool inverted)
- *Extract the value for ImagePositionPatient.*
- bool [ComputeMOSAICSliceNormal](#) (double dims[3], bool &inverted)
- *Extract the value for SliceNormalVector (CSA header)*
- bool [ComputeMOSAICSlicePosition](#) (double pos[3], bool inverted)
- [File](#) & [GetFile](#) ()
- const [File](#) & [GetFile](#) () const
- [Image](#) & [GetImage](#) ()
- const [Image](#) & [GetImage](#) () const
- void [SetFile](#) (const [File](#) &f)
- void [SetImage](#) (const [Image](#) &image)
- bool [Split](#) ()
- *Split the SIEMENS MOSAIC image.*

### Static Public Member Functions

- static bool [GetAcquisitionSize](#) (unsigned int size[2], [DataSet](#) const &ds)
- *Get the Acquisition Matrix (non zero value):*
- static unsigned int [GetNumberOfImagesInMosaic](#) ([File](#) const &file)
- *Return the value for NumberOfImagesInMosaic, or compute it from Acquisition Size.*

### 10.289.1 Detailed Description

[SplitMosaicFilter](#) class.

Class to reshuffle bytes for a SIEMENS Mosaic image Siemens CSA [Image](#) Header CSA:= Common Siemens Architecture, sometimes also known as Common syngo Architecture

#### Warning

when private attributes are not found, the acquisition matrix is used to compute the NumberOfImagesInMosaic. This means trailing black slices will be considered in the volume (instead of discarded). CSA 0029,1010 is needed for correct NumberOfImagesInMosaic CSA 0029,1020 is needed to compute the correct origin without above info default are taken (may not be accurate).

## 10.289.2 Constructor & Destructor Documentation

### 10.289.2.1 SplitMosaicFilter()

```
gdcm::SplitMosaicFilter::SplitMosaicFilter ( )
```

### 10.289.2.2 ~SplitMosaicFilter()

```
gdcm::SplitMosaicFilter::~SplitMosaicFilter ( )
```

## 10.289.3 Member Function Documentation

### 10.289.3.1 ComputeMOSAICDimensions()

```
bool gdcm::SplitMosaicFilter::ComputeMOSAICDimensions (
    unsigned int dims[3] )
```

Compute the new dimensions according to private information stored in the MOSAIC header.

### 10.289.3.2 ComputeMOSAICImagePositionPatient()

```
bool gdcm::SplitMosaicFilter::ComputeMOSAICImagePositionPatient (
    double pos[3],
    const double ipp[6],
    const double dircos[6],
    const double pixelspacing[3],
    const unsigned int image_dims[3],
    const unsigned int mosaic_dims[3],
    bool inverted )
```

Extract the value for ImagePositionPatient.

### 10.289.3.3 ComputeMOSAICSliceNormal()

```
bool gdcm::SplitMosaicFilter::ComputeMOSAICSliceNormal (
    double dims[3],
    bool & inverted )
```

Extract the value for SliceNormal/Vector (CSA header)

### 10.289.3.4 ComputeMOSAICSlicePosition()

```
bool gdcm::SplitMosaicFilter::ComputeMOSAICSlicePosition (
    double pos[3],
    bool inverted )
```

Extract the value for ImagePositionPatient (requires inverted flag) Deprecated

### 10.289.3.5 GetAcquisitionSize()

```
static bool gdcm::SplitMosaicFilter::GetAcquisitionSize (
    unsigned int size[2],
    DataSet const & ds ) [static]
```

Get the Acquisition Matrix (non zero value):

### 10.289.3.6 GetFile() [1/2]

```
File & gdcm::SplitMosaicFilter::GetFile ( ) [inline]
```

### 10.289.3.7 GetFile() [2/2]

```
const File & gdcm::SplitMosaicFilter::GetFile ( ) const [inline]
```

### 10.289.3.8 GetImage() [1/2]

```
Image & gdcm::SplitMosaicFilter::GetImage ( ) [inline]
```

### 10.289.3.9 GetImage() [2/2]

```
const Image & gdcm::SplitMosaicFilter::GetImage ( ) const [inline]
```

### 10.289.3.10 GetNumberOfImagesInMosaic()

```
static unsigned int gdcm::SplitMosaicFilter::GetNumberOfImagesInMosaic (
    File const & file ) [static]
```

Return the value for NumberOfImagesInMosaic, or compute it from Acquisition Size.

### 10.289.3.11 SetFile()

```
void gdcm::SplitMosaicFilter::SetFile (
    const File & f ) [inline]
```

### 10.289.3.12 SetImage()

```
void gdcm::SplitMosaicFilter::SetImage (
    const Image & image )
```

### 10.289.3.13 Split()

```
bool gdcmm::SplitMosaicFilter::Split ( )
```

Split the SIEMENS MOSAIC image.

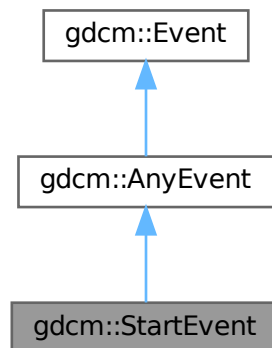
The documentation for this class was generated from the following file:

- [gdcmmSplitMosaicFilter.h](#)

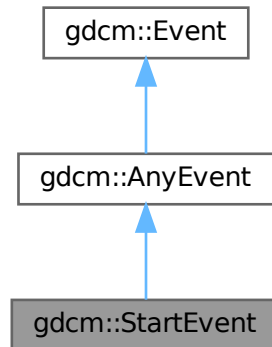
## 10.290 gdcmm::StartEvent Class Reference

```
#include <gdcmmEvent.h>
```

Inheritance diagram for gdcmm::StartEvent:



Collaboration diagram for `gdcm::StartEvent`:



#### Additional Inherited Members

#### Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()
- [Event](#) (const [Event](#) &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) \*) const =0
- virtual const char \* [GetEventName](#) () const =0
- virtual [Event](#) \* [MakeObject](#) () const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

The documentation for this class was generated from the following file:

- [gdcmEvent.h](#)

## 10.291 `gdcm::static_assert_test< x >` Struct Template Reference

```
#include <gdcmStaticAssert.h>
```

The documentation for this struct was generated from the following file:

- [gdcmStaticAssert.h](#)

## 10.292 gdcm::STATIC\_ASSERTION\_FAILURE< x > Struct Template Reference

The documentation for this struct was generated from the following file:

- [gdcmStaticAssert.h](#)

## 10.293 gdcm::STATIC\_ASSERTION\_FAILURE< true > Struct Reference

```
#include <gdcmStaticAssert.h>
```

### Public Types

- enum { [value](#) = 1 }

### 10.293.1 Member Enumeration Documentation

#### 10.293.1.1 anonymous enum

```
anonymous enum
```

#### Enumerator

value	
-------	--

The documentation for this struct was generated from the following file:

- [gdcmStaticAssert.h](#)

## 10.294 gdcm::StreamImageReader Class Reference

[StreamImageReader.](#)

```
#include <gdcmStreamImageReader.h>
```

## Public Member Functions

- [StreamImageReader](#) ()
- virtual [~StreamImageReader](#) ()
- bool [CanReadImage](#) () const
- void [DefinePixelExtent](#) (uint16\_t inXMin, uint16\_t inXMax, uint16\_t inYMin, uint16\_t inYMax, uint16\_t inZMin=0, uint16\_t inZMax=1)
- uint32\_t [DefineProperBufferLength](#) () const
- std::vector< unsigned int > [GetDimensionsValueForResolution](#) (unsigned int)
- [File](#) const & [GetFile](#) () const
- bool [Read](#) (char \*inReadBuffer, const std::size\_t &inBufferLength)
- virtual bool [ReadImageInformation](#) ()
- void [SetFileName](#) (const char \*inFileName)
- void [SetStream](#) (std::istream &inStream)

## 10.294.1 Detailed Description

[StreamImageReader](#).

### Note

its role is to convert the DICOM [DataSet](#) into a [Image](#) representation via an ITK streaming (ie, multithreaded) interface [Image](#) is different from [Pixmap](#) has it has a position and a direction in Space. Currently, this class is thread safe in that it can read a single extent in a single thread. Multiple versions can be used for multiple extents/threads.

### See also

[Image](#)

### Examples

[ExtractOneFrame.cs](#), and [StreamImageReaderTest.cxx](#).

## 10.294.2 Constructor & Destructor Documentation

### 10.294.2.1 StreamImageReader()

```
gdcm::StreamImageReader::StreamImageReader ( )
```

### 10.294.2.2 ~StreamImageReader()

```
virtual gdcm::StreamImageReader::~~StreamImageReader ( ) [virtual]
```



## 10.294.3 Member Function Documentation

### 10.294.3.1 CanReadImage()

```
bool gdcm::StreamImageReader::CanReadImage ( ) const
```

Only RAW images are currently readable by the stream reader. As more streaming codecs are added, then this function will be updated to reflect those changes. Calling this function prior to reading will ensure that only streamable files are streamed. Make sure to call ReadImageInformation prior to calling this function.

#### Examples

[StreamImageReaderTest.cxx](#).

### 10.294.3.2 DefinePixelExtent()

```
void gdcm::StreamImageReader::DefinePixelExtent (
    uint16_t inXMin,
    uint16_t inXMax,
    uint16_t inYMin,
    uint16_t inYMax,
    uint16_t inZMin = 0,
    uint16_t inZMax = 1 )
```

Defines an image extent for the Read function. DICOM states that an image can have no more than  $2^{16}$  pixels per edge (as of 2009) In this case, the pixel extents ignore the direction cosines entirely, and assumes that the origin of the image is at location 0,0 (regardless of the definition in space per the tags). So, if the first 100 pixels of the first row are to be read in, this function should be called with DefinePixelExtent(0, 100, 0, 1), regardless of pixel size or orientation.

#### Examples

[ExtractOneFrame.cs](#), and [StreamImageReaderTest.cxx](#).

### 10.294.3.3 DefineProperBufferLength()

```
uint32_t gdcm::StreamImageReader::DefineProperBufferLength ( ) const
```

Paying attention to the pixel format and so forth, define the proper buffer length for the user. The return amount is in bytes. Call this function to determine the size of the char\* buffer that will need to be passed in to ReadImageSubregion(). If the return is 0, then that means that the pixel extent was not defined prior

#### Examples

[ExtractOneFrame.cs](#), and [StreamImageReaderTest.cxx](#).

#### 10.294.3.4 GetDimensionsValueForResolution()

```
std::vector< unsigned int > gdcM::StreamImageReader::GetDimensionsValueForResolution (
    unsigned int )
```

#### 10.294.3.5 GetFile()

```
File const & gdcM::StreamImageReader::GetFile ( ) const
```

Returns the dataset read by ReadImageInformation Couple this with the [ImageHelper](#) to get statistics about the image, like pixel extent, to be able to initialize buffers for reading

##### Examples

[ExtractOneFrame.cs](#), and [StreamImageReaderTest.cxx](#).

#### 10.294.3.6 Read()

```
bool gdcM::StreamImageReader::Read (
    char * inReadBuffer,
    const std::size_t & inBufferLength )
```

Read the DICOM image. There are three reasons for failure:

1. The extent is not set
2. the conversion from char\* to std::ostream (internally) fails
3. the given buffer isn't large enough to accommodate the desired pixel extent. This method has been implemented to look similar to the metainageio in itk MUST have an extent defined, or else Read will return false. If no particular extent is required, use [ImageReader](#) instead.

##### Examples

[ExtractOneFrame.cs](#), and [StreamImageReaderTest.cxx](#).

#### 10.294.3.7 ReadImageInformation()

```
virtual bool gdcM::StreamImageReader::ReadImageInformation ( ) [virtual]
```

Set the spacing and dimension information for the set filename. returns false if the file is not initialized or not an image, with the pixel (7fe0,0010) tag.

##### Examples

[ExtractOneFrame.cs](#), and [StreamImageReaderTest.cxx](#).

### 10.294.3.8 SetFileName()

```
void gdcm::StreamImageReader::SetFileName (
    const char * inFileName )
```

One of either SetFileName or SetStream must be called prior to any other functions. These initialize an internal [Reader](#) class to be able to get non-pixel image information.

#### Examples

[ExtractOneFrame.cs](#), and [StreamImageReaderTest.cxx](#).

### 10.294.3.9 SetStream()

```
void gdcm::StreamImageReader::SetStream (
    std::istream & inStream )
```

The documentation for this class was generated from the following file:

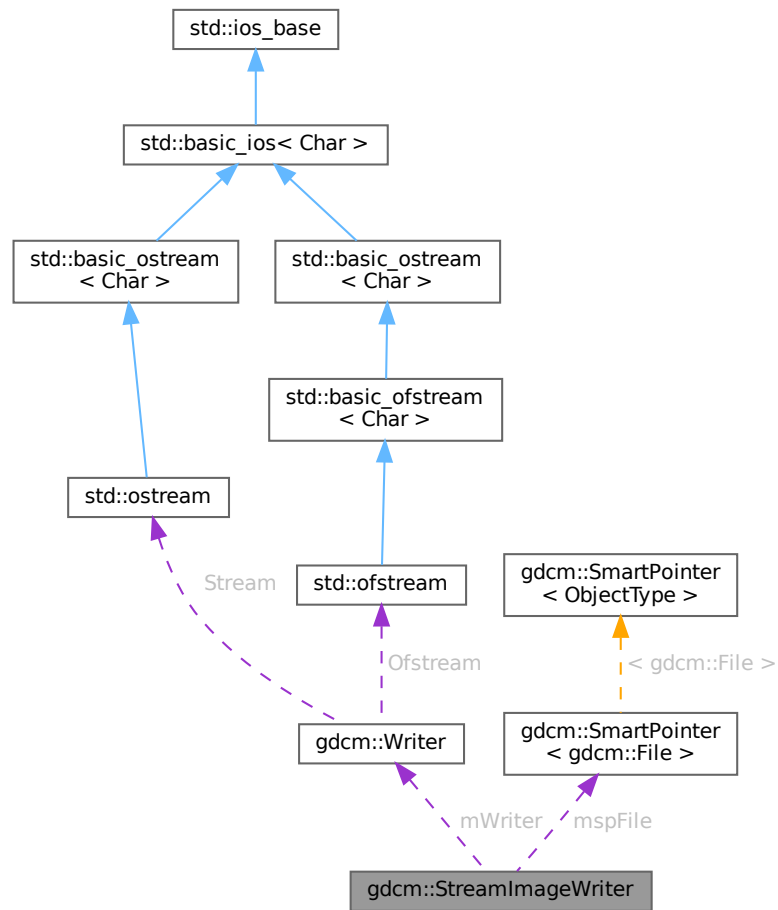
- [gdcmStreamImageReader.h](#)

## 10.295 gdcm::StreamImageWriter Class Reference

[StreamImageReader](#).

```
#include <gdcmStreamImageWriter.h>
```

Collaboration diagram for `gdcm::StreamImageWriter`:



## Public Member Functions

- [StreamImageWriter](#) ()
- virtual [~StreamImageWriter](#) ()
- bool [CanWriteFile](#) () const
- void [DefinePixelExtent](#) (uint16\_t inXMin, uint16\_t inXMax, uint16\_t inYMin, uint16\_t inYMax, uint16\_t inZMin=0, uint16\_t inZMax=1)
- uint32\_t [DefineProperBufferLength](#) ()
- void [SetFile](#) (const [File](#) &inFile)
- void [SetFileName](#) (const char \*inFileName)
- void [SetStream](#) (std::ostream &inStream)
- bool [Write](#) (void \*inWriteBuffer, const std::size\_t &inBufferLength)
- virtual bool [WriteImageInformation](#) ()

### Protected Member Functions

- virtual bool [WriteImageSubregionRAW](#) (char \*inWriteBuffer, const std::size\_t &inBufferLength)
- int [WriteRawHeader](#) ([RAWCodec](#) \*inCodec, std::ostream \*inStream)

### Protected Attributes

- int [mElementOffsets](#)
- int [mElementOffsets1](#)
- [SmartPointer](#)< [File](#) > [mspFile](#)
- [Writer](#) [mWriter](#)
- uint16\_t [mXMax](#)
- uint16\_t [mXMin](#)
- uint16\_t [mYMax](#)
- uint16\_t [mYMin](#)
- uint16\_t [mZMax](#)
- uint16\_t [mZMin](#)

## 10.295.1 Detailed Description

[StreamImageReader](#).

### Note

its role is to convert the DICOM [DataSet](#) into a [Image](#) representation via an ITK streaming (ie, multithreaded) interface [Image](#) is different from [Pixmap](#) has it has a position and a direction in Space. Currently, this class is threadsafe in that it can read a single extent in a single thread. Multiple versions can be used for multiple extents/threads.

### See also

[Image](#)

### Examples

[Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

## 10.295.2 Constructor & Destructor Documentation

### 10.295.2.1 StreamImageWriter()

```
gdcm::StreamImageWriter::StreamImageWriter ( )
```

### 10.295.2.2 ~StreamImageWriter()

```
virtual gdcm::StreamImageWriter::~~StreamImageWriter ( ) [virtual]
```

## 10.295.3 Member Function Documentation

### 10.295.3.1 CanWriteFile()

```
bool gdcM::StreamImageWriter::CanWriteFile ( ) const
```

This function determines if a file can even be written using the streaming writer unlike the reader, can be called before WriteImageInformation, but must be called after SetFile.

#### Examples

[Extracting\\_All\\_Resolution.cxx](#), and [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#).

### 10.295.3.2 DefinePixelExtent()

```
void gdcM::StreamImageWriter::DefinePixelExtent (
    uint16_t inXMin,
    uint16_t inXMax,
    uint16_t inYMin,
    uint16_t inYMax,
    uint16_t inZMin = 0,
    uint16_t inZMax = 1 )
```

Defines an image extent for the Read function. DICOM states that an image can have no more than  $2^{16}$  pixels per edge (as of 2009) In this case, the pixel extents ignore the direction cosines entirely, and assumes that the origin of the image is at location 0,0 (regardless of the definition in space per the tags). So, if the first 100 pixels of the first row are to be read in, this function should be called with DefinePixelExtent(0, 100, 0, 1), regardless of pixel size or orientation. 15 nov 2010: added z dimension, defaults to being 1 plane large

#### Examples

[Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

### 10.295.3.3 DefineProperBufferLength()

```
uint32_t gdcM::StreamImageWriter::DefineProperBufferLength ( )
```

Paying attention to the pixel format and so forth, define the proper buffer length for the user. The return amount is in bytes. If the return is 0, then that means that the pixel extent was not defined prior this return is for RAW inputs which are then encoded by the writer, but are used to ensure that the writer gets the proper buffer size

#### Examples

[Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

### 10.295.3.4 SetFile()

```
void gdcm::StreamImageWriter::SetFile (
    const File & inFile )
```

Set the image information to be written to disk that is everything but the pixel information: (7fe0,0010) PixelData

#### Examples

[Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

### 10.295.3.5 SetFileName()

```
void gdcm::StreamImageWriter::SetFileName (
    const char * inFileName )
```

One of either SetFileName or SetStream must be called prior to any other functions. These initialize an internal [Reader](#) class to be able to get non-pixel image information.

### 10.295.3.6 SetStream()

```
void gdcm::StreamImageWriter::SetStream (
    std::ostream & inStream )
```

#### Examples

[Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

### 10.295.3.7 Write()

```
bool gdcm::StreamImageWriter::Write (
    void * inWriteBuffer,
    const std::size_t & inBufferLength )
```

Read the DICOM image. There are three reasons for failure:

1. The extent is not set
2. the conversion from void\* to std::ostream (internally) fails
3. the given buffer isn't large enough to accommodate the desired pixel extent. This method has been implemented to look similar to the metaimageio in itk MUST have an extent defined, or else Read will return false. If no particular extent is required, use [ImageReader](#) instead.

#### Examples

[Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

### 10.295.3.8 WriteImageInformation()

```
virtual bool gdcm::StreamImageWriter::WriteImageInformation ( ) [virtual]
```

Write the header information to disk, and a bunch of zeros for the actual pixel information. Of course, if we're doing a non-compressed format, that works but if it's compressed, we have to force the ordering of chunks that are written.

#### Examples

[Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

### 10.295.3.9 WriteImageSubregionRAW()

```
virtual bool gdcm::StreamImageWriter::WriteImageSubregionRAW (
    char * inWriteBuffer,
    const std::size_t & inBufferLength ) [protected], [virtual]
```

Using the min, max, etc set by DefinePixelExtent, this will fill the given buffer. Make sure to call DefinePixelExtent and to initialize the buffer with the amount given by DefineProperBufferLength prior to calling this. Reads by the RAW codec; other codecs are added once implemented.

### 10.295.3.10 WriteRawHeader()

```
int gdcm::StreamImageWriter::WriteRawHeader (
    RAWCodec * inCodec,
    std::ostream * inStream ) [protected]
```

When writing a raw file, we know the full extent, and can just write the first 12 bytes out (the tag, the [VR](#), and the size). When we do compressed files, we'll do it in chunks, as described in 2009-3, part 5, Annex A, section 4. Pass the raw codec so that in the rare case of a bigendian explicit raw, the first 12 bytes written out should still be kosher. Returns -1 if there's any failure, or the complete offset (12 bytes) if it works. Those 12 bytes are then added to the position in order to determine where to write.

## 10.295.4 Member Data Documentation

### 10.295.4.1 mElementOffsets

```
int gdcm::StreamImageWriter::mElementOffsets [protected]
```

The result of WriteRawHeader (or another header, when that's implemented). This result is saved so that the first N bytes aren't constantly being rewritten for each chunk that's passed in. For compressed data, the offset table will require rewrites of data.

### 10.295.4.2 mElementOffsets1

```
int gdcm::StreamImageWriter::mElementOffsets1 [protected]
```



### 10.295.4.3 mspFile

`SmartPointer<File> gdcm::StreamImageWriter::mspFile` [protected]

### 10.295.4.4 mWriter

`Writer gdcm::StreamImageWriter::mWriter` [protected]

### 10.295.4.5 mXMax

`uint16_t gdcm::StreamImageWriter::mXMax` [protected]

### 10.295.4.6 mXMin

`uint16_t gdcm::StreamImageWriter::mXMin` [protected]

### 10.295.4.7 mYMax

`uint16_t gdcm::StreamImageWriter::mYMax` [protected]

### 10.295.4.8 mYMin

`uint16_t gdcm::StreamImageWriter::mYMin` [protected]

### 10.295.4.9 mZMax

`uint16_t gdcm::StreamImageWriter::mZMax` [protected]

### 10.295.4.10 mZMin

`uint16_t gdcm::StreamImageWriter::mZMin` [protected]

The documentation for this class was generated from the following file:

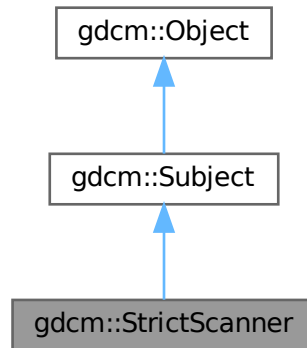
- [gdcmStreamImageWriter.h](#)

## 10.296 gdcmm::StrictScanner Class Reference

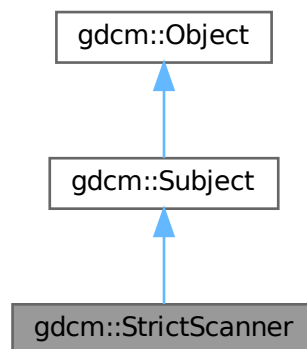
[StrictScanner](#).

```
#include <gdcmmStrictScanner.h>
```

Inheritance diagram for gdcmm::StrictScanner:



Collaboration diagram for gdcmm::StrictScanner:



### Classes

- struct [ltstr](#)

## Public Types

- typedef MappingType::const\_iterator [ConstIterator](#)
- typedef std::map< const char \*, [TagToValue](#), [Itstr](#) > [MappingType](#)
- typedef std::map< [Tag](#), const char \* > [TagToValue](#)
- typedef TagToValue::value\_type [TagToValueValueType](#)
- typedef std::set< std::string > [ValuesType](#)

## Public Member Functions

- [StrictScanner](#) ()
- [~StrictScanner](#) () override
- void [AddPrivateTag](#) ([PrivateTag](#) const &t)
- void [AddSkipTag](#) ([Tag](#) const &t)  
*Add a tag that will need to be skipped. Those are root level skip tags.*
- void [AddTag](#) ([Tag](#) const &t)  
*Add a tag that will need to be read. Those are root level skip tags.*
- [ConstIterator](#) [Begin](#) () const
- void [ClearSkipTags](#) ()
- void [ClearTags](#) ()
- [ConstIterator](#) [End](#) () const
- [Directory::FilenameType](#) [GetAllFileNamesFromTagToValue](#) ([Tag](#) const &t, const char \*valueref) const
- const char \* [GetFilenameFromTagToValue](#) ([Tag](#) const &t, const char \*valueref) const
- [Directory::FilenameType](#) const & [GetFileNames](#) () const
- [Directory::FilenameType](#) [GetKeys](#) () const
- [TagToValue](#) const & [GetMapping](#) (const char \*filename) const  
*Get the std::map mapping filenames to value for file 'filename'.*
- [TagToValue](#) const & [GetMappingFromTagToValue](#) ([Tag](#) const &t, const char \*value) const  
*See [GetFilenameFromTagToValue\(\)](#). This is simply [GetFilenameFromTagToValue](#) followed.*
- [MappingType](#) const & [GetMappings](#) () const  
*Mappings are the mapping from a particular tag to the map, mapping filename to value:*
- [Directory::FilenameType](#) [GetOrderedValues](#) ([Tag](#) const &t) const
- const char \* [GetValue](#) (const char \*filename, [Tag](#) const &t) const
- [ValuesType](#) const & [GetValues](#) () const  
*Get all the values found (in lexicographic order)*
- [ValuesType](#) [GetValues](#) ([Tag](#) const &t) const  
*Get all the values found (in lexicographic order) associated with [Tag](#) 't'.*
- bool [IsKey](#) (const char \*filename) const
- void [Print](#) (std::ostream &os) const override  
*Print result.*
- void [PrintTable](#) (std::ostream &os) const
- bool [Scan](#) ([Directory::FilenameType](#) const &filenames)  
*Start the scan !*

## Public Member Functions inherited from [gdcm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*) const
- [Command](#) \* [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

## Static Public Member Functions

- static [SmartPointer](#)< [StrictScanner](#) > [New](#) ()  
*for wrapped language: instantiate a reference counted object*

## Protected Member Functions

- void [ProcessPublicTag](#) ([StringFilter](#) &sf, const char \*filename)

## Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

## Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [StrictScanner](#) &s)

## 10.296.1 Detailed Description

[StrictScanner](#).

This filter is meant for quickly browsing a [FileSet](#) (a set of files on disk). Special consideration are taken so as to read the minimum amount of information in each file in order to retrieve the user specified set of DICOM [Attribute](#).

This filter is dealing with both VRASCII and VRBINARY element, thanks to the help of [StringFilter](#)

### Warning

IMPORTANT In case of file where tags are not ordered (illegal as per DICOM specification), the output will be missing information

### Note

implementation details. All values are stored in a `std::set of std::string`. Then the address of the `cstring` underlying the `std::string` is used in the `std::map`.

This class implement the Subject/Observer pattern trigger the following events:

- [ProgressEvent](#)
- [StartEvent](#)
- [EndEvent](#)

### Examples

[ScanDirectory.cs](#), and [SimpleScanner.cxx](#).

## 10.296.2 Member Typedef Documentation

### 10.296.2.1 ConstIterator

```
typedef MappingType::const_iterator gdcm::StrictScanner::ConstIterator
```

### 10.296.2.2 MappingType

```
typedef std::map<const char *,TagToValue, ltstr> gdcm::StrictScanner::MappingType
```

### 10.296.2.3 TagToValue

```
typedef std::map<Tag, const char*> gdcm::StrictScanner::TagToValue
```

struct to map a filename to a value Implementation note: all `std::map` in this class will be using `const char *` and not `std::string` since we are pointing to existing `std::string` (hold in a `std::vector`) this avoid an extra copy of the byte array. [Tag](#) are used as [Tag](#) class since `sizeof(tag) <= sizeof(pointer)`

#### 10.296.2.4 TagToValueValueType

```
typedef TagToValue::value_type gdc::StrictScanner::TagToValueValueType
```

#### 10.296.2.5 ValuesType

```
typedef std::set< std::string > gdc::StrictScanner::ValuesType
```

### 10.296.3 Constructor & Destructor Documentation

#### 10.296.3.1 StrictScanner()

```
gdc::StrictScanner::StrictScanner ( ) [inline]
```

#### 10.296.3.2 ~StrictScanner()

```
gdc::StrictScanner::~~StrictScanner ( ) [override]
```

### 10.296.4 Member Function Documentation

#### 10.296.4.1 AddPrivateTag()

```
void gdc::StrictScanner::AddPrivateTag (
    PrivateTag const & t )
```

#### 10.296.4.2 AddSkipTag()

```
void gdc::StrictScanner::AddSkipTag (
    Tag const & t )
```

Add a tag that will need to be skipped. Those are root level skip tags.

#### 10.296.4.3 AddTag()

```
void gdc::StrictScanner::AddTag (
    Tag const & t )
```

Add a tag that will need to be read. Those are root level skip tags.

#### Examples

[ScanDirectory.cs](#), and [SimpleScanner.cxx](#).

#### 10.296.4.4 Begin()

```
ConstIterator gdcm::StrictScanner::Begin ( ) const [inline]
```

#### 10.296.4.5 ClearSkipTags()

```
void gdcm::StrictScanner::ClearSkipTags ( )
```

#### 10.296.4.6 ClearTags()

```
void gdcm::StrictScanner::ClearTags ( )
```

#### 10.296.4.7 End()

```
ConstIterator gdcm::StrictScanner::End ( ) const [inline]
```

#### 10.296.4.8 GetAllFileNamesFromTagToValue()

```
Directory::FileNamesType gdcm::StrictScanner::GetAllFileNamesFromTagToValue (
    Tag const & t,
    const char * valueref ) const
```

Will loop over all files and return a vector of std::strings of filenames where value match the reference value 'valueref'

#### 10.296.4.9 GetFilenameFromTagToValue()

```
const char * gdcm::StrictScanner::GetFilenameFromTagToValue (
    Tag const & t,
    const char * valueref ) const
```

Will loop over all files and return the first file where value match the reference value 'valueref'

#### 10.296.4.10 GetFileNames()

```
Directory::FileNamesType const & gdcm::StrictScanner::GetFileNames ( ) const [inline]
```

#### 10.296.4.11 GetKeys()

```
Directory::FileNamesType gdcm::StrictScanner::GetKeys ( ) const
```

Return the list of filename that are key in the internal map, which means those filename were properly parsed

#### 10.296.4.12 GetMapping()

```
TagToValue const & gdcm::StrictScanner::GetMapping (
    const char * filename ) const
```

Get the std::map mapping filenames to value for file 'filename'.

##### Examples

[SimpleScanner.cxx](#).

#### 10.296.4.13 GetMappingFromTagToValue()

```
TagToValue const & gdcm::StrictScanner::GetMappingFromTagToValue (
    Tag const & t,
    const char * value ) const
```

See [GetFilenameFromTagToValue\(\)](#). This is simply GetFilenameFromTagToValue followed.

#### 10.296.4.14 GetMappings()

```
MappingType const & gdcm::StrictScanner::GetMappings ( ) const [inline]
```

Mappings are the mapping from a particular tag to the map, mapping filename to value:

#### 10.296.4.15 GetOrderedValues()

```
Directory::FileNamesType gdcm::StrictScanner::GetOrderedValues (
    Tag const & t ) const
```

Get all the values found (in a vector) associated with [Tag](#) 't' This function is identical to GetValues, but is accessible from the wrapped layer (python, C#, java)

#### 10.296.4.16 GetValue()

```
const char * gdcm::StrictScanner::GetValue (
    const char * filename,
    Tag const & t ) const
```

Retrieve the value found for tag: t associated with file: filename This is meant for a single short call. If multiple calls (multiple tags) should be done, prefer the GetMapping function, and then reuse the TagToValue hash table.

##### Warning

[Tag](#) 't' should have been added via [AddTag\(\)](#) prior to the [Scan\(\)](#) call !



**10.296.4.17 GetValues() [1/2]**

```
ValueType const & gdcm::StrictScanner::GetValues ( ) const [inline]
```

Get all the values found (in lexicographic order)

**10.296.4.18 GetValues() [2/2]**

```
ValueType gdcm::StrictScanner::GetValues (
    Tag const & t ) const
```

Get all the values found (in lexicographic order) associated with Tag 't'.

**10.296.4.19 IsKey()**

```
bool gdcm::StrictScanner::IsKey (
    const char * filename ) const
```

Check if filename is a key in the Mapping table. returns true only if file can be found, which means the file was indeed a DICOM file that could be processed

**Examples**

[ScanDirectory.cs](#), and [SimpleScanner.cxx](#).

**10.296.4.20 New()**

```
static SmartPointer< StrictScanner > gdcm::StrictScanner::New ( ) [inline], [static]
```

for wrapped language: instantiate a reference counted object

**Examples**

[ScanDirectory.cs](#).

**10.296.4.21 Print()**

```
void gdcm::StrictScanner::Print (
    std::ostream & os ) const [override], [virtual]
```

Print result.

Reimplemented from [gdcm::Object](#).

#### 10.296.4.22 PrintTable()

```
void gdcM::StrictScanner::PrintTable (
    std::ostream & os ) const
```

#### 10.296.4.23 ProcessPublicTag()

```
void gdcM::StrictScanner::ProcessPublicTag (
    StringFilter & sf,
    const char * filename ) [protected]
```

#### 10.296.4.24 Scan()

```
bool gdcM::StrictScanner::Scan (
    Directory::FileNamesType const & filenames )
```

Start the scan !

#### Examples

[ScanDirectory.cs](#), and [SimpleScanner.cxx](#).

### 10.296.5 Friends And Related Symbol Documentation

#### 10.296.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const StrictScanner & s ) [friend]
```

The documentation for this class was generated from the following file:

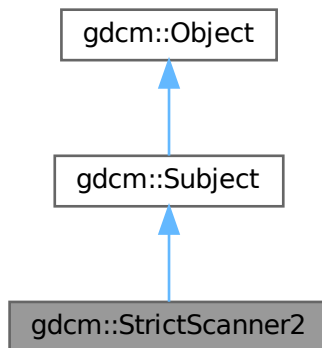
- [gdcMStrictScanner.h](#)

## 10.297 gdcm::StrictScanner2 Class Reference

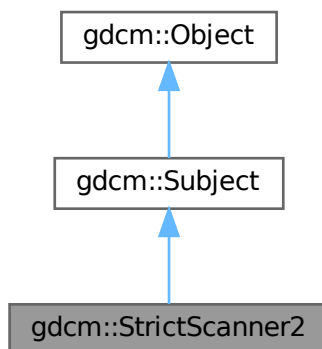
[StrictScanner2](#).

```
#include <gdcmStrictScanner2.h>
```

Inheritance diagram for gdcm::StrictScanner2:



Collaboration diagram for gdcm::StrictScanner2:



### Classes

- struct [ltstr](#)

## Public Types

- typedef PrivateMappingType::const\_iterator [PrivateConstIterator](#)
- typedef std::map< const char \*, [PrivateTagToValue](#), Itstr > [PrivateMappingType](#)
- typedef std::map< [PrivateTag](#), const char \* > [PrivateTagToValue](#)
- typedef PrivateTagToValue::value\_type [PrivateTagToValueValueType](#)
- typedef PublicMappingType::const\_iterator [PublicConstIterator](#)
- typedef std::map< const char \*, [PublicTagToValue](#), Itstr > [PublicMappingType](#)
- typedef std::map< [Tag](#), const char \* > [PublicTagToValue](#)
- typedef PublicTagToValue::value\_type [PublicTagToValueValueType](#)
- typedef std::set< std::string > [ValuesType](#)

## Public Member Functions

- [StrictScanner2](#) ()
- [~StrictScanner2](#) () override
- bool [AddPrivateTag](#) ([PrivateTag](#) const &pt)
- bool [AddPublicTag](#) ([Tag](#) const &t)  
*Add a tag that will need to be read. Those are root level tags.*
- bool [AddSkipTag](#) ([Tag](#) const &t)  
*Add a tag that will need to be skipped. Those are root level skip tags.*
- [PublicConstIterator](#) [Begin](#) () const
- void [ClearPrivateTags](#) ()
- void [ClearPublicTags](#) ()
- void [ClearSkipTags](#) ()
- [PublicConstIterator](#) [End](#) () const
- [Directory::FilenameType](#) [GetAllFilenamesFromPrivateTagToValue](#) ([PrivateTag](#) const &pt, const char \*valueref) const
- [Directory::FilenameType](#) [GetAllFilenamesFromPublicTagToValue](#) ([Tag](#) const &t, const char \*valueref) const
- const char \* [GetFilenameFromPrivateTagToValue](#) ([PrivateTag](#) const &pt, const char \*valueref) const
- const char \* [GetFilenameFromPublicTagToValue](#) ([Tag](#) const &t, const char \*valueref) const
- [Directory::FilenameType](#) const & [GetFilenames](#) () const  
*Return the list of filenames.*
- [Directory::FilenameType](#) [GetKeys](#) () const
- [PrivateTagToValue](#) const & [GetMappingFromPrivateTagToValue](#) ([PrivateTag](#) const &pt, const char \*value) const
- [PublicTagToValue](#) const & [GetMappingFromPublicTagToValue](#) ([Tag](#) const &t, const char \*value) const
- [PrivateTagToValue](#) const & [GetPrivateMapping](#) (const char \*filename) const
- [PrivateMappingType](#) const & [GetPrivateMappings](#) () const
- [Directory::FilenameType](#) [GetPrivateOrderedValues](#) ([PrivateTag](#) const &pt) const
- const char \* [GetPrivateValue](#) (const char \*filename, [PrivateTag](#) const &t) const
- [ValuesType](#) [GetPrivateValues](#) ([PrivateTag](#) const &pt) const
- [PublicTagToValue](#) const & [GetPublicMapping](#) (const char \*filename) const  
*Get the std::map mapping filenames to value for file 'filename'.*
- [PublicMappingType](#) const & [GetPublicMappings](#) () const
- [Directory::FilenameType](#) [GetPublicOrderedValues](#) ([Tag](#) const &t) const
- const char \* [GetPublicValue](#) (const char \*filename, [Tag](#) const &t) const
- [ValuesType](#) [GetPublicValues](#) ([Tag](#) const &t) const  
*Get all the values found (in lexicographic order) associated with [Tag](#) 't'.*
- [ValuesType](#) const & [GetValues](#) () const

*Get all the values found (in lexicographic order)*

- bool [IsKey](#) (const char \*filename) const
- void [Print](#) (std::ostream &os) const override

*Print result.*

- void [PrintTable](#) (std::ostream &os, bool header=false) const

*Print result as CSV table.*

- [PrivateConstIterator PrivateBegin](#) () const
- [PrivateConstIterator PrivateEnd](#) () const
- bool [Scan](#) ([Directory::FileNamesType](#) const &filenames)

*Start the scan !*

## Public Member Functions inherited from [gdcm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*) const
- [Command](#) \* [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
  - [Object](#) (const [Object](#) &)
- Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
  - void [operator=](#) (const [Object](#) &)

## Static Public Member Functions

- static [SmartPointer](#)< [StrictScanner2](#) > [New](#) ()
- for wrapped language: instantiate a reference counted object*

## Protected Member Functions

- void [ProcessPrivateTag](#) ([StringFilter](#) &sf, const char \*filename)
- void [ProcessPublicTag](#) ([StringFilter](#) &sf, const char \*filename)

## Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

## Friends

- `std::ostream & operator<< (std::ostream &_os, const StrictScanner2 &s)`

## 10.297.1 Detailed Description

### [StrictScanner2](#).

This filter is meant for quickly browsing a [FileSet](#) (a set of files on disk). Special consideration are taken so as to read the minimum amount of information in each file in order to retrieve the user specified set of DICOM [Attribute](#).

This filter is dealing with both VRASCII and VRBINARY element, thanks to the help of [StringFilter](#)

### Warning

IMPORTANT In case of file where tags are not ordered (illegal as per DICOM specification), the output will be missing information

### Note

implementation details. All values are stored in a `std::set of std::string`. Then the address of the `cstring` underlying the `std::string` is used in the `std::map`.

This class implement the Subject/Observer pattern trigger the following events:

- [ProgressEvent](#)
- [StartEvent](#)
- [EndEvent](#)

## 10.297.2 Member Typedef Documentation

### 10.297.2.1 PrivateConstIterator

```
typedef PrivateMappingType::const_iterator gdcm::StrictScanner2::PrivateConstIterator
```

### 10.297.2.2 PrivateMappingType

```
typedef std::map<const char *, PrivateTagToValue, ltstr> gdcm::StrictScanner2::PrivateMappingType
```

### 10.297.2.3 PrivateTagToValue

```
typedef std::map<PrivateTag, const char *> gdcm::StrictScanner2::PrivateTagToValue
```

#### 10.297.2.4 PrivateTagToValueValueType

```
typedef PrivateTagToValue::value_type gdcm::StrictScanner2::PrivateTagToValueValueType
```

#### 10.297.2.5 PublicConstIterator

```
typedef PublicMappingType::const_iterator gdcm::StrictScanner2::PublicConstIterator
```

#### 10.297.2.6 PublicMappingType

```
typedef std::map<const char *, PublicTagToValue, ltstr> gdcm::StrictScanner2::PublicMappingType
```

#### 10.297.2.7 PublicTagToValue

```
typedef std::map<Tag, const char *> gdcm::StrictScanner2::PublicTagToValue
```

struct to map a filename to a value Implementation note: all std::map in this class will be using const char \* and not std::string since we are pointing to existing std::string (held in a std::vector) this avoid an extra copy of the byte array. Tag are used as Tag class since sizeof(tag) <= sizeof(pointer)

#### 10.297.2.8 PublicTagToValueValueType

```
typedef PublicTagToValue::value_type gdcm::StrictScanner2::PublicTagToValueValueType
```

#### 10.297.2.9 ValueType

```
typedef std::set<std::string> gdcm::StrictScanner2::ValueType
```

### 10.297.3 Constructor & Destructor Documentation

#### 10.297.3.1 StrictScanner2()

```
gdcm::StrictScanner2::StrictScanner2 ( ) [inline]
```

#### 10.297.3.2 ~StrictScanner2()

```
gdcm::StrictScanner2::~~StrictScanner2 ( ) [override]
```

## 10.297.4 Member Function Documentation

### 10.297.4.1 AddPrivateTag()

```
bool gdcM::StrictScanner2::AddPrivateTag (
    PrivateTag const & pt )
```

### 10.297.4.2 AddPublicTag()

```
bool gdcM::StrictScanner2::AddPublicTag (
    Tag const & t )
```

Add a tag that will need to be read. Those are root level tags.

### 10.297.4.3 AddSkipTag()

```
bool gdcM::StrictScanner2::AddSkipTag (
    Tag const & t )
```

Add a tag that will need to be skipped. Those are root level skip tags.

### 10.297.4.4 Begin()

```
PublicConstIterator gdcM::StrictScanner2::Begin ( ) const [inline]
```

### 10.297.4.5 ClearPrivateTags()

```
void gdcM::StrictScanner2::ClearPrivateTags ( )
```

### 10.297.4.6 ClearPublicTags()

```
void gdcM::StrictScanner2::ClearPublicTags ( )
```

### 10.297.4.7 ClearSkipTags()

```
void gdcM::StrictScanner2::ClearSkipTags ( )
```

### 10.297.4.8 End()

```
PublicConstIterator gdcM::StrictScanner2::End ( ) const [inline]
```



#### 10.297.4.9 GetAllFileNamesFromPrivateTagToValue()

```
Directory::FileNamesType gdcm::StrictScanner2::GetAllFileNamesFromPrivateTagToValue (
    PrivateTag const & pt,
    const char * valueref ) const
```

#### 10.297.4.10 GetAllFileNamesFromPublicTagToValue()

```
Directory::FileNamesType gdcm::StrictScanner2::GetAllFileNamesFromPublicTagToValue (
    Tag const & t,
    const char * valueref ) const
```

Will loop over all files and return a vector of std::strings of filenames where value match the reference value 'valueref'

#### 10.297.4.11 GetFilenameFromPrivateTagToValue()

```
const char * gdcm::StrictScanner2::GetFilenameFromPrivateTagToValue (
    PrivateTag const & pt,
    const char * valueref ) const
```

#### 10.297.4.12 GetFilenameFromPublicTagToValue()

```
const char * gdcm::StrictScanner2::GetFilenameFromPublicTagToValue (
    Tag const & t,
    const char * valueref ) const
```

Will loop over all files and return the first file where value match the reference value 'valueref'

#### 10.297.4.13 GetFileNames()

```
Directory::FileNamesType const & gdcm::StrictScanner2::GetFileNames ( ) const [inline]
```

Return the list of filenames.

#### 10.297.4.14 GetKeys()

```
Directory::FileNamesType gdcm::StrictScanner2::GetKeys ( ) const
```

Return the list of filename that are key in the internal map, which means those filename were properly parsed

#### 10.297.4.15 GetMappingFromPrivateTagToValue()

```
PrivateTagToValue const & gdcm::StrictScanner2::GetMappingFromPrivateTagToValue (
    PrivateTag const & pt,
    const char * value ) const
```

**10.297.4.16 GetMappingFromPublicTagToValue()**

```
PublicTagToValue const & gdcM::StrictScanner2::GetMappingFromPublicTagToValue (
    Tag const & t,
    const char * value ) const
```

See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed

**10.297.4.17 GetPrivateMapping()**

```
PrivateTagToValue const & gdcM::StrictScanner2::GetPrivateMapping (
    const char * filename ) const
```

**10.297.4.18 GetPrivateMappings()**

```
PrivateMappingType const & gdcM::StrictScanner2::GetPrivateMappings ( ) const [inline]
```

**10.297.4.19 GetPrivateOrderedValues()**

```
Directory::FileNamesType gdcM::StrictScanner2::GetPrivateOrderedValues (
    PrivateTag const & pt ) const
```

**10.297.4.20 GetPrivateValue()**

```
const char * gdcM::StrictScanner2::GetPrivateValue (
    const char * filename,
    PrivateTag const & t ) const
```

**10.297.4.21 GetPrivateValues()**

```
ValuesType gdcM::StrictScanner2::GetPrivateValues (
    PrivateTag const & pt ) const
```

Get all the values found (in lexicographic order) associated with [PrivateTag](#) 'pt'

**10.297.4.22 GetPublicMapping()**

```
PublicTagToValue const & gdcM::StrictScanner2::GetPublicMapping (
    const char * filename ) const
```

Get the std::map mapping filenames to value for file 'filename'.

#### 10.297.4.23 GetPublicMappings()

```
PublicMappingType const & gdcm::StrictScanner2::GetPublicMappings ( ) const [inline]
```

Mappings are the mapping from a particular tag to the map, mapping filename to value:

#### 10.297.4.24 GetPublicOrderedValues()

```
Directory::FileNamesType gdcm::StrictScanner2::GetPublicOrderedValues (
    Tag const & t ) const
```

Get all the values found (in a vector) associated with Tag 't' This function is identical to GetValues, but is accessible from the wrapped layer (python, C#, java)

#### 10.297.4.25 GetPublicValue()

```
const char * gdcm::StrictScanner2::GetPublicValue (
    const char * filename,
    Tag const & t ) const
```

Retrieve the value found for tag: t associated with file: filename This is meant for a single short call. If multiple calls (multiple tags) should be done, prefer the GetMapping function, and then reuse the TagToValue hash table.

#### Warning

Tag 't' should have been added via AddTag() prior to the Scan() call !

#### 10.297.4.26 GetPublicValues()

```
ValuesType gdcm::StrictScanner2::GetPublicValues (
    Tag const & t ) const
```

Get all the values found (in lexicographic order) associated with Tag 't'.

#### 10.297.4.27 GetValues()

```
ValuesType const & gdcm::StrictScanner2::GetValues ( ) const [inline]
```

Get all the values found (in lexicographic order)

#### 10.297.4.28 IsKey()

```
bool gdcm::StrictScanner2::IsKey (
    const char * filename ) const
```

Check if filename is a key in the Mapping table. returns true only if file can be found, which means the file was indeed a DICOM file that could be processed

**10.297.4.29 New()**

```
static SmartPointer< StrictScanner2 > gdcM::StrictScanner2::New ( ) [inline], [static]
```

for wrapped language: instantiate a reference counted object

**10.297.4.30 Print()**

```
void gdcM::StrictScanner2::Print (
    std::ostream & os ) const [override], [virtual]
```

Print result.

Reimplemented from [gdcM::Object](#).

**10.297.4.31 PrintTable()**

```
void gdcM::StrictScanner2::PrintTable (
    std::ostream & os,
    bool header = false ) const
```

Print result as CSV table.

**10.297.4.32 PrivateBegin()**

```
PrivateConstIterator gdcM::StrictScanner2::PrivateBegin ( ) const [inline]
```

**10.297.4.33 PrivateEnd()**

```
PrivateConstIterator gdcM::StrictScanner2::PrivateEnd ( ) const [inline]
```

**10.297.4.34 ProcessPrivateTag()**

```
void gdcM::StrictScanner2::ProcessPrivateTag (
    StringFilter & sf,
    const char * filename ) [protected]
```

**10.297.4.35 ProcessPublicTag()**

```
void gdcM::StrictScanner2::ProcessPublicTag (
    StringFilter & sf,
    const char * filename ) [protected]
```

#### 10.297.4.36 Scan()

```
bool gdcm::StrictScanner2::Scan (
    Directory::FilenameType const & filenames )
```

Start the scan !

### 10.297.5 Friends And Related Symbol Documentation

#### 10.297.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const StrictScanner2 & s ) [friend]
```

The documentation for this class was generated from the following file:

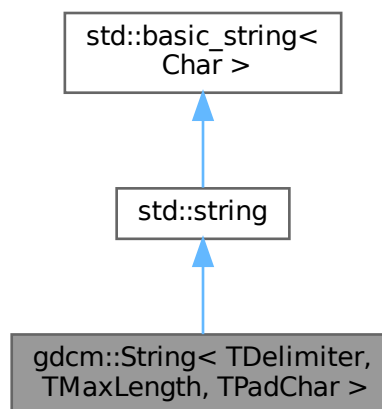
- [gdcmStrictScanner2.h](#)

## 10.298 gdcm::String< TDelimiter, TMaxLength, TPadChar > Class Template Reference

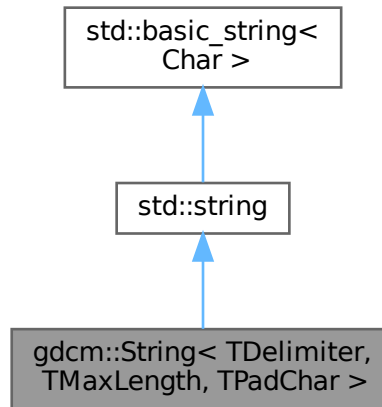
[String.](#)

```
#include <gdcmString.h>
```

Inheritance diagram for gdcm::String< TDelimiter, TMaxLength, TPadChar >:



Collaboration diagram for `gdcm::String< TDelimiter, TMaxLength, TPadChar >`:



## Public Types

- typedef `std::string::const_iterator` [const\\_iterator](#)
- typedef `std::string::const_reference` [const\\_reference](#)
- typedef `std::string::const_reverse_iterator` [const\\_reverse\\_iterator](#)
- typedef `std::string::difference_type` [difference\\_type](#)
- typedef `std::string::iterator` [iterator](#)
- typedef `std::string::pointer` [pointer](#)
- typedef `std::string::reference` [reference](#)
- typedef `std::string::reverse_iterator` [reverse\\_iterator](#)
- typedef `std::string::size_type` [size\\_type](#)
- typedef `std::string::value_type` [value\\_type](#)

## Public Member Functions

- [String](#) ()  
*String constructors.*
- [String](#) (const `std::string` &s, [size\\_type](#) pos=0, [size\\_type](#) n=npow)
- [String](#) (const [value\\_type](#) \*s)
- [String](#) (const [value\\_type](#) \*s, [size\\_type](#) n)
- bool [IsValid](#) () const  
*return if string is valid*
- [operator const char \\*](#) () const  
*WARNING: Trailing \0 might be lost in this operation:*
- `std::string` [Trim](#) () const
- [gdcm::String](#)< `TDelimiter`, `TMaxLength`, `TPadChar` > [Truncate](#) () const

## Static Public Member Functions

- static std::string [Trim](#) (const char \*input)

### 10.298.1 Detailed Description

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
class gdcm::String< TDelimiter, TMaxLength, TPadChar >
```

[String](#).

#### Note

TDelimiter template parameter is used to separate multiple [String](#) (VM1 >) TMaxLength is only a hint. No one actually respect the max length TPadChar is the string padding (0 or space)

#### Examples

[TemplateEmptyImage.cxx](#).

### 10.298.2 Member Typedef Documentation

#### 10.298.2.1 const\_iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::const_iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::const_↵
iterator
```

#### 10.298.2.2 const\_reference

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::const_reference gdcm::String< TDelimiter, TMaxLength, TPadChar >::const_↵
reference
```

#### 10.298.2.3 const\_reverse\_iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::const_reverse_iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >↵
::const_reverse_iterator
```

#### 10.298.2.4 difference\_type

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::difference_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::difference↵
_type
```

### 10.298.2.5 iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::iterator
```

### 10.298.2.6 pointer

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::pointer gdcm::String< TDelimiter, TMaxLength, TPadChar >::pointer
```

### 10.298.2.7 reference

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::reference gdcm::String< TDelimiter, TMaxLength, TPadChar >::reference
```

### 10.298.2.8 reverse\_iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::reverse_iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::reverse↵
_iterator
```

### 10.298.2.9 size\_type

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::size_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::size_type
```

### 10.298.2.10 value\_type

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::value_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::value_type
```

## 10.298.3 Constructor & Destructor Documentation

### 10.298.3.1 String() [1/4]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::String ( ) [inline]
```

[String](#) constructors.



### 10.298.3.2 String() [2/4]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (
    const value_type * s ) [inline]
```

### 10.298.3.3 String() [3/4]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (
    const value_type * s,
    size_type n ) [inline]
```

### 10.298.3.4 String() [4/4]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (
    const std::string & s,
    size_type pos = 0,
    size_type n = npos ) [inline]
```

## 10.298.4 Member Function Documentation

### 10.298.4.1 IsValid()

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
bool gdcm::String< TDelimiter, TMaxLength, TPadChar >::IsValid ( ) const [inline]
```

return if string is valid

Referenced by [gdcm::LO::IsValid\(\)](#), and [gdcm::String< TDelimiter, TMaxLength, TPadChar >::Truncate\(\)](#).

### 10.298.4.2 operator const char \*()

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::operator const char * ( ) const [inline]
```

WARNING: Trailing \0 might be lost in this operation:

#### 10.298.4.3 Trim() [1/2]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
std::string gdcm::String< TDelimiter, TMaxLength, TPadChar >::Trim ( ) const [inline]
```

Trim function is required to return a std::string object, otherwise we could not create a [gdcm::String](#) object with an odd number of bytes...

#### Examples

[DumpExamCard.cxx](#).

#### 10.298.4.4 Trim() [2/2]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
static std::string gdcm::String< TDelimiter, TMaxLength, TPadChar >::Trim (
    const char * input ) [inline], [static]
```

#### 10.298.4.5 Truncate()

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar > gdcm::String< TDelimiter, TMaxLength, TPadChar
>::Truncate ( ) const [inline]
```

References [gdcm::String< TDelimiter, TMaxLength, TPadChar >::IsValid\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmString.h](#)

## 10.299 gdcm::StringFilter Class Reference

[StringFilter](#).

```
#include <gdcmStringFilter.h>
```

## Public Member Functions

- [StringFilter](#) ()
- [~StringFilter](#) ()
- bool [ExecuteQuery](#) (std::string const &query, std::string &value) const
- std::string [FromString](#) (const [Tag](#) &t, const char \*value, size\_t len)  
*Convert to string the char array defined by the pair (value,len)*
- [File](#) & [GetFile](#) ()
- const [File](#) & [GetFile](#) () const
- void [SetDicts](#) (const [Dicts](#) &dicts)  
*Allow user to pass in there own dicts.*
- void [SetFile](#) (const [File](#) &f)  
*Set/Get File.*
- std::string [ToString](#) (const [DataElement](#) &de) const
- std::string [ToString](#) (const [PrivateTag](#) &t) const
- std::string [ToString](#) (const [Tag](#) &t) const  
*Directly from a Tag:*
- std::pair< std::string, std::string > [ToStringPair](#) (const [DataElement](#) &de) const
- std::pair< std::string, std::string > [ToStringPair](#) (const [Tag](#) &t) const  
*Directly from a Tag:*
- void [UseDictAlways](#) (bool)

## Protected Member Functions

- bool [ExecuteQuery](#) (std::string const &query, [DataSet](#) const &ds, std::string &value) const
- std::pair< std::string, std::string > [ToStringPair](#) (const [Tag](#) &t, [DataSet](#) const &ds) const

### 10.299.1 Detailed Description

[StringFilter](#).

[StringFilter](#) is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a [DataElement](#) into a string, typically this is a nice feature to have for wrapped language

#### Examples

[DumpVisusChange.cxx](#), [ReadAndPrintAttributes.cxx](#), and [SimplePrintPatientName.cs](#).

### 10.299.2 Constructor & Destructor Documentation

#### 10.299.2.1 StringFilter()

```
gdcm::StringFilter::StringFilter ( )
```

**10.299.2.2 ~StringFilter()**

```
gdcmm::StringFilter::~~StringFilter ( )
```

**10.299.3 Member Function Documentation****10.299.3.1 ExecuteQuery() [1/2]**

```
bool gdcmm::StringFilter::ExecuteQuery (
    std::string const & query,
    DataSet const & ds,
    std::string & value ) const [protected]
```

**10.299.3.2 ExecuteQuery() [2/2]**

```
bool gdcmm::StringFilter::ExecuteQuery (
    std::string const & query,
    std::string & value ) const
```

Execute the XPATH query to find a value (as string) return false when attribute is not found (or an error in the XPATH query) You need to make sure that your XPATH query is syntactically correct

**10.299.3.3 FromString()**

```
std::string gdcmm::StringFilter::FromString (
    const Tag & t,
    const char * value,
    size_t len )
```

Convert to string the char array defined by the pair (value,len)

**10.299.3.4 GetFile() [1/2]**

```
File & gdcmm::StringFilter::GetFile ( ) [inline]
```

**10.299.3.5 GetFile() [2/2]**

```
const File & gdcmm::StringFilter::GetFile ( ) const [inline]
```

**10.299.3.6 SetDicts()**

```
void gdcmm::StringFilter::SetDicts (
    const Dicts & dicts )
```

Allow user to pass in there own dicts.

### 10.299.3.7 SetFile()

```
void gdcm::StringFilter::SetFile (
    const File & f ) [inline]
```

Set/Get [File](#).

#### Examples

[DumpVisusChange.cxx](#), [ReadAndPrintAttributes.cxx](#), and [SimplePrintPatientName.cs](#).

### 10.299.3.8 ToString() [1/3]

```
std::string gdcm::StringFilter::ToString (
    const DataElement & de ) const
```

Convert to string the [ByteValue](#) contained in a [DataElement](#). The [DataElement](#) must be coming from the actual [DataSet](#) associated with [File](#) (see [SetFile](#)).

#### Examples

[DumpVisusChange.cxx](#), [ReadAndPrintAttributes.cxx](#), and [SimplePrintPatientName.cs](#).

### 10.299.3.9 ToString() [2/3]

```
std::string gdcm::StringFilter::ToString (
    const PrivateTag & t ) const
```

### 10.299.3.10 ToString() [3/3]

```
std::string gdcm::StringFilter::ToString (
    const Tag & t ) const
```

Directly from a [Tag](#):

### 10.299.3.11 ToStringPair() [1/3]

```
std::pair< std::string, std::string > gdcm::StringFilter::ToStringPair (
    const DataElement & de ) const
```

Convert to string the [ByteValue](#) contained in a [DataElement](#) the returned elements are: pair.first : the name as found in the dictionary of [DataElement](#) pari.second : the value encoded into a string (US,UL...) are properly converted

#### Examples

[ReadAndPrintAttributes.cxx](#).

**10.299.3.12 ToStringPair() [2/3]**

```
std::pair< std::string, std::string > gdc::StringFilter::ToStringPair (
    const Tag & t ) const
```

Directly from a [Tag](#):

**10.299.3.13 ToStringPair() [3/3]**

```
std::pair< std::string, std::string > gdc::StringFilter::ToStringPair (
    const Tag & t,
    DataSet const & ds ) const [protected]
```

**10.299.3.14 UseDictAlways()**

```
void gdc::StringFilter::UseDictAlways (
    bool ) [inline]
```

The documentation for this class was generated from the following file:

- [gdcStringFilter.h](#)

## 10.300 gdc::Study Class Reference

[Study](#).

```
#include <gdcStudy.h>
```

**Public Member Functions**

- [Study](#) ()=default

**10.300.1 Detailed Description**

[Study](#).

**10.300.2 Constructor & Destructor Documentation****10.300.2.1 Study()**

```
gdc::Study::Study ( ) [default]
```

The documentation for this class was generated from the following file:

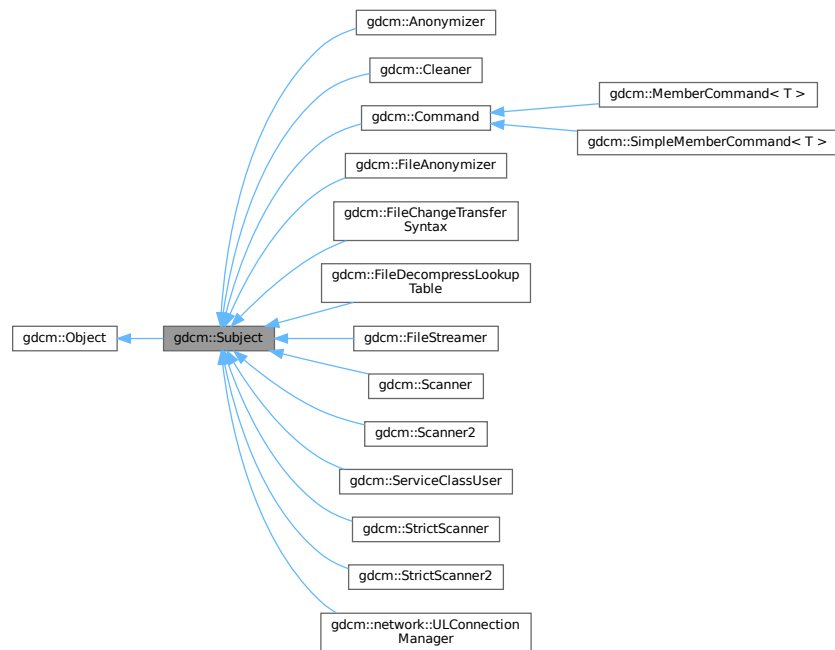
- [gdcStudy.h](#)

## 10.301 gdcm::Subject Class Reference

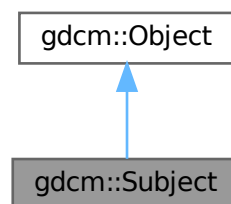
[Subject.](#)

```
#include <gdcmSubject.h>
```

Inheritance diagram for gdcm::Subject:



Collaboration diagram for gdcm::Subject:



**Public Member Functions**

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*) const
- [Command](#) \* [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

**Public Member Functions inherited from [gdcm::Object](#)**

- [Object](#) ()
- [Object](#) (const [Object](#) &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

**Additional Inherited Members****Protected Member Functions inherited from [gdcm::Object](#)**

- void [Register](#) ()
- void [UnRegister](#) ()

**10.301.1 Detailed Description**

[Subject](#).

See also

[Command Event](#)

Examples

[BasicAnonymizer.cs](#), [Cleaner.cs](#), [ClinicalTrialIdentificationWorkflow.cs](#), [ScanDirectory.cs](#), and [SimpleScanner.cxx](#).

**10.301.2 Constructor & Destructor Documentation****10.301.2.1 Subject()**

```
gdcm::Subject::Subject ( )
```



### 10.301.2.2 ~Subject()

```
gdcmm::Subject::~~Subject ( ) [override]
```

## 10.301.3 Member Function Documentation

### 10.301.3.1 AddObserver() [1/2]

```
unsigned long gdcmm::Subject::AddObserver (
    const Event & event,
    Command * )
```

Allow people to add/remove/invoke observers (callbacks) to any GDCM object. This is an implementation of the subject/observer design pattern. An observer is added by specifying an event to respond to and an [gdcmm::Command](#) to execute. It returns an unsigned long tag which can be used later to remove the event or retrieve the command. The memory for the [Command](#) becomes the responsibility of this object, so don't pass the same instance of a command to two different objects

### 10.301.3.2 AddObserver() [2/2]

```
unsigned long gdcmm::Subject::AddObserver (
    const Event & event,
    Command * ) const
```

### 10.301.3.3 GetCommand()

```
Command * gdcmm::Subject::GetCommand (
    unsigned long tag )
```

Get the command associated with the given tag. NOTE: This returns a pointer to a [Command](#), but it is safe to assign this to a [Command::Pointer](#). Since [Command](#) inherits from [LightObject](#), at this point in the code, only a pointer or a reference to the [Command](#) can be used.

### 10.301.3.4 HasObserver()

```
bool gdcmm::Subject::HasObserver (
    const Event & event ) const
```

Return true if an observer is registered for this event.

**10.301.3.5 InvokeEvent() [1/2]**

```
void gdcM::Subject::InvokeEvent (
    const Event & )
```

Call Execute on all the Commands observing this event id.

**10.301.3.6 InvokeEvent() [2/2]**

```
void gdcM::Subject::InvokeEvent (
    const Event & ) const
```

Call Execute on all the Commands observing this event id. The actions triggered by this call doesn't modify this object.

**10.301.3.7 RemoveAllObservers()**

```
void gdcM::Subject::RemoveAllObservers ( )
```

Remove all observers .

**10.301.3.8 RemoveObserver()**

```
void gdcM::Subject::RemoveObserver (
    unsigned long tag )
```

Remove the observer with this tag value.

The documentation for this class was generated from the following file:

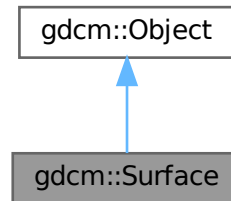
- [gdcMSubject.h](#)

## 10.302 gdcm::Surface Class Reference

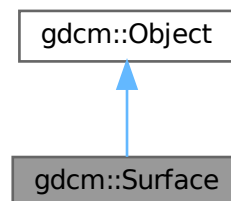
This class defines a SURFACE IE.

```
#include <gdcmSurface.h>
```

Inheritance diagram for gdcm::Surface:



Collaboration diagram for gdcm::Surface:



### Public Types

- enum [STATES](#) {  
    [NO](#) = 0 ,  
    [YES](#) ,  
    [UNKNOWN](#) ,  
    [STATES\\_END](#) }
- enum [VIEWType](#) {  
    [SURFACE](#) = 0 ,  
    [WIREFRAME](#) ,  
    [POINTS](#) ,  
    [VIEWType\\_END](#) }

*Enumeration for Recommended Presentation [Type](#).*

## Public Member Functions

- [Surface](#) ()
- [~Surface](#) () override
- [SegmentHelper::BasicCodedEntry](#) & [GetAlgorithmFamily](#) ()
- [SegmentHelper::BasicCodedEntry](#) const & [GetAlgorithmFamily](#) () const
- const char \* [GetAlgorithmName](#) () const
- const char \* [GetAlgorithmVersion](#) () const
- const float \* [GetAxisOfRotation](#) () const
- const float \* [GetCenterOfRotation](#) () const
- [STATES](#) [GetFiniteVolume](#) () const
- [STATES](#) [GetManifold](#) () const
- float [GetMaximumPointDistance](#) () const
- float [GetMeanPointDistance](#) () const
- [MeshPrimitive](#) & [GetMeshPrimitive](#) ()
- [MeshPrimitive](#) const & [GetMeshPrimitive](#) () const
- unsigned long [GetNumberOfSurfacePoints](#) () const
- unsigned long [GetNumberOfVectors](#) () const
- [DataElement](#) & [GetPointCoordinatesData](#) ()
- const [DataElement](#) & [GetPointCoordinatesData](#) () const
- const float \* [GetPointPositionAccuracy](#) () const
- const float \* [GetPointsBoundingBoxCoordinates](#) () const
- [SegmentHelper::BasicCodedEntry](#) & [GetProcessingAlgorithm](#) ()
- [SegmentHelper::BasicCodedEntry](#) const & [GetProcessingAlgorithm](#) () const
- const unsigned short \* [GetRecommendedDisplayCIELabValue](#) () const
- unsigned short [GetRecommendedDisplayCIELabValue](#) (const unsigned int idx) const
- unsigned short [GetRecommendedDisplayGrayscaleValue](#) () const
- float [GetRecommendedPresentationOpacity](#) () const
- [VIEWType](#) [GetRecommendedPresentationType](#) () const
- const char \* [GetSurfaceComments](#) () const
- unsigned long [GetSurfaceNumber](#) () const
- bool [GetSurfaceProcessing](#) () const
- const char \* [GetSurfaceProcessingDescription](#) () const
- float [GetSurfaceProcessingRatio](#) () const
- const float \* [GetVectorAccuracy](#) () const
- [DataElement](#) & [GetVectorCoordinateData](#) ()
- const [DataElement](#) & [GetVectorCoordinateData](#) () const
- unsigned short [GetVectorDimensionality](#) () const
- void [SetAlgorithmFamily](#) ([SegmentHelper::BasicCodedEntry](#) const &BSE)
- void [SetAlgorithmName](#) (const char \*str)
- void [SetAlgorithmVersion](#) (const char \*str)
- void [SetAxisOfRotation](#) (const float \*axis)
- void [SetCenterOfRotation](#) (const float \*center)
- void [SetFiniteVolume](#) ([STATES](#) state)
- void [SetManifold](#) ([STATES](#) state)
- void [SetMaximumPointDistance](#) (float maximum)
- void [SetMeanPointDistance](#) (float average)
- void [SetMeshPrimitive](#) ([MeshPrimitive](#) &mp)
- void [SetNumberOfSurfacePoints](#) (const unsigned long nb)
- void [SetNumberOfVectors](#) (const unsigned long nb)
- void [SetPointCoordinatesData](#) ([DataElement](#) const &de)

- void [SetPointPositionAccuracy](#) (const float \*accuracies)
- void [SetPointsBoundingBoxCoordinates](#) (const float \*coordinates)
- void [SetProcessingAlgorithm](#) ([SegmentHelper::BasicCodedEntry](#) const &BSE)
- void [SetRecommendedDisplayCIELabValue](#) (const std::vector< unsigned short > &vl)
- void [SetRecommendedDisplayCIELabValue](#) (const unsigned short vl, const unsigned int idx=0)
- void [SetRecommendedDisplayCIELabValue](#) (const unsigned short vl[3])
- void [SetRecommendedDisplayGrayscaleValue](#) (const unsigned short vl)
- void [SetRecommendedPresentationOpacity](#) (const float opacity)
- void [SetRecommendedPresentationType](#) ([VIEWType](#) type)
- void [SetSurfaceComments](#) (const char \*comment)
- void [SetSurfaceNumber](#) (const unsigned long nb)
- void [SetSurfaceProcessing](#) (bool b)
- void [SetSurfaceProcessingDescription](#) (const char \*description)
- void [SetSurfaceProcessingRatio](#) (const float ratio)
- void [SetVectorAccuracy](#) (const float \*accuracy)
- void [SetVectorCoordinateData](#) ([DataElement](#) const &de)
- void [SetVectorDimensionality](#) (const unsigned short dim)

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
- *Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

## Static Public Member Functions

- static [STATES](#) [GetSTATES](#) (const char \*state)
- static const char \* [GetSTATESString](#) ([STATES](#) state)
- static [VIEWType](#) [GetVIEWType](#) (const char \*type)
- static const char \* [GetVIEWTypeString](#) ([VIEWType](#) type)

## Additional Inherited Members

## Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

### 10.302.1 Detailed Description

This class defines a SURFACE IE.

This members are taken from required surface mesh module attributes.

See also

PS 3.3 A.1.2.18 , A.57 and C.27

## 10.302.2 Member Enumeration Documentation

### 10.302.2.1 STATES

enum `gdcm::Surface::STATES`

Enumerator

NO	
YES	
UNKNOWN	
STATES_END	

### 10.302.2.2 VIEWType

enum `gdcm::Surface::VIEWType`

Enumeration for Recommended Presentation [Type](#).

See also

Tag(0x0066, 0x000D) and PS 3.3 C.27.1.1.3

Enumerator

SURFACE	
WIREFRAME	
POINTS	
VIEWType_END	

## 10.302.3 Constructor & Destructor Documentation

### 10.302.3.1 Surface()

`gdcm::Surface::Surface ( )`

### 10.302.3.2 ~Surface()

`gdcm::Surface::~~Surface ( ) [override]`

## 10.302.4 Member Function Documentation

### 10.302.4.1 GetAlgorithmFamily() [1/2]

```
SegmentHelper::BasicCodedEntry & gdcm::Surface::GetAlgorithmFamily ( )
```

### 10.302.4.2 GetAlgorithmFamily() [2/2]

```
SegmentHelper::BasicCodedEntry const & gdcm::Surface::GetAlgorithmFamily ( ) const
```

### 10.302.4.3 GetAlgorithmName()

```
const char * gdcm::Surface::GetAlgorithmName ( ) const
```

### 10.302.4.4 GetAlgorithmVersion()

```
const char * gdcm::Surface::GetAlgorithmVersion ( ) const
```

### 10.302.4.5 GetAxisOfRotation()

```
const float * gdcm::Surface::GetAxisOfRotation ( ) const
```

#### Note

Pointer is null if undefined

### 10.302.4.6 GetCenterOfRotation()

```
const float * gdcm::Surface::GetCenterOfRotation ( ) const
```

#### Note

Pointer is null if undefined

### 10.302.4.7 GetFiniteVolume()

```
STATES gdcm::Surface::GetFiniteVolume ( ) const
```

**10.302.4.8 GetManifold()**

```
STATES gdcM::Surface::GetManifold ( ) const
```

**10.302.4.9 GetMaximumPointDistance()**

```
float gdcM::Surface::GetMaximumPointDistance ( ) const
```

**10.302.4.10 GetMeanPointDistance()**

```
float gdcM::Surface::GetMeanPointDistance ( ) const
```

**10.302.4.11 GetMeshPrimitive() [1/2]**

```
MeshPrimitive & gdcM::Surface::GetMeshPrimitive ( )
```

**10.302.4.12 GetMeshPrimitive() [2/2]**

```
MeshPrimitive const & gdcM::Surface::GetMeshPrimitive ( ) const
```

**10.302.4.13 GetNumberOfSurfacePoints()**

```
unsigned long gdcM::Surface::GetNumberOfSurfacePoints ( ) const
```

**10.302.4.14 GetNumberOfVectors()**

```
unsigned long gdcM::Surface::GetNumberOfVectors ( ) const
```

**10.302.4.15 GetPointCoordinatesData() [1/2]**

```
DataElement & gdcM::Surface::GetPointCoordinatesData ( )
```

**10.302.4.16 GetPointCoordinatesData() [2/2]**

```
const DataElement & gdcM::Surface::GetPointCoordinatesData ( ) const
```



**10.302.4.17 GetPointPositionAccuracy()**

```
const float * gdcm::Surface::GetPointPositionAccuracy ( ) const
```

**Note**

Pointer is null if undefined

**10.302.4.18 GetPointsBoundingBoxCoordinates()**

```
const float * gdcm::Surface::GetPointsBoundingBoxCoordinates ( ) const
```

**Note**

Pointer is null if undefined

**10.302.4.19 GetProcessingAlgorithm() [1/2]**

```
SegmentHelper::BasicCodedEntry & gdcm::Surface::GetProcessingAlgorithm ( )
```

**10.302.4.20 GetProcessingAlgorithm() [2/2]**

```
SegmentHelper::BasicCodedEntry const & gdcm::Surface::GetProcessingAlgorithm ( ) const
```

**10.302.4.21 GetRecommendedDisplayCIELabValue() [1/2]**

```
const unsigned short * gdcm::Surface::GetRecommendedDisplayCIELabValue ( ) const
```

**10.302.4.22 GetRecommendedDisplayCIELabValue() [2/2]**

```
unsigned short gdcm::Surface::GetRecommendedDisplayCIELabValue (
    const unsigned int idx ) const
```

**10.302.4.23 GetRecommendedDisplayGrayscaleValue()**

```
unsigned short gdcm::Surface::GetRecommendedDisplayGrayscaleValue ( ) const
```

**10.302.4.24 GetRecommendedPresentationOpacity()**

```
float gdcm::Surface::GetRecommendedPresentationOpacity ( ) const
```

**10.302.4.25 GetRecommendedPresentationType()**

```
VIEWType gdcm::Surface::GetRecommendedPresentationType ( ) const
```

**10.302.4.26 GetSTATES()**

```
static STATES gdcm::Surface::GetSTATES (
    const char * state ) [static]
```

**10.302.4.27 GetSTATESString()**

```
static const char * gdcm::Surface::GetSTATESString (
    STATES state ) [static]
```

**10.302.4.28 GetSurfaceComments()**

```
const char * gdcm::Surface::GetSurfaceComments ( ) const
```

**10.302.4.29 GetSurfaceNumber()**

```
unsigned long gdcm::Surface::GetSurfaceNumber ( ) const
```

**10.302.4.30 GetSurfaceProcessing()**

```
bool gdcm::Surface::GetSurfaceProcessing ( ) const
```

**10.302.4.31 GetSurfaceProcessingDescription()**

```
const char * gdcm::Surface::GetSurfaceProcessingDescription ( ) const
```

**10.302.4.32 GetSurfaceProcessingRatio()**

```
float gdcm::Surface::GetSurfaceProcessingRatio ( ) const
```

**10.302.4.33 GetVectorAccuracy()**

```
const float * gdcm::Surface::GetVectorAccuracy ( ) const
```

**10.302.4.34 GetVectorCoordinateData() [1/2]**

```
DataElement & gdcm::Surface::GetVectorCoordinateData ( )
```

**10.302.4.35 GetVectorCoordinateData() [2/2]**

```
const DataElement & gdcm::Surface::GetVectorCoordinateData ( ) const
```

**10.302.4.36 GetVectorDimensionality()**

```
unsigned short gdcm::Surface::GetVectorDimensionality ( ) const
```

**10.302.4.37 GetVIEWType()**

```
static VIEWType gdcm::Surface::GetVIEWType (
    const char * type ) [static]
```

**10.302.4.38 GetVIEWTypeString()**

```
static const char * gdcm::Surface::GetVIEWTypeString (
    VIEWType type ) [static]
```

**10.302.4.39 SetAlgorithmFamily()**

```
void gdcm::Surface::SetAlgorithmFamily (
    SegmentHelper::BasicCodedEntry const & BSE )
```

**10.302.4.40 SetAlgorithmName()**

```
void gdcm::Surface::SetAlgorithmName (
    const char * str )
```

**10.302.4.41 SetAlgorithmVersion()**

```
void gdcm::Surface::SetAlgorithmVersion (
    const char * str )
```

**10.302.4.42 SetAxisOfRotation()**

```
void gdcm::Surface::SetAxisOfRotation (
    const float * axis )
```

**10.302.4.43 SetCenterOfRotation()**

```
void gdcM::Surface::SetCenterOfRotation (
    const float * center )
```

**10.302.4.44 SetFiniteVolume()**

```
void gdcM::Surface::SetFiniteVolume (
    STATES state )
```

**10.302.4.45 SetManifold()**

```
void gdcM::Surface::SetManifold (
    STATES state )
```

**10.302.4.46 SetMaximumPointDistance()**

```
void gdcM::Surface::SetMaximumPointDistance (
    float maximum )
```

**10.302.4.47 SetMeanPointDistance()**

```
void gdcM::Surface::SetMeanPointDistance (
    float average )
```

**10.302.4.48 SetMeshPrimitive()**

```
void gdcM::Surface::SetMeshPrimitive (
    MeshPrimitive & mp )
```

**10.302.4.49 SetNumberOfSurfacePoints()**

```
void gdcM::Surface::SetNumberOfSurfacePoints (
    const unsigned long nb )
```

**10.302.4.50 SetNumberOfVectors()**

```
void gdcM::Surface::SetNumberOfVectors (
    const unsigned long nb )
```

**10.302.4.51 SetPointCoordinatesData()**

```
void gdcm::Surface::SetPointCoordinatesData (
    DataElement const & de )
```

**10.302.4.52 SetPointPositionAccuracy()**

```
void gdcm::Surface::SetPointPositionAccuracy (
    const float * accuracies )
```

**10.302.4.53 SetPointsBoundingBoxCoordinates()**

```
void gdcm::Surface::SetPointsBoundingBoxCoordinates (
    const float * coordinates )
```

**10.302.4.54 SetProcessingAlgorithm()**

```
void gdcm::Surface::SetProcessingAlgorithm (
    SegmentHelper::BasicCodedEntry const & BSE )
```

**10.302.4.55 SetRecommendedDisplayCIELabValue() [1/3]**

```
void gdcm::Surface::SetRecommendedDisplayCIELabValue (
    const std::vector< unsigned short > & vl )
```

**10.302.4.56 SetRecommendedDisplayCIELabValue() [2/3]**

```
void gdcm::Surface::SetRecommendedDisplayCIELabValue (
    const unsigned short vl,
    const unsigned int idx = 0 )
```

**10.302.4.57 SetRecommendedDisplayCIELabValue() [3/3]**

```
void gdcm::Surface::SetRecommendedDisplayCIELabValue (
    const unsigned short vl[3] )
```

**10.302.4.58 SetRecommendedDisplayGrayscaleValue()**

```
void gdcm::Surface::SetRecommendedDisplayGrayscaleValue (
    const unsigned short vl )
```

**10.302.4.59 SetRecommendedPresentationOpacity()**

```
void gdcm::Surface::SetRecommendedPresentationOpacity (
    const float opacity )
```

**10.302.4.60 SetRecommendedPresentationType()**

```
void gdcm::Surface::SetRecommendedPresentationType (
    VIEWType type )
```

**10.302.4.61 SetSurfaceComments()**

```
void gdcm::Surface::SetSurfaceComments (
    const char * comment )
```

**10.302.4.62 SetSurfaceNumber()**

```
void gdcm::Surface::SetSurfaceNumber (
    const unsigned long nb )
```

**10.302.4.63 SetSurfaceProcessing()**

```
void gdcm::Surface::SetSurfaceProcessing (
    bool b )
```

**10.302.4.64 SetSurfaceProcessingDescription()**

```
void gdcm::Surface::SetSurfaceProcessingDescription (
    const char * description )
```

**10.302.4.65 SetSurfaceProcessingRatio()**

```
void gdcm::Surface::SetSurfaceProcessingRatio (
    const float ratio )
```

**10.302.4.66 SetVectorAccuracy()**

```
void gdcm::Surface::SetVectorAccuracy (
    const float * accuracy )
```

**10.302.4.67 SetVectorCoordinateData()**

```
void gdcm::Surface::SetVectorCoordinateData (
    DataElement const & de )
```

**10.302.4.68 SetVectorDimensionality()**

```
void gdcm::Surface::SetVectorDimensionality (
    const unsigned short dim )
```

The documentation for this class was generated from the following file:

- [gdcmSurface.h](#)

**10.303 gdcm::SurfaceHelper Class Reference**

[SurfaceHelper](#).

```
#include <gdcmSurfaceHelper.h>
```

**Public Types**

- typedef std::vector< unsigned short > [ColorArray](#)

**Static Public Member Functions**

- template<typename T , typename U >  
static std::vector< T > [RecommendedDisplayCIELabToRGB](#) (const [ColorArray](#) &CIELab, const U range←  
Max=255)  
*Convert a DICOM CIE-Lab (after reading) color into RGB.*
- template<typename U >  
static std::vector< float > [RecommendedDisplayCIELabToRGB](#) (const [ColorArray](#) &CIELab, const U range←  
Max=255)  
*Convert a DICOM CIE-Lab (after reading) color into RGB.*
- template<typename T , typename U >  
static [ColorArray](#) [RGBToRecommendedDisplayCIELab](#) (const std::vector< T > &RGB, const U rangeMax=255)  
*Convert a RGB color into DICOM CIE-Lab (ready to write).*
- template<typename T , typename U >  
static unsigned short [RGBToRecommendedDisplayGrayscale](#) (const std::vector< T > &RGB, const U range←  
Max=255)  
*Convert a RGB color into DICOM grayscale (ready to write).*

### 10.303.1 Detailed Description

[SurfaceHelper](#).

Helper class for [Surface](#) object

### 10.303.2 Member Typedef Documentation

#### 10.303.2.1 ColorArray

```
typedef std::vector< unsigned short > gdcm::SurfaceHelper::ColorArray
```

### 10.303.3 Member Function Documentation

#### 10.303.3.1 RecommendedDisplayCIELabToRGB() [1/2]

```
template<typename T , typename U >
std::vector< T > gdcm::SurfaceHelper::RecommendedDisplayCIELabToRGB (
    const ColorArray & CIELab,
    const U rangeMax = 255 ) [static]
```

Convert a DICOM CIE-Lab (after reading) color into RGB.

See also

PS 3.3 C.10.7.1.1

#### Parameters

<i>CIELab</i>	DICOM CIE-Lab array.
<i>rangeMax</i>	Max value of the RGB range.

#### Template Parameters

<i>T</i>	<a href="#">Type</a> of CIELab components.
<i>U</i>	<a href="#">Type</a> of rangeMax value.

#### 10.303.3.2 RecommendedDisplayCIELabToRGB() [2/2]

```
template<typename U >
std::vector< float > gdcm::SurfaceHelper::RecommendedDisplayCIELabToRGB (
```



```
const ColorArray & CIELab,
const U rangeMax = 255 ) [static]
```

Convert a DICOM CIE-Lab (after reading) color into RGB.

See also

PS 3.3 C.10.7.1.1

Parameters

<i>CIELab</i>	DICOM CIE-Lab array.
<i>rangeMax</i>	Max value of the RGB range.

Template Parameters

<i>U</i>	Type of rangeMax value.
----------	-------------------------

### 10.303.3.3 RGBToRecommendedDisplayCIELab()

```
template<typename T , typename U >
SurfaceHelper::ColorArray gdcm::SurfaceHelper::RGBToRecommendedDisplayCIELab (
    const std::vector< T > & RGB,
    const U rangeMax = 255 ) [static]
```

Convert a RGB color into DICOM CIE-Lab (ready to write).

See also

PS 3.3 C.10.7.1.1

Parameters

<i>RGB</i>	RGB array.
<i>rangeMax</i>	Max value of the RGB range.

Template Parameters

<i>T</i>	Type of RGB components.
<i>U</i>	Type of rangeMax value.

### 10.303.3.4 RGBToRecommendedDisplayGrayscale()

```
template<typename T , typename U >
```

```

unsigned short gdcM::SurfaceHelper::RGBToRecommendedDisplayGrayscale (
    const std::vector< T > & RGB,
    const U rangeMax = 255 ) [static]

```

Convert a RGB color into DICOM grayscale (ready to write).

#### See also

PS 3.3 C.27.1 tag(0062,000C)

#### Parameters

<i>RGB</i>	RGB array.
<i>rangeMax</i>	Max value of the RGB range.

#### Template Parameters

<i>T</i>	Type of RGB components.
<i>U</i>	Type of rangeMax value.

The documentation for this class was generated from the following file:

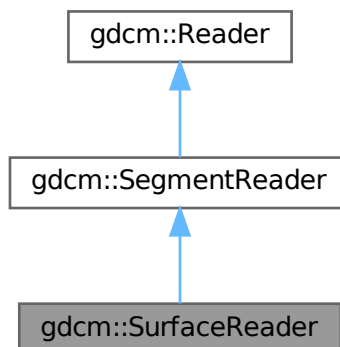
- [gdcMSurfaceHelper.h](#)

## 10.304 gdcM::SurfaceReader Class Reference

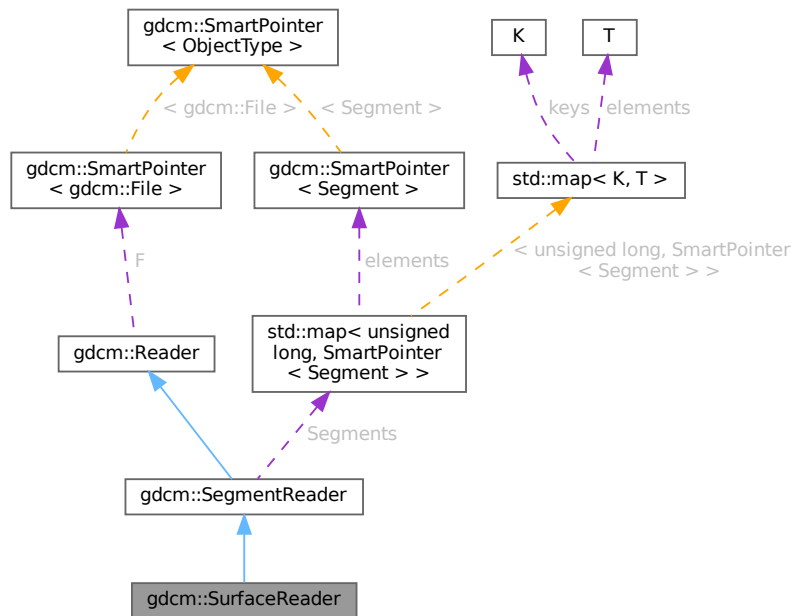
This class defines a SURFACE IE reader.

```
#include <gdcMSurfaceReader.h>
```

Inheritance diagram for gdcM::SurfaceReader:



Collaboration diagram for gdcm::SurfaceReader:



### Public Member Functions

- [SurfaceReader](#) ()
- [~SurfaceReader](#) () override
- unsigned long [GetNumberOfSurfaces](#) () const
- bool [Read](#) () override

*Read.*

### Public Member Functions inherited from [gdcm::SegmentReader](#)

- [SegmentReader](#) ()
- [~SegmentReader](#) () override
- [SegmentVector](#) [GetSegments](#) ()
- [SegmentVector](#) [GetSegments](#) () const

### Public Member Functions inherited from [gdcm::Reader](#)

- [Reader](#) ()
- virtual [~Reader](#) ()
- bool [CanRead](#) () const
- [File](#) & [GetFile](#) ()

*Set/Get File.*

- const [File](#) & [GetFile](#) () const

*Set/Get File.*

- size\_t [GetStreamCurrentPosition](#) () const
- bool [ReadSelectedPrivateTags](#) (std::set< [PrivateTag](#) > const &ptags, bool readvalues=true)

*Will only read the specified selected private tags.*

- bool [ReadSelectedTags](#) (std::set< [Tag](#) > const &tags, bool readvalues=true)

*Will only read the specified selected tags.*

- bool [ReadUpToTag](#) (const [Tag](#) &tag, std::set< [Tag](#) > const &skiptags=std::set< [Tag](#) >())
- void [SetFile](#) ([File](#) &file)

*Set/Get File.*

- void [SetFileName](#) (const char \*filename\_native)
- void [SetStream](#) (std::istream &input\_stream)

*Set the open-ed stream directly.*

### Protected Member Functions

- bool [ReadPointMacro](#) ([SmartPointer](#)< [Surface](#) > surface, const [DataSet](#) &surfaceDS)
- bool [ReadSurface](#) (const [Item](#) &surfaceItem, const unsigned long idx)
- bool [ReadSurfaces](#) ()

### Protected Member Functions inherited from [gdcm::SegmentReader](#)

- bool [ReadSegment](#) (const [Item](#) &segmentItem, const unsigned int idx)
- bool [ReadSegments](#) ()

### Protected Member Functions inherited from [gdcm::Reader](#)

- std::istream \* [GetStreamPtr](#) () const
- bool [ReadDataSet](#) ()
- bool [ReadMetaInformation](#) ()
- bool [ReadPreamble](#) ()

### Additional Inherited Members

### Public Types inherited from [gdcm::SegmentReader](#)

- typedef std::vector< [SmartPointer](#)< [Segment](#) > > [SegmentVector](#)

### Protected Types inherited from [gdcm::SegmentReader](#)

- typedef std::map< unsigned long, [SmartPointer](#)< [Segment](#) > > [SegmentMap](#)

## Protected Attributes inherited from [gdcm::SegmentReader](#)

- [SegmentMap Segments](#)

## Protected Attributes inherited from [gdcm::Reader](#)

- [SmartPointer< File > F](#)

### 10.304.1 Detailed Description

This class defines a SURFACE IE reader.

It reads surface mesh module attributes.

See also

PS 3.3 A.1.2.18 , A.57 and C.27

### 10.304.2 Constructor & Destructor Documentation

#### 10.304.2.1 SurfaceReader()

```
gdcm::SurfaceReader::SurfaceReader ( )
```

#### 10.304.2.2 ~SurfaceReader()

```
gdcm::SurfaceReader::~~SurfaceReader ( ) [override]
```

### 10.304.3 Member Function Documentation

#### 10.304.3.1 GetNumberOfSurfaces()

```
unsigned long gdcm::SurfaceReader::GetNumberOfSurfaces ( ) const
```

#### 10.304.3.2 Read()

```
bool gdcm::SurfaceReader::Read ( ) [override], [virtual]
```

Read.

Reimplemented from [gdcm::SegmentReader](#).

### 10.304.3.3 ReadPointMacro()

```
bool gdcM::SurfaceReader::ReadPointMacro (
    SmartPointer< Surface > surface,
    const DataSet & surfaceDS ) [protected]
```

### 10.304.3.4 ReadSurface()

```
bool gdcM::SurfaceReader::ReadSurface (
    const Item & surfaceItem,
    const unsigned long idx ) [protected]
```

### 10.304.3.5 ReadSurfaces()

```
bool gdcM::SurfaceReader::ReadSurfaces ( ) [protected]
```

The documentation for this class was generated from the following file:

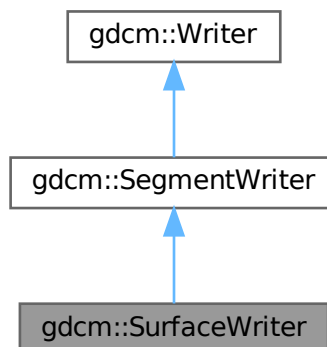
- [gdcMSurfaceReader.h](#)

## 10.305 gdcM::SurfaceWriter Class Reference

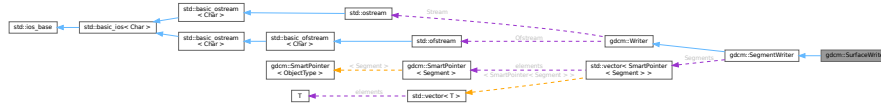
This class defines a SURFACE IE writer.

```
#include <gdcMSurfaceWriter.h>
```

Inheritance diagram for gdcM::SurfaceWriter:



Collaboration diagram for gdcm::SurfaceWriter:



## Public Member Functions

- [SurfaceWriter](#) ()
- [~SurfaceWriter](#) () override
- unsigned long [GetNumberOfSurfaces](#) ()
- void [SetNumberOfSurfaces](#) (const unsigned long nb)
- bool [Write](#) () override

*Write.*

## Public Member Functions inherited from [gdcm::SegmentWriter](#)

- [SegmentWriter](#) ()
- [~SegmentWriter](#) () override
- void [AddSegment](#) ([SmartPointer< Segment >](#) segment)
- unsigned int [GetNumberOfSegments](#) () const
- [SmartPointer< Segment >](#) [GetSegment](#) (const unsigned int idx=0) const
- [SegmentVector](#) & [GetSegments](#) ()
- const [SegmentVector](#) & [GetSegments](#) () const
- void [SetNumberOfSegments](#) (const unsigned int size)
- void [SetSegments](#) ([SegmentVector](#) &segments)

## Public Member Functions inherited from [gdcm::Writer](#)

- [Writer](#) ()
- virtual [~Writer](#) ()
- void [CheckFileMetaInformationOff](#) ()
- void [CheckFileMetaInformationOn](#) ()
- [File](#) & [GetFile](#) ()
- void [SetCheckFileMetaInformation](#) (bool b)
- *Undocumented function, do not use (= leave default)*
- void [SetFile](#) (const [File](#) &f)
- *Set/Get the DICOM file ([DataSet](#) + Header)*
- void [SetFileName](#) (const char \*filename\_native)
- *Set the filename of DICOM file to write:*
- void [SetStream](#) (std::ostream &output\_stream)
- *Set user ostream buffer.*

### Protected Member Functions

- void [ComputeNumberOfSurfaces](#) ()
- bool [PrepareWrite](#) ()
- bool [PrepareWritePointMacro](#) ([SmartPointer](#)< [Surface](#) > surface, [DataSet](#) &surfaceDS, const [TransferSyntax](#) &ts)

### Protected Member Functions inherited from [gdcm::SegmentWriter](#)

- bool [PrepareWrite](#) ()

### Protected Member Functions inherited from [gdcm::Writer](#)

- bool [GetCheckFileMetaInformation](#) () const
- std::ostream \* [GetStreamPtr](#) () const
- void [SetWriteDataSetOnly](#) (bool b)

### Protected Attributes

- unsigned long [NumberOfSurfaces](#)

### Protected Attributes inherited from [gdcm::SegmentWriter](#)

- [SegmentVector](#) [Segments](#)

### Protected Attributes inherited from [gdcm::Writer](#)

- std::ofstream \* [Ofstream](#)
- std::ostream \* [Stream](#)

### Additional Inherited Members

### Public Types inherited from [gdcm::SegmentWriter](#)

- typedef std::vector< [SmartPointer](#)< [Segment](#) > > [SegmentVector](#)

## 10.305.1 Detailed Description

This class defines a SURFACE IE writer.

It writes surface mesh module attributes.

See also

PS 3.3 A.1.2.18 , A.57 and C.27



## 10.305.2 Constructor & Destructor Documentation

### 10.305.2.1 SurfaceWriter()

```
gdcm::SurfaceWriter::SurfaceWriter ( )
```

### 10.305.2.2 ~SurfaceWriter()

```
gdcm::SurfaceWriter::~~SurfaceWriter ( ) [override]
```

## 10.305.3 Member Function Documentation

### 10.305.3.1 ComputeNumberOfSurfaces()

```
void gdcm::SurfaceWriter::ComputeNumberOfSurfaces ( ) [protected]
```

### 10.305.3.2 GetNumberOfSurfaces()

```
unsigned long gdcm::SurfaceWriter::GetNumberOfSurfaces ( )
```

### 10.305.3.3 PrepareWrite()

```
bool gdcm::SurfaceWriter::PrepareWrite ( ) [protected]
```

### 10.305.3.4 PrepareWritePointMacro()

```
bool gdcm::SurfaceWriter::PrepareWritePointMacro (
    SmartPointer< Surface > surface,
    DataSet & surfaceDS,
    const TransferSyntax & ts ) [protected]
```

### 10.305.3.5 SetNumberOfSurfaces()

```
void gdcm::SurfaceWriter::SetNumberOfSurfaces (
    const unsigned long nb )
```

### 10.305.3.6 Write()

```
bool gdcm::SurfaceWriter::Write ( ) [override], [virtual]
```

Write.

Reimplemented from [gdcm::SegmentWriter](#).

## 10.305.4 Member Data Documentation

### 10.305.4.1 NumberOfSurfaces

```
unsigned long gdcm::SurfaceWriter::NumberOfSurfaces [protected]
```

The documentation for this class was generated from the following file:

- [gdcmSurfaceWriter.h](#)

## 10.306 gdcm::SwapCode Class Reference

[SwapCode](#) representation.

```
#include <gdcmSwapCode.h>
```

### Public Types

- enum [SwapCodeType](#) {  
    [Unknown](#) = 0 ,  
    [LittleEndian](#) = 1234 ,  
    [BigEndian](#) = 4321 ,  
    [BadLittleEndian](#) = 3412 ,  
    [BadBigEndian](#) = 2143 }

### Public Member Functions

- [SwapCode](#) ([SwapCodeType](#) sc=[Unknown](#))
- [operator SwapCode::SwapCodeType](#) () const

### Static Public Member Functions

- static const char \* [GetSwapCodeString](#) ([SwapCode](#) const &sc)

### Static Protected Member Functions

- static int [GetIndex](#) ([SwapCode](#) const &sc)

### Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [SwapCode](#) &sc)

## 10.306.1 Detailed Description

[SwapCode](#) representation.

Examples

[TestByteSwap.cxx](#).

## 10.306.2 Member Enumeration Documentation

### 10.306.2.1 SwapCodeType

```
enum gdcm::SwapCode::SwapCodeType
```

Enumerator

Unknown	
LittleEndian	
BigEndian	
BadLittleEndian	
BadBigEndian	

## 10.306.3 Constructor & Destructor Documentation

### 10.306.3.1 SwapCode()

```
gdcm::SwapCode::SwapCode (  
    SwapCodeType sc = Unknown ) [inline]
```

## 10.306.4 Member Function Documentation

### 10.306.4.1 GetIndex()

```
static int gdcm::SwapCode::GetIndex (  
    SwapCode const & sc ) [static], [protected]
```

### 10.306.4.2 GetSwapCodeString()

```
static const char * gdcm::SwapCode::GetSwapCodeString (  
    SwapCode const & sc ) [static]
```

References [gdcm::operator<<\(\)](#).

### 10.306.4.3 operator SwapCode::SwapCodeType()

```
gdcM::SwapCode::operator SwapCode::SwapCodeType ( ) const [inline]
```

## 10.306.5 Friends And Related Symbol Documentation

### 10.306.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & os,
    const SwapCode & sc ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcMSwapCode.h](#)

## 10.307 gdcM::SwapperDoOp Class Reference

```
#include <gdcMSwapper.h>
```

### Static Public Member Functions

- `template<typename T >`  
static T [Swap](#) (T val)
- `template<typename T >`  
static void [SwapArray](#) (T \*array, size\_t n)

### 10.307.1 Member Function Documentation

#### 10.307.1.1 Swap()

```
template<typename T >
static T gdcM::SwapperDoOp::Swap (
    T val ) [static]
```

#### 10.307.1.2 SwapArray()

```
template<typename T >
static void gdcM::SwapperDoOp::SwapArray (
    T * array,
    size_t n ) [inline], [static]
```

The documentation for this class was generated from the following file:

- [gdcMSwapper.h](#)

## 10.308 gdcm::SwapperNoOp Class Reference

```
#include <gdcmSwapper.h>
```

### Static Public Member Functions

- `template<typename T >`  
`static T Swap (T val)`
- `template<typename T >`  
`static void SwapArray (T *, size_t)`

### 10.308.1 Detailed Description

#### Examples

[DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), and [ReadExplicitLengthSQIVR.cxx](#).

### 10.308.2 Member Function Documentation

#### 10.308.2.1 [Swap\(\)](#)

```
template<typename T >
static T gdcm::SwapperNoOp::Swap (
    T val ) [inline], [static]
```

#### 10.308.2.2 [SwapArray\(\)](#)

```
template<typename T >
static void gdcm::SwapperNoOp::SwapArray (
    T * ,
    size_t ) [inline], [static]
```

The documentation for this class was generated from the following file:

- [gdcmSwapper.h](#)

## 10.309 gdcm::System Class Reference

Class to do system operation.

```
#include <gdcmSystem.h>
```

## Static Public Member Functions

- static std::wstring [ConvertToUNC](#) (const char \*utf8path)
- static bool [DeleteDirectory](#) (const char \*source)  
*remove a directory named source*
- static size\_t [EncodeBytes](#) (char \*out, const unsigned char \*data, int size)
- static bool [FileExists](#) (const char \*filename)  
*Check whether the specified file exist on the system.*
- static bool [FileIsDirectory](#) (const char \*name)  
*Check whether the file specified is a directory:*
- static bool [FileIsSymlink](#) (const char \*name)  
*Check whether name is a symlink.*
- static size\_t [FileSize](#) (const char \*filename)
- static time\_t [FileTime](#) (const char \*filename)
- static bool [FormatDateTime](#) (char date[22], time\_t t, long milliseconds=0)
- static bool [GetCurrentDateTime](#) (char date[22])
- static const char \* [GetCurrentModuleFileName](#) ()
- static const char \* [GetCurrentProcessFileName](#) ()
- static const char \* [GetCurrentResourcesDirectory](#) ()
- static const char \* [GetCWD](#) ()
- static bool [GetHostName](#) (char hostname[255])
- static const char \* [GetLastError](#) ()  
*Return the last error.*
- static const char \* [GetLocaleCharset](#) ()  
*return locale charmap*
- static const char \* [GetTimezoneOffsetFromUTC](#) ()
- static bool [MakeDirectory](#) (const char \*path)  
*Create a directory name path.*
- static bool [ParseDateTime](#) (time\_t &timep, const char date[22])  
*Parse a date stored as ASCII text into a time\_t structured (discard millisecond if any)*
- static bool [ParseDateTime](#) (time\_t &timep, long &milliseconds, const char date[22])
- static bool [RemoveFile](#) (const char \*source)  
*remove a file named source*
- static int [StrCaseCmp](#) (const char \*s1, const char \*s2)  
*consistent func for C99 spec of strcasecmp/strncasecmp*
- static int [StrNCaseCmp](#) (const char \*s1, const char \*s2, size\_t n)
- static char \* [StrSep](#) (char \*\*stringp, const char \*delim)
- static char \* [StrTokR](#) (char \*ptr, const char \*sep, char \*\*end)  
*strtok\_r*

## Static Protected Member Functions

- static bool [GetPermissions](#) (const char \*file, unsigned short &mode)  
*NOT THREAD SAFE.*
- static bool [SetPermissions](#) (const char \*file, unsigned short mode)

## 10.309.1 Detailed Description

Class to do system operation.

OS independent functionalities

### Examples

[BasicAnonymizer.cs](#), [BasicImageAnonymizer.cs](#), [Cleaner.cs](#), [ClinicalTrialIdentificationWorkflow.cs](#), [CompressLossyJPEG.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [DumpCSA.cs](#), [ExtractEncapsulatedFile.cs](#), [ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), [ExtractOneFrame.cs](#), [FileAnonymize.cs](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [FileStreaming.cs](#), [GetArray.cs](#), [MetaImageMD5Activiz.cs](#), [MpegVideoInfo.cs](#), [ReformatFile.cs](#), [RescaleImage.cs](#), [ScanDirectory.cs](#), [SimplePrint.cs](#), and [StandardizeFiles.cs](#).

## 10.309.2 Member Function Documentation

### 10.309.2.1 ConvertToUNC()

```
static std::wstring gdcm::System::ConvertToUNC (
    const char * utf8path ) [static]
```

When needed convert a PATH into a UNC equivalent. This allow transparent support for path longer than MAX\_PATH. Only on \_MSC\_VER compiler, return empty string otherwise.

### 10.309.2.2 DeleteDirectory()

```
static bool gdcm::System::DeleteDirectory (
    const char * source ) [static]
```

remove a directory named source

### 10.309.2.3 EncodeBytes()

```
static size_t gdcm::System::EncodeBytes (
    char * out,
    const unsigned char * data,
    int size ) [static]
```

Used internally by the [UIDGenerator](#) class to convert a uuid tape to a DICOM VR:UI type

### 10.309.2.4 FileExists()

```
static bool gdcm::System::FileExists (
    const char * filename ) [static]
```

Check whether the specified file exist on the system.

### Examples

[DumpVisusChange.cxx](#), [EncapsulateFileInRawData.cxx](#), [MagnifyFile.cxx](#), and [gdcmorthoplanes.cxx](#).

### 10.309.2.5 FileIsDirectory()

```
static bool gdcM::System::FileIsDirectory (
    const char * name ) [static]
```

Check whether the file specified is a directory:

#### Examples

[DumpVisusChange.cxx](#), [gdcMorthoplanes.cxx](#), and [threadgdcM.cxx](#).

### 10.309.2.6 FileIsSymlink()

```
static bool gdcM::System::FileIsSymlink (
    const char * name ) [static]
```

Check whether name is a symlink.

### 10.309.2.7 FileSize()

```
static size_t gdcM::System::FileSize (
    const char * filename ) [static]
```

Return the filesize. 0 if file does not exist.

#### Warning

you need to use FileExists to differentiate between empty file and missing file.

for very large size file and on system where size\_t is not appropriate to store off\_t value the function will return 0.

#### Examples

[CheckBigEndianBug.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [EncapsulateFileInRawData.cxx](#), and [SimpleScanner.cxx](#).

### 10.309.2.8 FileTime()

```
static time_t gdcM::System::FileTime (
    const char * filename ) [static]
```

Return the time of last modification of file 0 if the file does not exist



### 10.309.2.9 FormatDateTime()

```
static bool gdcm::System::FormatDateTime (
    char date[22],
    time_t t,
    long milliseconds = 0 ) [static]
```

format as ASCII text a time\_t with milliseconds See [VR::DT](#) from DICOM PS 3.5 milliseconds is in the range [0, 999999]

### 10.309.2.10 GetCurrentDateTime()

```
static bool gdcm::System::GetCurrentDateTime (
    char date[22] ) [static]
```

Return the current data time, and format it as ASCII text. This is simply a call to gettimeofday + FormatDateTime, since WIN32 do not have an implementation for gettimeofday, this is more portable. The call time(0) is not precise for our resolution

#### Examples

[TemplateEmptyImage.cxx](#).

### 10.309.2.11 GetCurrentModuleFileName()

```
static const char * gdcm::System::GetCurrentModuleFileName ( ) [static]
```

Return the directory the current module is located: NOT THREAD SAFE

### 10.309.2.12 GetCurrentProcessFileName()

```
static const char * gdcm::System::GetCurrentProcessFileName ( ) [static]
```

Return the directory the current process (executable) is located: NOT THREAD SAFE

### 10.309.2.13 GetCurrentResourcesDirectory()

```
static const char * gdcm::System::GetCurrentResourcesDirectory ( ) [static]
```

On some system (Apple) return the path to the current bundled 'Resources' directory NOT THREAD SAFE

### 10.309.2.14 GetCWD()

```
static const char * gdcm::System::GetCWD ( ) [static]
```

Return current working directory Warning: if current working path is too long (>2048 bytes) the call will fail and call will return NULL NOT THREAD SAFE

**10.309.2.15 GetHostName()**

```
static bool gdcm::System::GetHostName (
    char hostname[255] ) [static]
```

Retrieve the hostname, only the first 255 byte are copied. This may come handy to specify the Station Name

**10.309.2.16 GetLastSystemError()**

```
static const char * gdcm::System::GetLastSystemError ( ) [static]
```

Return the last error.

**10.309.2.17 GetLocaleCharset()**

```
static const char * gdcm::System::GetLocaleCharset ( ) [static]
```

return locale charmap

**10.309.2.18 GetPermissions()**

```
static bool gdcm::System::GetPermissions (
    const char * file,
    unsigned short & mode ) [static], [protected]
```

NOT THREAD SAFE.

**10.309.2.19 GetTimezoneOffsetFromUTC()**

```
static const char * gdcm::System::GetTimezoneOffsetFromUTC ( ) [static]
```

Return the value for Timezone Offset From UTC as string.

**Warning**

not thread safe

**10.309.2.20 MakeDirectory()**

```
static bool gdcm::System::MakeDirectory (
    const char * path ) [static]
```

Create a directory name path.

**10.309.2.21 ParseDateTime() [1/2]**

```
static bool gdcm::System::ParseDateTime (
    time_t & timep,
    const char date[22] ) [static]
```

Parse a date stored as ASCII text into a time\_t structured (discard millisecond if any)

**10.309.2.22 ParseDateTime() [2/2]**

```
static bool gdcm::System::ParseDateTime (
    time_t & timep,
    long & milliseconds,
    const char date[22] ) [static]
```

Parse a date stored as ASCII text into a time\_t structured and millisecond

See also

[FormatDateTime](#)

**10.309.2.23 RemoveFile()**

```
static bool gdcm::System::RemoveFile (
    const char * source ) [static]
```

remove a file named source

**10.309.2.24 SetPermissions()**

```
static bool gdcm::System::SetPermissions (
    const char * file,
    unsigned short mode ) [static], [protected]
```

**10.309.2.25 StrCaseCmp()**

```
static int gdcm::System::StrCaseCmp (
    const char * s1,
    const char * s2 ) [static]
```

consistent func for C99 spec of strcasecmp/strncasecmp

### 10.309.2.26 StrNCaseCmp()

```
static int gdcM::System::StrNCaseCmp (
    const char * s1,
    const char * s2,
    size_t n ) [static]
```

#### Precondition

n != 0

### 10.309.2.27 StrSep()

```
static char * gdcM::System::StrSep (
    char ** stringp,
    const char * delim ) [static]
```

strsep param stringp is passed by pointer, it may be modified, you'll need to make a copy, in case you want to free the memory pointed at

### 10.309.2.28 StrTokR()

```
static char * gdcM::System::StrTokR (
    char * ptr,
    const char * sep,
    char ** end ) [static]
```

strtok\_r

The documentation for this class was generated from the following file:

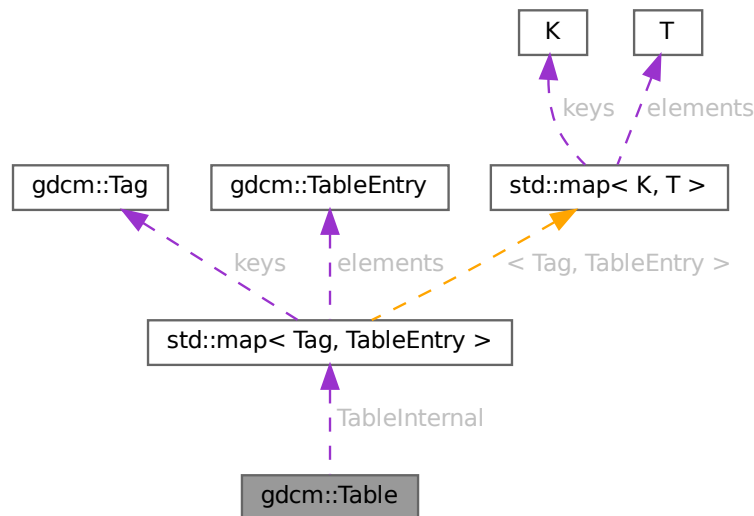
- [gdcMSystem.h](#)

## 10.310 gdcM::Table Class Reference

[Table](#).

```
#include <gdcMTable.h>
```

Collaboration diagram for gdcm::Table:



## Public Types

- typedef std::map< [Tag](#), [TableEntry](#) > [MapTableEntry](#)

## Public Member Functions

- [Table](#) ()=default
- [Table](#) (const [Table](#) &\_val)=delete
- [~Table](#) ()=default
- const [TableEntry](#) & [GetTableEntry](#) (const [Tag](#) &tag) const
- void [InsertEntry](#) ([Tag](#) const &tag, [TableEntry](#) const &te)
- [Table](#) & [operator=](#) (const [Table](#) &\_val)=delete

## Public Attributes

- [MapTableEntry](#) [TableInternal](#)

## Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [Table](#) &\_val)

### 10.310.1 Detailed Description

[Table](#).

### 10.310.2 Member Typedef Documentation

#### 10.310.2.1 MapTableEntry

```
typedef std::map<Tag, TableEntry> gdcM::Table::MapTableEntry
```

### 10.310.3 Constructor & Destructor Documentation

#### 10.310.3.1 Table() [1/2]

```
gdcM::Table::Table ( ) [default]
```

#### 10.310.3.2 ~Table()

```
gdcM::Table::~~Table ( ) [default]
```

#### 10.310.3.3 Table() [2/2]

```
gdcM::Table::Table (
    const Table & _val ) [delete]
```

### 10.310.4 Member Function Documentation

#### 10.310.4.1 GetTableEntry()

```
const TableEntry & gdcM::Table::GetTableEntry (
    const Tag & tag ) const [inline]
```

References [GetTableEntry\(\)](#), and [TableInternal](#).

Referenced by [GetTableEntry\(\)](#).

#### 10.310.4.2 InsertEntry()

```
void gdcM::Table::InsertEntry (
    Tag const & tag,
    TableEntry const & te ) [inline]
```

References [TableInternal](#).

#### 10.310.4.3 operator=()

```
Table & gdcm::Table::operator= (
    const Table & _val ) [delete]
```

### 10.310.5 Friends And Related Symbol Documentation

#### 10.310.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const Table & _val ) [friend]
```

### 10.310.6 Member Data Documentation

#### 10.310.6.1 TableInternal

[MapTableEntry](#) gdcm::Table::TableInternal

Referenced by [GetTableEntry\(\)](#), and [InsertEntry\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmTable.h](#)

## 10.311 gdcm::TableEntry Class Reference

[TableEntry](#).

```
#include <gdcmTableEntry.h>
```

### Public Member Functions

- [TableEntry](#) (const char \*attribute=nullptr, [Type](#) const &type=[Type](#)(), const char \*des=nullptr)
- [~TableEntry](#) ()=default

#### 10.311.1 Detailed Description

[TableEntry](#).

## 10.311.2 Constructor & Destructor Documentation

### 10.311.2.1 TableEntry()

```
gdcm::TableEntry::TableEntry (
    const char * attribute = nullptr,
    Type const & type = Type(),
    const char * des = nullptr ) [inline]
```

### 10.311.2.2 ~TableEntry()

```
gdcm::TableEntry::~~TableEntry ( ) [default]
```

The documentation for this class was generated from the following file:

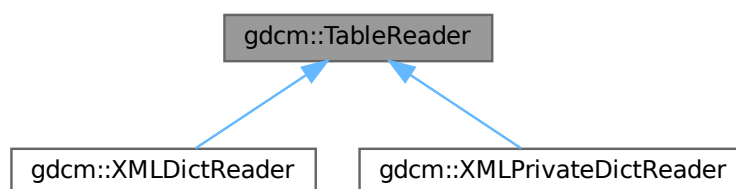
- [gdcmTableEntry.h](#)

## 10.312 gdcm::TableReader Class Reference

Class for representing a [TableReader](#).

```
#include <gdcmTableReader.h>
```

Inheritance diagram for gdcm::TableReader:





**Public Member Functions**

- [TableReader](#) ([Defs](#) &defs)
- virtual [~TableReader](#) ()=default
- virtual void [CharacterDataHandler](#) (const char \*data, int length)
- virtual void [EndElement](#) (const char \*name)
- const [Defs](#) & [GetDefs](#) () const
- const char \* [GetFilename](#) ()
- void [HandleIOD](#) (const char \*\*atts)
- void [HandleIODEntry](#) (const char \*\*atts)
- void [HandleMacro](#) (const char \*\*atts)
- void [HandleMacroEntry](#) (const char \*\*atts)
- void [HandleMacroEntryDescription](#) (const char \*\*atts)
- void [HandleModule](#) (const char \*\*atts)
- void [HandleModuleEntry](#) (const char \*\*atts)
- void [HandleModuleEntryDescription](#) (const char \*\*atts)
- void [HandleModuleInclude](#) (const char \*\*atts)
- int [Read](#) ()
- void [SetFilename](#) (const char \*filename)
- virtual void [StartElement](#) (const char \*name, const char \*\*atts)

**10.312.1 Detailed Description**

Class for representing a [TableReader](#).

**Note**

This class is an empty shell meant to be derived

**10.312.2 Constructor & Destructor Documentation****10.312.2.1 TableReader()**

```
gdcm::TableReader::TableReader (
    Defs & defs ) [inline]
```

**10.312.2.2 ~TableReader()**

```
virtual gdcm::TableReader::~~TableReader ( ) [virtual], [default]
```

**10.312.3 Member Function Documentation****10.312.3.1 CharacterDataHandler()**

```
virtual void gdcm::TableReader::CharacterDataHandler (
    const char * data,
    int length ) [virtual]
```

Reimplemented in [gdcm::XMLDictReader](#), and [gdcm::XMLPrivateDictReader](#).

### 10.312.3.2 EndElement()

```
virtual void gdcm::TableReader::EndElement (
    const char * name ) [virtual]
```

Reimplemented in [gdcm::XMLDictReader](#), and [gdcm::XMLPrivateDictReader](#).

### 10.312.3.3 GetDefs()

```
const Defs & gdcm::TableReader::GetDefs ( ) const [inline]
```

### 10.312.3.4 GetFilename()

```
const char * gdcm::TableReader::GetFilename ( ) [inline]
```

### 10.312.3.5 HandleIOD()

```
void gdcm::TableReader::HandleIOD (
    const char ** atts )
```

### 10.312.3.6 HandleIODEntry()

```
void gdcm::TableReader::HandleIODEntry (
    const char ** atts )
```

### 10.312.3.7 HandleMacro()

```
void gdcm::TableReader::HandleMacro (
    const char ** atts )
```

### 10.312.3.8 HandleMacroEntry()

```
void gdcm::TableReader::HandleMacroEntry (
    const char ** atts )
```

### 10.312.3.9 HandleMacroEntryDescription()

```
void gdcm::TableReader::HandleMacroEntryDescription (
    const char ** atts )
```

### 10.312.3.10 HandleModule()

```
void gdcm::TableReader::HandleModule (
    const char ** atts )
```

### 10.312.3.11 HandleModuleEntry()

```
void gdcm::TableReader::HandleModuleEntry (
    const char ** atts )
```

### 10.312.3.12 HandleModuleEntryDescription()

```
void gdcm::TableReader::HandleModuleEntryDescription (
    const char ** atts )
```

### 10.312.3.13 HandleModuleInclude()

```
void gdcm::TableReader::HandleModuleInclude (
    const char ** atts )
```

### 10.312.3.14 Read()

```
int gdcm::TableReader::Read ( )
```

### 10.312.3.15 SetFilename()

```
void gdcm::TableReader::SetFilename (
    const char * filename ) [inline]
```

### 10.312.3.16 StartElement()

```
virtual void gdcm::TableReader::StartElement (
    const char * name,
    const char ** atts ) [virtual]
```

Reimplemented in [gdcm::XMLDictReader](#), and [gdcm::XMLPrivateDictReader](#).

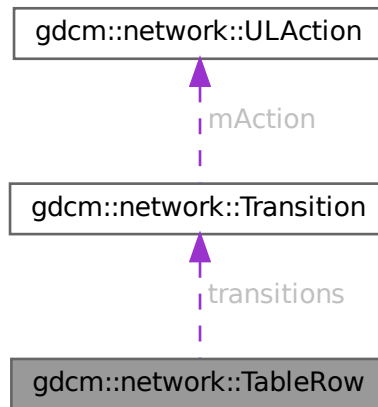
The documentation for this class was generated from the following file:

- [gdcmTableReader.h](#)

## 10.313 gdcmm::network::TableRow Class Reference

```
#include <gdcmmULTransitionTable.h>
```

Collaboration diagram for gdcmm::network::TableRow:



### Public Member Functions

- [TableRow](#) ()
- [~TableRow](#) ()

### Public Attributes

- [Transition](#) \* [transitions](#) [[cMaxStateID](#)]

## 10.313.1 Constructor & Destructor Documentation

### 10.313.1.1 TableRow()

```
gdcmm::network::TableRow::TableRow ( ) [inline]
```

References [gdcmm::network::cMaxStateID](#), and [transitions](#).

### 10.313.1.2 ~TableRow()

```
gdcmm::network::TableRow::~~TableRow ( ) [inline]
```

References [gdcmm::network::cMaxStateID](#), and [transitions](#).

## 10.313.2 Member Data Documentation

### 10.313.2.1 transitions

[Transition\\*](#) gdcmm::network::TableRow::transitions[cMaxStateID]

Referenced by [TableRow\(\)](#), and [~TableRow\(\)](#).

The documentation for this class was generated from the following file:

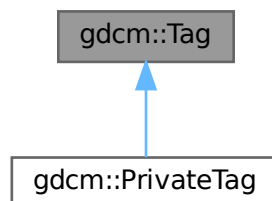
- [gdcmmULTransitionTable.h](#)

## 10.314 gdcmm::Tag Class Reference

Class to represent a DICOM Data [Element](#) ([Attribute](#)) [Tag](#) (Group, [Element](#)).

```
#include <gdcmmTag.h>
```

Inheritance diagram for gdcmm::Tag:



### Public Member Functions

- [Tag](#) (const [Tag](#) &\_val)
- [Tag](#) (uint16\_t group, uint16\_t element)  
*Constructor with 2\*uint16\_t.*
- [Tag](#) (uint32\_t tag=0)  
*Constructor with 1\*uint32\_t Prefer the ctor that takes two uint16\_t.*
- uint16\_t [GetElement](#) () const  
*Returns the 'Element number' of the given Tag.*
- uint32\_t [GetElementTag](#) () const  
*Returns the full tag value of the given Tag.*
- uint16\_t [GetGroup](#) () const

- Returns the 'Group number' of the given [Tag](#).*

  - `uint32_t GetLength () const`  
*return the length of tag (read: size on disk)*
  - `Tag GetPrivateCreator () const`  
*Return the Private Creator Data [Element](#) tag of a private data element.*
  - `bool IsGroupLength () const`  
*return whether the tag correspond to a group length tag:*
  - `bool IsGroupXX (const Tag &t) const`  
*e.g 6002,3000 belong to groupXX: 6000,3000*
  - `bool IsIllegal () const`  
*return if the tag is considered to be an illegal tag*
  - `bool IsPrivate () const`
  - `bool IsPrivateCreator () const`
  - `bool IsPublic () const`
  - `bool operator!= (const Tag &_val) const`
  - `bool operator< (const Tag &_val) const`
  - `bool operator<= (const Tag &t2) const`
  - `Tag & operator= (const Tag &_val)`
  - `bool operator== (const Tag &_val) const`
  - `uint16_t & operator[] (const unsigned int &_id)`  
*Returns the Group or [Element](#) of the given [Tag](#), depending on id (0/1)*
  - `const uint16_t & operator[] (const unsigned int &_id) const`  
*Returns the Group or [Element](#) of the given [Tag](#), depending on id (0/1)*
  - `std::string PrintAsContinuousString () const`
  - `std::string PrintAsContinuousUpperCaseString () const`  
*Same as PrintAsContinuousString, but hexadecimal [a-f] are printed using upper case.*
  - `std::string PrintAsPipeSeparatedString () const`
  - `template<typename TSwap >`  
`std::istream & Read (std::istream &is)`  
*Read a tag from binary representation.*
  - `bool ReadFromCommaSeparatedString (const char *str)`
  - `bool ReadFromContinuousString (const char *str)`
  - `bool ReadFromPipeSeparatedString (const char *str)`
  - `void SetElement (uint16_t element)`  
*Sets the '[Element](#) number' of the given [Tag](#).*
  - `void SetElementTag (uint16_t group, uint16_t element)`  
*Sets the 'Group number' & '[Element](#) number' of the given [Tag](#).*
  - `void SetElementTag (uint32_t tag)`  
*Sets the full tag value of the given [Tag](#).*
  - `void SetGroup (uint16_t group)`  
*Sets the 'Group number' of the given [Tag](#).*
  - `void SetPrivateCreator (Tag const &t)`  
*Set private creator:*
  - `template<typename TSwap >`  
`const std::ostream & Write (std::ostream &os) const`  
*Write a tag in binary rep.*

## Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [Tag](#) &\_val)
- std::istream & [operator>>](#) (std::istream &\_is, [Tag](#) &\_val)

## 10.314.1 Detailed Description

Class to represent a DICOM Data [Element](#) ([Attribute](#) [Tag](#) (Group, [Element](#)).

Basically an uint32\_t which can also be expressed as two uint16\_t (group and element)

### Note

DATA ELEMENT TAG: A unique identifier for a Data [Element](#) composed of an ordered pair of numbers (a Group Number followed by an [Element](#) Number). GROUP NUMBER: The first number in the ordered pair of numbers that makes up a Data [Element Tag](#). ELEMENT NUMBER: The second number in the ordered pair of numbers that makes up a Data [Element Tag](#).

### Examples

[BasicAnonymizer.cs](#), [BasicImageAnonymizer.cs](#), [ChangeSequenceUltrasound.cxx](#), [Cleaner.cs](#), [ClinicalTrialAnnotate.cxx](#), [ClinicalTrialIdentificationWorkflow.cs](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [DecompressImage.cs](#), [DeriveSeries.cxx](#), [DumpToSQLITE3.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FileAnonymize.cs](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenAllIVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSEExplicit.cxx](#), [MakeTemplate.cxx](#), [ManipulateFile.cs](#), [MergeTwoFiles.cxx](#), [MpegVideoInfo.cs](#), [PatchFile.cxx](#), [PublicDict.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ScanDirectory.cs](#), [SimpleScanner.cxx](#), [SortImage.cxx](#), [StreamImageReaderTest.cxx](#), [TraverseModules.cxx](#), [VolumeSorter.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

## 10.314.2 Constructor & Destructor Documentation

### 10.314.2.1 Tag() [1/3]

```
gdcm::Tag::Tag (
    uint16_t group,
    uint16_t element ) [inline]
```

Constructor with 2\*uint16\_t.

### 10.314.2.2 Tag() [2/3]

```
gdcm::Tag::Tag (
    uint32_t tag = 0 ) [inline]
```

Constructor with 1\*uint32\_t Prefer the ctor that takes two uint16\_t.

### 10.314.2.3 Tag() [3/3]

```
gdcM::Tag::Tag (
    const Tag & _val ) [inline]
```

References [tag](#).

## 10.314.3 Member Function Documentation

### 10.314.3.1 GetElement()

```
uint16_t gdcM::Tag::GetElement ( ) const [inline]
```

Returns the 'Element number' of the given [Tag](#).

#### Examples

[DuplicatePCDE.cxx](#), and [PublicDict.cxx](#).

Referenced by [gdcM::PrivateTag::PrivateTag\(\)](#), [gdcM::DataSet::ComputeGroupLength\(\)](#), [IsGroupXX\(\)](#), [gdcM::PrivateDict::PrintXML\(\)](#), [gdcM::SequenceOfFragments::ReadValue\(\)](#), and [SetPrivateCreator\(\)](#).

### 10.314.3.2 GetElementTag()

```
uint32_t gdcM::Tag::GetElementTag ( ) const [inline]
```

Returns the full tag value of the given [Tag](#).

Referenced by [gdcM::PrivateTag::operator!=\(\)](#), [gdcM::PrivateTag::operator!=\(\)](#), [gdcM::PrivateTag::operator=\(\)](#), [gdcM::PrivateTag::operator==\(\)](#), and [gdcM::PrivateTag::operator==\(\)](#).

### 10.314.3.3 GetGroup()

```
uint16_t gdcM::Tag::GetGroup ( ) const [inline]
```

Returns the 'Group number' of the given [Tag](#).

#### Examples

[DuplicatePCDE.cxx](#), and [GenAllVR.cxx](#).

Referenced by [gdcM::DataSet::ComputeGroupLength\(\)](#), [gdcM::DataSet::Insert\(\)](#), [gdcM::FileMetaInformation::Insert\(\)](#), [gdcM::CommandDataSet::Insert\(\)](#), [IsGroupXX\(\)](#), [gdcM::PrivateDict::PrintXML\(\)](#), [gdcM::SequenceOfFragments::ReadValue\(\)](#), and [SetPrivateCreator\(\)](#).



#### 10.314.3.4 GetLength()

```
uint32_t gdcm::Tag::GetLength ( ) const [inline]
```

return the length of tag (read: size on disk)

#### 10.314.3.5 GetPrivateCreator()

```
Tag gdcm::Tag::GetPrivateCreator ( ) const [inline]
```

Return the Private Creator Data [Element](#) tag of a private data element.

References [SetElement\(\)](#).

#### 10.314.3.6 IsGroupLength()

```
bool gdcm::Tag::IsGroupLength ( ) const [inline]
```

return whether the tag correspond to a group length tag:

#### 10.314.3.7 IsGroupXX()

```
bool gdcm::Tag::IsGroupXX (
    const Tag & t ) const [inline]
```

e.g 6002,3000 belong to groupXX: 6000,3000

References [GetElement\(\)](#), [GetGroup\(\)](#), and [IsPrivate\(\)](#).

#### 10.314.3.8 IsIllegal()

```
bool gdcm::Tag::IsIllegal ( ) const [inline]
```

return if the tag is considered to be an illegal tag

#### 10.314.3.9 IsPrivate()

```
bool gdcm::Tag::IsPrivate ( ) const [inline]
```

PRIVATE DATA ELEMENT: Additional Data [Element](#), defined by an implementor, to communicate information that is not contained in Standard Data Elements. Private Data elements have odd Group Numbers.

#### Examples

[DuplicatePCDE.cxx](#).

Referenced by [IsGroupXX\(\)](#), and [SetPrivateCreator\(\)](#).

#### 10.314.3.10 IsPrivateCreator()

```
bool gdcm::Tag::IsPrivateCreator ( ) const [inline]
```

Returns if tag is a Private Creator (xxxx,00yy), where xxxx is odd number and yy in [0x10,0xFF]

##### Examples

[DuplicatePCDE.cxx](#).

#### 10.314.3.11 IsPublic()

```
bool gdcm::Tag::IsPublic ( ) const [inline]
```

STANDARD DATA ELEMENT: A Data [Element](#) defined in the DICOM Standard, and therefore listed in the DICOM Data [Element](#) Dictionary in PS 3.6. Is the [Tag](#) from the Public dict...well the implementation is buggy it does not prove the element is indeed in the dict...

#### 10.314.3.12 operator!=(())

```
bool gdcm::Tag::operator!= (
    const Tag & _val ) const [inline]
```

References [tag](#).

#### 10.314.3.13 operator<()

```
bool gdcm::Tag::operator< (
    const Tag & _val ) const [inline]
```

DICOM Standard expects the Data [Element](#) to be sorted by Tags All other comparison can be constructed from this one and operator ==

References [tag](#), and [tags](#).

#### 10.314.3.14 operator<=()

```
bool gdcm::Tag::operator<= (
    const Tag & t2 ) const [inline]
```

#### 10.314.3.15 operator=()

```
Tag & gdcm::Tag::operator= (
    const Tag & _val ) [inline]
```

References [tag](#).

#### 10.314.3.16 operator==( )

```
bool gdcmm::Tag::operator==(
    const Tag & _val ) const [inline]
```

References [tag](#).

#### 10.314.3.17 operator[]( ) [1/2]

```
uint16_t & gdcmm::Tag::operator[](
    const unsigned int & _id ) [inline]
```

Returns the Group or [Element](#) of the given [Tag](#), depending on id (0/1)

#### 10.314.3.18 operator[]( ) [2/2]

```
const uint16_t & gdcmm::Tag::operator[](
    const unsigned int & _id ) const [inline]
```

Returns the Group or [Element](#) of the given [Tag](#), depending on id (0/1)

#### 10.314.3.19 PrintAsContinuousString( )

```
std::string gdcmm::Tag::PrintAsContinuousString ( ) const
```

Print tag value with no separating comma: eg. tag = "12345678" It comes in useful when reading tag values from XML file(in NativeDICOMModel)

#### 10.314.3.20 PrintAsContinuousUpperCaseString( )

```
std::string gdcmm::Tag::PrintAsContinuousUpperCaseString ( ) const
```

Same as PrintAsContinuousString, but hexadecimal [a-f] are printed using upper case.

#### 10.314.3.21 PrintAsPipeSeparatedString( )

```
std::string gdcmm::Tag::PrintAsPipeSeparatedString ( ) const
```

Print as a pipe separated string (GDCM 1.x compat only). Do not use in newer code

See also

[ReadFromPipeSeparatedString](#)

### 10.314.3.22 Read()

```
template<typename TSwap >
std::istream & gdcM::Tag::Read (
    std::istream & is ) [inline]
```

Read a tag from binary representation.

### 10.314.3.23 ReadFromCommaSeparatedString()

```
bool gdcM::Tag::ReadFromCommaSeparatedString (
    const char * str )
```

Read from a comma separated string. This is a highly user oriented function, the string should be formatted as↵: 1234,5678 to specify the tag (0x1234,0x5678) The notation comes from the DICOM standard, and is handy to use from a command line program

### 10.314.3.24 ReadFromContinuousString()

```
bool gdcM::Tag::ReadFromContinuousString (
    const char * str )
```

Read From XML formatted tag value eg. tag = "12345678" It comes in useful when reading tag values from XML file(in NativeDICOMModel)

### 10.314.3.25 ReadFromPipeSeparatedString()

```
bool gdcM::Tag::ReadFromPipeSeparatedString (
    const char * str )
```

Read from a pipe separated string (GDCM 1.x compat only). Do not use in newer code

See also

[ReadFromCommaSeparatedString](#)

### 10.314.3.26 SetElement()

```
void gdcM::Tag::SetElement (
    uint16_t element ) [inline]
```

Sets the '[Element](#) number' of the given [Tag](#).

Examples

[DuplicatePCDE.cxx](#), and [PublicDict.cxx](#).

Referenced by [GetPrivateCreator\(\)](#).

**10.314.3.27 SetElementTag()** [1/2]

```
void gdcm::Tag::SetElementTag (
    uint16_t group,
    uint16_t element ) [inline]
```

Sets the 'Group number' & 'Element number' of the given [Tag](#).

**10.314.3.28 SetElementTag()** [2/2]

```
void gdcm::Tag::SetElementTag (
    uint32_t tag ) [inline]
```

Sets the full tag value of the given [Tag](#).

**10.314.3.29 SetGroup()**

```
void gdcm::Tag::SetGroup (
    uint16_t group ) [inline]
```

Sets the 'Group number' of the given [Tag](#).

**10.314.3.30 SetPrivateCreator()**

```
void gdcm::Tag::SetPrivateCreator (
    Tag const & t ) [inline]
```

Set private creator:

**Examples**

[DuplicatePCDE.cxx](#).

References [GetElement\(\)](#), [GetGroup\(\)](#), and [IsPrivate\(\)](#).

**10.314.3.31 Write()**

```
template<typename TSwap >
const std::ostream & gdcm::Tag::Write (
    std::ostream & os ) const [inline]
```

Write a tag in binary rep.

Referenced by [gdcm::Item::Write\(\)](#), [gdcm::SequenceOfFragments::Write\(\)](#), and [gdcm::SequenceOfItems::Write\(\)](#).

## 10.314.4 Friends And Related Symbol Documentation

### 10.314.4.1 `operator<<`

```
std::ostream & operator<< (  
    std::ostream & _os,  
    const Tag & _val ) [friend]
```

### 10.314.4.2 `operator>>`

```
std::istream & operator>> (  
    std::istream & _is,  
    Tag & _val ) [friend]
```

## 10.314.5 Member Data Documentation

### 10.314.5.1 `bytes`

```
char gdcM::Tag::bytes[4]
```

### 10.314.5.2 `tag`

```
uint32_t gdcM::Tag::tag
```

Referenced by [Tag\(\)](#), [operator!=\(\)](#), [operator<\(\)](#), [operator=\(\)](#), and [operator==\(\)](#).

### 10.314.5.3 `tags`

```
uint16_t gdcM::Tag::tags[2]
```

Referenced by [operator<\(\)](#).

The documentation for this class was generated from the following file:

- [gdcMTag.h](#)

## 10.315 `gdcM::TagPath` Class Reference

class to handle a path of tag.

```
#include <gdcMTagPath.h>
```

## Public Member Functions

- [TagPath](#) ()
- [~TagPath](#) ()
- bool [ConstructFromString](#) (const char \*path)
- bool [ConstructFromTagList](#) (Tag const \*l, unsigned int n)  
*Construct from a list of tags.*
- void [Print](#) (std::ostream &) const
- bool [Push](#) (Tag const &t)
- bool [Push](#) (unsigned int itemnum)

## Static Public Member Functions

- static bool [IsValid](#) (const char \*path)  
*Return if path is valid or not.*

### 10.315.1 Detailed Description

class to handle a path of tag.

Any Resemblance to Existing XPath is Purely Coincidental [ftp://medical.nema.org/medical/dicom/supps/sup118←\\_pc.pdf](ftp://medical.nema.org/medical/dicom/supps/sup118←_pc.pdf)

### 10.315.2 Constructor & Destructor Documentation

#### 10.315.2.1 TagPath()

```
gdcm::TagPath::TagPath ( )
```

#### 10.315.2.2 ~TagPath()

```
gdcm::TagPath::~~TagPath ( )
```

### 10.315.3 Member Function Documentation

#### 10.315.3.1 ConstructFromString()

```
bool gdcm::TagPath::ConstructFromString (
    const char * path )
```

"/0018,0018/"... No space allowed, comma is use to separate tag group from tag element and slash is used to separate tag return false if invalid

### 10.315.3.2 ConstructFromTagList()

```
bool gdcM::TagPath::ConstructFromTagList (
    Tag const * l,
    unsigned int n )
```

Construct from a list of tags.

### 10.315.3.3 IsValid()

```
static bool gdcM::TagPath::IsValid (
    const char * path ) [static]
```

Return if path is valid or not.

### 10.315.3.4 Print()

```
void gdcM::TagPath::Print (
    std::ostream & ) const
```

### 10.315.3.5 Push() [1/2]

```
bool gdcM::TagPath::Push (
    Tag const & t )
```

### 10.315.3.6 Push() [2/2]

```
bool gdcM::TagPath::Push (
    unsigned int itemnum )
```

The documentation for this class was generated from the following file:

- [gdcMTagPath.h](#)

## 10.316 gdcM::Testing Class Reference

class for testing

```
#include <gdcMTesting.h>
```



## Public Types

- typedef const char \*const (\* [MD5DataImagesType](#))[2]
- typedef const char \*const (\* [MediaStorageDataFilesType](#))[2]  
*return the table that map the media storage (as string) of a filename (gdcmData)*

## Public Member Functions

- [Testing](#) ()=default
- [~Testing](#) ()=default
- void [Print](#) (std::ostream &os=std::cout)  
*Print.*

## Static Public Member Functions

- static bool [ComputeFileMD5](#) (const char \*filename, char digest\_str[33])
- static bool [ComputeMD5](#) (const char \*buffer, size\_t buf\_len, char digest\_str[33])
- static const char \* [GetDataExtraRoot](#) ()  
*Return the GDCM DATA EXTRA ROOT.*
- static const char \* [GetDataRoot](#) ()  
*Return the GDCM DATA ROOT.*
- static const char \* [GetFileName](#) (unsigned int file)
- static const char \*const \* [GetFileNames](#) ()  
*return the table of fullpath to gdcmData DICOM files:*
- static int [GetLossyFlagFromFile](#) (const char \*filepath)
- static const char \*const \* [GetMD5DataImage](#) (unsigned int file)
- static [MD5DataImagesType](#) [GetMD5DataImages](#) ()
- static const char \* [GetMD5FromBrokenFile](#) (const char \*filepath)
- static const char \* [GetMD5FromFile](#) (const char \*filepath)
- static const char \*const \* [GetMediaStorageDataFile](#) (unsigned int file)
- static [MediaStorageDataFilesType](#) [GetMediaStorageDataFiles](#) ()
- static const char \* [GetMediaStorageFromFile](#) (const char \*filepath)
- static unsigned int [GetNumberOfFileNames](#) ()
- static unsigned int [GetNumberOfMD5DataImages](#) ()
- static unsigned int [GetNumberOfMediaStorageDataFiles](#) ()
- static const char \* [GetPixelSpacingDataRoot](#) ()  
*Return the GDCM PIXEL SPACING DATA ROOT (See David Clunie website for dataset)*
- static std::streamoff [GetSelectedPrivateGroupOffsetFromFile](#) (const char \*filepath)
- static std::streamoff [GetSelectedTagsOffsetFromFile](#) (const char \*filepath)
- static const char \* [GetSourceDirectory](#) ()
- static std::streamoff [GetStreamOffsetFromFile](#) (const char \*filepath)
- static const char \* [GetTempDirectory](#) (const char \*subdir=nullptr)
- static const wchar\_t \* [GetTempDirectoryW](#) (const wchar\_t \*subdir=nullptr)  
*NOT THREAD SAFE.*
- static const char \* [GetTempFilename](#) (const char \*filename, const char \*subdir=nullptr)  
*NOT THREAD SAFE.*
- static const wchar\_t \* [GetTempFilenameW](#) (const wchar\_t \*filename, const wchar\_t \*subdir=nullptr)  
*NOT THREAD SAFE.*

### 10.316.1 Detailed Description

class for testing

this class is used for the nightly regression system for GDCM It makes heavily use of md5 computation

See also

[gdcm::MD5](#) class for md5 computation

### 10.316.2 Member Typedef Documentation

#### 10.316.2.1 MD5DataImagesType

```
typedef const char* const(* gdcm::Testing::MD5DataImagesType)[2]
```

return the table that map the md5 (as in md5sum) of the Pixel Data associated to a filename

#### 10.316.2.2 MediaStorageDataFileType

```
typedef const char* const(* gdcm::Testing::MediaStorageDataFileType)[2]
```

return the table that map the media storage (as string) of a filename (gdcmData)

### 10.316.3 Constructor & Destructor Documentation

#### 10.316.3.1 Testing()

```
gdcm::Testing::Testing ( ) [default]
```

#### 10.316.3.2 ~Testing()

```
gdcm::Testing::~~Testing ( ) [default]
```

### 10.316.4 Member Function Documentation

#### 10.316.4.1 ComputeFileMD5()

```
static bool gdcm::Testing::ComputeFileMD5 (
    const char * filename,
    char digest_str[33] ) [static]
```

Examples

[MetaImageMD5Activiz.cs](#).

#### 10.316.4.2 ComputeMD5()

```
static bool gdcmm::Testing::ComputeMD5 (
    const char * buffer,
    size_t buf_len,
    char digest_str[33] ) [static]
```

MD5 stuff *digest\_str* needs to be at least :  $\text{strlen} = [2 \times 16 + 1]$ ; string will be \0 padded. (md5 are 32 bytes long) [Testing](#) is not meant to be shipped with an installed GDCM release, always prefer the [gdcmm::MD5](#) API when doing md5 computation.

#### 10.316.4.3 GetDataExtraRoot()

```
static const char * gdcmm::Testing::GetDataExtraRoot ( ) [static]
```

Return the GDCM DATA EXTRA ROOT.

##### Examples

[DiscriminateVolume.cxx](#), [VolumeSorter.cxx](#), and [reslicesphere.cxx](#).

#### 10.316.4.4 GetDataRoot()

```
static const char * gdcmm::Testing::GetDataRoot ( ) [static]
```

Return the GDCM DATA ROOT.

##### Examples

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), and [MagnifyFile.cxx](#).

#### 10.316.4.5 GetFileName()

```
static const char * gdcmm::Testing::GetFileName (
    unsigned int file ) [static]
```

##### Examples

[MetalImageMD5Activiz.cs](#).

#### 10.316.4.6 GetFileNames()

```
static const char *const * gdcM::Testing::GetFileNames ( ) [static]
```

return the table of fullpath to gdcMData DICOM files:

##### Examples

[TestReader.cxx](#).

#### 10.316.4.7 GetLossyFlagFromFile()

```
static int gdcM::Testing::GetLossyFlagFromFile (
    const char * filepath ) [static]
```

Return the lossy flag of the given filename -1 -> Error 0 -> Lossless 1 -> Lossy

#### 10.316.4.8 GetMD5DataImage()

```
static const char *const * gdcM::Testing::GetMD5DataImage (
    unsigned int file ) [static]
```

#### 10.316.4.9 GetMD5DataImages()

```
static MD5DataImagesType gdcM::Testing::GetMD5DataImages ( ) [static]
```

#### 10.316.4.10 GetMD5FromBrokenFile()

```
static const char * gdcM::Testing::GetMD5FromBrokenFile (
    const char * filepath ) [static]
```

Return what should have been the md5 of file 'filepath' This is based on current GDCM implementation to decipher a broken DICOM file.

#### 10.316.4.11 GetMD5FromFile()

```
static const char * gdcM::Testing::GetMD5FromFile (
    const char * filepath ) [static]
```

#### 10.316.4.12 GetMediaStorageDataFile()

```
static const char *const * gdcM::Testing::GetMediaStorageDataFile (
    unsigned int file ) [static]
```

#### 10.316.4.13 GetMediaStorageDataFiles()

```
static MediaStorageDataFileType gdcm::Testing::GetMediaStorageDataFiles ( ) [static]
```

#### 10.316.4.14 GetMediaStorageFromFile()

```
static const char * gdcm::Testing::GetMediaStorageFromFile (
    const char * filepath ) [static]
```

##### Examples

[MetaImageMD5Activiz.cs](#), and [TestReader.cxx](#).

#### 10.316.4.15 GetNumberOfFileNames()

```
static unsigned int gdcm::Testing::GetNumberOfFileNames ( ) [static]
```

##### Examples

[MetaImageMD5Activiz.cs](#).

#### 10.316.4.16 GetNumberOfMD5DataImages()

```
static unsigned int gdcm::Testing::GetNumberOfMD5DataImages ( ) [static]
```

#### 10.316.4.17 GetNumberOfMediaStorageDataFiles()

```
static unsigned int gdcm::Testing::GetNumberOfMediaStorageDataFiles ( ) [static]
```

#### 10.316.4.18 GetPixelSpacingDataRoot()

```
static const char * gdcm::Testing::GetPixelSpacingDataRoot ( ) [static]
```

Return the GDCM PIXEL SPACING DATA ROOT (See David Clunie website for dataset)

#### 10.316.4.19 GetSelectedPrivateGroupOffsetFromFile()

```
static std::streamoff gdcm::Testing::GetSelectedPrivateGroupOffsetFromFile (
    const char * filepath ) [static]
```

Return the offset just after private attribute (0009,0010,"GEMS\_IDEN\_01") if found. Otherwise the offset of the next attribute -1 if not found

#### 10.316.4.20 GetSelectedTagsOffsetFromFile()

```
static std::streamoff gdcm::Testing::GetSelectedTagsOffsetFromFile (
    const char * filepath ) [static]
```

Return the offset just after Pixel Data Length (7fe0,0000) if found. Otherwise the offset of the very first pixel cell in Pixel Data -1 if not found

#### 10.316.4.21 GetSourceDirectory()

```
static const char * gdcm::Testing::GetSourceDirectory ( ) [static]
```

##### Examples

[BasicAnonymizer.cs](#), and [ClinicalTrialIdentificationWorkflow.cs](#).

#### 10.316.4.22 GetStreamOffsetFromFile()

```
static std::streamoff gdcm::Testing::GetStreamOffsetFromFile (
    const char * filepath ) [static]
```

Return the offset of the very first pixel cell in the PixelData -1 if not found

#### 10.316.4.23 GetTempDirectory()

```
static const char * gdcm::Testing::GetTempDirectory (
    const char * subdir = nullptr ) [static]
```

NOT THREAD SAFE Returns the temp directory as used in testing needing to output data:

##### Examples

[MetalImageMD5Activiz.cs](#).

#### 10.316.4.24 GetTempDirectoryW()

```
static const wchar_t * gdcm::Testing::GetTempDirectoryW (
    const wchar_t * subdir = nullptr ) [static]
```

NOT THREAD SAFE.

#### 10.316.4.25 GetTempFilename()

```
static const char * gdcm::Testing::GetTempFilename (
    const char * filename,
    const char * subdir = nullptr ) [static]
```

NOT THREAD SAFE.

##### Examples

[MetaImageMD5Activiz.cs](#).

#### 10.316.4.26 GetTempFilenameW()

```
static const wchar_t * gdcm::Testing::GetTempFilenameW (
    const wchar_t * filename,
    const wchar_t * subdir = nullptr ) [static]
```

NOT THREAD SAFE.

#### 10.316.4.27 Print()

```
void gdcm::Testing::Print (
    std::ostream & os = std::cout )
```

Print.

The documentation for this class was generated from the following file:

- [gdcmTesting.h](#)

## 10.317 gdcm::Trace Class Reference

[Trace](#).

```
#include <gdcmTrace.h>
```

### Public Member Functions

- [Trace](#) ()
- [~Trace](#) ()

### Static Public Member Functions

- static void [DebugOff](#) ()
- static void [DebugOn](#) ()
- static void [ErrorOff](#) ()
- static void [ErrorOn](#) ()
- static bool [GetDebugFlag](#) ()
- static std::ostream & [GetDebugStream](#) ()
- static bool [GetErrorFlag](#) ()
- static std::ostream & [GetErrorStream](#) ()
- static std::ostream & [GetStream](#) ()
- static bool [GetWarningFlag](#) ()
- static std::ostream & [GetWarningStream](#) ()
- static void [SetDebug](#) (bool debug)  
*Turn debug messages on (default: false)*
- static void [SetDebugStream](#) (std::ostream &os)  
*Explicitly set the stream which receive Debug messages:*
- static void [SetError](#) (bool debug)  
*Turn error messages on (default: true)*
- static void [SetErrorStream](#) (std::ostream &os)  
*Explicitly set the stream which receive Error messages:*
- static void [SetStream](#) (std::ostream &os)
- static void [SetStreamToFile](#) (const char \*filename)
- static void [SetWarning](#) (bool debug)  
*Turn warning messages on (default: true)*
- static void [SetWarningStream](#) (std::ostream &os)  
*Explicitly set the stream which receive Warning messages:*
- static void [WarningOff](#) ()
- static void [WarningOn](#) ()

### 10.317.1 Detailed Description

[Trace](#).

Debug / Warning and Error are encapsulated in this class by default the [Trace](#) class will redirect any debug/warning/error to std::cerr. Unless SetStream was specified with another (open) stream or SetStreamToFile was specified to a writable file on the system.

#### Warning

All string messages are removed during compilation time when compiled with CMAKE\_BUILD\_TYPE being set to either:

- Release
- MinSizeRel It is recommended to compile with RelWithDebInfo and/or Debug during prototyping of applications.

#### Examples

[DecompressJPEGFile.cs](#).



## 10.317.2 Constructor & Destructor Documentation

### 10.317.2.1 Trace()

```
gdcmm::Trace::Trace ( )
```

### 10.317.2.2 ~Trace()

```
gdcmm::Trace::~~Trace ( )
```

## 10.317.3 Member Function Documentation

### 10.317.3.1 DebugOff()

```
static void gdcmm::Trace::DebugOff ( ) [static]
```

#### Examples

[MetalImageMD5Activiz.cs](#), and [TestReader.cxx](#).

### 10.317.3.2 DebugOn()

```
static void gdcmm::Trace::DebugOn ( ) [static]
```

#### Examples

[CreateFakePET.cxx](#), [DecompressJPEGFile.cs](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

### 10.317.3.3 ErrorOff()

```
static void gdcmm::Trace::ErrorOff ( ) [static]
```

#### Examples

[MetalImageMD5Activiz.cs](#).

### 10.317.3.4 ErrorOn()

```
static void gdcmm::Trace::ErrorOn ( ) [static]
```

#### 10.317.3.5 GetDebugFlag()

```
static bool gdcm::Trace::GetDebugFlag ( ) [static]
```

#### 10.317.3.6 GetDebugStream()

```
static std::ostream & gdcm::Trace::GetDebugStream ( ) [static]
```

#### 10.317.3.7 GetErrorFlag()

```
static bool gdcm::Trace::GetErrorFlag ( ) [static]
```

#### 10.317.3.8 GetErrorStream()

```
static std::ostream & gdcm::Trace::GetErrorStream ( ) [static]
```

#### 10.317.3.9 GetStream()

```
static std::ostream & gdcm::Trace::GetStream ( ) [static]
```

#### 10.317.3.10 GetWarningFlag()

```
static bool gdcm::Trace::GetWarningFlag ( ) [static]
```

#### 10.317.3.11 GetWarningStream()

```
static std::ostream & gdcm::Trace::GetWarningStream ( ) [static]
```

#### 10.317.3.12 SetDebug()

```
static void gdcm::Trace::SetDebug (
    bool debug ) [static]
```

Turn debug messages on (default: false)

#### Examples

[DumpToSQLITE3.cxx](#).

### 10.317.3.13 SetDebugStream()

```
static void gdcm::Trace::SetDebugStream (
    std::ostream & os ) [static]
```

Explicitly set the stream which receive Debug messages:

### 10.317.3.14 SetError()

```
static void gdcm::Trace::SetError (
    bool debug ) [static]
```

Turn error messages on (default: true)

### 10.317.3.15 SetErrorStream()

```
static void gdcm::Trace::SetErrorStream (
    std::ostream & os ) [static]
```

Explicitly set the stream which receive Error messages:

#### Examples

[CStoreQtProgress.cxx](#).

### 10.317.3.16 SetStream()

```
static void gdcm::Trace::SetStream (
    std::ostream & os ) [static]
```

Explicitly set the ostream for [gdcm::Trace](#) to report to This will set the DebugStream, WarningStream and ErrorStream at once:

### 10.317.3.17 SetStreamToFile()

```
static void gdcm::Trace::SetStreamToFile (
    const char * filename ) [static]
```

Explicitly set the filename for [gdcm::Trace](#) to report to The file will be created (it will not append to existing file)

#### 10.317.3.18 SetWarning()

```
static void gdcm::Trace::SetWarning (
    bool debug ) [static]
```

Turn warning messages on (default: true)

##### Examples

[DumpToSQLITE3.cxx](#).

#### 10.317.3.19 SetWarningStream()

```
static void gdcm::Trace::SetWarningStream (
    std::ostream & os ) [static]
```

Explicitly set the stream which receive Warning messages:

#### 10.317.3.20 WarningOff()

```
static void gdcm::Trace::WarningOff ( ) [static]
```

##### Examples

[MetalImageMD5Activiz.cs](#), and [TestReader.cxx](#).

#### 10.317.3.21 WarningOn()

```
static void gdcm::Trace::WarningOn ( ) [static]
```

##### Examples

[Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

The documentation for this class was generated from the following file:

- [gdcmTrace.h](#)

## 10.318 gdcm::TransferSyntax Class Reference

Class to manipulate Transfer Syntax.

```
#include <gdcmTransferSyntax.h>
```

## Public Types

- enum [NegociatedType](#) {  
    [Unknown](#) = 0 ,  
    [Explicit](#) ,  
    [Implicit](#) }
- enum [TSType](#) {  
    [ImplicitVRLittleEndian](#) = 0 ,  
    [ImplicitVRBigEndianPrivateGE](#) ,  
    [ExplicitVRLittleEndian](#) ,  
    [DeflatedExplicitVRLittleEndian](#) ,  
    [ExplicitVRBigEndian](#) ,  
    [JPEGBaselineProcess1](#) ,  
    [JPEGExtendedProcess2\\_4](#) ,  
    [JPEGExtendedProcess3\\_5](#) ,  
    [JPEGSpectralSelectionProcess6\\_8](#) ,  
    [JPEGFullProgressionProcess10\\_12](#) ,  
    [JPEGLosslessProcess14](#) ,  
    [JPEGLosslessProcess14\\_1](#) ,  
    [JPEGLSLossless](#) ,  
    [JPEGLSNearLossless](#) ,  
    [JPEG2000Lossless](#) ,  
    [JPEG2000](#) ,  
    [JPEG2000Part2Lossless](#) ,  
    [JPEG2000Part2](#) ,  
    [RLELossless](#) ,  
    [MPEG2MainProfile](#) ,  
    [ImplicitVRBigEndianACRNEMA](#) ,  
    [WeirdPapryus](#) ,  
    [CT\\_private\\_ELE](#) ,  
    [JPIPReferenced](#) ,  
    [MPEG2MainProfileHighLevel](#) ,  
    [MPEG4AVCH264HighProfileLevel4\\_1](#) ,  
    [MPEG4AVCH264BDcompatibleHighProfileLevel4\\_1](#) ,  
    [TS\\_END](#) }

## Public Member Functions

- [TransferSyntax](#) ([TSType](#) type=[ImplicitVRLittleEndian](#))
- bool [CanStoreLossy](#) () const
- [NegociatedType](#) [GetNegociatedType](#) () const
- const char \* [GetString](#) () const
- [SwapCode](#) [GetSwapCode](#) () const
- bool [IsEncapsulated](#) () const
- bool [IsEncoded](#) () const
- bool [IsExplicit](#) () const
- bool [IsImplicit](#) () const
- bool [IsLossless](#) () const
- bool [IsLossy](#) () const
- bool [IsValid](#) () const
- [operator TSType](#) () const

### Static Public Member Functions

- static const char \* [GetTSSString](#) (TSType ts)
- static TSType [GetTSType](#) (const char \*str)

### Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [TransferSyntax](#) &ts)

## 10.318.1 Detailed Description

Class to manipulate Transfer Syntax.

### Note

TRANSFER SYNTAX (Standard and Private): A set of encoding rules that allow Application Entities to unambiguously negotiate the encoding techniques (e.g., Data [Element](#) structure, byte ordering, compression) they are able to support, thereby allowing these Application Entities to communicate.

**Todo** : The implementation is completely retarded -> see [gdcm::UIDs](#) for a replacement We need: IsSupported We need preprocess of raw/xml file We need GetFullName()

Need a notion of Private Syntax. As defined in PS 3.5. Section 9.2

### See also

[UIDs](#)

### Examples

[BasicImageAnonymizer.cs](#), [CompressLossyJPEG.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [GetJPEGSamplePrecision.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [MpegVideoInfo.cs](#), and [StandardizeFiles.cs](#).

## 10.318.2 Member Enumeration Documentation

### 10.318.2.1 NegotiatedType

```
enum gdcm::TransferSyntax::NegociatedType
```

#### Enumerator

Unknown	
Explicit	
Implicit	

### 10.318.2.2 TSType

```
enum gdcm::TransferSyntax::TSType
```

#### Enumerator

ImplicitVRLittleEndian	
ImplicitVRBigEndianPrivateGE	
ExplicitVRLittleEndian	
DeflatedExplicitVRLittleEndian	
ExplicitVRBigEndian	
JPEGBaselineProcess1	
JPEGExtendedProcess2_4	
JPEGExtendedProcess3_5	
JPEGSpectralSelectionProcess6_8	
JPEGFullProgressionProcess10_12	
JPEGLosslessProcess14	
JPEGLosslessProcess14_1	
JPEGLSLossless	
JPEGLSNearLossless	
JPEG2000Lossless	
JPEG2000	
JPEG2000Part2Lossless	
JPEG2000Part2	
RLELossless	
MPEG2MainProfile	
ImplicitVRBigEndianACRNEMA	
WeirdPapryus	
CT_private_ELE	
JPIPRreferenced	
MPEG2MainProfileHighLevel	
MPEG4AVCH264HighProfileLevel4_1	
MPEG4AVCH264BDcompatibleHighProfileLevel4↔ _1	
TS_END	

#### Examples

[BasicImageAnonymizer.cs](#), [CompressLossyJPEG.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [MpegVideoInfo.cs](#), and [StandardizeFiles.cs](#).

## 10.318.3 Constructor & Destructor Documentation

### 10.318.3.1 TransferSyntax()

```
gdcm::TransferSyntax::TransferSyntax (
    TSType type = ImplicitVRLittleEndian ) [inline]
```

## 10.318.4 Member Function Documentation

### 10.318.4.1 CanStoreLossy()

```
bool gdcm::TransferSyntax::CanStoreLossy ( ) const
```

return true if TransFer Syntax Allow storing of Lossy Pixel Data

### 10.318.4.2 GetNegociatedType()

```
NegociatedType gdcm::TransferSyntax::GetNegociatedType ( ) const
```

### 10.318.4.3 GetString()

```
const char * gdcm::TransferSyntax::GetString ( ) const [inline]
```

### 10.318.4.4 GetSwapCode()

```
SwapCode gdcm::TransferSyntax::GetSwapCode ( ) const
```

**Deprecated** Return the [SwapCode](#) associated with the Transfer Syntax. Be careful with the special GE private syntax the [DataSet](#) is written in little endian but the Pixel Data is in Big Endian.

### 10.318.4.5 GetTSString()

```
static const char * gdcm::TransferSyntax::GetTSString (
    TSType ts ) [static]
```

#### Examples

[LargeVRDSExplicit.cxx](#).

### 10.318.4.6 GetTSType()

```
static TSType gdcm::TransferSyntax::GetTSType (
    const char * str ) [static]
```



#### 10.318.4.7 IsEncapsulated()

```
bool gdcm::TransferSyntax::IsEncapsulated ( ) const
```

##### Examples

[ExtractIconFromFile.cxx](#).

#### 10.318.4.8 IsEncoded()

```
bool gdcm::TransferSyntax::IsEncoded ( ) const
```

#### 10.318.4.9 IsExplicit()

```
bool gdcm::TransferSyntax::IsExplicit ( ) const
```

#### 10.318.4.10 IsImplicit()

```
bool gdcm::TransferSyntax::IsImplicit ( ) const
```

#### 10.318.4.11 IsLossless()

```
bool gdcm::TransferSyntax::IsLossless ( ) const
```

Return true if the transfer syntax algorithm is a lossless algorithm

#### 10.318.4.12 IsLossy()

```
bool gdcm::TransferSyntax::IsLossy ( ) const
```

Return true if the transfer syntax algorithm is a lossy algorithm

#### 10.318.4.13 IsValid()

```
bool gdcm::TransferSyntax::IsValid ( ) const [inline]
```

#### 10.318.4.14 operator TSType()

```
gdcm::TransferSyntax::operator TSType ( ) const [inline]
```

## 10.318.5 Friends And Related Symbol Documentation

### 10.318.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & os,
    const TransferSyntax & ts ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmTransferSyntax.h](#)

## 10.319 gdcm::network::TransferSyntaxSub Class Reference

[TransferSyntaxSub.](#)

```
#include <gdcmTransferSyntaxSub.h>
```

### Public Member Functions

- [TransferSyntaxSub](#) ()
- const char \* [GetName](#) () const
- bool [operator==](#) (const [TransferSyntaxSub](#) &ts) const
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [SetName](#) (const char \*name)
- void [SetNameFromUID](#) (UIDs::TSName tsname)
- size\_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

### 10.319.1 Detailed Description

[TransferSyntaxSub.](#)

[Table](#) 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS

TODO what is the goal of :

[Table](#) 9-19 TRANSFER SYNTAX SUB-ITEM FIELDS

## 10.319.2 Constructor & Destructor Documentation

### 10.319.2.1 TransferSyntaxSub()

```
gdcm::network::TransferSyntaxSub::TransferSyntaxSub ( )
```

## 10.319.3 Member Function Documentation

### 10.319.3.1 GetName()

```
const char * gdcm::network::TransferSyntaxSub::GetName ( ) const [inline]
```

### 10.319.3.2 operator==(

```
bool gdcm::network::TransferSyntaxSub::operator== (
    const TransferSyntaxSub & ts ) const [inline]
```

### 10.319.3.3 Print()

```
void gdcm::network::TransferSyntaxSub::Print (
    std::ostream & os ) const
```

### 10.319.3.4 Read()

```
std::istream & gdcm::network::TransferSyntaxSub::Read (
    std::istream & is )
```

### 10.319.3.5 SetName()

```
void gdcm::network::TransferSyntaxSub::SetName (
    const char * name )
```

### 10.319.3.6 SetNameFromUID()

```
void gdcm::network::TransferSyntaxSub::SetNameFromUID (
    UIDs::TSName tsname )
```

### 10.319.3.7 Size()

```
size_t gdcm::network::TransferSyntaxSub::Size ( ) const
```

### 10.319.3.8 Write()

```
const std::ostream & gdcm::network::TransferSyntaxSub::Write (
    std::ostream & os ) const
```

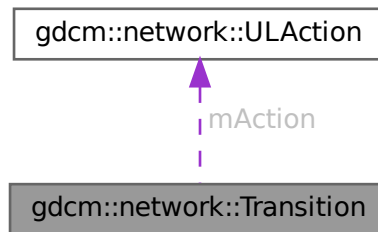
The documentation for this class was generated from the following file:

- [gdcmTransferSyntaxSub.h](#)

## 10.320 gdcmm::network::Transition Struct Reference

```
#include <gdcmmULTransitionTable.h>
```

Collaboration diagram for gdcmm::network::Transition:



### Public Member Functions

- [Transition](#) ()
- [Transition](#) (int inEndState, [ULAction](#) \*inAction)
- [~Transition](#) ()

### Static Public Member Functions

- static [Transition](#) \* [MakeNew](#) (int inEndState, [ULAction](#) \*inAction)

### Public Attributes

- [ULAction](#) \* [mAction](#)
- int [mEnd](#)

## 10.320.1 Constructor & Destructor Documentation

### 10.320.1.1 Transition() [1/2]

```
gdcmm::network::Transition::Transition ( ) [inline]
```

References [gdcmm::network::eStaDoesNotExist](#), [mAction](#), and [mEnd](#).

Referenced by [MakeNew\(\)](#).

### 10.320.1.2 ~Transition()

```
gdcm::network::Transition::~~Transition ( ) [inline]
```

References [mAction](#).

### 10.320.1.3 Transition() [2/2]

```
gdcm::network::Transition::Transition (
    int inEndState,
    ULAction * inAction ) [inline]
```

References [mAction](#), and [mEnd](#).

## 10.320.2 Member Function Documentation

### 10.320.2.1 MakeNew()

```
static Transition * gdcm::network::Transition::MakeNew (
    int inEndState,
    ULAction * inAction ) [inline], [static]
```

References [Transition\(\)](#).

## 10.320.3 Member Data Documentation

### 10.320.3.1 mAction

```
ULAction* gdcm::network::Transition::mAction
```

Referenced by [Transition\(\)](#), [Transition\(\)](#), and [~Transition\(\)](#).

### 10.320.3.2 mEnd

```
int gdcm::network::Transition::mEnd
```

Referenced by [Transition\(\)](#), and [Transition\(\)](#).

The documentation for this struct was generated from the following file:

- [gdcmULTransitionTable.h](#)

## 10.321 gdcm::Type Class Reference

Type.

```
#include <gdcmType.h>
```

### Public Types

- enum [TypeType](#) {  
    [T1](#) = 0 ,  
    [T1C](#) ,  
    [T2](#) ,  
    [T2C](#) ,  
    [T3](#) ,  
    [UNKNOWN](#) }

### Public Member Functions

- [Type](#) ([TypeType](#) type=[UNKNOWN](#))
- [operator TypeType](#) () const

### Static Public Member Functions

- static const char \* [GetTypeString](#) ([TypeType](#) type)
- static [TypeType](#) [GetTypeType](#) (const char \*type)

### Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [Type](#) &vr)

### 10.321.1 Detailed Description

Type.

#### Note

PS 3.5 7.4 DATA ELEMENT TYPE 7.4.1 TYPE 1 REQUIRED DATA ELEMENTS 7.4.2 TYPE 1C CONDITIONAL DATA ELEMENTS 7.4.3 TYPE 2 REQUIRED DATA ELEMENTS 7.4.4 TYPE 2C CONDITIONAL DATA ELEMENTS 7.4.5 TYPE 3 OPTIONAL DATA ELEMENTS

The intent of [Type](#) 2 Data Elements is to allow a zero length to be conveyed when the operator or application does not know its value or has a specific reason for not specifying its value. It is the intent that the device should support these Data Elements.

#### Examples

[TraverseModules.cxx](#).

### 10.321.2 Member Enumeration Documentation

#### 10.321.2.1 TypeType

```
enum gdcm::Type::TypeType
```

## Enumerator

T1	
T1C	
T2	
T2C	
T3	
UNKNOWN	

### 10.321.3 Constructor & Destructor Documentation

#### 10.321.3.1 Type()

```
gdcm::Type::Type (  
    TypeType type = UNKNOWN ) [inline]
```

### 10.321.4 Member Function Documentation

#### 10.321.4.1 GetTypeString()

```
static const char * gdcm::Type::GetTypeString (  
    TypeType type ) [static]
```

#### 10.321.4.2 GetTypeType()

```
static TypeType gdcm::Type::GetTypeType (  
    const char * type ) [static]
```

#### 10.321.4.3 operator TypeType()

```
gdcm::Type::operator TypeType ( ) const [inline]
```

### 10.321.5 Friends And Related Symbol Documentation

#### 10.321.5.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & os,  
    const Type & vr ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmType.h](#)

## 10.322 gdcm::UI Struct Reference

```
#include <gdcmVR.h>
```

### Public Attributes

- char [Internal](#) [64+1]

### Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [UI](#) &\_val)

## 10.322.1 Friends And Related Symbol Documentation

### 10.322.1.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & _os,  
    const UI & _val ) [friend]
```

## 10.322.2 Member Data Documentation

### 10.322.2.1 Internal

```
char gdcm::UI::Internal[64+1]
```

The documentation for this struct was generated from the following file:

- [gdcmVR.h](#)

## 10.323 gdcm::UIDGenerator Class Reference

Class for generating unique UID.

```
#include <gdcmUIDGenerator.h>
```

### Public Member Functions

- [UIDGenerator](#) ()  
*By default the root of a UID is a GDCM Root...*
- const char \* [Generate](#) ()



## Static Public Member Functions

- static const char \* [GetGDCMUID](#) ()  
*Return the default (GDCM) root UID:*
- static const char \* [GetRoot](#) ()
- static bool [IsValid](#) (const char \*uid)
- static void [SetRoot](#) (const char \*root)

## Static Protected Member Functions

- static bool [GenerateUUID](#) (unsigned char \*uuid\_data)

### 10.323.1 Detailed Description

Class for generating unique UID.

When constructing a [Series](#) or [Study](#) UID, user *has* to keep around the UID, otherwise the UID Generator will simply forget the value and create a new UID.

#### Examples

[CreateJPIPDataSet.cxx](#), [EncapsulateFileInRawData.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [ManipulateFile.cs](#), [MpegVideoInfo.cs](#), [ReformatFile.cs](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), and [uid\\_unique.cxx](#).

### 10.323.2 Constructor & Destructor Documentation

#### 10.323.2.1 UIDGenerator()

```
gdcm::UIDGenerator::UIDGenerator ( ) [inline]
```

By default the root of a UID is a GDCM Root...

### 10.323.3 Member Function Documentation

#### 10.323.3.1 Generate()

```
const char * gdcm::UIDGenerator::Generate ( )
```

Internally uses a std::string, so two calls have the same pointer ! save into a std::string In summary do not write code like that: const char \*uid1 = uid.Generate(); const char \*uid2 = uid.Generate(); since uid1 == uid2

#### Examples

[CreateJPIPDataSet.cxx](#), [EncapsulateFileInRawData.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [ManipulateFile.cs](#), [ReformatFile.cs](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), and [uid\\_unique.cxx](#).

### 10.323.3.2 GenerateUUID()

```
static bool gdcM::UIDGenerator::GenerateUUID (
    unsigned char * uuid_data ) [static], [protected]
```

### 10.323.3.3 GetGDCMUID()

```
static const char * gdcM::UIDGenerator::GetGDCMUID ( ) [static]
```

Return the default (GDCM) root UID:

### 10.323.3.4 GetRoot()

```
static const char * gdcM::UIDGenerator::GetRoot ( ) [static]
```

#### Examples

[ClinicalTrialIdentificationWorkflow.cs](#), [ReformatFile.cs](#), and [StandardizeFiles.cs](#).

### 10.323.3.5 IsValid()

```
static bool gdcM::UIDGenerator::IsValid (
    const char * uid ) [static]
```

Find out if the string is a valid UID or not

**Todo** : Move that in DataStructureAndEncoding (see FileMetaInformation::CheckFileMetaInformation)

### 10.323.3.6 SetRoot()

```
static void gdcM::UIDGenerator::SetRoot (
    const char * root ) [static]
```

The current implementation in GDCM make use of the UUID implementation (RFC 4122) and has been successfully been tested for a root of size 26 bytes. Any longer root should work (the [Generate\(\)](#) function will return a string), but will truncate the high bits of the 128bits UUID until the generated string fits on 64 bits. The authors disclaims any responsability for guaranteeing uniqueness of [UIDs](#) when the root is longer than 26 bytes.

#### Examples

[ClinicalTrialIdentificationWorkflow.cs](#), [ReformatFile.cs](#), [StandardizeFiles.cs](#), and [uid\\_unique.cxx](#).

The documentation for this class was generated from the following file:

- [gdcMUIDGenerator.h](#)

## 10.324 gdcm::UIDs Class Reference

all known uids

```
#include <gdcmUIDs.h>
```

### Public Types

- typedef const char \*const (\* [TransferSyntaxStringsType](#))[2]
- enum [TSName](#) {
  - [VerificationSOPClass](#) = 1 ,
  - [ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM](#) = 2 ,
  - [ExplicitVRLittleEndian](#) = 3 ,
  - [DeflatedExplicitVRLittleEndian](#) = 4 ,
  - [ExplicitVRBigEndian](#) = 5 ,
  - [JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImageCompression](#) = 6 ,
  - [JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG12BitImageCompressionProcess4only](#) = 7 ,
  - [JPEGExtendedProcess35Retired](#) = 8 ,
  - [JPEGSpectralSelectionNonHierarchicalProcess68Retired](#) = 9 ,
  - [JPEGSpectralSelectionNonHierarchicalProcess79Retired](#) = 10 ,
  - [JPEGFullProgressionNonHierarchicalProcess1012Retired](#) = 11 ,
  - [JPEGFullProgressionNonHierarchicalProcess1113Retired](#) = 12 ,
  - [JPEGLosslessNonHierarchicalProcess14](#) = 13 ,
  - [JPEGLosslessNonHierarchicalProcess15Retired](#) = 14 ,
  - [JPEGExtendedHierarchicalProcess1618Retired](#) = 15 ,
  - [JPEGExtendedHierarchicalProcess1719Retired](#) = 16 ,
  - [JPEGSpectralSelectionHierarchicalProcess2022Retired](#) = 17 ,
  - [JPEGSpectralSelectionHierarchicalProcess2123Retired](#) = 18 ,
  - [JPEGFullProgressionHierarchicalProcess2426Retired](#) = 19 ,
  - [JPEGFullProgressionHierarchicalProcess2527Retired](#) = 20 ,
  - [JPEGLosslessHierarchicalProcess28Retired](#) = 21 ,
  - [JPEGLosslessHierarchicalProcess29Retired](#) = 22 ,
  - [JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLosslessJPEGImageCompression](#) = 23 ,
  - [JPEGLSLosslessImageCompression](#) = 24 ,
  - [JPEGLSLossyNearLosslessImageCompression](#) = 25 ,
  - [JPEG2000ImageCompressionLosslessOnly](#) = 26 ,
  - [JPEG2000ImageCompression](#) = 27 ,
  - [JPEG2000Part2MulticomponentImageCompressionLosslessOnly](#) = 28 ,
  - [JPEG2000Part2MulticomponentImageCompression](#) = 29 ,
  - [JPIPIReferenced](#) = 30 ,
  - [JPIPIReferencedDeflate](#) = 31 ,
  - [MPEG2MainProfileMainLevel](#) = 32 ,
  - [RLELossless](#) = 33 ,
  - [RFC2557MIMEencapsulation](#) = 34 ,
  - [XMLEncoding](#) = 35 ,
  - [MediaStorageDirectoryStorage](#) = 36 ,
  - [TalairachBrainAtlasFrameofReference](#) = 37 ,
  - [SPM2T1FrameofReference](#) = 38 ,
  - [SPM2T2FrameofReference](#) = 39 ,
  - [SPM2PDFFrameofReference](#) = 40 ,
  - [SPM2EPIFrameofReference](#) = 41 ,

[SPM2FILT1FrameofReference](#) = 42 ,  
[SPM2PETFrameofReference](#) = 43 ,  
[SPM2TRANSMFrameofReference](#) = 44 ,  
[SPM2SPECTFrameofReference](#) = 45 ,  
[SPM2GRAYFrameofReference](#) = 46 ,  
[SPM2WHITEFrameofReference](#) = 47 ,  
[SPM2CSFFFrameofReference](#) = 48 ,  
[SPM2BRAINMASKFrameofReference](#) = 49 ,  
[SPM2AVG305T1FrameofReference](#) = 50 ,  
[SPM2AVG152T1FrameofReference](#) = 51 ,  
[SPM2AVG152T2FrameofReference](#) = 52 ,  
[SPM2AVG152PDFrameofReference](#) = 53 ,  
[SPM2SINGLESUBJT1FrameofReference](#) = 54 ,  
[ICBM452T1FrameofReference](#) = 55 ,  
[ICBMSingleSubjectMRIFrameofReference](#) = 56 ,  
[BasicStudyContentNotificationSOPClassRetired](#) = 57 ,  
[StorageCommitmentPushModelSOPClass](#) = 58 ,  
[StorageCommitmentPushModelSOPInstance](#) = 59 ,  
[StorageCommitmentPullModelSOPClassRetired](#) = 60 ,  
[StorageCommitmentPullModelSOPInstanceRetired](#) = 61 ,  
[ProceduralEventLoggingSOPClass](#) = 62 ,  
[ProceduralEventLoggingSOPInstance](#) = 63 ,  
[SubstanceAdministrationLoggingSOPClass](#) = 64 ,  
[SubstanceAdministrationLoggingSOPInstance](#) = 65 ,  
[DICOMUIDRegistry](#) = 66 ,  
[DICOMControlledTerminology](#) = 67 ,  
[DICOMApplicationContextName](#) = 68 ,  
[DetachedPatientManagementSOPClassRetired](#) = 69 ,  
[DetachedPatientManagementMetaSOPClassRetired](#) = 70 ,  
[DetachedVisitManagementSOPClassRetired](#) = 71 ,  
[DetachedStudyManagementSOPClassRetired](#) = 72 ,  
[StudyComponentManagementSOPClassRetired](#) = 73 ,  
[ModalityPerformedProcedureStepSOPClass](#) = 74 ,  
[ModalityPerformedProcedureStepRetrieveSOPClass](#) = 75 ,  
[ModalityPerformedProcedureStepNotificationSOPClass](#) = 76 ,  
[DetachedResultsManagementSOPClassRetired](#) = 77 ,  
[DetachedResultsManagementMetaSOPClassRetired](#) = 78 ,  
[DetachedStudyManagementMetaSOPClassRetired](#) = 79 ,  
[DetachedInterpretationManagementSOPClassRetired](#) = 80 ,  
[StorageServiceClass](#) = 81 ,  
[BasicFilmSessionSOPClass](#) = 82 ,  
[BasicFilmBoxSOPClass](#) = 83 ,  
[BasicGrayscaleImageBoxSOPClass](#) = 84 ,  
[BasicColorImageBoxSOPClass](#) = 85 ,  
[ReferencedImageBoxSOPClassRetired](#) = 86 ,  
[BasicGrayscalePrintManagementMetaSOPClass](#) = 87 ,  
[ReferencedGrayscalePrintManagementMetaSOPClassRetired](#) = 88 ,  
[PrintJobSOPClass](#) = 89 ,  
[BasicAnnotationBoxSOPClass](#) = 90 ,  
[PrinterSOPClass](#) = 91 ,  
[PrinterConfigurationRetrievalSOPClass](#) = 92 ,  
[PrinterSOPInstance](#) = 93 ,  
[PrinterConfigurationRetrievalSOPInstance](#) = 94 ,  
[BasicColorPrintManagementMetaSOPClass](#) = 95 ,

[ReferencedColorPrintManagementMetaSOPClassRetired](#) = 96 ,  
[VOILUTBoxSOPClass](#) = 97 ,  
[PresentationLUTSOPClass](#) = 98 ,  
[ImageOverlayBoxSOPClassRetired](#) = 99 ,  
[BasicPrintImageOverlayBoxSOPClassRetired](#) = 100 ,  
[PrintQueueSOPInstanceRetired](#) = 101 ,  
[PrintQueueManagementSOPClassRetired](#) = 102 ,  
[StoredPrintStorageSOPClassRetired](#) = 103 ,  
[HardcopyGrayscaleImageStorageSOPClassRetired](#) = 104 ,  
[HardcopyColorImageStorageSOPClassRetired](#) = 105 ,  
[PullPrintRequestSOPClassRetired](#) = 106 ,  
[PullStoredPrintManagementMetaSOPClassRetired](#) = 107 ,  
[MediaCreationManagementSOPClassUID](#) = 108 ,  
[ComputedRadiographyImageStorage](#) = 109 ,  
[DigitalXRayImageStorageForPresentation](#) = 110 ,  
[DigitalXRayImageStorageForProcessing](#) = 111 ,  
[DigitalMammographyXRayImageStorageForPresentation](#) = 112 ,  
[DigitalMammographyXRayImageStorageForProcessing](#) = 113 ,  
[DigitalIntraoralXRayImageStorageForPresentation](#) = 114 ,  
[DigitalIntraoralXRayImageStorageForProcessing](#) = 115 ,  
[CTImageStorage](#) = 116 ,  
[EnhancedCTImageStorage](#) = 117 ,  
[UltrasoundMultiframeImageStorageRetired](#) = 118 ,  
[UltrasoundMultiframeImageStorage](#) = 119 ,  
[MRIImageStorage](#) = 120 ,  
[EnhancedMRIImageStorage](#) = 121 ,  
[MRSpectroscopyStorage](#) = 122 ,  
[NuclearMedicineImageStorageRetired](#) = 123 ,  
[UltrasoundImageStorageRetired](#) = 124 ,  
[UltrasoundImageStorage](#) = 125 ,  
[SecondaryCaptureImageStorage](#) = 126 ,  
[MultiframeSingleBitSecondaryCaptureImageStorage](#) = 127 ,  
[MultiframeGrayscaleByteSecondaryCaptureImageStorage](#) = 128 ,  
[MultiframeGrayscaleWordSecondaryCaptureImageStorage](#) = 129 ,  
[MultiframeTrueColorSecondaryCaptureImageStorage](#) = 130 ,  
[StandaloneOverlayStorageRetired](#) = 131 ,  
[StandaloneCurveStorageRetired](#) = 132 ,  
[WaveformStorageTrialRetired](#) = 133 ,  
[ECG12leadWaveformStorage](#) = 134 ,  
[GeneralECGWaveformStorage](#) = 135 ,  
[AmbulatoryECGWaveformStorage](#) = 136 ,  
[HemodynamicWaveformStorage](#) = 137 ,  
[CardiacElectrophysiologyWaveformStorage](#) = 138 ,  
[BasicVoiceAudioWaveformStorage](#) = 139 ,  
[StandaloneModalityLUTStorageRetired](#) = 140 ,  
[StandaloneVOILUTStorageRetired](#) = 141 ,  
[GrayscaleSoftcopyPresentationStateStorageSOPClass](#) = 142 ,  
[ColorSoftcopyPresentationStateStorageSOPClass](#) = 143 ,  
[PseudoColorSoftcopyPresentationStateStorageSOPClass](#) = 144 ,  
[BlendingSoftcopyPresentationStateStorageSOPClass](#) = 145 ,  
[XRayAngiographicImageStorage](#) = 146 ,  
[EnhancedXAImageStorage](#) = 147 ,  
[XRayRadiofluoroscopicImageStorage](#) = 148 ,  
[EnhancedXRFImageStorage](#) = 149 ,

[XRay3DAngiographicImageStorage](#) = 150 ,  
[XRay3DCraniofacialImageStorage](#) = 151 ,  
[XRayAngiographicBiPlaneImageStorageRetired](#) = 152 ,  
[NuclearMedicineImageStorage](#) = 153 ,  
[RawDataStorage](#) = 154 ,  
[SpatialRegistrationStorage](#) = 155 ,  
[SpatialFiducialsStorage](#) = 156 ,  
[DeformableSpatialRegistrationStorage](#) = 157 ,  
[SegmentationStorage](#) = 158 ,  
[RealWorldValueMappingStorage](#) = 159 ,  
[VLImageStorageTrialRetired](#) = 160 ,  
[VLMultiframeImageStorageTrialRetired](#) = 161 ,  
[VLEndoscopicImageStorage](#) = 162 ,  
[VideoEndoscopicImageStorage](#) = 163 ,  
[VLMicroscopicImageStorage](#) = 164 ,  
[VideoMicroscopicImageStorage](#) = 165 ,  
[VLSlideCoordinatesMicroscopicImageStorage](#) = 166 ,  
[VLPhotographicImageStorage](#) = 167 ,  
[VideoPhotographicImageStorage](#) = 168 ,  
[OphthalmicPhotography8BitImageStorage](#) = 169 ,  
[OphthalmicPhotography16BitImageStorage](#) = 170 ,  
[StereometricRelationshipStorage](#) = 171 ,  
[OphthalmicTomographyImageStorage](#) = 172 ,  
[TextSRStorageTrialRetired](#) = 173 ,  
[AudioSRStorageTrialRetired](#) = 174 ,  
[DetailSRStorageTrialRetired](#) = 175 ,  
[ComprehensiveSRStorageTrialRetired](#) = 176 ,  
[BasicTextSRStorage](#) = 177 ,  
[EnhancedSRStorage](#) = 178 ,  
[ComprehensiveSRStorage](#) = 179 ,  
[ProcedureLogStorage](#) = 180 ,  
[MammographyCADSRStorage](#) = 181 ,  
[KeyObjectSelectionDocumentStorage](#) = 182 ,  
[ChestCADSRStorage](#) = 183 ,  
[XRayRadiationDoseSRStorage](#) = 184 ,  
[EncapsulatedPDFStorage](#) = 185 ,  
[EncapsulatedCDASStorage](#) = 186 ,  
[PositronEmissionTomographyImageStorage](#) = 187 ,  
[StandalonePETCurveStorageRetired](#) = 188 ,  
[RTImageStorage](#) = 189 ,  
[RTDoseStorage](#) = 190 ,  
[RTStructureSetStorage](#) = 191 ,  
[RTBeamsTreatmentRecordStorage](#) = 192 ,  
[RTPlanStorage](#) = 193 ,  
[RTBrachyTreatmentRecordStorage](#) = 194 ,  
[RTTreatmentSummaryRecordStorage](#) = 195 ,  
[RTIonPlanStorage](#) = 196 ,  
[RTIonBeamsTreatmentRecordStorage](#) = 197 ,  
[PatientRootQueryRetrieveInformationModelFIND](#) = 198 ,  
[PatientRootQueryRetrieveInformationModelMOVE](#) = 199 ,  
[PatientRootQueryRetrieveInformationModelGET](#) = 200 ,  
[StudyRootQueryRetrieveInformationModelFIND](#) = 201 ,  
[StudyRootQueryRetrieveInformationModelMOVE](#) = 202 ,  
[StudyRootQueryRetrieveInformationModelGET](#) = 203 ,

[PatientStudyOnlyQueryRetrieveInformationModelFINDRetired](#) = 204 ,  
[PatientStudyOnlyQueryRetrieveInformationModelMOVERetired](#) = 205 ,  
[PatientStudyOnlyQueryRetrieveInformationModelGETRetired](#) = 206 ,  
[ModalityWorklistInformationModelFIND](#) = 207 ,  
[GeneralPurposeWorklistInformationModelFIND](#) = 208 ,  
[GeneralPurposeScheduledProcedureStepSOPClass](#) = 209 ,  
[GeneralPurposePerformedProcedureStepSOPClass](#) = 210 ,  
[GeneralPurposeWorklistManagementMetaSOPClass](#) = 211 ,  
[InstanceAvailabilityNotificationSOPClass](#) = 212 ,  
[RTBeamsDeliveryInstructionStorageSupplement74FrozenDraft](#) = 213 ,  
[RTConventionalMachineVerificationSupplement74FrozenDraft](#) = 214 ,  
[RTIonMachineVerificationSupplement74FrozenDraft](#) = 215 ,  
[UnifiedWorklistandProcedureStepServiceClass](#) = 216 ,  
[UnifiedProcedureStepPushSOPClass](#) = 217 ,  
[UnifiedProcedureStepWatchSOPClass](#) = 218 ,  
[UnifiedProcedureStepPullSOPClass](#) = 219 ,  
[UnifiedProcedureStepEventSOPClass](#) = 220 ,  
[UnifiedWorklistandProcedureStepSOPInstance](#) = 221 ,  
[GeneralRelevantPatientInformationQuery](#) = 222 ,  
[BreastImagingRelevantPatientInformationQuery](#) = 223 ,  
[CardiacRelevantPatientInformationQuery](#) = 224 ,  
[HangingProtocolStorage](#) = 225 ,  
[HangingProtocolInformationModelFIND](#) = 226 ,  
[HangingProtocolInformationModelMOVE](#) = 227 ,  
[ProductCharacteristicsQuerySOPClass](#) = 228 ,  
[SubstanceApprovalQuerySOPClass](#) = 229 ,  
[dicomDeviceName](#) = 230 ,  
[dicomDescription](#) = 231 ,  
[dicomManufacturer](#) = 232 ,  
[dicomManufacturerModelName](#) = 233 ,  
[dicomSoftwareVersion](#) = 234 ,  
[dicomVendorData](#) = 235 ,  
[dicomAETitle](#) = 236 ,  
[dicomNetworkConnectionReference](#) = 237 ,  
[dicomApplicationCluster](#) = 238 ,  
[dicomAssociationInitiator](#) = 239 ,  
[dicomAssociationAcceptor](#) = 240 ,  
[dicomHostname](#) = 241 ,  
[dicomPort](#) = 242 ,  
[dicomSOPClass](#) = 243 ,  
[dicomTransferRole](#) = 244 ,  
[dicomTransferSyntax](#) = 245 ,  
[dicomPrimaryDeviceType](#) = 246 ,  
[dicomRelatedDeviceReference](#) = 247 ,  
[dicomPreferredCalledAETitle](#) = 248 ,  
[dicomTLSCyphersuite](#) = 249 ,  
[dicomAuthorizedNodeCertificateReference](#) = 250 ,  
[dicomThisNodeCertificateReference](#) = 251 ,  
[dicomInstalled](#) = 252 ,  
[dicomStationName](#) = 253 ,  
[dicomDeviceSerialNumber](#) = 254 ,  
[dicomInstitutionName](#) = 255 ,  
[dicomInstitutionAddress](#) = 256 ,  
[dicomInstitutionDepartmentName](#) = 257 ,

[dicomIssuerOfPatientID](#) = 258 ,  
[dicomPreferredCallingAETitle](#) = 259 ,  
[dicomSupportedCharacterSet](#) = 260 ,  
[dicomConfigurationRoot](#) = 261 ,  
[dicomDevicesRoot](#) = 262 ,  
[dicomUniqueAETitlesRegistryRoot](#) = 263 ,  
[dicomDevice](#) = 264 ,  
[dicomNetworkAE](#) = 265 ,  
[dicomNetworkConnection](#) = 266 ,  
[dicomUniqueAETitle](#) = 267 ,  
[dicomTransferCapability](#) = 268 ,  
[VLWholeSlideMicroscopyImageStorage](#) = 269 ,  
[EnhancedUSVolumeStorage](#) = 270 ,  
[SurfaceSegmentationStorage](#) = 271 ,  
[BreastTomosynthesisImageStorage](#) = 272 ,  
[LegacyConvertedEnhancedCTImageStorage](#) = 273 ,  
[LegacyConvertedEnhancedMRImageStorage](#) = 274 ,  
[LegacyConvertedEnhancedPETImageStorage](#) = 275 ,  
[MPEG2MainProfileHighLevel](#) = 276 ,  
[MPEG4AVCH\\_264HighProfileLevel4\\_1](#) = 277 ,  
[MPEG4AVCH\\_264BDcompatibleHighProfileLevel4\\_1](#) = 278 ,  
[PETColorPaletteSOPInstance](#) = 279 ,  
[HotMetalBlueColorPaletteSOPInstance](#) = 280 ,  
[PET20StepColorPaletteSOPInstance](#) = 281 ,  
[SpringColorPaletteSOPInstance](#) = 282 ,  
[SummerColorPaletteSOPInstance](#) = 283 ,  
[FallColorPaletteSOPInstance](#) = 284 ,  
[WinterColorPaletteSOPInstance](#) = 285 ,  
[Papyrus3ImplicitVRLittleEndian](#) = 286 ,  
[AdultMouseAnatomyOntology](#) = 287 ,  
[UberonOntology](#) = 288 ,  
[IntegratedTaxonomicInformationSystemITISTaxonomicSerialNumberTSN](#) = 289 ,  
[MouseGenomeInitiativeMGI](#) = 290 ,  
[PubChemCompoundCID](#) = 291 ,  
[ICD11](#) = 292 ,  
[NewYorkUniversityMelanomaClinicalCooperativeGroup](#) = 293 ,  
[MayoClinicNonradiologicalImagesSBSAnatomicalSurfaceRegionGuide](#) = 294 ,  
[ImageBiomarkerStandardisationInitiative](#) = 295 ,  
[RadiomicsOntology](#) = 296 ,  
[DisplaySystemSOPClass](#) = 297 ,  
[DisplaySystemSOPInstance](#) = 298 ,  
[GeneralAudioWaveformStorage](#) = 299 ,  
[ArterialPulseWaveformStorage](#) = 300 ,  
[RespiratoryWaveformStorage](#) = 301 ,  
[XAXRFGrayscaleSoftcopyPresentationStateStorage](#) = 302 ,  
[GrayscalePlanarMPRVolumetricPresentationStateStorage](#) = 303 ,  
[MPEG4AVCH\\_264HighProfileLevel4\\_2For2DVideo](#) = 304 ,  
[MPEG4AVCH\\_264HighProfileLevel4\\_2For3DVideo](#) = 305 ,  
[MPEG4AVCH\\_264StereoHighProfileLevel4\\_2](#) = 306 ,  
[HEVCH\\_265MainProfileLevel5\\_1](#) = 307 ,  
[HEVCH\\_265Main10ProfileLevel5\\_1](#) = 308 ,  
[HotIronColorPaletteSOPInstance](#) = 309 ,  
[CompositingPlanarMPRVolumetricPresentationStateStorage](#) = 310 ,  
[AdvancedBlendingPresentationStateStorage](#) = 311 ,



VolumeRenderingVolumetricPresentationStateStorage = 312 ,  
SegmentedVolumeRenderingVolumetricPresentationStateStorage = 313 ,  
MultipleVolumeRenderingVolumetricPresentationStateStorage = 314 ,  
Null0 = 315 ,  
BreastProjectionXRayImageStorageForPresentation = 316 ,  
BreastProjectionXRayImageStorageForProcessing = 317 ,  
IntravascularOpticalCoherenceTomographyImageStorageForPresentation = 318 ,  
IntravascularOpticalCoherenceTomographyImageStorageForProcessing = 319 ,  
ParametricMapStorage = 320 ,  
Null1 = 321 ,  
TractographyResultsStorage = 322 ,  
SurfaceScanMeshStorage = 323 ,  
SurfaceScanPointCloudStorage = 324 ,  
WideFieldOphthalmicPhotographyStereographicProjectionImageStorage = 325 ,  
WideFieldOphthalmicPhotography3DCoordinatesImageStorage = 326 ,  
OphthalmicOpticalCoherenceTomographyEnFaceImageStorage = 327 ,  
OphthalmicOpticalCoherenceTomographyBscanVolumeAnalysisStorage = 328 ,  
LensometryMeasurementsStorage = 329 ,  
AutorefractionMeasurementsStorage = 330 ,  
KeratometryMeasurementsStorage = 331 ,  
SubjectiveRefractionMeasurementsStorage = 332 ,  
VisualAcuityMeasurementsStorage = 333 ,  
SpectaclePrescriptionReportStorage = 334 ,  
OphthalmicAxialMeasurementsStorage = 335 ,  
IntraocularLensCalculationsStorage = 336 ,  
MacularGridThicknessandVolumeReportStorage = 337 ,  
OphthalmicVisualFieldStaticPerimetryMeasurementsStorage = 338 ,  
OphthalmicThicknessMapStorage = 339 ,  
CornealTopographyMapStorage = 340 ,  
Comprehensive3DSRStorage = 341 ,  
ExtensibleSRStorage = 342 ,  
RadiopharmaceuticalRadiationDoseSRStorage = 343 ,  
ColonCADSRStorage = 344 ,  
ImplantationPlanSRStorage = 345 ,  
AcquisitionContextSRStorage = 346 ,  
SimplifiedAdultEchoSRStorage = 347 ,  
PatientRadiationDoseSRStorage = 348 ,  
PlannedImagingAgentAdministrationSRStorage = 349 ,  
PerformedImagingAgentAdministrationSRStorage = 350 ,  
ContentAssessmentResultsStorage = 351 ,  
EncapsulatedSTLStorage = 352 ,  
EnhancedPETImageStorage = 353 ,  
BasicStructuredDisplayStorage = 354 ,  
CTDefinedProcedureProtocolStorage = 355 ,  
CTPerformedProcedureProtocolStorage = 356 ,  
ProtocolApprovalStorage = 357 ,  
ProtocolApprovalInformationModelFIND = 358 ,  
ProtocolApprovalInformationModelMOVE = 359 ,  
ProtocolApprovalInformationModelGET = 360 ,  
RTPhysicianIntentStorage = 361 ,  
RTSegmentAnnotationStorage = 362 ,  
DICOSCTImageStorage = 363 ,  
DICOSDigitalXRayImageStorageForPresentation = 364 ,  
DICOSDigitalXRayImageStorageForProcessing = 365 ,

```

DICOSThreatDetectionReportStorage = 366 ,
DICOS2DAITStorage = 367 ,
DICOS3DAITStorage = 368 ,
DICOSQuadrupoleResonanceQRStorage = 369 ,
EddyCurrentImageStorage = 370 ,
EddyCurrentMultiframeImageStorage = 371 ,
CompositeInstanceRootRetrieveMOVE = 372 ,
CompositeInstanceRootRetrieveGET = 373 ,
CompositeInstanceRetrieveWithoutBulkDataGET = 374 ,
DefinedProcedureProtocolInformationModelFIND = 375 ,
DefinedProcedureProtocolInformationModelMOVE = 376 ,
DefinedProcedureProtocolInformationModelGET = 377 ,
UPSFilteredGlobalSubscriptionSOPInstance = 378 ,
UnifiedWorklistandProcedureStepServiceClass1 = 379 ,
UnifiedProcedureStepPushSOPClass1 = 380 ,
UnifiedProcedureStepWatchSOPClass1 = 381 ,
UnifiedProcedureStepPullSOPClass1 = 382 ,
UnifiedProcedureStepEventSOPClass1 = 383 ,
RTBeamsDeliveryInstructionStorage = 384 ,
RTConventionalMachineVerification = 385 ,
RTIonMachineVerification = 386 ,
RTBrachyApplicationSetupDeliveryInstructionStorage = 387 ,
HangingProtocolInformationModelGET = 388 ,
ColorPaletteStorage = 389 ,
ColorPaletteQueryRetrieveInformationModelFIND = 390 ,
ColorPaletteQueryRetrieveInformationModelMOVE = 391 ,
ColorPaletteQueryRetrieveInformationModelGET = 392 ,
GenericImplantTemplateStorage = 393 ,
GenericImplantTemplateInformationModelFIND = 394 ,
GenericImplantTemplateInformationModelMOVE = 395 ,
GenericImplantTemplateInformationModelGET = 396 ,
ImplantAssemblyTemplateStorage = 397 ,
ImplantAssemblyTemplateInformationModelFIND = 398 ,
ImplantAssemblyTemplateInformationModelMOVE = 399 ,
ImplantAssemblyTemplateInformationModelGET = 400 ,
ImplantTemplateGroupStorage = 401 ,
ImplantTemplateGroupInformationModelFIND = 402 ,
ImplantTemplateGroupInformationModelMOVE = 403 ,
ImplantTemplateGroupInformationModelGET = 404 ,
NativeDICOMModel = 405 ,
AbstractMultiDimensionalImageModel = 406 ,
DICOMContentMappingResource = 407 ,
EnhancedMRColorImageStorage = 408 ,
UniversalCoordinatedTime = 409 }
• enum TSType {
uid_1_2_840_10008_1_1 = 1 ,
uid_1_2_840_10008_1_2 = 2 ,
uid_1_2_840_10008_1_2_1 = 3 ,
uid_1_2_840_10008_1_2_1_99 = 4 ,
uid_1_2_840_10008_1_2_2 = 5 ,
uid_1_2_840_10008_1_2_4_50 = 6 ,
uid_1_2_840_10008_1_2_4_51 = 7 ,
uid_1_2_840_10008_1_2_4_52 = 8 ,
uid_1_2_840_10008_1_2_4_53 = 9 ,

```

```
uid_1_2_840_10008_1_2_4_54 = 10 ,
uid_1_2_840_10008_1_2_4_55 = 11 ,
uid_1_2_840_10008_1_2_4_56 = 12 ,
uid_1_2_840_10008_1_2_4_57 = 13 ,
uid_1_2_840_10008_1_2_4_58 = 14 ,
uid_1_2_840_10008_1_2_4_59 = 15 ,
uid_1_2_840_10008_1_2_4_60 = 16 ,
uid_1_2_840_10008_1_2_4_61 = 17 ,
uid_1_2_840_10008_1_2_4_62 = 18 ,
uid_1_2_840_10008_1_2_4_63 = 19 ,
uid_1_2_840_10008_1_2_4_64 = 20 ,
uid_1_2_840_10008_1_2_4_65 = 21 ,
uid_1_2_840_10008_1_2_4_66 = 22 ,
uid_1_2_840_10008_1_2_4_70 = 23 ,
uid_1_2_840_10008_1_2_4_80 = 24 ,
uid_1_2_840_10008_1_2_4_81 = 25 ,
uid_1_2_840_10008_1_2_4_90 = 26 ,
uid_1_2_840_10008_1_2_4_91 = 27 ,
uid_1_2_840_10008_1_2_4_92 = 28 ,
uid_1_2_840_10008_1_2_4_93 = 29 ,
uid_1_2_840_10008_1_2_4_94 = 30 ,
uid_1_2_840_10008_1_2_4_95 = 31 ,
uid_1_2_840_10008_1_2_4_100 = 32 ,
uid_1_2_840_10008_1_2_5 = 33 ,
uid_1_2_840_10008_1_2_6_1 = 34 ,
uid_1_2_840_10008_1_2_6_2 = 35 ,
uid_1_2_840_10008_1_3_10 = 36 ,
uid_1_2_840_10008_1_4_1_1 = 37 ,
uid_1_2_840_10008_1_4_1_2 = 38 ,
uid_1_2_840_10008_1_4_1_3 = 39 ,
uid_1_2_840_10008_1_4_1_4 = 40 ,
uid_1_2_840_10008_1_4_1_5 = 41 ,
uid_1_2_840_10008_1_4_1_6 = 42 ,
uid_1_2_840_10008_1_4_1_7 = 43 ,
uid_1_2_840_10008_1_4_1_8 = 44 ,
uid_1_2_840_10008_1_4_1_9 = 45 ,
uid_1_2_840_10008_1_4_1_10 = 46 ,
uid_1_2_840_10008_1_4_1_11 = 47 ,
uid_1_2_840_10008_1_4_1_12 = 48 ,
uid_1_2_840_10008_1_4_1_13 = 49 ,
uid_1_2_840_10008_1_4_1_14 = 50 ,
uid_1_2_840_10008_1_4_1_15 = 51 ,
uid_1_2_840_10008_1_4_1_16 = 52 ,
uid_1_2_840_10008_1_4_1_17 = 53 ,
uid_1_2_840_10008_1_4_1_18 = 54 ,
uid_1_2_840_10008_1_4_2_1 = 55 ,
uid_1_2_840_10008_1_4_2_2 = 56 ,
uid_1_2_840_10008_1_9 = 57 ,
uid_1_2_840_10008_1_20_1 = 58 ,
uid_1_2_840_10008_1_20_1_1 = 59 ,
uid_1_2_840_10008_1_20_2 = 60 ,
uid_1_2_840_10008_1_20_2_1 = 61 ,
uid_1_2_840_10008_1_40 = 62 ,
uid_1_2_840_10008_1_40_1 = 63 ,
```

```
uid_1_2_840_10008_1_42 = 64 ,  
uid_1_2_840_10008_1_42_1 = 65 ,  
uid_1_2_840_10008_2_6_1 = 66 ,  
uid_1_2_840_10008_2_16_4 = 67 ,  
uid_1_2_840_10008_3_1_1_1 = 68 ,  
uid_1_2_840_10008_3_1_2_1_1 = 69 ,  
uid_1_2_840_10008_3_1_2_1_4 = 70 ,  
uid_1_2_840_10008_3_1_2_2_1 = 71 ,  
uid_1_2_840_10008_3_1_2_3_1 = 72 ,  
uid_1_2_840_10008_3_1_2_3_2 = 73 ,  
uid_1_2_840_10008_3_1_2_3_3 = 74 ,  
uid_1_2_840_10008_3_1_2_3_4 = 75 ,  
uid_1_2_840_10008_3_1_2_3_5 = 76 ,  
uid_1_2_840_10008_3_1_2_5_1 = 77 ,  
uid_1_2_840_10008_3_1_2_5_4 = 78 ,  
uid_1_2_840_10008_3_1_2_5_5 = 79 ,  
uid_1_2_840_10008_3_1_2_6_1 = 80 ,  
uid_1_2_840_10008_4_2 = 81 ,  
uid_1_2_840_10008_5_1_1_1 = 82 ,  
uid_1_2_840_10008_5_1_1_2 = 83 ,  
uid_1_2_840_10008_5_1_1_4 = 84 ,  
uid_1_2_840_10008_5_1_1_4_1 = 85 ,  
uid_1_2_840_10008_5_1_1_4_2 = 86 ,  
uid_1_2_840_10008_5_1_1_9 = 87 ,  
uid_1_2_840_10008_5_1_1_9_1 = 88 ,  
uid_1_2_840_10008_5_1_1_14 = 89 ,  
uid_1_2_840_10008_5_1_1_15 = 90 ,  
uid_1_2_840_10008_5_1_1_16 = 91 ,  
uid_1_2_840_10008_5_1_1_16_376 = 92 ,  
uid_1_2_840_10008_5_1_1_17 = 93 ,  
uid_1_2_840_10008_5_1_1_17_376 = 94 ,  
uid_1_2_840_10008_5_1_1_18 = 95 ,  
uid_1_2_840_10008_5_1_1_18_1 = 96 ,  
uid_1_2_840_10008_5_1_1_22 = 97 ,  
uid_1_2_840_10008_5_1_1_23 = 98 ,  
uid_1_2_840_10008_5_1_1_24 = 99 ,  
uid_1_2_840_10008_5_1_1_24_1 = 100 ,  
uid_1_2_840_10008_5_1_1_25 = 101 ,  
uid_1_2_840_10008_5_1_1_26 = 102 ,  
uid_1_2_840_10008_5_1_1_27 = 103 ,  
uid_1_2_840_10008_5_1_1_29 = 104 ,  
uid_1_2_840_10008_5_1_1_30 = 105 ,  
uid_1_2_840_10008_5_1_1_31 = 106 ,  
uid_1_2_840_10008_5_1_1_32 = 107 ,  
uid_1_2_840_10008_5_1_1_33 = 108 ,  
uid_1_2_840_10008_5_1_4_1_1_1 = 109 ,  
uid_1_2_840_10008_5_1_4_1_1_1_1 = 110 ,  
uid_1_2_840_10008_5_1_4_1_1_1_1_1 = 111 ,  
uid_1_2_840_10008_5_1_4_1_1_1_2 = 112 ,  
uid_1_2_840_10008_5_1_4_1_1_1_2_1 = 113 ,  
uid_1_2_840_10008_5_1_4_1_1_1_3 = 114 ,  
uid_1_2_840_10008_5_1_4_1_1_1_3_1 = 115 ,  
uid_1_2_840_10008_5_1_4_1_1_2 = 116 ,  
uid_1_2_840_10008_5_1_4_1_1_2_1 = 117 ,
```

```
uid_1_2_840_10008_5_1_4_1_1_3 = 118 ,  
uid_1_2_840_10008_5_1_4_1_1_3_1 = 119 ,  
uid_1_2_840_10008_5_1_4_1_1_4 = 120 ,  
uid_1_2_840_10008_5_1_4_1_1_4_1 = 121 ,  
uid_1_2_840_10008_5_1_4_1_1_4_2 = 122 ,  
uid_1_2_840_10008_5_1_4_1_1_5 = 123 ,  
uid_1_2_840_10008_5_1_4_1_1_6 = 124 ,  
uid_1_2_840_10008_5_1_4_1_1_6_1 = 125 ,  
uid_1_2_840_10008_5_1_4_1_1_7 = 126 ,  
uid_1_2_840_10008_5_1_4_1_1_7_1 = 127 ,  
uid_1_2_840_10008_5_1_4_1_1_7_2 = 128 ,  
uid_1_2_840_10008_5_1_4_1_1_7_3 = 129 ,  
uid_1_2_840_10008_5_1_4_1_1_7_4 = 130 ,  
uid_1_2_840_10008_5_1_4_1_1_8 = 131 ,  
uid_1_2_840_10008_5_1_4_1_1_9 = 132 ,  
uid_1_2_840_10008_5_1_4_1_1_9_1 = 133 ,  
uid_1_2_840_10008_5_1_4_1_1_9_1_1 = 134 ,  
uid_1_2_840_10008_5_1_4_1_1_9_1_2 = 135 ,  
uid_1_2_840_10008_5_1_4_1_1_9_1_3 = 136 ,  
uid_1_2_840_10008_5_1_4_1_1_9_2_1 = 137 ,  
uid_1_2_840_10008_5_1_4_1_1_9_3_1 = 138 ,  
uid_1_2_840_10008_5_1_4_1_1_9_4_1 = 139 ,  
uid_1_2_840_10008_5_1_4_1_1_10 = 140 ,  
uid_1_2_840_10008_5_1_4_1_1_11 = 141 ,  
uid_1_2_840_10008_5_1_4_1_1_11_1 = 142 ,  
uid_1_2_840_10008_5_1_4_1_1_11_2 = 143 ,  
uid_1_2_840_10008_5_1_4_1_1_11_3 = 144 ,  
uid_1_2_840_10008_5_1_4_1_1_11_4 = 145 ,  
uid_1_2_840_10008_5_1_4_1_1_12_1 = 146 ,  
uid_1_2_840_10008_5_1_4_1_1_12_1_1 = 147 ,  
uid_1_2_840_10008_5_1_4_1_1_12_2 = 148 ,  
uid_1_2_840_10008_5_1_4_1_1_12_2_1 = 149 ,  
uid_1_2_840_10008_5_1_4_1_1_13_1_1 = 150 ,  
uid_1_2_840_10008_5_1_4_1_1_13_1_2 = 151 ,  
uid_1_2_840_10008_5_1_4_1_1_12_3 = 152 ,  
uid_1_2_840_10008_5_1_4_1_1_20 = 153 ,  
uid_1_2_840_10008_5_1_4_1_1_66 = 154 ,  
uid_1_2_840_10008_5_1_4_1_1_66_1 = 155 ,  
uid_1_2_840_10008_5_1_4_1_1_66_2 = 156 ,  
uid_1_2_840_10008_5_1_4_1_1_66_3 = 157 ,  
uid_1_2_840_10008_5_1_4_1_1_66_4 = 158 ,  
uid_1_2_840_10008_5_1_4_1_1_67 = 159 ,  
uid_1_2_840_10008_5_1_4_1_1_77_1 = 160 ,  
uid_1_2_840_10008_5_1_4_1_1_77_2 = 161 ,  
uid_1_2_840_10008_5_1_4_1_1_77_1_1 = 162 ,  
uid_1_2_840_10008_5_1_4_1_1_77_1_1_1 = 163 ,  
uid_1_2_840_10008_5_1_4_1_1_77_1_2 = 164 ,  
uid_1_2_840_10008_5_1_4_1_1_77_1_2_1 = 165 ,  
uid_1_2_840_10008_5_1_4_1_1_77_1_3 = 166 ,  
uid_1_2_840_10008_5_1_4_1_1_77_1_4 = 167 ,  
uid_1_2_840_10008_5_1_4_1_1_77_1_4_1 = 168 ,  
uid_1_2_840_10008_5_1_4_1_1_77_1_5_1 = 169 ,  
uid_1_2_840_10008_5_1_4_1_1_77_1_5_2 = 170 ,  
uid_1_2_840_10008_5_1_4_1_1_77_1_5_3 = 171 ,
```

```
uid_1_2_840_10008_5_1_4_1_1_77_1_5_4 = 172 ,
uid_1_2_840_10008_5_1_4_1_1_88_1 = 173 ,
uid_1_2_840_10008_5_1_4_1_1_88_2 = 174 ,
uid_1_2_840_10008_5_1_4_1_1_88_3 = 175 ,
uid_1_2_840_10008_5_1_4_1_1_88_4 = 176 ,
uid_1_2_840_10008_5_1_4_1_1_88_11 = 177 ,
uid_1_2_840_10008_5_1_4_1_1_88_22 = 178 ,
uid_1_2_840_10008_5_1_4_1_1_88_33 = 179 ,
uid_1_2_840_10008_5_1_4_1_1_88_40 = 180 ,
uid_1_2_840_10008_5_1_4_1_1_88_50 = 181 ,
uid_1_2_840_10008_5_1_4_1_1_88_59 = 182 ,
uid_1_2_840_10008_5_1_4_1_1_88_65 = 183 ,
uid_1_2_840_10008_5_1_4_1_1_88_67 = 184 ,
uid_1_2_840_10008_5_1_4_1_1_104_1 = 185 ,
uid_1_2_840_10008_5_1_4_1_1_104_2 = 186 ,
uid_1_2_840_10008_5_1_4_1_1_128 = 187 ,
uid_1_2_840_10008_5_1_4_1_1_129 = 188 ,
uid_1_2_840_10008_5_1_4_1_1_481_1 = 189 ,
uid_1_2_840_10008_5_1_4_1_1_481_2 = 190 ,
uid_1_2_840_10008_5_1_4_1_1_481_3 = 191 ,
uid_1_2_840_10008_5_1_4_1_1_481_4 = 192 ,
uid_1_2_840_10008_5_1_4_1_1_481_5 = 193 ,
uid_1_2_840_10008_5_1_4_1_1_481_6 = 194 ,
uid_1_2_840_10008_5_1_4_1_1_481_7 = 195 ,
uid_1_2_840_10008_5_1_4_1_1_481_8 = 196 ,
uid_1_2_840_10008_5_1_4_1_1_481_9 = 197 ,
uid_1_2_840_10008_5_1_4_1_2_1_1 = 198 ,
uid_1_2_840_10008_5_1_4_1_2_1_2 = 199 ,
uid_1_2_840_10008_5_1_4_1_2_1_3 = 200 ,
uid_1_2_840_10008_5_1_4_1_2_2_1 = 201 ,
uid_1_2_840_10008_5_1_4_1_2_2_2 = 202 ,
uid_1_2_840_10008_5_1_4_1_2_2_3 = 203 ,
uid_1_2_840_10008_5_1_4_1_2_3_1 = 204 ,
uid_1_2_840_10008_5_1_4_1_2_3_2 = 205 ,
uid_1_2_840_10008_5_1_4_1_2_3_3 = 206 ,
uid_1_2_840_10008_5_1_4_31 = 207 ,
uid_1_2_840_10008_5_1_4_32_1 = 208 ,
uid_1_2_840_10008_5_1_4_32_2 = 209 ,
uid_1_2_840_10008_5_1_4_32_3 = 210 ,
uid_1_2_840_10008_5_1_4_32 = 211 ,
uid_1_2_840_10008_5_1_4_33 = 212 ,
uid_1_2_840_10008_5_1_4_34_1 = 213 ,
uid_1_2_840_10008_5_1_4_34_2 = 214 ,
uid_1_2_840_10008_5_1_4_34_3 = 215 ,
uid_1_2_840_10008_5_1_4_34_4 = 216 ,
uid_1_2_840_10008_5_1_4_34_4_1 = 217 ,
uid_1_2_840_10008_5_1_4_34_4_2 = 218 ,
uid_1_2_840_10008_5_1_4_34_4_3 = 219 ,
uid_1_2_840_10008_5_1_4_34_4_4 = 220 ,
uid_1_2_840_10008_5_1_4_34_5 = 221 ,
uid_1_2_840_10008_5_1_4_37_1 = 222 ,
uid_1_2_840_10008_5_1_4_37_2 = 223 ,
uid_1_2_840_10008_5_1_4_37_3 = 224 ,
uid_1_2_840_10008_5_1_4_38_1 = 225 ,
```

uid\_1\_2\_840\_10008\_5\_1\_4\_38\_2 = 226 ,  
uid\_1\_2\_840\_10008\_5\_1\_4\_38\_3 = 227 ,  
uid\_1\_2\_840\_10008\_5\_1\_4\_41 = 228 ,  
uid\_1\_2\_840\_10008\_5\_1\_4\_42 = 229 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_1 = 230 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_2 = 231 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_3 = 232 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_4 = 233 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_5 = 234 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_6 = 235 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_7 = 236 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_8 = 237 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_9 = 238 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_10 = 239 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_11 = 240 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_12 = 241 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_13 = 242 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_14 = 243 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_15 = 244 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_16 = 245 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_17 = 246 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_18 = 247 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_19 = 248 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_20 = 249 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_21 = 250 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_22 = 251 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_23 = 252 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_24 = 253 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_25 = 254 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_26 = 255 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_27 = 256 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_28 = 257 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_29 = 258 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_30 = 259 ,  
uid\_1\_2\_840\_10008\_15\_0\_3\_31 = 260 ,  
uid\_1\_2\_840\_10008\_15\_0\_4\_1 = 261 ,  
uid\_1\_2\_840\_10008\_15\_0\_4\_2 = 262 ,  
uid\_1\_2\_840\_10008\_15\_0\_4\_3 = 263 ,  
uid\_1\_2\_840\_10008\_15\_0\_4\_4 = 264 ,  
uid\_1\_2\_840\_10008\_15\_0\_4\_5 = 265 ,  
uid\_1\_2\_840\_10008\_15\_0\_4\_6 = 266 ,  
uid\_1\_2\_840\_10008\_15\_0\_4\_7 = 267 ,  
uid\_1\_2\_840\_10008\_15\_0\_4\_8 = 268 ,  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_6 = 269 ,  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_6\_2 = 270 ,  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_5 = 271 ,  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_3 = 272 ,  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_2\_2 = 273 ,  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4\_4 = 274 ,  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_128\_1 = 275 ,  
uid\_1\_2\_840\_10008\_1\_2\_4\_101 = 276 ,  
uid\_1\_2\_840\_10008\_1\_2\_4\_102 = 277 ,  
uid\_1\_2\_840\_10008\_1\_2\_4\_103 = 278 ,  
uid\_1\_2\_840\_10008\_1\_5\_2 = 279 ,

```
uid_1_2_840_10008_1_5_3 = 280 ,
uid_1_2_840_10008_1_5_4 = 281 ,
uid_1_2_840_10008_1_5_5 = 282 ,
uid_1_2_840_10008_1_5_6 = 283 ,
uid_1_2_840_10008_1_5_7 = 284 ,
uid_1_2_840_10008_1_5_8 = 285 ,
uid_1_2_840_10008_1_20 = 286 ,
uid_1_2_840_10008_2_16_5 = 287 ,
uid_1_2_840_10008_2_16_6 = 288 ,
uid_1_2_840_10008_2_16_7 = 289 ,
uid_1_2_840_10008_2_16_8 = 290 ,
uid_1_2_840_10008_2_16_9 = 291 ,
uid_1_2_840_10008_2_16_10 = 292 ,
uid_1_2_840_10008_2_16_11 = 293 ,
uid_1_2_840_10008_2_16_12 = 294 ,
uid_1_2_840_10008_2_16_13 = 295 ,
uid_1_2_840_10008_2_16_14 = 296 ,
uid_1_2_840_10008_5_1_1_40 = 297 ,
uid_1_2_840_10008_5_1_1_40_1 = 298 ,
uid_1_2_840_10008_5_1_4_1_1_9_4_2 = 299 ,
uid_1_2_840_10008_5_1_4_1_1_9_5_1 = 300 ,
uid_1_2_840_10008_5_1_4_1_1_9_6_1 = 301 ,
uid_1_2_840_10008_5_1_4_1_1_11_5 = 302 ,
uid_1_2_840_10008_5_1_4_1_1_11_6 = 303 ,
uid_1_2_840_10008_1_2_4_104 = 304 ,
uid_1_2_840_10008_1_2_4_105 = 305 ,
uid_1_2_840_10008_1_2_4_106 = 306 ,
uid_1_2_840_10008_1_2_4_107 = 307 ,
uid_1_2_840_10008_1_2_4_108 = 308 ,
uid_1_2_840_10008_1_5_1 = 309 ,
uid_1_2_840_10008_5_1_4_1_1_11_7 = 310 ,
uid_1_2_840_10008_5_1_4_1_1_11_8 = 311 ,
uid_1_2_840_10008_5_1_4_1_1_11_9 = 312 ,
uid_1_2_840_10008_5_1_4_1_1_11_10 = 313 ,
uid_1_2_840_10008_5_1_4_1_1_11_11 = 314 ,
uid_1_2_840_10008_5_1_4_1_1_12_77 = 315 ,
uid_1_2_840_10008_5_1_4_1_1_13_1_4 = 316 ,
uid_1_2_840_10008_5_1_4_1_1_13_1_5 = 317 ,
uid_1_2_840_10008_5_1_4_1_1_14_1 = 318 ,
uid_1_2_840_10008_5_1_4_1_1_14_2 = 319 ,
uid_1_2_840_10008_5_1_4_1_1_30 = 320 ,
uid_1_2_840_10008_5_1_4_1_1_40 = 321 ,
uid_1_2_840_10008_5_1_4_1_1_66_6 = 322 ,
uid_1_2_840_10008_5_1_4_1_1_68_1 = 323 ,
uid_1_2_840_10008_5_1_4_1_1_68_2 = 324 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_5 = 325 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_6 = 326 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_7 = 327 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_8 = 328 ,
uid_1_2_840_10008_5_1_4_1_1_78_1 = 329 ,
uid_1_2_840_10008_5_1_4_1_1_78_2 = 330 ,
uid_1_2_840_10008_5_1_4_1_1_78_3 = 331 ,
uid_1_2_840_10008_5_1_4_1_1_78_4 = 332 ,
uid_1_2_840_10008_5_1_4_1_1_78_5 = 333 ,
```



```
uid_1_2_840_10008_5_1_4_1_1_78_6 = 334 ,  
uid_1_2_840_10008_5_1_4_1_1_78_7 = 335 ,  
uid_1_2_840_10008_5_1_4_1_1_78_8 = 336 ,  
uid_1_2_840_10008_5_1_4_1_1_79_1 = 337 ,  
uid_1_2_840_10008_5_1_4_1_1_80_1 = 338 ,  
uid_1_2_840_10008_5_1_4_1_1_81_1 = 339 ,  
uid_1_2_840_10008_5_1_4_1_1_82_1 = 340 ,  
uid_1_2_840_10008_5_1_4_1_1_88_34 = 341 ,  
uid_1_2_840_10008_5_1_4_1_1_88_35 = 342 ,  
uid_1_2_840_10008_5_1_4_1_1_88_68 = 343 ,  
uid_1_2_840_10008_5_1_4_1_1_88_69 = 344 ,  
uid_1_2_840_10008_5_1_4_1_1_88_70 = 345 ,  
uid_1_2_840_10008_5_1_4_1_1_88_71 = 346 ,  
uid_1_2_840_10008_5_1_4_1_1_88_72 = 347 ,  
uid_1_2_840_10008_5_1_4_1_1_88_73 = 348 ,  
uid_1_2_840_10008_5_1_4_1_1_88_74 = 349 ,  
uid_1_2_840_10008_5_1_4_1_1_88_75 = 350 ,  
uid_1_2_840_10008_5_1_4_1_1_90_1 = 351 ,  
uid_1_2_840_10008_5_1_4_1_1_104_3 = 352 ,  
uid_1_2_840_10008_5_1_4_1_1_130 = 353 ,  
uid_1_2_840_10008_5_1_4_1_1_131 = 354 ,  
uid_1_2_840_10008_5_1_4_1_1_200_1 = 355 ,  
uid_1_2_840_10008_5_1_4_1_1_200_2 = 356 ,  
uid_1_2_840_10008_5_1_4_1_1_200_3 = 357 ,  
uid_1_2_840_10008_5_1_4_1_1_200_4 = 358 ,  
uid_1_2_840_10008_5_1_4_1_1_200_5 = 359 ,  
uid_1_2_840_10008_5_1_4_1_1_200_6 = 360 ,  
uid_1_2_840_10008_5_1_4_1_1_481_10 = 361 ,  
uid_1_2_840_10008_5_1_4_1_1_481_11 = 362 ,  
uid_1_2_840_10008_5_1_4_1_1_501_1 = 363 ,  
uid_1_2_840_10008_5_1_4_1_1_501_2_1 = 364 ,  
uid_1_2_840_10008_5_1_4_1_1_501_2_2 = 365 ,  
uid_1_2_840_10008_5_1_4_1_1_501_3 = 366 ,  
uid_1_2_840_10008_5_1_4_1_1_501_4 = 367 ,  
uid_1_2_840_10008_5_1_4_1_1_501_5 = 368 ,  
uid_1_2_840_10008_5_1_4_1_1_501_6 = 369 ,  
uid_1_2_840_10008_5_1_4_1_1_601_1 = 370 ,  
uid_1_2_840_10008_5_1_4_1_1_601_2 = 371 ,  
uid_1_2_840_10008_5_1_4_1_2_4_2 = 372 ,  
uid_1_2_840_10008_5_1_4_1_2_4_3 = 373 ,  
uid_1_2_840_10008_5_1_4_1_2_5_3 = 374 ,  
uid_1_2_840_10008_5_1_4_20_1 = 375 ,  
uid_1_2_840_10008_5_1_4_20_2 = 376 ,  
uid_1_2_840_10008_5_1_4_20_3 = 377 ,  
uid_1_2_840_10008_5_1_4_34_5_1 = 378 ,  
uid_1_2_840_10008_5_1_4_34_6 = 379 ,  
uid_1_2_840_10008_5_1_4_34_6_1 = 380 ,  
uid_1_2_840_10008_5_1_4_34_6_2 = 381 ,  
uid_1_2_840_10008_5_1_4_34_6_3 = 382 ,  
uid_1_2_840_10008_5_1_4_34_6_4 = 383 ,  
uid_1_2_840_10008_5_1_4_34_7 = 384 ,  
uid_1_2_840_10008_5_1_4_34_8 = 385 ,  
uid_1_2_840_10008_5_1_4_34_9 = 386 ,  
uid_1_2_840_10008_5_1_4_34_10 = 387 ,
```

```

uid_1_2_840_10008_5_1_4_38_4 = 388 ,
uid_1_2_840_10008_5_1_4_39_1 = 389 ,
uid_1_2_840_10008_5_1_4_39_2 = 390 ,
uid_1_2_840_10008_5_1_4_39_3 = 391 ,
uid_1_2_840_10008_5_1_4_39_4 = 392 ,
uid_1_2_840_10008_5_1_4_43_1 = 393 ,
uid_1_2_840_10008_5_1_4_43_2 = 394 ,
uid_1_2_840_10008_5_1_4_43_3 = 395 ,
uid_1_2_840_10008_5_1_4_43_4 = 396 ,
uid_1_2_840_10008_5_1_4_44_1 = 397 ,
uid_1_2_840_10008_5_1_4_44_2 = 398 ,
uid_1_2_840_10008_5_1_4_44_3 = 399 ,
uid_1_2_840_10008_5_1_4_44_4 = 400 ,
uid_1_2_840_10008_5_1_4_45_1 = 401 ,
uid_1_2_840_10008_5_1_4_45_2 = 402 ,
uid_1_2_840_10008_5_1_4_45_3 = 403 ,
uid_1_2_840_10008_5_1_4_45_4 = 404 ,
uid_1_2_840_10008_7_1_1 = 405 ,
uid_1_2_840_10008_7_1_2 = 406 ,
uid_1_2_840_10008_8_1_1 = 407 ,
uid_1_2_840_10008_5_1_4_1_1_4_3 = 408 ,
uid_1_2_840_10008_15_1_1 = 409 }

```

### Public Member Functions

- const char \* [GetName](#) () const
- const char \* [GetString](#) () const
- [operator TSType](#) () const
- bool [SetFromUID](#) (const char \*str)

### Static Public Member Functions

- static unsigned int [GetNumberOfTransferSyntaxStrings](#) ()
- static const char \*const \* [GetTransferSyntaxString](#) (unsigned int ts)
- static [TransferSyntaxStringsType](#) [GetTransferSyntaxStrings](#) ()
- static const char \* [GetUIDName](#) (unsigned int ts)
- static const char \* [GetUIDString](#) (unsigned int ts)

## 10.324.1 Detailed Description

all known uids

### Examples

[GenerateStandardSOPClasses.cxx](#).

## 10.324.2 Member Typedef Documentation

### 10.324.2.1 TransferSyntaxStringsType

```
typedef const char* const(* gdcm::UIDs::TransferSyntaxStringsType)[2]
```

## 10.324.3 Member Enumeration Documentation

### 10.324.3.1 TSName

```
enum gdcm::UIDs::TSName
```

Enumerator

VerificationSOPClass
ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM
ExplicitVRLittleEndian
DeflatedExplicitVRLittleEndian
ExplicitVRBigEndian
JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImageCompression
JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG12BitImageCompressionProcess4only
JPEGExtendedProcess35Retired
JPEGSpectralSelectionNonHierarchicalProcess68Retired
JPEGSpectralSelectionNonHierarchicalProcess79Retired
JPEGFullProgressionNonHierarchicalProcess1012Retired
JPEGFullProgressionNonHierarchicalProcess1113Retired
JPEGLosslessNonHierarchicalProcess14
JPEGLosslessNonHierarchicalProcess15Retired
JPEGExtendedHierarchicalProcess1618Retired
JPEGExtendedHierarchicalProcess1719Retired
JPEGSpectralSelectionHierarchicalProcess2022Retired
JPEGSpectralSelectionHierarchicalProcess2123Retired
JPEGFullProgressionHierarchicalProcess2426Retired
JPEGFullProgressionHierarchicalProcess2527Retired
JPEGLosslessHierarchicalProcess28Retired
JPEGLosslessHierarchicalProcess29Retired
JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLossless↔ JPEGImageCompression
JPEGLSLosslessImageCompression
JPEGLSLossyNearLosslessImageCompression
JPEG2000ImageCompressionLosslessOnly
JPEG2000ImageCompression
JPEG2000Part2MulticomponentImageCompressionLosslessOnly
JPEG2000Part2MulticomponentImageCompression
JPIPReferenced

## Enumerator

JPIPReferencedDeflate
MPEG2MainProfileMainLevel
RLELossless
RFC2557MIMEencapsulation
XMLEncoding
MediaStorageDirectoryStorage
TalairachBrainAtlasFrameofReference
SPM2T1FrameofReference
SPM2T2FrameofReference
SPM2PDFFrameofReference
SPM2EPIFrameofReference
SPM2FILT1FrameofReference
SPM2PETFrameofReference
SPM2TRANSMFrameofReference
SPM2SPECTFrameofReference
SPM2GRAYFrameofReference
SPM2WHITEFrameofReference
SPM2CSFFFrameofReference
SPM2BRAINMASKFrameofReference
SPM2AVG305T1FrameofReference
SPM2AVG152T1FrameofReference
SPM2AVG152T2FrameofReference
SPM2AVG152PDFFrameofReference
SPM2SINGLESUBJT1FrameofReference
ICBM452T1FrameofReference
ICBMSingleSubjectMRIFrameofReference
BasicStudyContentNotificationSOPClassRetired
StorageCommitmentPushModelSOPClass
StorageCommitmentPushModelSOPInstance
StorageCommitmentPullModelSOPClassRetired
StorageCommitmentPullModelSOPInstanceRetired
ProceduralEventLoggingSOPClass
ProceduralEventLoggingSOPInstance
SubstanceAdministrationLoggingSOPClass
SubstanceAdministrationLoggingSOPInstance
DICOMUIDRegistry
DICOMControlledTerminology
DICOMApplicationContextName
DetachedPatientManagementSOPClassRetired
DetachedPatientManagementMetaSOPClassRetired
DetachedVisitManagementSOPClassRetired
DetachedStudyManagementSOPClassRetired
StudyComponentManagementSOPClassRetired
ModalityPerformedProcedureStepSOPClass
ModalityPerformedProcedureStepRetrieveSOPClass

## Enumerator

ModalityPerformedProcedureStepNotificationSOPClass	
DetachedResultsManagementSOPClassRetired	
DetachedResultsManagementMetaSOPClassRetired	
DetachedStudyManagementMetaSOPClassRetired	
DetachedInterpretationManagementSOPClassRetired	
StorageServiceClass	
BasicFilmSessionSOPClass	
BasicFilmBoxSOPClass	
BasicGrayscaleImageBoxSOPClass	
BasicColorImageBoxSOPClass	
ReferencedImageBoxSOPClassRetired	
BasicGrayscalePrintManagementMetaSOPClass	
ReferencedGrayscalePrintManagementMetaSOPClassRetired	
PrintJobSOPClass	
BasicAnnotationBoxSOPClass	
PrinterSOPClass	
PrinterConfigurationRetrievalSOPClass	
PrinterSOPInstance	
PrinterConfigurationRetrievalSOPInstance	
BasicColorPrintManagementMetaSOPClass	
ReferencedColorPrintManagementMetaSOPClassRetired	
VOILUTBoxSOPClass	
PresentationLUTSOPClass	
ImageOverlayBoxSOPClassRetired	
BasicPrintImageOverlayBoxSOPClassRetired	
PrintQueueSOPInstanceRetired	
PrintQueueManagementSOPClassRetired	
StoredPrintStorageSOPClassRetired	
HardcopyGrayscaleImageStorageSOPClassRetired	
HardcopyColorImageStorageSOPClassRetired	
PullPrintRequestSOPClassRetired	
PullStoredPrintManagementMetaSOPClassRetired	
MediaCreationManagementSOPClassUID	
ComputedRadiographyImageStorage	
DigitalXRayImageStorageForPresentation	
DigitalXRayImageStorageForProcessing	
DigitalMammographyXRayImageStorageForPresentation	
DigitalMammographyXRayImageStorageForProcessing	
DigitalIntraoralXRayImageStorageForPresentation	
DigitalIntraoralXRayImageStorageForProcessing	
CTImageStorage	
EnhancedCTImageStorage	
UltrasoundMultiframeImageStorageRetired	

## Enumerator

UltrasoundMultiframeImageStorage	
MRImageStorage	
EnhancedMRImageStorage	
MRSpectroscopyStorage	
NuclearMedicineImageStorageRetired	
UltrasoundImageStorageRetired	
UltrasoundImageStorage	
SecondaryCaptureImageStorage	
MultiframeSingleBitSecondaryCaptureImageStorage	
MultiframeGrayscaleByteSecondaryCaptureImageStorage	
MultiframeGrayscaleWordSecondaryCaptureImageStorage	
MultiframeTrueColorSecondaryCaptureImageStorage	
StandaloneOverlayStorageRetired	
StandaloneCurveStorageRetired	
WaveformStorageTrialRetired	
ECG12leadWaveformStorage	
GeneralECGWaveformStorage	
AmbulatoryECGWaveformStorage	
HemodynamicWaveformStorage	
CardiacElectrophysiologyWaveformStorage	
BasicVoiceAudioWaveformStorage	
StandaloneModalityLUTStorageRetired	
StandaloneVOILUTStorageRetired	
GrayscaleSoftcopyPresentationStateStorageSOPClass	
ColorSoftcopyPresentationStateStorageSOPClass	
PseudoColorSoftcopyPresentationStateStorageSOPClass	
BlendingSoftcopyPresentationStateStorageSOPClass	
XRyAngiographicImageStorage	
EnhancedXAImageStorage	
XRyRadiofluoroscopicImageStorage	
EnhancedXRFIImageStorage	
XRy3DAngiographicImageStorage	
XRy3DCraniofacialImageStorage	
XRyAngiographicBiPlaneImageStorageRetired	
NuclearMedicineImageStorage	
RawDataStorage	
SpatialRegistrationStorage	
SpatialFiducialsStorage	
DeformableSpatialRegistrationStorage	
SegmentationStorage	
RealWorldValueMappingStorage	
VLIImageStorageTrialRetired	
VLMultiframeImageStorageTrialRetired	

## Enumerator

VLEndoscopicImageStorage	
VideoEndoscopicImageStorage	
VLMicroscopicImageStorage	
VideoMicroscopicImageStorage	
VLSlideCoordinatesMicroscopicImageStorage	
VLPhotographicImageStorage	
VideoPhotographicImageStorage	
OphthalmicPhotography8BitImageStorage	
OphthalmicPhotography16BitImageStorage	
StereometricRelationshipStorage	
OphthalmicTomographyImageStorage	
TextSRStorageTrialRetired	
AudioSRStorageTrialRetired	
DetailSRStorageTrialRetired	
ComprehensiveSRStorageTrialRetired	
BasicTextSRStorage	
EnhancedSRStorage	
ComprehensiveSRStorage	
ProcedureLogStorage	
MammographyCADSRStorage	
KeyObjectSelectionDocumentStorage	
ChestCADSRStorage	
XRayRadiationDoseSRStorage	
EncapsulatedPDFStorage	
EncapsulatedCDASStorage	
PositronEmissionTomographyImageStorage	
StandalonePETCurveStorageRetired	
RTImageStorage	
RTDoseStorage	
RTStructureSetStorage	
RTBeamsTreatmentRecordStorage	
RTPlanStorage	
RTBrachyTreatmentRecordStorage	
RTTreatmentSummaryRecordStorage	
RTIonPlanStorage	
RTIonBeamsTreatmentRecordStorage	
PatientRootQueryRetrieveInformationModelFIND	
PatientRootQueryRetrieveInformationModelMOVE	
PatientRootQueryRetrieveInformationModelGET	
StudyRootQueryRetrieveInformationModelFIND	
StudyRootQueryRetrieveInformationModelMOVE	
StudyRootQueryRetrieveInformationModelGET	
PatientStudyOnlyQueryRetrieveInformationModelFINDRetired	

## Enumerator

PatientStudyOnlyQueryRetrieveInformationModelMOVERetired	
PatientStudyOnlyQueryRetrieveInformationModelGETRetired	
ModalityWorklistInformationModelFIND	
GeneralPurposeWorklistInformationModelFIND	
GeneralPurposeScheduledProcedureStepSOPClass	
GeneralPurposePerformedProcedureStepSOPClass	
GeneralPurposeWorklistManagementMetaSOPClass	
InstanceAvailabilityNotificationSOPClass	
RTBeamsDeliveryInstructionStorageSupplement74FrozenDraft	
RTConventionalMachineVerificationSupplement74FrozenDraft	
RTIonMachineVerificationSupplement74FrozenDraft	
UnifiedWorklistandProcedureStepServiceClass	
UnifiedProcedureStepPushSOPClass	
UnifiedProcedureStepWatchSOPClass	
UnifiedProcedureStepPullSOPClass	
UnifiedProcedureStepEventSOPClass	
UnifiedWorklistandProcedureStepSOPInstance	
GeneralRelevantPatientInformationQuery	
BreastImagingRelevantPatientInformationQuery	
CardiacRelevantPatientInformationQuery	
HangingProtocolStorage	
HangingProtocolInformationModelFIND	
HangingProtocolInformationModelMOVE	
ProductCharacteristicsQuerySOPClass	
SubstanceApprovalQuerySOPClass	
dicomDeviceName	
dicomDescription	
dicomManufacturer	
dicomManufacturerModelName	
dicomSoftwareVersion	
dicomVendorData	
dicomAETitle	
dicomNetworkConnectionReference	
dicomApplicationCluster	
dicomAssociationInitiator	
dicomAssociationAcceptor	
dicomHostname	
dicomPort	
dicomSOPClass	
dicomTransferRole	
dicomTransferSyntax	
dicomPrimaryDeviceType	
dicomRelatedDeviceReference	
dicomPreferredCalledAETitle	



## Enumerator

dicomTLSCyphersuite	
dicomAuthorizedNodeCertificateReference	
dicomThisNodeCertificateReference	
dicomInstalled	
dicomStationName	
dicomDeviceSerialNumber	
dicomInstitutionName	
dicomInstitutionAddress	
dicomInstitutionDepartmentName	
dicomIssuerOfPatientID	
dicomPreferredCallingAETitle	
dicomSupportedCharacterSet	
dicomConfigurationRoot	
dicomDevicesRoot	
dicomUniqueAETitlesRegistryRoot	
dicomDevice	
dicomNetworkAE	
dicomNetworkConnection	
dicomUniqueAETitle	
dicomTransferCapability	
VLWholeSlideMicroscopyImageStorage	
EnhancedUSVolumeStorage	
SurfaceSegmentationStorage	
BreastTomosynthesisImageStorage	
LegacyConvertedEnhancedCTImageStorage	
LegacyConvertedEnhancedMRIImageStorage	
LegacyConvertedEnhancedPETImageStorage	
MPEG2MainProfileHighLevel	
MPEG4AVCH_264HighProfileLevel4_1	
MPEG4AVCH_264BDcompatibleHighProfileLevel4_1	
PETColorPaletteSOPInstance	
HotMetalBlueColorPaletteSOPInstance	
PET20StepColorPaletteSOPInstance	
SpringColorPaletteSOPInstance	
SummerColorPaletteSOPInstance	
FallColorPaletteSOPInstance	
WinterColorPaletteSOPInstance	
Papyrus3ImplicitVRLittleEndian	
AdultMouseAnatomyOntology	
UberonOntology	
IntegratedTaxonomicInformationSystemITISTaxonomicSerialNumberTSN	
MouseGenomeInitiativeMGI	
PubChemCompoundCID	
ICD11	

## Enumerator

NewYorkUniversityMelanomaClinicalCooperativeGroup
MayoClinicNonradiologicalImagesSBSAnatomicalSurfaceRegionGuide
ImageBiomarkerStandardisationInitiative
RadiomicsOntology
DisplaySystemSOPClass
DisplaySystemSOPInstance
GeneralAudioWaveformStorage
ArterialPulseWaveformStorage
RespiratoryWaveformStorage
XAXRFGrayscaleSoftcopyPresentationStateStorage
GrayscalePlanarMPRVolumetricPresentationStateStorage
MPEG4AVCH_264HighProfileLevel4_2For2DVideo
MPEG4AVCH_264HighProfileLevel4_2For3DVideo
MPEG4AVCH_264StereoHighProfileLevel4_2
HEVCH_265MainProfileLevel5_1
HEVCH_265Main10ProfileLevel5_1
HotIronColorPaletteSOPInstance
CompositingPlanarMPRVolumetricPresentationStateStorage
AdvancedBlendingPresentationStateStorage
VolumeRenderingVolumetricPresentationStateStorage
SegmentedVolumeRenderingVolumetricPresentationStateStorage
MultipleVolumeRenderingVolumetricPresentationStateStorage
Null0
BreastProjectionXRayImageStorageForPresentation
BreastProjectionXRayImageStorageForProcessing
IntravascularOpticalCoherenceTomographyImageStorageForPresentation
IntravascularOpticalCoherenceTomographyImageStorageForProcessing
ParametricMapStorage
Null1
TractographyResultsStorage
SurfaceScanMeshStorage
SurfaceScanPointCloudStorage
WideFieldOphthalmicPhotographyStereographicProjectionImageStorage
WideFieldOphthalmicPhotography3DCoordinatesImageStorage
OphthalmicOpticalCoherenceTomographyEnFaceImageStorage
OphthalmicOpticalCoherenceTomographyBscanVolumeAnalysisStorage
LensometryMeasurementsStorage
AutorefractionMeasurementsStorage
KeratometryMeasurementsStorage
SubjectiveRefractionMeasurementsStorage
VisualAcuityMeasurementsStorage
SpectaclePrescriptionReportStorage
OphthalmicAxialMeasurementsStorage

## Enumerator

IntraocularLensCalculationsStorage	
MacularGridThicknessandVolumeReportStorage	
OphthalmicVisualFieldStaticPerimetryMeasurementsStorage	
OphthalmicThicknessMapStorage	
CornealTopographyMapStorage	
Comprehensive3DSRStorage	
ExtensibleSRStorage	
RadiopharmaceuticalRadiationDoseSRStorage	
ColonCADSRStorage	
ImplantationPlanSRStorage	
AcquisitionContextSRStorage	
SimplifiedAdultEchoSRStorage	
PatientRadiationDoseSRStorage	
PlannedImagingAgentAdministrationSRStorage	
PerformedImagingAgentAdministrationSRStorage	
ContentAssessmentResultsStorage	
EncapsulatedSTLStorage	
EnhancedPETImageStorage	
BasicStructuredDisplayStorage	
CTDefinedProcedureProtocolStorage	
CTPerformedProcedureProtocolStorage	
ProtocolApprovalStorage	
ProtocolApprovalInformationModelFIND	
ProtocolApprovalInformationModelMOVE	
ProtocolApprovalInformationModelGET	
RTPhysicianIntentStorage	
RTSegmentAnnotationStorage	
DICOSCTImageStorage	
DICOSDigitalXRayImageStorageForPresentation	
DICOSDigitalXRayImageStorageForProcessing	
DICOSThreatDetectionReportStorage	
DICOS2DAITStorage	
DICOS3DAITStorage	
DICOSQuadrupoleResonanceQRStorage	
EddyCurrentImageStorage	
EddyCurrentMultiframeImageStorage	
CompositeInstanceRootRetrieveMOVE	
CompositeInstanceRootRetrieveGET	
CompositeInstanceRetrieveWithoutBulkDataGET	
DefinedProcedureProtocolInformationModelFIND	
DefinedProcedureProtocolInformationModelMOVE	
DefinedProcedureProtocolInformationModelGET	
UPSFilteredGlobalSubscriptionSOPInstance	

## Enumerator

UnifiedWorklistandProcedureStepServiceClass1	
UnifiedProcedureStepPushSOPClass1	
UnifiedProcedureStepWatchSOPClass1	
UnifiedProcedureStepPullSOPClass1	
UnifiedProcedureStepEventSOPClass1	
RTBeamsDeliveryInstructionStorage	
RTConventionalMachineVerification	
RTIonMachineVerification	
RTBrachyApplicationSetupDeliveryInstructionStorage	
HangingProtocolInformationModelGET	
ColorPaletteStorage	
ColorPaletteQueryRetrieveInformationModelFIND	
ColorPaletteQueryRetrieveInformationModelMOVE	
ColorPaletteQueryRetrieveInformationModelGET	
GenericImplantTemplateStorage	
GenericImplantTemplateInformationModelFIND	
GenericImplantTemplateInformationModelMOVE	
GenericImplantTemplateInformationModelGET	
ImplantAssemblyTemplateStorage	
ImplantAssemblyTemplateInformationModelFIND	
ImplantAssemblyTemplateInformationModelMOVE	
ImplantAssemblyTemplateInformationModelGET	
ImplantTemplateGroupStorage	
ImplantTemplateGroupInformationModelFIND	
ImplantTemplateGroupInformationModelMOVE	
ImplantTemplateGroupInformationModelGET	
NativeDICOMModel	
AbstractMultiDimensionalImageModel	
DICOMContentMappingResource	
EnhancedMRColorImageStorage	
UniversalCoordinatedTime	

## 10.324.3.2 TSType

```
enum gdcm::UIDs::TSType
```

## Enumerator

uid_1_2_840_10008_1_1	
uid_1_2_840_10008_1_2	
uid_1_2_840_10008_1_2_1	
uid_1_2_840_10008_1_2_1_99	
uid_1_2_840_10008_1_2_2	

## Enumerator

uid_1_2_840_10008_1_2_4_50	
uid_1_2_840_10008_1_2_4_51	
uid_1_2_840_10008_1_2_4_52	
uid_1_2_840_10008_1_2_4_53	
uid_1_2_840_10008_1_2_4_54	
uid_1_2_840_10008_1_2_4_55	
uid_1_2_840_10008_1_2_4_56	
uid_1_2_840_10008_1_2_4_57	
uid_1_2_840_10008_1_2_4_58	
uid_1_2_840_10008_1_2_4_59	
uid_1_2_840_10008_1_2_4_60	
uid_1_2_840_10008_1_2_4_61	
uid_1_2_840_10008_1_2_4_62	
uid_1_2_840_10008_1_2_4_63	
uid_1_2_840_10008_1_2_4_64	
uid_1_2_840_10008_1_2_4_65	
uid_1_2_840_10008_1_2_4_66	
uid_1_2_840_10008_1_2_4_70	
uid_1_2_840_10008_1_2_4_80	
uid_1_2_840_10008_1_2_4_81	
uid_1_2_840_10008_1_2_4_90	
uid_1_2_840_10008_1_2_4_91	
uid_1_2_840_10008_1_2_4_92	
uid_1_2_840_10008_1_2_4_93	
uid_1_2_840_10008_1_2_4_94	
uid_1_2_840_10008_1_2_4_95	
uid_1_2_840_10008_1_2_4_100	
uid_1_2_840_10008_1_2_5	
uid_1_2_840_10008_1_2_6_1	
uid_1_2_840_10008_1_2_6_2	
uid_1_2_840_10008_1_3_10	
uid_1_2_840_10008_1_4_1_1	
uid_1_2_840_10008_1_4_1_2	
uid_1_2_840_10008_1_4_1_3	
uid_1_2_840_10008_1_4_1_4	
uid_1_2_840_10008_1_4_1_5	
uid_1_2_840_10008_1_4_1_6	
uid_1_2_840_10008_1_4_1_7	
uid_1_2_840_10008_1_4_1_8	
uid_1_2_840_10008_1_4_1_9	
uid_1_2_840_10008_1_4_1_10	
uid_1_2_840_10008_1_4_1_11	
uid_1_2_840_10008_1_4_1_12	
uid_1_2_840_10008_1_4_1_13	
uid_1_2_840_10008_1_4_1_14	
uid_1_2_840_10008_1_4_1_15	

## Enumerator

uid_1_2_840_10008_1_4_1_16	
uid_1_2_840_10008_1_4_1_17	
uid_1_2_840_10008_1_4_1_18	
uid_1_2_840_10008_1_4_2_1	
uid_1_2_840_10008_1_4_2_2	
uid_1_2_840_10008_1_9	
uid_1_2_840_10008_1_20_1	
uid_1_2_840_10008_1_20_1_1	
uid_1_2_840_10008_1_20_2	
uid_1_2_840_10008_1_20_2_1	
uid_1_2_840_10008_1_40	
uid_1_2_840_10008_1_40_1	
uid_1_2_840_10008_1_42	
uid_1_2_840_10008_1_42_1	
uid_1_2_840_10008_2_6_1	
uid_1_2_840_10008_2_16_4	
uid_1_2_840_10008_3_1_1_1	
uid_1_2_840_10008_3_1_2_1_1	
uid_1_2_840_10008_3_1_2_1_4	
uid_1_2_840_10008_3_1_2_2_1	
uid_1_2_840_10008_3_1_2_3_1	
uid_1_2_840_10008_3_1_2_3_2	
uid_1_2_840_10008_3_1_2_3_3	
uid_1_2_840_10008_3_1_2_3_4	
uid_1_2_840_10008_3_1_2_3_5	
uid_1_2_840_10008_3_1_2_5_1	
uid_1_2_840_10008_3_1_2_5_4	
uid_1_2_840_10008_3_1_2_5_5	
uid_1_2_840_10008_3_1_2_6_1	
uid_1_2_840_10008_4_2	
uid_1_2_840_10008_5_1_1_1	
uid_1_2_840_10008_5_1_1_2	
uid_1_2_840_10008_5_1_1_4	
uid_1_2_840_10008_5_1_1_4_1	
uid_1_2_840_10008_5_1_1_4_2	
uid_1_2_840_10008_5_1_1_9	
uid_1_2_840_10008_5_1_1_9_1	
uid_1_2_840_10008_5_1_1_14	
uid_1_2_840_10008_5_1_1_15	
uid_1_2_840_10008_5_1_1_16	
uid_1_2_840_10008_5_1_1_16_376	
uid_1_2_840_10008_5_1_1_17	
uid_1_2_840_10008_5_1_1_17_376	
uid_1_2_840_10008_5_1_1_18	
uid_1_2_840_10008_5_1_1_18_1	
uid_1_2_840_10008_5_1_1_22	

## Enumerator

uid_1_2_840_10008_5_1_1_23	
uid_1_2_840_10008_5_1_1_24	
uid_1_2_840_10008_5_1_1_24_1	
uid_1_2_840_10008_5_1_1_25	
uid_1_2_840_10008_5_1_1_26	
uid_1_2_840_10008_5_1_1_27	
uid_1_2_840_10008_5_1_1_29	
uid_1_2_840_10008_5_1_1_30	
uid_1_2_840_10008_5_1_1_31	
uid_1_2_840_10008_5_1_1_32	
uid_1_2_840_10008_5_1_1_33	
uid_1_2_840_10008_5_1_4_1_1_1	
uid_1_2_840_10008_5_1_4_1_1_1_1	
uid_1_2_840_10008_5_1_4_1_1_1_1_1	
uid_1_2_840_10008_5_1_4_1_1_1_2	
uid_1_2_840_10008_5_1_4_1_1_1_2_1	
uid_1_2_840_10008_5_1_4_1_1_1_3	
uid_1_2_840_10008_5_1_4_1_1_1_3_1	
uid_1_2_840_10008_5_1_4_1_1_2	
uid_1_2_840_10008_5_1_4_1_1_2_1	
uid_1_2_840_10008_5_1_4_1_1_3	
uid_1_2_840_10008_5_1_4_1_1_3_1	
uid_1_2_840_10008_5_1_4_1_1_4	
uid_1_2_840_10008_5_1_4_1_1_4_1	
uid_1_2_840_10008_5_1_4_1_1_4_2	
uid_1_2_840_10008_5_1_4_1_1_5	
uid_1_2_840_10008_5_1_4_1_1_6	
uid_1_2_840_10008_5_1_4_1_1_6_1	
uid_1_2_840_10008_5_1_4_1_1_7	
uid_1_2_840_10008_5_1_4_1_1_7_1	
uid_1_2_840_10008_5_1_4_1_1_7_2	
uid_1_2_840_10008_5_1_4_1_1_7_3	
uid_1_2_840_10008_5_1_4_1_1_7_4	
uid_1_2_840_10008_5_1_4_1_1_8	
uid_1_2_840_10008_5_1_4_1_1_9	
uid_1_2_840_10008_5_1_4_1_1_9_1	
uid_1_2_840_10008_5_1_4_1_1_9_1_1	
uid_1_2_840_10008_5_1_4_1_1_9_1_2	
uid_1_2_840_10008_5_1_4_1_1_9_1_3	
uid_1_2_840_10008_5_1_4_1_1_9_2_1	
uid_1_2_840_10008_5_1_4_1_1_9_3_1	
uid_1_2_840_10008_5_1_4_1_1_9_4_1	
uid_1_2_840_10008_5_1_4_1_1_10	
uid_1_2_840_10008_5_1_4_1_1_11	
uid_1_2_840_10008_5_1_4_1_1_11_1	
uid_1_2_840_10008_5_1_4_1_1_11_2	

## Enumerator

uid_1_2_840_10008_5_1_4_1_1_11_3	
uid_1_2_840_10008_5_1_4_1_1_11_4	
uid_1_2_840_10008_5_1_4_1_1_12_1	
uid_1_2_840_10008_5_1_4_1_1_12_1_1	
uid_1_2_840_10008_5_1_4_1_1_12_2	
uid_1_2_840_10008_5_1_4_1_1_12_2_1	
uid_1_2_840_10008_5_1_4_1_1_13_1_1	
uid_1_2_840_10008_5_1_4_1_1_13_1_2	
uid_1_2_840_10008_5_1_4_1_1_12_3	
uid_1_2_840_10008_5_1_4_1_1_20	
uid_1_2_840_10008_5_1_4_1_1_66	
uid_1_2_840_10008_5_1_4_1_1_66_1	
uid_1_2_840_10008_5_1_4_1_1_66_2	
uid_1_2_840_10008_5_1_4_1_1_66_3	
uid_1_2_840_10008_5_1_4_1_1_66_4	
uid_1_2_840_10008_5_1_4_1_1_67	
uid_1_2_840_10008_5_1_4_1_1_77_1	
uid_1_2_840_10008_5_1_4_1_1_77_2	
uid_1_2_840_10008_5_1_4_1_1_77_1_1	
uid_1_2_840_10008_5_1_4_1_1_77_1_1↵ _1	
uid_1_2_840_10008_5_1_4_1_1_77_1_2	
uid_1_2_840_10008_5_1_4_1_1_77_1_2↵ _1	
uid_1_2_840_10008_5_1_4_1_1_77_1_3	
uid_1_2_840_10008_5_1_4_1_1_77_1_4	
uid_1_2_840_10008_5_1_4_1_1_77_1_4↵ _1	
uid_1_2_840_10008_5_1_4_1_1_77_1_5↵ _1	
uid_1_2_840_10008_5_1_4_1_1_77_1_5↵ _2	
uid_1_2_840_10008_5_1_4_1_1_77_1_5↵ _3	
uid_1_2_840_10008_5_1_4_1_1_77_1_5↵ _4	
uid_1_2_840_10008_5_1_4_1_1_88_1	
uid_1_2_840_10008_5_1_4_1_1_88_2	
uid_1_2_840_10008_5_1_4_1_1_88_3	
uid_1_2_840_10008_5_1_4_1_1_88_4	
uid_1_2_840_10008_5_1_4_1_1_88_11	
uid_1_2_840_10008_5_1_4_1_1_88_22	
uid_1_2_840_10008_5_1_4_1_1_88_33	
uid_1_2_840_10008_5_1_4_1_1_88_40	
uid_1_2_840_10008_5_1_4_1_1_88_50	
uid_1_2_840_10008_5_1_4_1_1_88_59	
uid_1_2_840_10008_5_1_4_1_1_88_65	



## Enumerator

uid_1_2_840_10008_5_1_4_1_1_88_67	
uid_1_2_840_10008_5_1_4_1_1_104_1	
uid_1_2_840_10008_5_1_4_1_1_104_2	
uid_1_2_840_10008_5_1_4_1_1_128	
uid_1_2_840_10008_5_1_4_1_1_129	
uid_1_2_840_10008_5_1_4_1_1_481_1	
uid_1_2_840_10008_5_1_4_1_1_481_2	
uid_1_2_840_10008_5_1_4_1_1_481_3	
uid_1_2_840_10008_5_1_4_1_1_481_4	
uid_1_2_840_10008_5_1_4_1_1_481_5	
uid_1_2_840_10008_5_1_4_1_1_481_6	
uid_1_2_840_10008_5_1_4_1_1_481_7	
uid_1_2_840_10008_5_1_4_1_1_481_8	
uid_1_2_840_10008_5_1_4_1_1_481_9	
uid_1_2_840_10008_5_1_4_1_2_1_1	
uid_1_2_840_10008_5_1_4_1_2_1_2	
uid_1_2_840_10008_5_1_4_1_2_1_3	
uid_1_2_840_10008_5_1_4_1_2_2_1	
uid_1_2_840_10008_5_1_4_1_2_2_2	
uid_1_2_840_10008_5_1_4_1_2_2_3	
uid_1_2_840_10008_5_1_4_1_2_3_1	
uid_1_2_840_10008_5_1_4_1_2_3_2	
uid_1_2_840_10008_5_1_4_1_2_3_3	
uid_1_2_840_10008_5_1_4_31	
uid_1_2_840_10008_5_1_4_32_1	
uid_1_2_840_10008_5_1_4_32_2	
uid_1_2_840_10008_5_1_4_32_3	
uid_1_2_840_10008_5_1_4_32	
uid_1_2_840_10008_5_1_4_33	
uid_1_2_840_10008_5_1_4_34_1	
uid_1_2_840_10008_5_1_4_34_2	
uid_1_2_840_10008_5_1_4_34_3	
uid_1_2_840_10008_5_1_4_34_4	
uid_1_2_840_10008_5_1_4_34_4_1	
uid_1_2_840_10008_5_1_4_34_4_2	
uid_1_2_840_10008_5_1_4_34_4_3	
uid_1_2_840_10008_5_1_4_34_4_4	
uid_1_2_840_10008_5_1_4_34_5	
uid_1_2_840_10008_5_1_4_37_1	
uid_1_2_840_10008_5_1_4_37_2	
uid_1_2_840_10008_5_1_4_37_3	
uid_1_2_840_10008_5_1_4_38_1	
uid_1_2_840_10008_5_1_4_38_2	
uid_1_2_840_10008_5_1_4_38_3	
uid_1_2_840_10008_5_1_4_41	
uid_1_2_840_10008_5_1_4_42	

## Enumerator

uid_1_2_840_10008_15_0_3_1	
uid_1_2_840_10008_15_0_3_2	
uid_1_2_840_10008_15_0_3_3	
uid_1_2_840_10008_15_0_3_4	
uid_1_2_840_10008_15_0_3_5	
uid_1_2_840_10008_15_0_3_6	
uid_1_2_840_10008_15_0_3_7	
uid_1_2_840_10008_15_0_3_8	
uid_1_2_840_10008_15_0_3_9	
uid_1_2_840_10008_15_0_3_10	
uid_1_2_840_10008_15_0_3_11	
uid_1_2_840_10008_15_0_3_12	
uid_1_2_840_10008_15_0_3_13	
uid_1_2_840_10008_15_0_3_14	
uid_1_2_840_10008_15_0_3_15	
uid_1_2_840_10008_15_0_3_16	
uid_1_2_840_10008_15_0_3_17	
uid_1_2_840_10008_15_0_3_18	
uid_1_2_840_10008_15_0_3_19	
uid_1_2_840_10008_15_0_3_20	
uid_1_2_840_10008_15_0_3_21	
uid_1_2_840_10008_15_0_3_22	
uid_1_2_840_10008_15_0_3_23	
uid_1_2_840_10008_15_0_3_24	
uid_1_2_840_10008_15_0_3_25	
uid_1_2_840_10008_15_0_3_26	
uid_1_2_840_10008_15_0_3_27	
uid_1_2_840_10008_15_0_3_28	
uid_1_2_840_10008_15_0_3_29	
uid_1_2_840_10008_15_0_3_30	
uid_1_2_840_10008_15_0_3_31	
uid_1_2_840_10008_15_0_4_1	
uid_1_2_840_10008_15_0_4_2	
uid_1_2_840_10008_15_0_4_3	
uid_1_2_840_10008_15_0_4_4	
uid_1_2_840_10008_15_0_4_5	
uid_1_2_840_10008_15_0_4_6	
uid_1_2_840_10008_15_0_4_7	
uid_1_2_840_10008_15_0_4_8	
uid_1_2_840_10008_5_1_4_1_1_77_1_6	
uid_1_2_840_10008_5_1_4_1_1_6_2	
uid_1_2_840_10008_5_1_4_1_1_66_5	
uid_1_2_840_10008_5_1_4_1_1_13_1_3	
uid_1_2_840_10008_5_1_4_1_1_2_2	
uid_1_2_840_10008_5_1_4_1_1_4_4	
uid_1_2_840_10008_5_1_4_1_1_128_1	

## Enumerator

uid_1_2_840_10008_1_2_4_101	
uid_1_2_840_10008_1_2_4_102	
uid_1_2_840_10008_1_2_4_103	
uid_1_2_840_10008_1_5_2	
uid_1_2_840_10008_1_5_3	
uid_1_2_840_10008_1_5_4	
uid_1_2_840_10008_1_5_5	
uid_1_2_840_10008_1_5_6	
uid_1_2_840_10008_1_5_7	
uid_1_2_840_10008_1_5_8	
uid_1_2_840_10008_1_20	
uid_1_2_840_10008_2_16_5	
uid_1_2_840_10008_2_16_6	
uid_1_2_840_10008_2_16_7	
uid_1_2_840_10008_2_16_8	
uid_1_2_840_10008_2_16_9	
uid_1_2_840_10008_2_16_10	
uid_1_2_840_10008_2_16_11	
uid_1_2_840_10008_2_16_12	
uid_1_2_840_10008_2_16_13	
uid_1_2_840_10008_2_16_14	
uid_1_2_840_10008_5_1_1_40	
uid_1_2_840_10008_5_1_1_40_1	
uid_1_2_840_10008_5_1_4_1_1_9_4_2	
uid_1_2_840_10008_5_1_4_1_1_9_5_1	
uid_1_2_840_10008_5_1_4_1_1_9_6_1	
uid_1_2_840_10008_5_1_4_1_1_11_5	
uid_1_2_840_10008_5_1_4_1_1_11_6	
uid_1_2_840_10008_1_2_4_104	
uid_1_2_840_10008_1_2_4_105	
uid_1_2_840_10008_1_2_4_106	
uid_1_2_840_10008_1_2_4_107	
uid_1_2_840_10008_1_2_4_108	
uid_1_2_840_10008_1_5_1	
uid_1_2_840_10008_5_1_4_1_1_11_7	
uid_1_2_840_10008_5_1_4_1_1_11_8	
uid_1_2_840_10008_5_1_4_1_1_11_9	
uid_1_2_840_10008_5_1_4_1_1_11_10	
uid_1_2_840_10008_5_1_4_1_1_11_11	
uid_1_2_840_10008_5_1_4_1_1_12_77	
uid_1_2_840_10008_5_1_4_1_1_13_1_4	
uid_1_2_840_10008_5_1_4_1_1_13_1_5	
uid_1_2_840_10008_5_1_4_1_1_14_1	
uid_1_2_840_10008_5_1_4_1_1_14_2	
uid_1_2_840_10008_5_1_4_1_1_30	
uid_1_2_840_10008_5_1_4_1_1_40	

## Enumerator

uid_1_2_840_10008_5_1_4_1_1_66_6	
uid_1_2_840_10008_5_1_4_1_1_68_1	
uid_1_2_840_10008_5_1_4_1_1_68_2	
uid_1_2_840_10008_5_1_4_1_1_77_1_5↔ _5	
uid_1_2_840_10008_5_1_4_1_1_77_1_5↔ _6	
uid_1_2_840_10008_5_1_4_1_1_77_1_5↔ _7	
uid_1_2_840_10008_5_1_4_1_1_77_1_5↔ _8	
uid_1_2_840_10008_5_1_4_1_1_78_1	
uid_1_2_840_10008_5_1_4_1_1_78_2	
uid_1_2_840_10008_5_1_4_1_1_78_3	
uid_1_2_840_10008_5_1_4_1_1_78_4	
uid_1_2_840_10008_5_1_4_1_1_78_5	
uid_1_2_840_10008_5_1_4_1_1_78_6	
uid_1_2_840_10008_5_1_4_1_1_78_7	
uid_1_2_840_10008_5_1_4_1_1_78_8	
uid_1_2_840_10008_5_1_4_1_1_79_1	
uid_1_2_840_10008_5_1_4_1_1_80_1	
uid_1_2_840_10008_5_1_4_1_1_81_1	
uid_1_2_840_10008_5_1_4_1_1_82_1	
uid_1_2_840_10008_5_1_4_1_1_88_34	
uid_1_2_840_10008_5_1_4_1_1_88_35	
uid_1_2_840_10008_5_1_4_1_1_88_68	
uid_1_2_840_10008_5_1_4_1_1_88_69	
uid_1_2_840_10008_5_1_4_1_1_88_70	
uid_1_2_840_10008_5_1_4_1_1_88_71	
uid_1_2_840_10008_5_1_4_1_1_88_72	
uid_1_2_840_10008_5_1_4_1_1_88_73	
uid_1_2_840_10008_5_1_4_1_1_88_74	
uid_1_2_840_10008_5_1_4_1_1_88_75	
uid_1_2_840_10008_5_1_4_1_1_90_1	
uid_1_2_840_10008_5_1_4_1_1_104_3	
uid_1_2_840_10008_5_1_4_1_1_130	
uid_1_2_840_10008_5_1_4_1_1_131	
uid_1_2_840_10008_5_1_4_1_1_200_1	
uid_1_2_840_10008_5_1_4_1_1_200_2	
uid_1_2_840_10008_5_1_4_1_1_200_3	
uid_1_2_840_10008_5_1_4_1_1_200_4	
uid_1_2_840_10008_5_1_4_1_1_200_5	
uid_1_2_840_10008_5_1_4_1_1_200_6	
uid_1_2_840_10008_5_1_4_1_1_481_10	
uid_1_2_840_10008_5_1_4_1_1_481_11	
uid_1_2_840_10008_5_1_4_1_1_501_1	
uid_1_2_840_10008_5_1_4_1_1_501_2_1	

## Enumerator

uid_1_2_840_10008_5_1_4_1_1_501_2_2	
uid_1_2_840_10008_5_1_4_1_1_501_3	
uid_1_2_840_10008_5_1_4_1_1_501_4	
uid_1_2_840_10008_5_1_4_1_1_501_5	
uid_1_2_840_10008_5_1_4_1_1_501_6	
uid_1_2_840_10008_5_1_4_1_1_601_1	
uid_1_2_840_10008_5_1_4_1_1_601_2	
uid_1_2_840_10008_5_1_4_1_2_4_2	
uid_1_2_840_10008_5_1_4_1_2_4_3	
uid_1_2_840_10008_5_1_4_1_2_5_3	
uid_1_2_840_10008_5_1_4_20_1	
uid_1_2_840_10008_5_1_4_20_2	
uid_1_2_840_10008_5_1_4_20_3	
uid_1_2_840_10008_5_1_4_34_5_1	
uid_1_2_840_10008_5_1_4_34_6	
uid_1_2_840_10008_5_1_4_34_6_1	
uid_1_2_840_10008_5_1_4_34_6_2	
uid_1_2_840_10008_5_1_4_34_6_3	
uid_1_2_840_10008_5_1_4_34_6_4	
uid_1_2_840_10008_5_1_4_34_7	
uid_1_2_840_10008_5_1_4_34_8	
uid_1_2_840_10008_5_1_4_34_9	
uid_1_2_840_10008_5_1_4_34_10	
uid_1_2_840_10008_5_1_4_38_4	
uid_1_2_840_10008_5_1_4_39_1	
uid_1_2_840_10008_5_1_4_39_2	
uid_1_2_840_10008_5_1_4_39_3	
uid_1_2_840_10008_5_1_4_39_4	
uid_1_2_840_10008_5_1_4_43_1	
uid_1_2_840_10008_5_1_4_43_2	
uid_1_2_840_10008_5_1_4_43_3	
uid_1_2_840_10008_5_1_4_43_4	
uid_1_2_840_10008_5_1_4_44_1	
uid_1_2_840_10008_5_1_4_44_2	
uid_1_2_840_10008_5_1_4_44_3	
uid_1_2_840_10008_5_1_4_44_4	
uid_1_2_840_10008_5_1_4_45_1	
uid_1_2_840_10008_5_1_4_45_2	
uid_1_2_840_10008_5_1_4_45_3	
uid_1_2_840_10008_5_1_4_45_4	
uid_1_2_840_10008_7_1_1	
uid_1_2_840_10008_7_1_2	
uid_1_2_840_10008_8_1_1	
uid_1_2_840_10008_5_1_4_1_1_4_3	
uid_1_2_840_10008_15_1_1	

## 10.324.4 Member Function Documentation

### 10.324.4.1 GetName()

```
const char * gdcm::UIDs::GetName ( ) const
```

When object is Initialize function return the well known name associated with uid return NULL when not initialized

#### Examples

[GenerateStandardSOPClasses.cxx](#).

Referenced by [gdcm::operator<<\(\)](#).

### 10.324.4.2 GetNumberOfTransferSyntaxStrings()

```
static unsigned int gdcm::UIDs::GetNumberOfTransferSyntaxStrings ( ) [static]
```

### 10.324.4.3 GetString()

```
const char * gdcm::UIDs::GetString ( ) const
```

When object is Initialize function return the uid return NULL when not initialized

#### Examples

[GenerateStandardSOPClasses.cxx](#).

Referenced by [gdcm::operator<<\(\)](#).

### 10.324.4.4 GetTransferSyntaxString()

```
static const char *const * gdcm::UIDs::GetTransferSyntaxString (
    unsigned int ts ) [static]
```

### 10.324.4.5 GetTransferSyntaxStrings()

```
static TransferSyntaxStringsType gdcm::UIDs::GetTransferSyntaxStrings ( ) [static]
```

### 10.324.4.6 GetUIDName()

```
static const char * gdcm::UIDs::GetUIDName (
    unsigned int ts ) [static]
```

#### 10.324.4.7 GetUIDString()

```
static const char * gdcm::UIDs::GetUIDString (
    unsigned int ts ) [static]
```

#### 10.324.4.8 operator TType()

```
gdcm::UIDs::operator TType ( ) const [inline]
```

#### 10.324.4.9 SetFromUID()

```
bool gdcm::UIDs::SetFromUID (
    const char * str )
```

Initialize object from a string (a uid number) return false on error, and internal state is set to 0

#### Examples

[GenerateStandardSOPClasses.cxx](#).

The documentation for this class was generated from the following file:

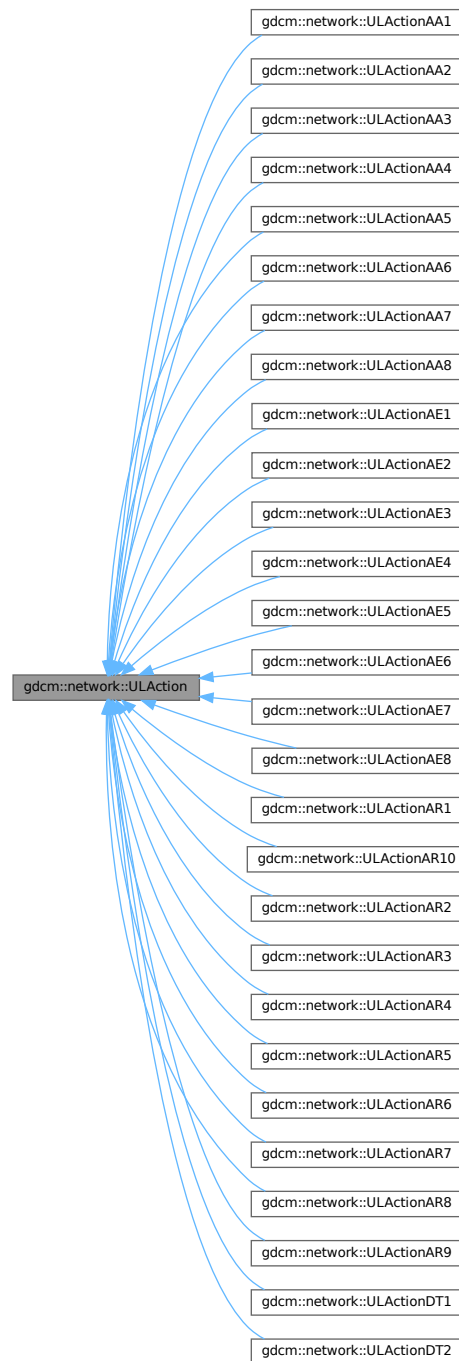
- [gdcmUIDs.h](#)

## 10.325 gdcm::network::ULAction Class Reference

[ULAction](#).

```
#include <gdcmULAction.h>
```

Inheritance diagram for `gdc::network::ULAction`:



## Public Member Functions

- [ULAction](#) ()=default



- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete
- virtual [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaiting←  
ForEvent, [EEventID](#) &outRaisedEvent)=0

### 10.325.1 Detailed Description

[ULAction](#).

A [ULConnection](#) in a given ULState can perform certain ULActions. This base class provides the interface for running those ULActions on a given [ULConnection](#).

Essentially, the [ULConnectionManager](#) will take this object, determined from the current ULState of the [ULConnection](#), and pass the [ULConnection](#) object to the [ULAction](#). The [ULAction](#) will then invoke whatever necessary commands are required by a given action.

The result of a [ULAction](#) is a [ULEvent](#) (ie, what happened as a result of the action).

This [ULEvent](#) is passed to the ULState, so that the transition to the next state can occur.

Actions are associated with Payloads – be those filestreams, AETitles to establish connections, whatever. The actual parameters that the user will pass via an action will come through a Payload object, which should, in itself, be some gdcmm-based object (but not all objects can be payloads; sending a single dataelement as a payload isn't meaningful). As such, each action has its own particular payload.

For the sake of keeping files together, both the particular payload class and the action class will be defined in the same header file. Payloads should JUST be data (or streams), NO METHODS.

Some actions perform changes that should raise events on the local system, and some actions perform changes that will require waiting for events from the remote system.

Therefore, this base action has been modified so that those events are set by each action. When the event loop runs an action, it will then test to see if a local event was raised by the action, and if so, perform the appropriate subsequent action. If the action requires waiting for a response from the remote system, then the event loop will sit there (presumably with the ARTIM timer running) and wait for a response from the remote system. Once a response is obtained, then the the rest of the state transitions can happen.

### 10.325.2 Constructor & Destructor Documentation

#### 10.325.2.1 [ULAction](#)() [1/2]

```
gdcmm::network::ULAction::ULAction ( ) [default]
```

#### 10.325.2.2 [~ULAction](#)()

```
virtual gdcmm::network::ULAction::~~ULAction ( ) [virtual], [default]
```

### 10.325.2.3 ULAction() [2/2]

```
gdcmm::network::ULAction::ULAction (
    const ULAction & inAction ) [delete]
```

## 10.325.3 Member Function Documentation

### 10.325.3.1 operator=()

```
void gdcmm::network::ULAction::operator= (
    const ULAction & ) [delete]
```

### 10.325.3.2 PerformAction()

```
virtual EStateID gdcmm::network::ULAction::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [pure virtual]
```

Implemented in [gdcmm::network::ULActionAA1](#), [gdcmm::network::ULActionAA2](#), [gdcmm::network::ULActionAA3](#), [gdcmm::network::ULActionAA4](#), [gdcmm::network::ULActionAA5](#), [gdcmm::network::ULActionAA6](#), [gdcmm::network::ULActionAA7](#), [gdcmm::network::ULActionAA8](#), [gdcmm::network::ULActionAE1](#), [gdcmm::network::ULActionAE2](#), [gdcmm::network::ULActionAE3](#), [gdcmm::network::ULActionAE4](#), [gdcmm::network::ULActionAE5](#), [gdcmm::network::ULActionAE6](#), [gdcmm::network::ULActionAE7](#), [gdcmm::network::ULActionAE8](#), [gdcmm::network::ULActionAR1](#), [gdcmm::network::ULActionAR2](#), [gdcmm::network::ULActionAR3](#), [gdcmm::network::ULActionAR4](#), [gdcmm::network::ULActionAR5](#), [gdcmm::network::ULActionAR6](#), [gdcmm::network::ULActionAR7](#), [gdcmm::network::ULActionAR8](#), [gdcmm::network::ULActionAR9](#), [gdcmm::network::ULActionAR10](#), [gdcmm::network::ULActionDT1](#), and [gdcmm::network::ULActionDT2](#).

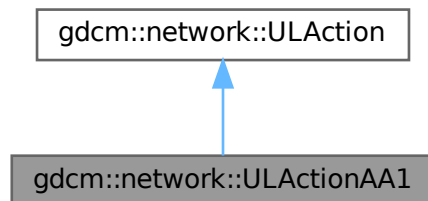
The documentation for this class was generated from the following file:

- [gdcmmULAction.h](#)

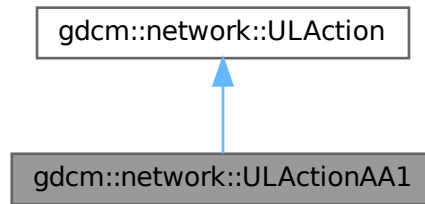
## 10.326 gdcmm::network::ULActionAA1 Class Reference

```
#include <gdcmmULActionAA.h>
```

Inheritance diagram for gdcmm::network::ULActionAA1:



Collaboration diagram for gdcm::network::ULActionAA1:



### Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

### Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

## 10.326.1 Member Function Documentation

### 10.326.1.1 PerformAction()

```

EStateID gdcm::network::ULActionAA1::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
  
```

Implements [gdcm::network::ULAction](#).

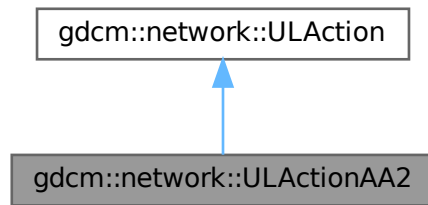
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

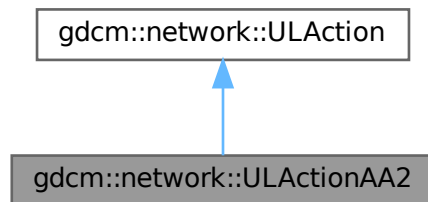
## 10.327 gdcmm::network::ULActionAA2 Class Reference

```
#include <gdcmmULActionAA.h>
```

Inheritance diagram for gdcmm::network::ULActionAA2:



Collaboration diagram for gdcmm::network::ULActionAA2:



### Public Member Functions

- `EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)` override

### Public Member Functions inherited from gdcmm::network::ULAction

- `ULAction ()`=default
- `ULAction (const ULAction &inAction)`=delete
- virtual `~ULAction ()`=default
- void `operator= (const ULAction &)`=delete

## 10.327.1 Member Function Documentation

### 10.327.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA2::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

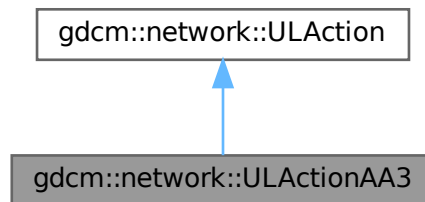
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

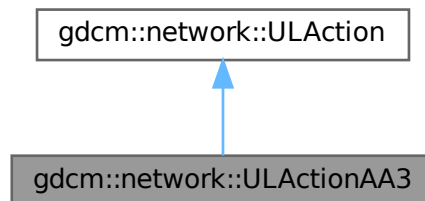
## 10.328 gdcm::network::ULActionAA3 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA3:



Collaboration diagram for gdcm::network::ULActionAA3:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

## Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

## 10.328.1 Member Function Documentation

### 10.328.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA3::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

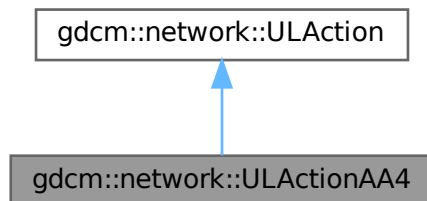
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

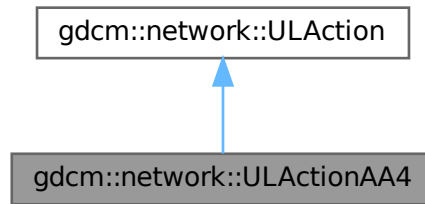
## 10.329 gdcm::network::ULActionAA4 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for [gdcm::network::ULActionAA4](#):



Collaboration diagram for gdcm::network::ULActionAA4:



### Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

### Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

## 10.329.1 Member Function Documentation

### 10.329.1.1 PerformAction()

```

EStateID gdcm::network::ULActionAA4::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
  
```

Implements [gdcm::network::ULAction](#).

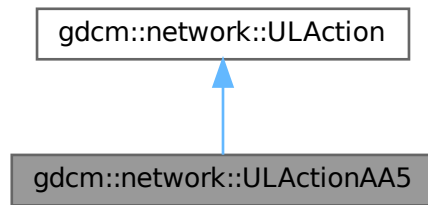
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

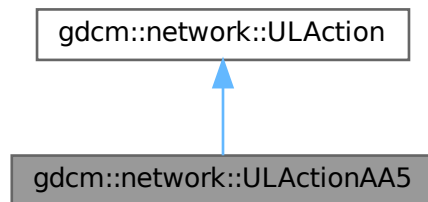
## 10.330 gdcmm::network::ULActionAA5 Class Reference

```
#include <gdcmmULActionAA.h>
```

Inheritance diagram for gdcmm::network::ULActionAA5:



Collaboration diagram for gdcmm::network::ULActionAA5:



### Public Member Functions

- `EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)` override

### Public Member Functions inherited from gdcmm::network::ULAction

- `ULAction ()`=default
- `ULAction (const ULAction &inAction)`=delete
- virtual `~ULAction ()`=default
- void `operator= (const ULAction &)`=delete



## 10.330.1 Member Function Documentation

### 10.330.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA5::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

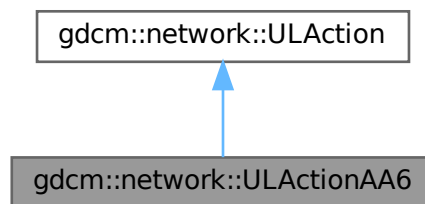
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

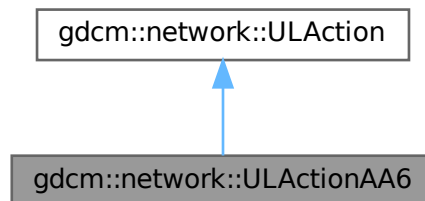
## 10.331 gdcm::network::ULActionAA6 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA6:



Collaboration diagram for gdcm::network::ULActionAA6:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

## Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

## 10.331.1 Member Function Documentation

### 10.331.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA6::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

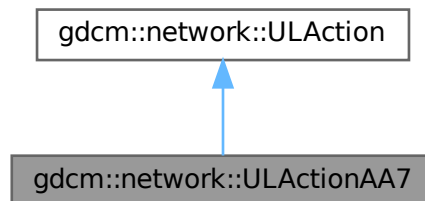
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

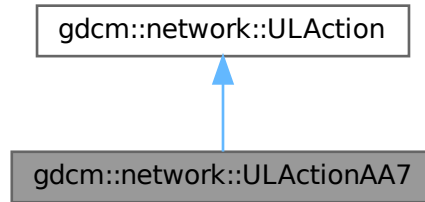
## 10.332 [gdcm::network::ULActionAA7](#) Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for [gdcm::network::ULActionAA7](#):



Collaboration diagram for gdcm::network::ULActionAA7:



### Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

### Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

## 10.332.1 Member Function Documentation

### 10.332.1.1 PerformAction()

```

EStateID gdcm::network::ULActionAA7::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
  
```

Implements [gdcm::network::ULAction](#).

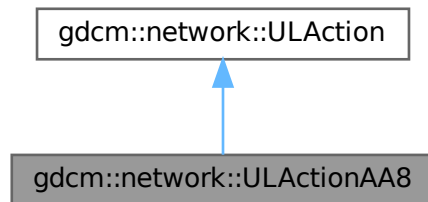
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

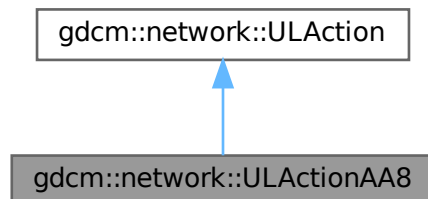
### 10.333 gdcm::network::ULActionAA8 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA8:



Collaboration diagram for gdcm::network::ULActionAA8:



#### Public Member Functions

- `EStateID PerformAction` (`Subject *s`, `ULEvent &inEvent`, `ULConnection &inConnection`, `bool &outWaitingForEvent`, `EEventID &outRaisedEvent`) override

#### Public Member Functions inherited from `gdcm::network::ULAction`

- `ULAction` ()=default
- `ULAction` (const `ULAction` &inAction)=delete
- virtual `~ULAction` ()=default
- void `operator=` (const `ULAction` &)=delete

## 10.333.1 Member Function Documentation

### 10.333.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA8::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

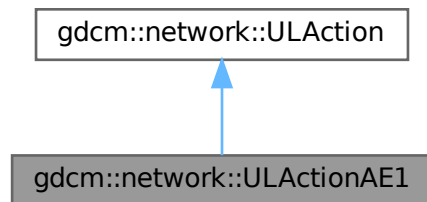
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

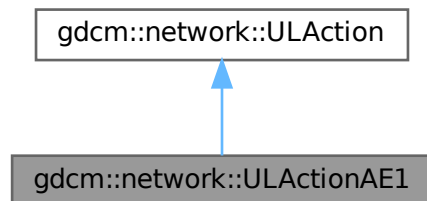
## 10.334 gdcm::network::ULActionAE1 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE1:



Collaboration diagram for gdcm::network::ULActionAE1:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

## Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

## 10.334.1 Member Function Documentation

### 10.334.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE1::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

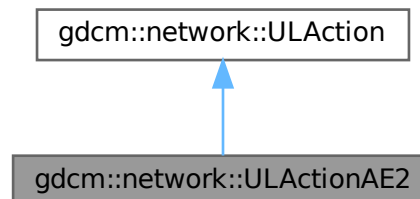
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

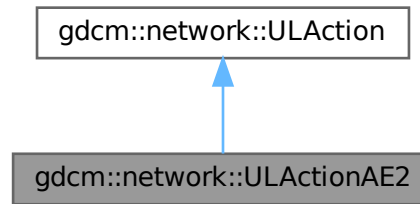
## 10.335 gdcm::network::ULActionAE2 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for [gdcm::network::ULActionAE2](#):



Collaboration diagram for gdcm::network::ULActionAE2:



### Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

### Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

## 10.335.1 Member Function Documentation

### 10.335.1.1 PerformAction()

```

EStateID gdcm::network::ULActionAE2::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
  
```

Implements [gdcm::network::ULAction](#).

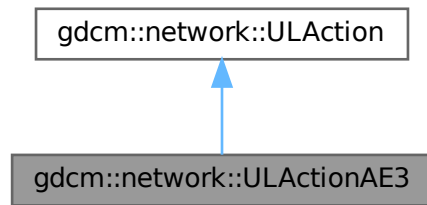
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

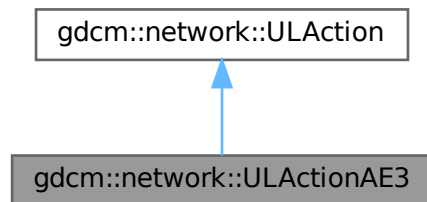
## 10.336 gdcm::network::ULActionAE3 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE3:



Collaboration diagram for gdcm::network::ULActionAE3:



### Public Member Functions

- `EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)` override

### Public Member Functions inherited from `gdcm::network::ULAction`

- `ULAction ()`=default
- `ULAction (const ULAction &inAction)`=delete
- virtual `~ULAction ()`=default
- void `operator= (const ULAction &)`=delete



## 10.336.1 Member Function Documentation

### 10.336.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE3::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

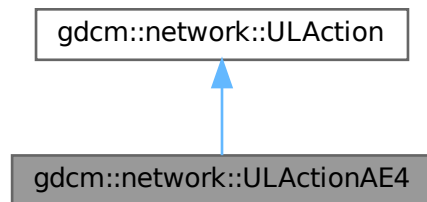
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

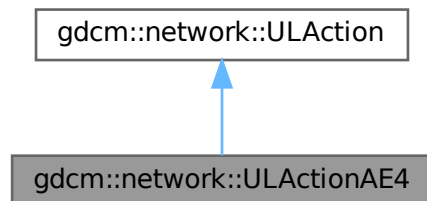
## 10.337 gdcm::network::ULActionAE4 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE4:



Collaboration diagram for gdcm::network::ULActionAE4:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

## Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

## 10.337.1 Member Function Documentation

### 10.337.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE4::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

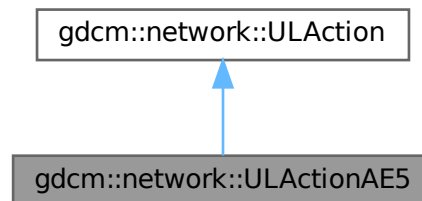
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

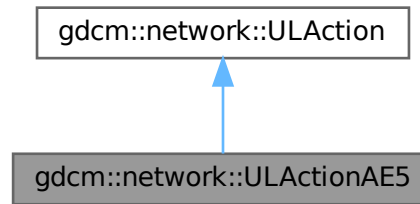
## 10.338 gdcm::network::ULActionAE5 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for [gdcm::network::ULActionAE5](#):



Collaboration diagram for gdcm::network::ULActionAE5:



### Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

### Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

## 10.338.1 Member Function Documentation

### 10.338.1.1 PerformAction()

```

EStateID gdcm::network::ULActionAE5::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
  
```

Implements [gdcm::network::ULAction](#).

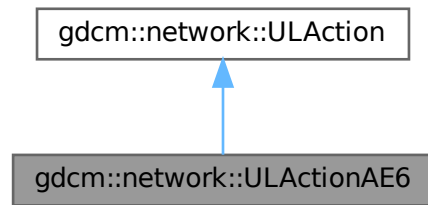
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

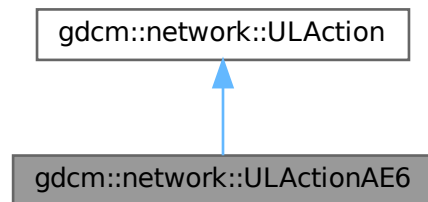
## 10.339 gdcm::network::ULActionAE6 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for `gdcm::network::ULActionAE6`:



Collaboration diagram for `gdcm::network::ULActionAE6`:



### Public Member Functions

- `EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)` override

### Public Member Functions inherited from `gdcm::network::ULAction`

- `ULAction ()`=default
- `ULAction (const ULAction &inAction)`=delete
- virtual `~ULAction ()`=default
- void `operator= (const ULAction &)`=delete

## 10.339.1 Member Function Documentation

### 10.339.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE6::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

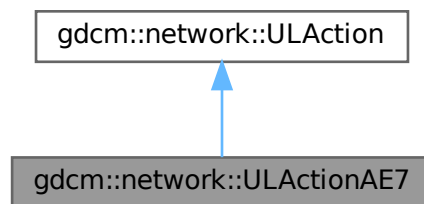
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

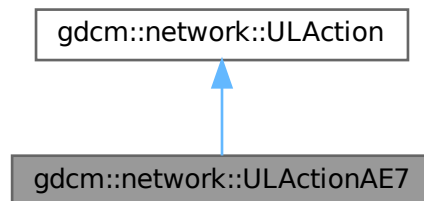
## 10.340 gdcm::network::ULActionAE7 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE7:



Collaboration diagram for gdcm::network::ULActionAE7:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

## Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

## 10.340.1 Member Function Documentation

### 10.340.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE7::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

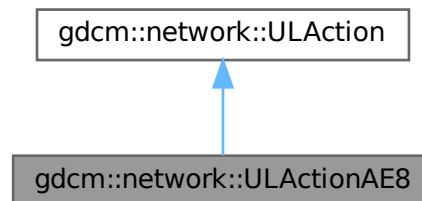
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

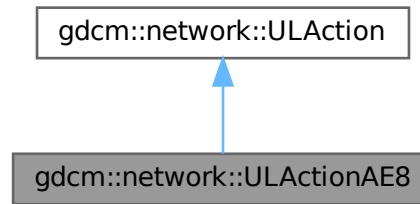
## 10.341 gdcm::network::ULActionAE8 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for [gdcm::network::ULActionAE8](#):



Collaboration diagram for gdcm::network::ULActionAE8:



### Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

### Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

## 10.341.1 Member Function Documentation

### 10.341.1.1 PerformAction()

```

EStateID gdcm::network::ULActionAE8::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
  
```

Implements [gdcm::network::ULAction](#).

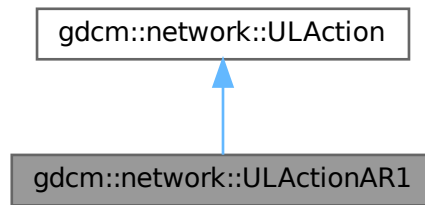
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

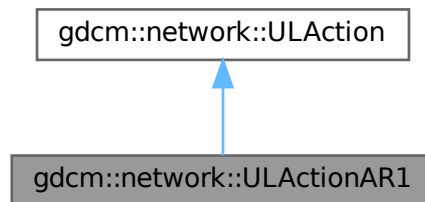
## 10.342 gdcm::network::ULActionAR1 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR1:



Collaboration diagram for gdcm::network::ULActionAR1:



### Public Member Functions

- `EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)` override

### Public Member Functions inherited from `gdcm::network::ULAction`

- `ULAction ()`=default
- `ULAction (const ULAction &inAction)`=delete
- virtual `~ULAction ()`=default
- void `operator= (const ULAction &)`=delete



## 10.342.1 Member Function Documentation

### 10.342.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR1::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

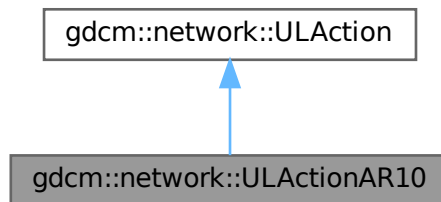
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

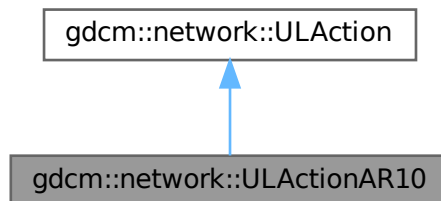
## 10.343 gdcm::network::ULActionAR10 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR10:



Collaboration diagram for gdcm::network::ULActionAR10:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

## Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

## 10.343.1 Member Function Documentation

### 10.343.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR10::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

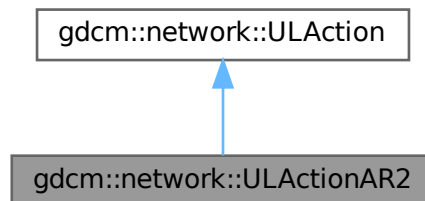
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

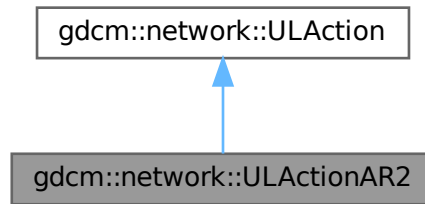
## 10.344 gdcm::network::ULActionAR2 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for [gdcm::network::ULActionAR2](#):



Collaboration diagram for gdcm::network::ULActionAR2:



### Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

### Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

## 10.344.1 Member Function Documentation

### 10.344.1.1 PerformAction()

```

EStateID gdcm::network::ULActionAR2::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
  
```

Implements [gdcm::network::ULAction](#).

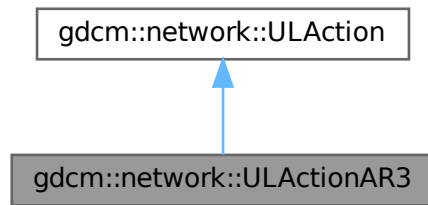
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

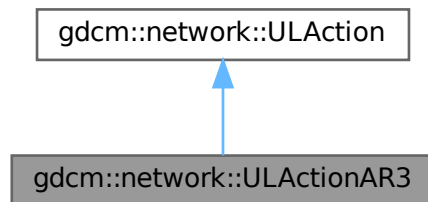
## 10.345 gdcm::network::ULActionAR3 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR3:



Collaboration diagram for gdcm::network::ULActionAR3:



### Public Member Functions

- `EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)` override

### Public Member Functions inherited from `gdcm::network::ULAction`

- `ULAction ()`=default
- `ULAction (const ULAction &inAction)`=delete
- virtual `~ULAction ()`=default
- void `operator= (const ULAction &)`=delete

## 10.345.1 Member Function Documentation

### 10.345.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR3::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

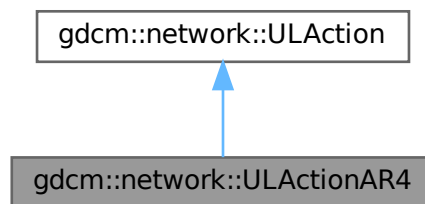
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

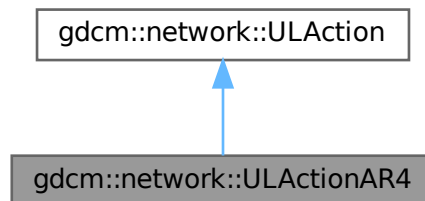
## 10.346 gdcm::network::ULActionAR4 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR4:



Collaboration diagram for gdcm::network::ULActionAR4:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

## Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

## 10.346.1 Member Function Documentation

### 10.346.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR4::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

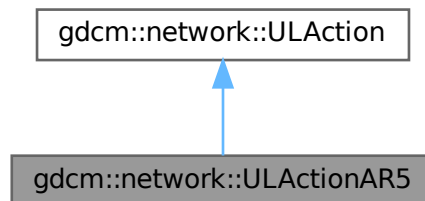
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

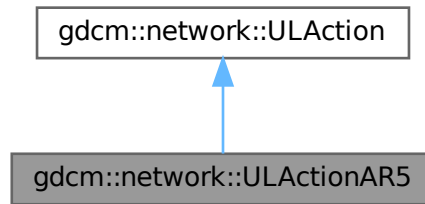
## 10.347 gdcm::network::ULActionAR5 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for [gdcm::network::ULActionAR5](#):



Collaboration diagram for gdcm::network::ULActionAR5:



### Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

### Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

## 10.347.1 Member Function Documentation

### 10.347.1.1 PerformAction()

```

EStateID gdcm::network::ULActionAR5::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
  
```

Implements [gdcm::network::ULAction](#).

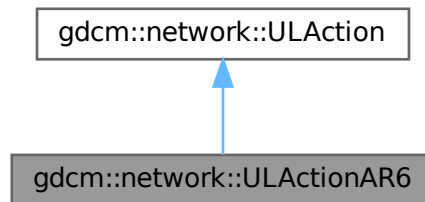
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

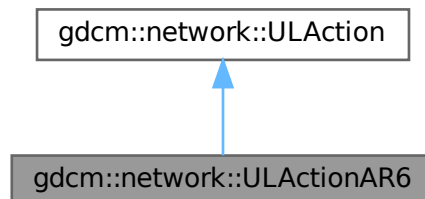
## 10.348 gdcm::network::ULActionAR6 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR6:



Collaboration diagram for gdcm::network::ULActionAR6:



### Public Member Functions

- `EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)` override

### Public Member Functions inherited from `gdcm::network::ULAction`

- `ULAction ()`=default
- `ULAction (const ULAction &inAction)`=delete
- virtual `~ULAction ()`=default
- void `operator= (const ULAction &)`=delete



## 10.348.1 Member Function Documentation

### 10.348.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR6::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

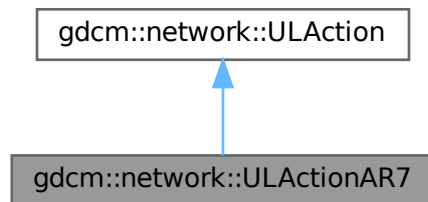
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

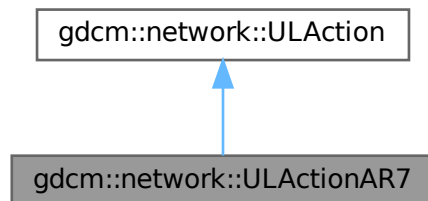
## 10.349 gdcm::network::ULActionAR7 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR7:



Collaboration diagram for gdcm::network::ULActionAR7:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

## Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

## 10.349.1 Member Function Documentation

### 10.349.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR7::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

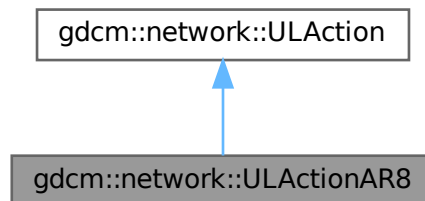
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

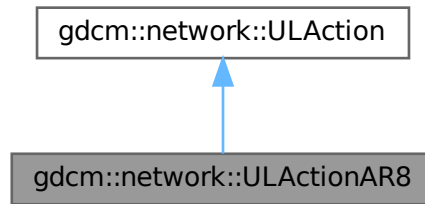
## 10.350 gdcm::network::ULActionAR8 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for [gdcm::network::ULActionAR8](#):



Collaboration diagram for gdcm::network::ULActionAR8:



### Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

### Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

## 10.350.1 Member Function Documentation

### 10.350.1.1 PerformAction()

```

EStateID gdcm::network::ULActionAR8::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
  
```

Implements [gdcm::network::ULAction](#).

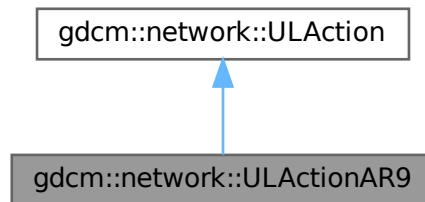
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

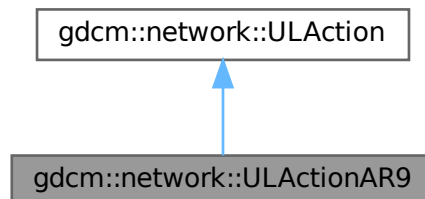
## 10.351 gdcm::network::ULActionAR9 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR9:



Collaboration diagram for gdcm::network::ULActionAR9:



### Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

### Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

## 10.351.1 Member Function Documentation

### 10.351.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR9::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

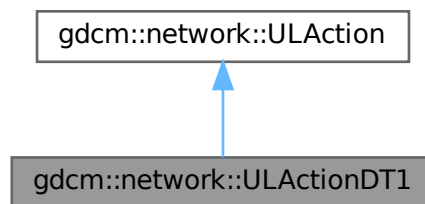
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

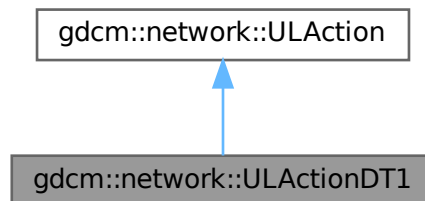
## 10.352 gdcm::network::ULActionDT1 Class Reference

```
#include <gdcmULActionDT.h>
```

Inheritance diagram for gdcm::network::ULActionDT1:



Collaboration diagram for gdcm::network::ULActionDT1:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

## Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

## 10.352.1 Member Function Documentation

### 10.352.1.1 PerformAction()

```
EStateID gdcm::network::ULActionDT1::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

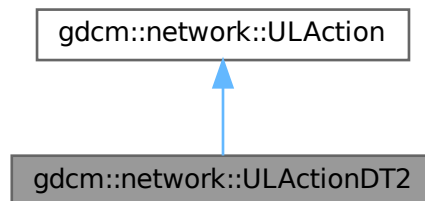
The documentation for this class was generated from the following file:

- [gdcmULActionDT.h](#)

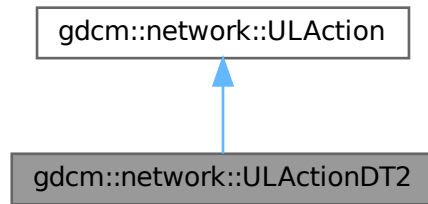
## 10.353 gdcm::network::ULActionDT2 Class Reference

```
#include <gdcmULActionDT.h>
```

Inheritance diagram for [gdcm::network::ULActionDT2](#):



Collaboration diagram for gdcm::network::ULActionDT2:



### Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

### Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

## 10.353.1 Member Function Documentation

### 10.353.1.1 PerformAction()

```

EStateID gdcm::network::ULActionDT2::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
  
```

Implements [gdcm::network::ULAction](#).

The documentation for this class was generated from the following file:

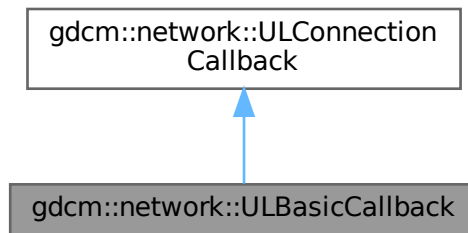
- [gdcmULActionDT.h](#)

## 10.354 gdcm::network::ULBasicCallback Class Reference

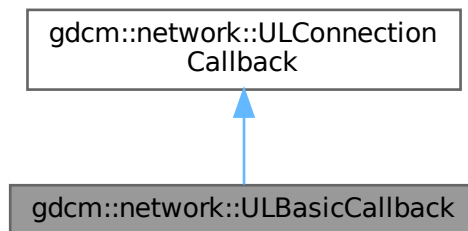
[ULBasicCallback](#).

```
#include <gdcmULBasicCallback.h>
```

Inheritance diagram for gdcm::network::ULBasicCallback:



Collaboration diagram for gdcm::network::ULBasicCallback:



### Public Member Functions

- [ULBasicCallback](#) ()=default
- [~ULBasicCallback](#) () override=default
- `std::vector< DataSet > const & GetDataSets () const`
- `std::vector< DataSet > const & GetResponses () const`
- `void HandleDataSet (const DataSet &inDataSet) override`
- `void HandleResponse (const DataSet &inDataSet) override`



**Public Member Functions inherited from [gdcm::network::ULConnectionCallback](#)**

- [ULConnectionCallback](#) ()
- virtual [~ULConnectionCallback](#) ()=default
- bool [DataSetHandles](#) () const
- void [ResetHandledDataSet](#) ()
- void [SetImplicitFlag](#) (const bool imp)

**Additional Inherited Members****Protected Member Functions inherited from [gdcm::network::ULConnectionCallback](#)**

- void [DataSetHandled](#) ()

**Protected Attributes inherited from [gdcm::network::ULConnectionCallback](#)**

- bool [mImplicit](#)

**10.354.1 Detailed Description**

[ULBasicCallback](#).

This is the most basic of callbacks for how the [ULConnectionManager](#) handles incoming datasets. DataSets are just concatenated to the mDataSets vector, and the result can be pulled out of the vector by later code. Alternatives to this method include progress updates, saving to disk, etc. This class is NOT THREAD SAFE. Access the dataset vector after the entire set of datasets has been returned by the [ULConnectionManager](#).

**10.354.2 Constructor & Destructor Documentation****10.354.2.1 [ULBasicCallback](#)()**

```
gdcm::network::ULBasicCallback::ULBasicCallback ( ) [default]
```

**10.354.2.2 [~ULBasicCallback](#)()**

```
gdcm::network::ULBasicCallback::~~ULBasicCallback ( ) [override], [default]
```

**10.354.3 Member Function Documentation****10.354.3.1 [GetDataSets](#)()**

```
std::vector< DataSet > const & gdcm::network::ULBasicCallback::GetDataSets ( ) const
```

### 10.354.3.2 GetResponses()

```
std::vector< DataSet > const & gdcn::network::ULBasicCallback::GetResponses ( ) const
```

### 10.354.3.3 HandleDataSet()

```
void gdcn::network::ULBasicCallback::HandleDataSet (
    const DataSet & inDataSet ) [override], [virtual]
```

Implements [gdcn::network::ULConnectionCallback](#).

### 10.354.3.4 HandleResponse()

```
void gdcn::network::ULBasicCallback::HandleResponse (
    const DataSet & inDataSet ) [override], [virtual]
```

Implements [gdcn::network::ULConnectionCallback](#).

The documentation for this class was generated from the following file:

- [gdcnULBasicCallback.h](#)

## 10.355 gdcn::network::ULConnection Class Reference

[ULConnection](#).

```
#include <gdcnULConnection.h>
```

### Public Member Functions

- [ULConnection](#) (const [ULConnection](#) &)=delete
- [ULConnection](#) (const [ULConnectionInfo](#) &inUserInformation)
- virtual [~ULConnection](#) ()
- void [AddAcceptedPresentationContext](#) (const [PresentationContextAC](#) &inPC)
- [PresentationContextRQ](#) [FindContext](#) (const [DataElement](#) &de) const
- std::vector< [PresentationContextAC](#) > & [GetAcceptedPresentationContexts](#) ()
- std::vector< [PresentationContextAC](#) > const & [GetAcceptedPresentationContexts](#) () const
- const [ULConnectionInfo](#) & [GetConnectionInfo](#) () const
- uint32\_t [GetMaxPDUSize](#) () const
- const [PresentationContextAC](#) \* [GetPresentationContextACByID](#) (uint8\_t id) const
- uint8\_t [GetPresentationContextIDFromPresentationContext](#) ([PresentationContextRQ](#) const &pc) const  
*return 0 upon error*
- const [PresentationContextRQ](#) \* [GetPresentationContextRQByID](#) (uint8\_t id) const
- std::vector< [PresentationContextRQ](#) > const & [GetPresentationContexts](#) () const
- std::iostream \* [GetProtocol](#) ()

- [EStateID GetState](#) () const
- [ARTIMTimer & GetTimer](#) ()
- bool [InitializeConnection](#) ()  
*used to establish scu connections*
- bool [InitializeIncomingConnection](#) ()  
*used to establish scp connections*
- void [operator=](#) (const [ULConnection](#) &)=delete
- void [SetMaxPDUSize](#) (uint32\_t inSize)
- void [SetPresentationContexts](#) (const std::vector< [PresentationContext](#) > &inContexts)
- void [SetPresentationContexts](#) (const std::vector< [PresentationContextRQ](#) > &inContexts)
- void [SetState](#) (const [EStateID](#) &inState)
- void [StopProtocol](#) ()

## Friends

- class [ULActionAE6](#)
- class [ULConnectionManager](#)

## 10.355.1 Detailed Description

### [ULConnection](#).

This is the class that contains the socket to another machine, and passes data through itself, as well as maintaining a sense of state.

The [ULConnectionManager](#) tells the [ULConnection](#) what data can actually be sent.

This class is done this way so that it can be eventually be replaced with a [ULSecureConnection](#), if such a protocol is warranted, so that all data that passes through can be managed through a secure connection. For now, this class provides a simple pass-through mechanism to the socket itself.

So, for instance, a [gdcm](#) object will be passes to this object, and it will then get passed along the connection, if that connection is in the proper state to do so.

For right now, this class is not directly intended to be inherited from, but the potential for future [ULSecureConnection](#) warrants the addition, rather than having everything be managed from within the [ULConnectionManager](#) (or this class) without a wrapper.

## 10.355.2 Constructor & Destructor Documentation

### 10.355.2.1 [ULConnection](#)() [1/2]

```
gdcm::network::ULConnection::ULConnection (
    const ULConnectionInfo & inUserInformation )
```

### 10.355.2.2 ~ULConnection()

```
virtual gdcmm::network::ULConnection::~~ULConnection ( ) [virtual]
```

### 10.355.2.3 ULConnection() [2/2]

```
gdcmm::network::ULConnection::ULConnection (
    const ULConnection & ) [delete]
```

## 10.355.3 Member Function Documentation

### 10.355.3.1 AddAcceptedPresentationContext()

```
void gdcmm::network::ULConnection::AddAcceptedPresentationContext (
    const PresentationContextAC & inPC )
```

### 10.355.3.2 FindContext()

```
PresentationContextRQ gdcmm::network::ULConnection::FindContext (
    const DataElement & de ) const
```

### 10.355.3.3 GetAcceptedPresentationContexts() [1/2]

```
std::vector< PresentationContextAC > & gdcmm::network::ULConnection::GetAcceptedPresentation←
Contexts ( )
```

### 10.355.3.4 GetAcceptedPresentationContexts() [2/2]

```
std::vector< PresentationContextAC > const & gdcmm::network::ULConnection::GetAcceptedPresentation←
Contexts ( ) const
```

### 10.355.3.5 GetConnectionInfo()

```
const ULConnectionInfo & gdcmm::network::ULConnection::GetConnectionInfo ( ) const
```

### 10.355.3.6 GetMaxPDUSize()

```
uint32_t gdcmm::network::ULConnection::GetMaxPDUSize ( ) const
```

### 10.355.3.7 GetPresentationContextACByID()

```
const PresentationContextAC * gdcm::network::ULConnection::GetPresentationContextACByID (
    uint8_t id ) const
```

### 10.355.3.8 GetPresentationContextIDFromPresentationContext()

```
uint8_t gdcm::network::ULConnection::GetPresentationContextIDFromPresentationContext (
    PresentationContextRQ const & pc ) const
```

return 0 upon error

### 10.355.3.9 GetPresentationContextRQByID()

```
const PresentationContextRQ * gdcm::network::ULConnection::GetPresentationContextRQByID (
    uint8_t id ) const
```

### 10.355.3.10 GetPresentationContexts()

```
std::vector< PresentationContextRQ > const & gdcm::network::ULConnection::GetPresentationContexts
( ) const
```

### 10.355.3.11 GetProtocol()

```
std::iostream * gdcm::network::ULConnection::GetProtocol ( )
```

### 10.355.3.12 GetState()

```
EStateID gdcm::network::ULConnection::GetState ( ) const
```

### 10.355.3.13 GetTimer()

```
ARTIMTimer & gdcm::network::ULConnection::GetTimer ( )
```

### 10.355.3.14 InitializeConnection()

```
bool gdcm::network::ULConnection::InitializeConnection ( )
```

used to establish scu connections

**10.355.3.15 InitializeIncomingConnection()**

```
bool gdcm::network::ULConnection::InitializeIncomingConnection ( )
```

used to establish scp connections

**10.355.3.16 operator=()**

```
void gdcm::network::ULConnection::operator= (
    const ULConnection & ) [delete]
```

**10.355.3.17 SetMaxPDUSize()**

```
void gdcm::network::ULConnection::SetMaxPDUSize (
    uint32_t inSize )
```

**10.355.3.18 SetPresentationContexts() [1/2]**

```
void gdcm::network::ULConnection::SetPresentationContexts (
    const std::vector< PresentationContext > & inContexts )
```

**10.355.3.19 SetPresentationContexts() [2/2]**

```
void gdcm::network::ULConnection::SetPresentationContexts (
    const std::vector< PresentationContextRQ > & inContexts )
```

**10.355.3.20 SetState()**

```
void gdcm::network::ULConnection::SetState (
    const EStateID & inState )
```

**10.355.3.21 StopProtocol()**

```
void gdcm::network::ULConnection::StopProtocol ( )
```

**10.355.4 Friends And Related Symbol Documentation****10.355.4.1 ULActionAE6**

```
friend class ULActionAE6 [friend]
```

### 10.355.4.2 ULConnectionManager

```
friend class ULConnectionManager [friend]
```

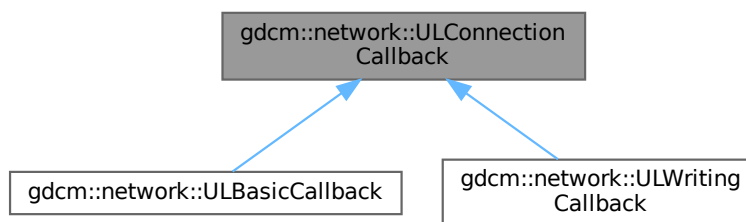
The documentation for this class was generated from the following file:

- [gdcmULConnection.h](#)

## 10.356 gdcm::network::ULConnectionCallback Class Reference

```
#include <gdcmULConnectionCallback.h>
```

Inheritance diagram for gdcm::network::ULConnectionCallback:



### Public Member Functions

- [ULConnectionCallback](#) ()
- virtual [~ULConnectionCallback](#) ()=default
- bool [DataSetHandles](#) () const
- virtual void [HandleDataSet](#) (const [DataSet](#) &inDataSet)=0
- virtual void [HandleResponse](#) (const [DataSet](#) &inDataSet)=0
- void [ResetHandledDataSet](#) ()
- void [SetImplicitFlag](#) (const bool imp)

### Protected Member Functions

- void [DataSetHandled](#) ()

### Protected Attributes

- bool [mImplicit](#)

### 10.356.1 Detailed Description

When a dataset comes back from a query/move/etc, the result can either be stored entirely in memory, or could be stored on disk. This class provides a mechanism to indicate what the [ULConnectionManager](#) should do with datasets that are produced through query results. The [ULConnectionManager](#) will call the `HandleDataSet` function during the course of receiving datasets. Particular implementations should fill in what that function does, including updating progress, etc. NOTE: since `cmove` requires that multiple event loops be employed, the callback function MUST set `mHandledData` to true. otherwise, the `cmove` event loop handler will not know data was received, and proceed to end the loop prematurely.

### 10.356.2 Constructor & Destructor Documentation

#### 10.356.2.1 [ULConnectionCallback\(\)](#)

```
gdcmm::network::ULConnectionCallback::ULConnectionCallback ( ) [inline]
```

#### 10.356.2.2 [~ULConnectionCallback\(\)](#)

```
virtual gdcmm::network::ULConnectionCallback::~~ULConnectionCallback ( ) [virtual], [default]
```

### 10.356.3 Member Function Documentation

#### 10.356.3.1 [DataSetHandled\(\)](#)

```
void gdcmm::network::ULConnectionCallback::DataSetHandled ( ) [inline], [protected]
```

#### 10.356.3.2 [DataSetHandles\(\)](#)

```
bool gdcmm::network::ULConnectionCallback::DataSetHandles ( ) const [inline]
```

#### 10.356.3.3 [HandleDataSet\(\)](#)

```
virtual void gdcmm::network::ULConnectionCallback::HandleDataSet (
    const DataSet & inDataSet ) [pure virtual]
```

Implemented in [gdcmm::network::ULBasicCallback](#), and [gdcmm::network::ULWritingCallback](#).

#### 10.356.3.4 [HandleResponse\(\)](#)

```
virtual void gdcmm::network::ULConnectionCallback::HandleResponse (
    const DataSet & inDataSet ) [pure virtual]
```

Implemented in [gdcmm::network::ULBasicCallback](#), and [gdcmm::network::ULWritingCallback](#).



### 10.356.3.5 ResetHandledDataSet()

```
void gdcm::network::ULConnectionCallback::ResetHandledDataSet ( ) [inline]
```

### 10.356.3.6 SetImplicitFlag()

```
void gdcm::network::ULConnectionCallback::SetImplicitFlag (
    const bool imp ) [inline]
```

## 10.356.4 Member Data Documentation

### 10.356.4.1 mImplicit

```
bool gdcm::network::ULConnectionCallback::mImplicit [protected]
```

The documentation for this class was generated from the following file:

- [gdcmULConnectionCallback.h](#)

## 10.357 gdcm::network::ULConnectionInfo Class Reference

[ULConnectionInfo](#).

```
#include <gdcmULConnectionInfo.h>
```

### Public Member Functions

- [ULConnectionInfo](#) ()
- const char \* [GetCalledAETitle](#) () const
- std::string [GetCalledComputerName](#) () const
- unsigned long [GetCalledIPAddress](#) () const
- int [GetCalledIPPort](#) () const
- const char \* [GetCallingAETitle](#) () const
- unsigned long [GetMaxPDULength](#) () const
- bool [Initialize](#) ([UserInformation](#) const &inUserInformation, const char \*inCalledAETitle, const char \*inCallingAETitle, unsigned long inCalledIPAddress, int inCalledIPPort, std::string inCalledComputerName)
- void [SetMaxPDULength](#) (unsigned long inMaxPDULength)

### 10.357.1 Detailed Description

[ULConnectionInfo](#).

this class contains all the information about a particular connection as established by the user. That is, it's: User Information Calling AE Title Called AE Title IP address/computer name IP Port A connection must be established with this information, that's subsequently placed into various primitives for actual communication.

## 10.357.2 Constructor & Destructor Documentation

### 10.357.2.1 ULConnectionInfo()

```
gdcm::network::ULConnectionInfo::ULConnectionInfo ( )
```

## 10.357.3 Member Function Documentation

### 10.357.3.1 GetCalledAETitle()

```
const char * gdcm::network::ULConnectionInfo::GetCalledAETitle ( ) const
```

### 10.357.3.2 GetCalledComputerName()

```
std::string gdcm::network::ULConnectionInfo::GetCalledComputerName ( ) const
```

### 10.357.3.3 GetCalledIPAddress()

```
unsigned long gdcm::network::ULConnectionInfo::GetCalledIPAddress ( ) const
```

### 10.357.3.4 GetCalledIPPort()

```
int gdcm::network::ULConnectionInfo::GetCalledIPPort ( ) const
```

### 10.357.3.5 GetCallingAETitle()

```
const char * gdcm::network::ULConnectionInfo::GetCallingAETitle ( ) const
```

### 10.357.3.6 GetMaxPDULength()

```
unsigned long gdcm::network::ULConnectionInfo::GetMaxPDULength ( ) const
```

### 10.357.3.7 Initialize()

```
bool gdcm::network::ULConnectionInfo::Initialize (
    UserInformation const & inUserInformation,
    const char * inCalledAETitle,
    const char * inCallingAETitle,
    unsigned long inCalledIPAddress,
    int inCalledIPPort,
    std::string inCalledComputerName )
```

### 10.357.3.8 SetMaxPDULength()

```
void gdcm::network::ULConnectionInfo::SetMaxPDULength (
    unsigned long inMaxPDULength )
```

The documentation for this class was generated from the following file:

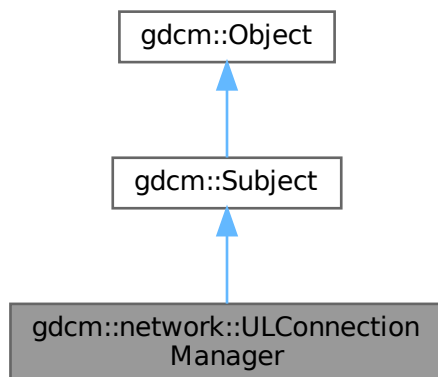
- [gdcmULConnectionInfo.h](#)

## 10.358 gdcm::network::ULConnectionManager Class Reference

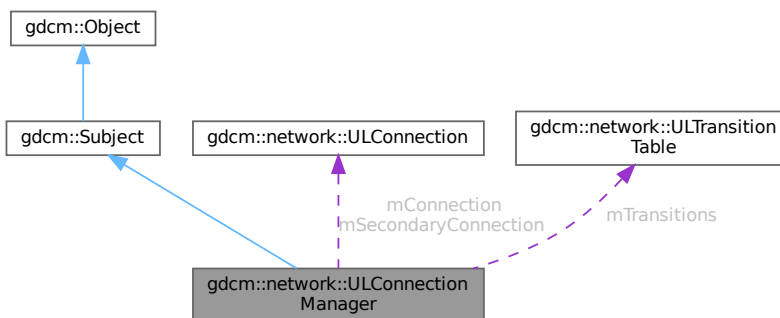
[ULConnectionManager](#).

```
#include <gdcmULConnectionManager.h>
```

Inheritance diagram for gdcm::network::ULConnectionManager:



Collaboration diagram for gdcm::network::ULConnectionManager:



## Public Member Functions

- [ULConnectionManager](#) ()
- [~ULConnectionManager](#) () override
- bool [BreakConnection](#) (const double &inTimeout)
- void [BreakConnectionNow](#) ()
- bool [EstablishConnection](#) (const std::string &inAETitle, const std::string &inConnectAETitle, const std::string &inComputerName, long inIPAddress, uint16\_t inConnectPort, double inTimeout, std::vector< [PresentationContext](#) > const &pcVector)
- bool [EstablishConnectionMove](#) (const std::string &inAETitle, const std::string &inConnectAETitle, const std::string &inComputerName, long inIPAddress, uint16\_t inConnectPort, double inTimeout, uint16\_t inReturnPort, std::vector< [PresentationContext](#) > const &pcVector)
- std::vector< [PresentationDataValue](#) > [SendEcho](#) ()
- std::vector< [DataSet](#) > [SendFind](#) (const [BaseRootQuery](#) \*inRootQuery)
- void [SendFind](#) (const [BaseRootQuery](#) \*inRootQuery, [ULConnectionCallback](#) \*inCallback)
- std::vector< [DataSet](#) > [SendMove](#) (const [BaseRootQuery](#) \*inRootQuery)
- bool [SendMove](#) (const [BaseRootQuery](#) \*inRootQuery, [ULConnectionCallback](#) \*inCallback)  
*return false upon error*
- std::vector< [DataSet](#) > [SendNAction](#) (const [BaseQuery](#) \*inQuery)
- void [SendNAction](#) (const [BaseQuery](#) \*inQuery, [ULConnectionCallback](#) \*inCallback)
- std::vector< [DataSet](#) > [SendNCreate](#) (const [BaseQuery](#) \*inQuery)
- void [SendNCreate](#) (const [BaseQuery](#) \*inQuery, [ULConnectionCallback](#) \*inCallback)
- std::vector< [DataSet](#) > [SendNDelete](#) (const [BaseQuery](#) \*inQuery)
- void [SendNDelete](#) (const [BaseQuery](#) \*inQuery, [ULConnectionCallback](#) \*inCallback)
- std::vector< [DataSet](#) > [SendNEventReport](#) (const [BaseQuery](#) \*inQuery)
- void [SendNEventReport](#) (const [BaseQuery](#) \*inQuery, [ULConnectionCallback](#) \*inCallback)
- std::vector< [DataSet](#) > [SendNGet](#) (const [BaseQuery](#) \*inQuery)
- void [SendNGet](#) (const [BaseQuery](#) \*inQuery, [ULConnectionCallback](#) \*inCallback)
- std::vector< [DataSet](#) > [SendNSet](#) (const [BaseQuery](#) \*inQuery)
- void [SendNSet](#) (const [BaseQuery](#) \*inQuery, [ULConnectionCallback](#) \*inCallback)
- std::vector< [DataSet](#) > [SendStore](#) (const [File](#) &file, std::istream \*pStream=nullptr, std::streampos dataSetOffset=0)
- void [SendStore](#) (const [File](#) &file, [ULConnectionCallback](#) \*inCallback, std::istream \*pStream=nullptr, std::streampos dataSetOffset=0)  
*callback based API*

## Public Member Functions inherited from [gdcm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) \*) const
- [Command](#) \* [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

**Public Member Functions inherited from [gdcm::Object](#)**

- [Object](#) ()
- [Object](#) (const [Object](#) &)
- *Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

**Protected Member Functions**

- [ULConnectionManager](#) (const [ULConnectionManager](#) &inCM)
- [EStateID RunEventLoop](#) ([ULEvent](#) &inEvent, [ULConnection](#) \*inWhichConnection, [ULConnectionCallback](#) \*inCallback, const bool &startWaiting)
- [EStateID RunMoveEventLoop](#) ([ULEvent](#) &inEvent, [ULConnectionCallback](#) \*inCallback)

**Protected Member Functions inherited from [gdcm::Object](#)**

- void [Register](#) ()
- void [UnRegister](#) ()

**Protected Attributes**

- [ULConnection](#) \* mConnection
- [ULConnection](#) \* mSecondaryConnection
- [ULTransitionTable](#) mTransitions

**10.358.1 Detailed Description**

[ULConnectionManager](#).

The [ULConnectionManager](#) performs actions on the [ULConnection](#) given inputs from the user and from the state of what's going on around the connection (ie, timeouts of the ARTIM timer, responses from the peer across the connection, etc).

Its inputs are ULEvents, and it performs ULActions.

**10.358.2 Constructor & Destructor Documentation****10.358.2.1 [ULConnectionManager](#)() [1/2]**

```
gdcm::network::ULConnectionManager::ULConnectionManager (
    const ULConnectionManager & inCM ) [protected]
```

**10.358.2.2 ULConnectionManager()** [2/2]

```
gdcmm::network::ULConnectionManager::ULConnectionManager ( )
```

**10.358.2.3 ~ULConnectionManager()**

```
gdcmm::network::ULConnectionManager::~~ULConnectionManager ( ) [override]
```

**10.358.3 Member Function Documentation****10.358.3.1 BreakConnection()**

```
bool gdcmm::network::ULConnectionManager::BreakConnection (
    const double & inTimeout )
```

**10.358.3.2 BreakConnectionNow()**

```
void gdcmm::network::ULConnectionManager::BreakConnectionNow ( )
```

**10.358.3.3 EstablishConnection()**

```
bool gdcmm::network::ULConnectionManager::EstablishConnection (
    const std::string & inAETitle,
    const std::string & inConnectAETitle,
    const std::string & inComputerName,
    long inIPAddress,
    uint16_t inConnectPort,
    double inTimeout,
    std::vector< PresentationContext > const & pcVector )
```

returns true if a connection of the given AETitle (ie, 'this' program) is able to connect to the given AETitle and Port in a certain amount of time providing the connection type will establish the proper exchange syntax with a server; if a different functionality is required, a different connection should be established. returns false if the connection type is 'move'— have to give a return port for move to work as specified.

**10.358.3.4 EstablishConnectionMove()**

```
bool gdcmm::network::ULConnectionManager::EstablishConnectionMove (
    const std::string & inAETitle,
    const std::string & inConnectAETitle,
    const std::string & inComputerName,
    long inIPAddress,
    uint16_t inConnectPort,
    double inTimeout,
    uint16_t inReturnPort,
    std::vector< PresentationContext > const & pcVector )
```

returns true for above reasons, but contains the special 'move' port

### 10.358.3.5 RunEventLoop()

```
EStateID gdcmm::network::ULConnectionManager::RunEventLoop (
    ULEvent & inEvent,
    ULConnection * inWhichConnection,
    ULConnectionCallback * inCallback,
    const bool & startWaiting ) [protected]
```

### 10.358.3.6 RunMoveEventLoop()

```
EStateID gdcmm::network::ULConnectionManager::RunMoveEventLoop (
    ULEvent & inEvent,
    ULConnectionCallback * inCallback ) [protected]
```

### 10.358.3.7 SendEcho()

```
std::vector< PresentationDataValue > gdcmm::network::ULConnectionManager::SendEcho ( )
```

### 10.358.3.8 SendFind() [1/2]

```
std::vector< DataSet > gdcmm::network::ULConnectionManager::SendFind (
    const BaseRootQuery * inRootQuery )
```

### 10.358.3.9 SendFind() [2/2]

```
void gdcmm::network::ULConnectionManager::SendFind (
    const BaseRootQuery * inRootQuery,
    ULConnectionCallback * inCallback )
```

### 10.358.3.10 SendMove() [1/2]

```
std::vector< DataSet > gdcmm::network::ULConnectionManager::SendMove (
    const BaseRootQuery * inRootQuery )
```

### 10.358.3.11 SendMove() [2/2]

```
bool gdcmm::network::ULConnectionManager::SendMove (
    const BaseRootQuery * inRootQuery,
    ULConnectionCallback * inCallback )
```

return false upon error

**10.358.3.12 SendNAction()** [1/2]

```
std::vector< DataSet > gdc::network::ULConnectionManager::SendNAction (
    const BaseQuery * inQuery )
```

**10.358.3.13 SendNAction()** [2/2]

```
void gdc::network::ULConnectionManager::SendNAction (
    const BaseQuery * inQuery,
    ULConnectionCallback * inCallback )
```

**10.358.3.14 SendNCreate()** [1/2]

```
std::vector< DataSet > gdc::network::ULConnectionManager::SendNCreate (
    const BaseQuery * inQuery )
```

**10.358.3.15 SendNCreate()** [2/2]

```
void gdc::network::ULConnectionManager::SendNCreate (
    const BaseQuery * inQuery,
    ULConnectionCallback * inCallback )
```

**10.358.3.16 SendNDelete()** [1/2]

```
std::vector< DataSet > gdc::network::ULConnectionManager::SendNDelete (
    const BaseQuery * inQuery )
```

**10.358.3.17 SendNDelete()** [2/2]

```
void gdc::network::ULConnectionManager::SendNDelete (
    const BaseQuery * inQuery,
    ULConnectionCallback * inCallback )
```

**10.358.3.18 SendNEventReport()** [1/2]

```
std::vector< DataSet > gdc::network::ULConnectionManager::SendNEventReport (
    const BaseQuery * inQuery )
```

**10.358.3.19 SendNEventReport()** [2/2]

```
void gdc::network::ULConnectionManager::SendNEventReport (
    const BaseQuery * inQuery,
    ULConnectionCallback * inCallback )
```



### 10.358.3.20 SendNGet() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendNGet (
    const BaseQuery * inQuery )
```

### 10.358.3.21 SendNGet() [2/2]

```
void gdcm::network::ULConnectionManager::SendNGet (
    const BaseQuery * inQuery,
    ULConnectionCallback * inCallback )
```

### 10.358.3.22 SendNSet() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendNSet (
    const BaseQuery * inQuery )
```

### 10.358.3.23 SendNSet() [2/2]

```
void gdcm::network::ULConnectionManager::SendNSet (
    const BaseQuery * inQuery,
    ULConnectionCallback * inCallback )
```

### 10.358.3.24 SendStore() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendStore (
    const File & file,
    std::istream * pStream = nullptr,
    std::streampos dataSetOffset = 0 )
```

### 10.358.3.25 SendStore() [2/2]

```
void gdcm::network::ULConnectionManager::SendStore (
    const File & file,
    ULConnectionCallback * inCallback,
    std::istream * pStream = nullptr,
    std::streampos dataSetOffset = 0 )
```

callback based API

## 10.358.4 Member Data Documentation

### 10.358.4.1 mConnection

`ULConnection*` gdcm::network::ULConnectionManager::mConnection [protected]

### 10.358.4.2 mSecondaryConnection

```
ULConnection* gdcmm::network::ULConnectionManager::mSecondaryConnection [protected]
```

### 10.358.4.3 mTransitions

```
ULTransitionTable gdcmm::network::ULConnectionManager::mTransitions [protected]
```

The documentation for this class was generated from the following file:

- [gdcmmULConnectionManager.h](#)

## 10.359 gdcmm::network::ULEvent Class Reference

[ULEvent](#).

```
#include <gdcmmULEvent.h>
```

### Public Member Functions

- [ULEvent](#) (const [EEventID](#) &inEventID, [BasePDU](#) \*inBasePDU, std::istream \*iStream=nullptr, std::streampos posDataSet=0)
- [ULEvent](#) (const [EEventID](#) &inEventID, std::vector< [BasePDU](#) \* > inBasePDU, std::istream \*iStream=nullptr, std::streampos posDataSet=0)
- [~ULEvent](#) ()
- std::streampos [GetDataSetPos](#) () const
- [EEventID](#) [GetEvent](#) () const
- std::istream \* [GetStream](#) () const
- std::vector< [BasePDU](#) \* > const & [GetPDUs](#) () const
- void [SetEvent](#) (const [EEventID](#) &inEvent)
- void [SetPDU](#) (std::vector< [BasePDU](#) \* > const &inPDU)

### 10.359.1 Detailed Description

[ULEvent](#).

base class for network events.

An event consists of the event ID and the data associated with that event.

Note that once a PDU is created, it is now the responsibility of the associated event to destroy it!

## 10.359.2 Constructor & Destructor Documentation

### 10.359.2.1 ULEvent() [1/2]

```
gdcm::network::ULEvent::ULEvent (
    const EEventID & inEventID,
    std::vector< BasePDU * > inBasePDU,
    std::istream * iStream = nullptr,
    std::streampos posDataSet = 0 ) [inline]
```

### 10.359.2.2 ULEvent() [2/2]

```
gdcm::network::ULEvent::ULEvent (
    const EEventID & inEventID,
    BasePDU * inBasePDU,
    std::istream * iStream = nullptr,
    std::streampos posDataSet = 0 ) [inline]
```

### 10.359.2.3 ~ULEvent()

```
gdcm::network::ULEvent::~~ULEvent ( ) [inline]
```

## 10.359.3 Member Function Documentation

### 10.359.3.1 GetDataSetPos()

```
std::streampos gdcm::network::ULEvent::GetDataSetPos ( ) const [inline]
```

### 10.359.3.2 GetEvent()

```
EEventID gdcm::network::ULEvent::GetEvent ( ) const [inline]
```

### 10.359.3.3 GetIStream()

```
std::istream * gdcm::network::ULEvent::GetIStream ( ) const [inline]
```

### 10.359.3.4 GetPDUs()

```
std::vector< BasePDU * > const & gdcm::network::ULEvent::GetPDUs ( ) const [inline]
```

### 10.359.3.5 SetEvent()

```
void gdcmm::network::ULEvent::SetEvent (
    const EEventID & inEvent ) [inline]
```

### 10.359.3.6 SetPDU()

```
void gdcmm::network::ULEvent::SetPDU (
    std::vector< BasePDU * > const & inPDU ) [inline]
```

The documentation for this class was generated from the following file:

- [gdcmmULEvent.h](#)

## 10.360 gdcmm::network::ULTransitionTable Class Reference

[ULTransitionTable](#) The transition table of all the ULEvents, new ULActions, and ULStates.

```
#include <gdcmmULTransitionTable.h>
```

### Public Member Functions

- [ULTransitionTable](#) ()
- void [HandleEvent](#) (Subject \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) const
- void [PrintTable](#) () const

### 10.360.1 Detailed Description

[ULTransitionTable](#) The transition table of all the ULEvents, new ULActions, and ULStates.

Based roughly on the solutions in player2.cpp in the boost examples and this so question: <http://stackoverflow.com/questions/1647631/c-state-machine-design>

The transition table is constructed of TableRows. Each row is based on an event, and an event handler in the TransitionTable object takes a given event, and then finds the given row.

Then, given the current state of the connection, determines the appropriate action to take and then the state to transition to next.

### 10.360.2 Constructor & Destructor Documentation

#### 10.360.2.1 ULTransitionTable()

```
gdcmm::network::ULTransitionTable::ULTransitionTable ( )
```

### 10.360.3 Member Function Documentation

#### 10.360.3.1 HandleEvent()

```
void gdcm::network::ULTransitionTable::HandleEvent (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) const
```

#### 10.360.3.2 PrintTable()

```
void gdcm::network::ULTransitionTable::PrintTable ( ) const
```

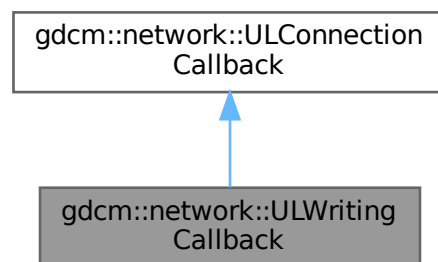
The documentation for this class was generated from the following file:

- [gdcmULTransitionTable.h](#)

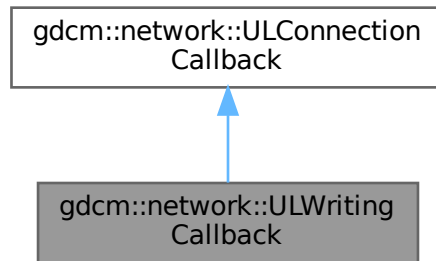
## 10.361 gdcm::network::ULWritingCallback Class Reference

```
#include <gdcmULWritingCallback.h>
```

Inheritance diagram for gdcm::network::ULWritingCallback:



Collaboration diagram for `gdcm::network::ULWritingCallback`:



### Public Member Functions

- [ULWritingCallback](#) ()=default
- [~ULWritingCallback](#) () override=default
- void [HandleDataSet](#) (const [DataSet](#) &inDataSet) override
- void [HandleResponse](#) (const [DataSet](#) &inDataSet) override
- void [SetDirectory](#) (const std::string &inDirectoryName)  
*provide the directory into which all files are written.*

### Public Member Functions inherited from [gdcm::network::ULConnectionCallback](#)

- [ULConnectionCallback](#) ()
- virtual [~ULConnectionCallback](#) ()=default
- bool [DataSetHandles](#) () const
- void [ResetHandledDataSet](#) ()
- void [SetImplicitFlag](#) (const bool imp)

### Additional Inherited Members

### Protected Member Functions inherited from [gdcm::network::ULConnectionCallback](#)

- void [DataSetHandled](#) ()

### Protected Attributes inherited from [gdcm::network::ULConnectionCallback](#)

- bool [mImplicit](#)

## 10.361.1 Constructor & Destructor Documentation

### 10.361.1.1 ULWritingCallback()

```
gdcm::network::ULWritingCallback::ULWritingCallback ( ) [default]
```

### 10.361.1.2 ~ULWritingCallback()

```
gdcm::network::ULWritingCallback::~~ULWritingCallback ( ) [override], [default]
```

## 10.361.2 Member Function Documentation

### 10.361.2.1 HandleDataSet()

```
void gdcm::network::ULWritingCallback::HandleDataSet (
    const DataSet & inDataSet ) [override], [virtual]
```

Implements [gdcm::network::ULConnectionCallback](#).

### 10.361.2.2 HandleResponse()

```
void gdcm::network::ULWritingCallback::HandleResponse (
    const DataSet & inDataSet ) [override], [virtual]
```

Implements [gdcm::network::ULConnectionCallback](#).

### 10.361.2.3 SetDirectory()

```
void gdcm::network::ULWritingCallback::SetDirectory (
    const std::string & inDirectoryName ) [inline]
```

provide the directory into which all files are written.

The documentation for this class was generated from the following file:

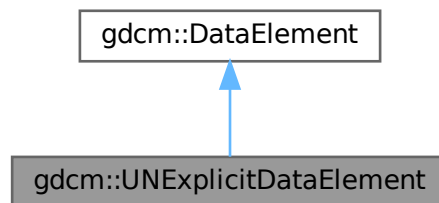
- [gdcmULWritingCallback.h](#)

## 10.362 gdcm::UNExplicitDataElement Class Reference

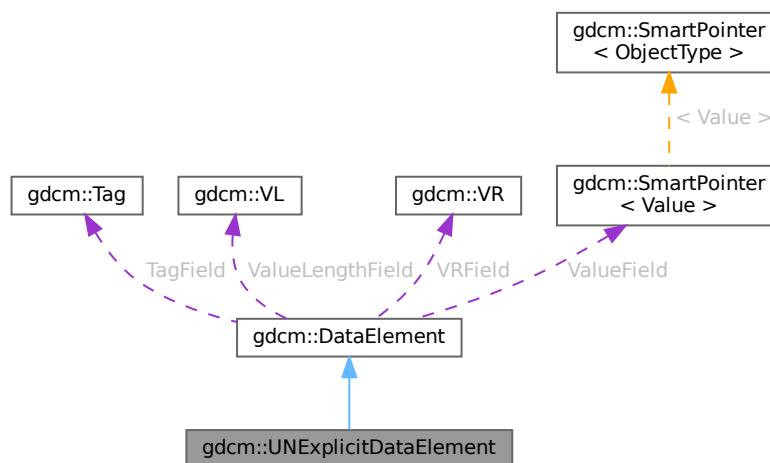
Class to read/write a [DataElement](#) as UNExplicit Data [Element](#).

```
#include <gdcmUNExplicitDataElement.h>
```

Inheritance diagram for gdcm::UNExplicitDataElement:



Collaboration diagram for gdcm::UNExplicitDataElement:



### Public Member Functions

- [VL GetLength](#) () const
- template<typename TSwap >  
std::istream & [Read](#) (std::istream &is)



- template<typename TSwap >  
std::istream & [ReadPreValue](#) (std::istream &is)
- template<typename TSwap >  
std::istream & [ReadValue](#) (std::istream &is, bool readvalues=true)
- template<typename TSwap >  
std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)

## Public Member Functions inherited from [gdcm::DataElement](#)

- [DataElement](#) (const [DataElement](#) &\_val)
- [DataElement](#) (const [Tag](#) &t=[Tag](#)(0), const [VL](#) &vl=0, const [VR](#) &vr=[VR::INVALID](#))
- void [Clear](#) ()  
*Clear Data [Element](#) (make [Value](#) empty and invalidate [Tag](#) & [VR](#))*
- void [Empty](#) ()  
*Make Data [Element](#) empty (no [Value](#))*
- const [ByteValue](#) \* [GetByteValue](#) () const
- template<typename TDE >  
[VL](#) [GetLength](#) () const
- [SequenceOfFragments](#) \* [GetSequenceOfFragments](#) ()
- const [SequenceOfFragments](#) \* [GetSequenceOfFragments](#) () const
- [Tag](#) & [GetTag](#) ()
- const [Tag](#) & [GetTag](#) () const  
*Get [Tag](#).*
- [Value](#) & [GetValue](#) ()
- [Value](#) const & [GetValue](#) () const  
*Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):*
- [SmartPointer](#)< [SequenceOfItems](#) > [GetValueAsSQ](#) () const
- [VL](#) & [GetVL](#) ()
- const [VL](#) & [GetVL](#) () const  
*Get [VL](#).*
- [VR](#) const & [GetVR](#) () const
- bool [IsEmpty](#) () const  
*Check if Data [Element](#) is empty.*
- bool [IsUndefinedLength](#) () const  
*return if [Value](#) Length if of undefined length*
- bool [operator<](#) (const [DataElement](#) &de) const
- [DataElement](#) & [operator=](#) (const [DataElement](#) &)=default
- bool [operator==](#) (const [DataElement](#) &de) const
- template<typename TDE , typename TSwap >  
std::istream & [Read](#) (std::istream &is)
- template<typename TDE , typename TSwap >  
std::istream & [ReadOrSkip](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE , typename TSwap >  
std::istream & [ReadPreValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE , typename TSwap >  
std::istream & [ReadValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE , typename TSwap >  
std::istream & [ReadValueWithLength](#) (std::istream &is, [VL](#) &length, std::set< [Tag](#) > const &skiptags)

- `template<typename TDE , typename TSwap >`  
`std::istream & ReadWithLength (std::istream &is, VL &length)`
- `void SetByteValue (const char *array, VL length)`
- `void SetTag (const Tag &t)`
- `void SetValue (Value const &vl)`
- `void SetVL (const VL &vl)`
- `void SetVLToUndefined ()`
- `void SetVR (VR const &vr)`
- `template<typename TDE , typename TSwap >`  
`const std::ostream & Write (std::ostream &os) const`

### Additional Inherited Members

### Protected Types inherited from `gdcm::DataElement`

- `typedef SmartPointer< Value > ValuePtr`

### Protected Member Functions inherited from `gdcm::DataElement`

- `void SetValueFieldLength (VL vl, bool readvalues)`

### Protected Attributes inherited from `gdcm::DataElement`

- `Tag TagField`
- `ValuePtr ValueField`
- `VL ValueLengthField`
- `VR VRField`

## 10.362.1 Detailed Description

Class to read/write a `DataElement` as UNExplicit Data `Element`.

#### Note

bla

## 10.362.2 Member Function Documentation

### 10.362.2.1 GetLength()

```
VL gdcm::UNExplicitDataElement::GetLength ( ) const
```

### 10.362.2.2 Read()

```
template<typename TSwap >
std::istream & gdcm::UNExplicitDataElement::Read (
    std::istream & is )
```

### 10.362.2.3 ReadPreValue()

```
template<typename TSwap >
std::istream & gdcm::UNExplicitDataElement::ReadPreValue (
    std::istream & is )
```

### 10.362.2.4 ReadValue()

```
template<typename TSwap >
std::istream & gdcm::UNExplicitDataElement::ReadValue (
    std::istream & is,
    bool readvalues = true )
```

### 10.362.2.5 ReadWithLength()

```
template<typename TSwap >
std::istream & gdcm::UNExplicitDataElement::ReadWithLength (
    std::istream & is,
    VL & length )
```

The documentation for this class was generated from the following file:

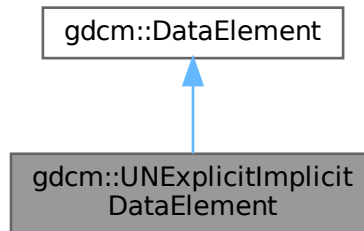
- [gdcmUNExplicitDataElement.h](#)

## 10.363 gdcm::UNExplicitImplicitDataElement Class Reference

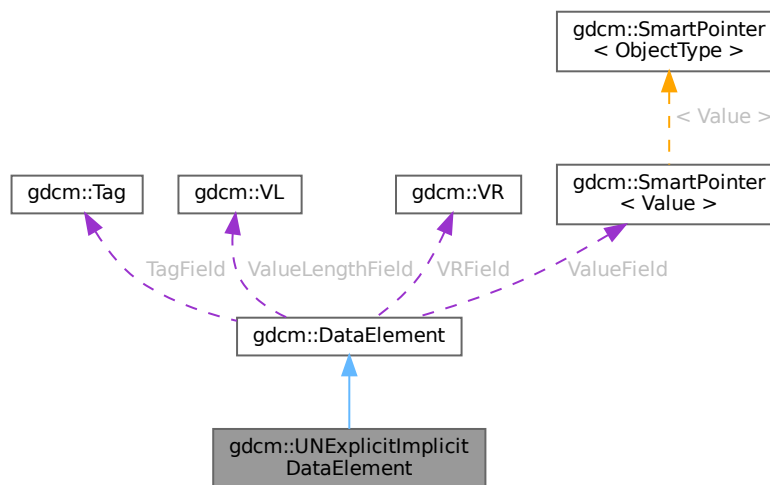
Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#).

```
#include <gdcmUNExplicitImplicitDataElement.h>
```

Inheritance diagram for `gdc::UNExplicitImplicitDataElement`:



Collaboration diagram for `gdc::UNExplicitImplicitDataElement`:



## Public Member Functions

- [VL.GetLength](#) () const
- template<typename TSwap >  
std::istream & [Read](#) (std::istream &is)
- template<typename TSwap >  
std::istream & [ReadPreValue](#) (std::istream &is)
- template<typename TSwap >  
std::istream & [ReadValue](#) (std::istream &is)

## Public Member Functions inherited from gdcm::DataElement

- [DataElement](#) (const [DataElement](#) &\_val)
- [DataElement](#) (const [Tag](#) &t=[Tag](#)(0), const [VL](#) &vl=0, const [VR](#) &vr=[VR::INVALID](#))
- void [Clear](#) ()
  - Clear Data [Element](#) (make [Value](#) empty and invalidate [Tag](#) & [VR](#))*
- void [Empty](#) ()
  - Make Data [Element](#) empty (no [Value](#))*
- const [ByteValue](#) \* [GetByteValue](#) () const
- template<typename TDE >
 [VL](#) [GetLength](#) () const
- [SequenceOfFragments](#) \* [GetSequenceOfFragments](#) ()
- const [SequenceOfFragments](#) \* [GetSequenceOfFragments](#) () const
- [Tag](#) & [GetTag](#) ()
- const [Tag](#) & [GetTag](#) () const
  - Get [Tag](#).*
- [Value](#) & [GetValue](#) ()
- [Value](#) const & [GetValue](#) () const
  - Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):*
- [SmartPointer](#)< [SequenceOfItems](#) > [GetValueAsSQ](#) () const
- [VL](#) & [GetVL](#) ()
- const [VL](#) & [GetVL](#) () const
  - Get [VL](#).*
- [VR](#) const & [GetVR](#) () const
- bool [IsEmpty](#) () const
  - Check if Data [Element](#) is empty.*
- bool [IsUndefinedLength](#) () const
  - return if [Value](#) Length if of undefined length*
- bool [operator](#)< (const [DataElement](#) &de) const
- [DataElement](#) & [operator](#)= (const [DataElement](#) &)=default
- bool [operator](#)== (const [DataElement](#) &de) const
- template<typename TDE , typename TSwap >
 std::istream & [Read](#) (std::istream &is)
- template<typename TDE , typename TSwap >
 std::istream & [ReadOrSkip](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE , typename TSwap >
 std::istream & [ReadPreValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE , typename TSwap >
 std::istream & [ReadValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE , typename TSwap >
 std::istream & [ReadValueWithLength](#) (std::istream &is, [VL](#) &length, std::set< [Tag](#) > const &skiptags)
- template<typename TDE , typename TSwap >
 std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)
- void [SetByteValue](#) (const char \*array, [VL](#) length)
- void [SetTag](#) (const [Tag](#) &t)
- void [SetValue](#) ([Value](#) const &vl)
- void [SetVL](#) (const [VL](#) &vl)
- void [SetVLToUndefined](#) ()
- void [SetVR](#) ([VR](#) const &vr)
- template<typename TDE , typename TSwap >
 const std::ostream & [Write](#) (std::ostream &os) const

## Additional Inherited Members

## Protected Types inherited from [gdcm::DataElement](#)

- typedef [SmartPointer](#)< [Value](#) > [ValuePtr](#)

## Protected Member Functions inherited from [gdcm::DataElement](#)

- void [SetValueFieldLength](#) ([VL](#) vl, bool readvalues)

## Protected Attributes inherited from [gdcm::DataElement](#)

- [Tag](#) [TagField](#)
- [ValuePtr](#) [ValueField](#)
- [VL](#) [ValueLengthField](#)
- [VR](#) [VRField](#)

## 10.363.1 Detailed Description

Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#).

This class gather two known bugs:

1. GDCM 1.2.0 would rewrite [VR](#)=UN [Value](#) Length on 2 bytes instead of 4 bytes
2. GDCM 1.2.0 would also rewrite [DataElement](#) as Implicit when the [VR](#) would not be known this would only happen in some very rare cases. gdcm 2.X design could handle bug #1 or #2 exclusively, this class can now handle file which have both issues. See: [gdcmData/TheralysGDCM120Bug.dcm](#)

## 10.363.2 Member Function Documentation

### 10.363.2.1 GetLength()

```
VL gdcm::UNExplicitImplicitDataElement::GetLength ( ) const
```

### 10.363.2.2 Read()

```
template<typename TSwap >
std::istream & gdcm::UNExplicitImplicitDataElement::Read (
    std::istream & is )
```

### 10.363.2.3 ReadPreValue()

```
template<typename TSwap >
std::istream & gdcm::UNExplicitImplicitDataElement::ReadPreValue (
    std::istream & is )
```

### 10.363.2.4 ReadValue()

```
template<typename TSwap >
std::istream & gdcm::UNExplicitImplicitDataElement::ReadValue (
    std::istream & is )
```

The documentation for this class was generated from the following file:

- [gdcmUNExplicitImplicitDataElement.h](#)

## 10.364 gdcm::Unpacker12Bits Class Reference

Pack/Unpack 12 bits pixel into 16bits.

```
#include <gdcmUnpacker12Bits.h>
```

### Static Public Member Functions

- static bool [Pack](#) (char \*out, const char \*in, size\_t n)
- static bool [Unpack](#) (char \*out, const char \*in, size\_t n)

### 10.364.1 Detailed Description

Pack/Unpack 12 bits pixel into 16bits.

- You can only pack an even number of 16bits, which means a multiple of 4 (expressed in bytes)
- You can only unpack a multiple of 3 bytes

This class has no purpose in general purpose DICOM implementation. However to be able to cope with some early ACR-NEMA file generated by a well-known private vendor, one would need to unpack 12bits Stored Pixel [Value](#) into a more standard 16bits Stored Pixel [Value](#).

See also

[Rescaler](#)

## 10.364.2 Member Function Documentation

### 10.364.2.1 Pack()

```
static bool gdcm::Unpacker12Bits::Pack (  
    char * out,  
    const char * in,  
    size_t n ) [static]
```

Pack an array of 16bits where all values are 12bits into a pack form. n is the length in bytes of array in, out will be a fake 8bits array of size  $(n / 2) * 3$

### 10.364.2.2 Unpack()

```
static bool gdcm::Unpacker12Bits::Unpack (  
    char * out,  
    const char * in,  
    size_t n ) [static]
```

Unpack an array of 'packed' 12bits data into a more conventional 16bits array. n is the length in bytes of array in, out will be a 16bits array of size  $(n / 3) * 2$

The documentation for this class was generated from the following file:

- [gdcmUnpacker12Bits.h](#)

## 10.365 gdcm::Usage Class Reference

[Usage.](#)

```
#include <gdcmUsage.h>
```

### Public Types

- enum [UsageType](#) {  
    [Mandatory](#) ,  
    [Conditional](#) ,  
    [UserOption](#) ,  
    [Invalid](#) }

### Public Member Functions

- [Usage](#) ([UsageType](#) type=[Invalid](#))
- [operator UsageType](#) () const



## Static Public Member Functions

- static const char \* [GetUsageString](#) ([UsageType](#) type)
- static [UsageType](#) [GetUsageType](#) (const char \*type)

## Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [Usage](#) &vr)

## 10.365.1 Detailed Description

[Usage](#).

### Note

A.1.3 [IOD Module Table](#) and Functional Group [Macro Table](#) This Section of each [IOD](#) defines in a tabular form the [Modules](#) comprising the [IOD](#). The following information must be specified for each [Module](#) in the table:

- The name of the [Module](#) or Functional Group
- A reference to the Section in Annex C which defines the [Module](#) or Functional Group
- The usage of the [Module](#) or Functional Group; whether it is:
  - Mandatory (see A.1.3.1) , abbreviated M
  - Conditional (see A.1.3.2) , abbreviated C
  - User Option (see A.1.3.3) , abbreviated U The [Modules](#) referenced are defined in Annex C. A.1.3.1 MANDATORY MODULES For each [IOD](#), Mandatory [Modules](#) shall be supported per the definitions, semantics and requirements defined in Annex C.

A.1.3.2 CONDITIONAL MODULES Conditional [Modules](#) are Mandatory [Modules](#) if specific conditions are met. If the specified conditions are not met, this [Module](#) shall not be supported; that is, no information defined in that [Module](#) shall be sent. A.1.3.3 USER OPTION MODULES User Option [Modules](#) may or may not be supported. If an optional [Module](#) is supported, the [Attribute](#) Types specified in the [Modules](#) in Annex C shall be supported.

## 10.365.2 Member Enumeration Documentation

### 10.365.2.1 UsageType

```
enum gdcmm::Usage::UsageType
```

#### Enumerator

Mandatory	
Conditional	
UserOption	
Invalid	

### 10.365.3 Constructor & Destructor Documentation

#### 10.365.3.1 Usage()

```
gdcM::Usage::Usage (
    UsageType type = Invalid ) [inline]
```

### 10.365.4 Member Function Documentation

#### 10.365.4.1 GetUsageString()

```
static const char * gdcM::Usage::GetUsageString (
    UsageType type ) [static]
```

#### 10.365.4.2 GetUsageType()

```
static UsageType gdcM::Usage::GetUsageType (
    const char * type ) [static]
```

#### 10.365.4.3 operator UsageType()

```
gdcM::Usage::operator UsageType ( ) const [inline]
```

### 10.365.5 Friends And Related Symbol Documentation

#### 10.365.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & os,
    const Usage & vr ) [friend]
```

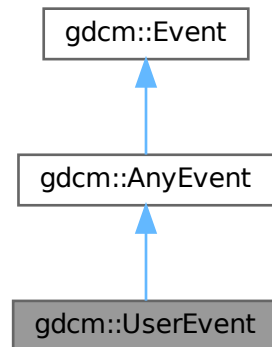
The documentation for this class was generated from the following file:

- [gdcMUsage.h](#)

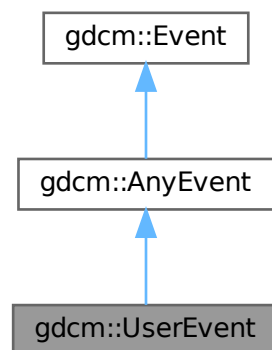
## 10.366 gdcm::UserEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::UserEvent:



Collaboration diagram for gdcm::UserEvent:



### Additional Inherited Members

### Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()

- [Event](#) (const [Event](#) &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) \*) const =0
- virtual const char \* [GetEventName](#) () const =0
- virtual [Event](#) \* [MakeObject](#) () const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

The documentation for this class was generated from the following file:

- [gdcmEvent.h](#)

## 10.367 gdcm::network::UserInformation Class Reference

[UserInformation](#).

```
#include <gdcmUserInformation.h>
```

### Public Member Functions

- [UserInformation](#) ()
- [UserInformation](#) (const [UserInformation](#) &)=delete
- [~UserInformation](#) ()
- void [AddRoleSelectionSub](#) ([RoleSelectionSub](#) const &r)
- void [AddSOPClassExtendedNegociationSub](#) ([SOPClassExtendedNegociationSub](#) const &s)
- [MaximumLengthSub](#) & [GetMaximumLengthSub](#) ()
- const [MaximumLengthSub](#) & [GetMaximumLengthSub](#) () const
- [UserInformation](#) & [operator=](#) (const [UserInformation](#) &)
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- size\_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

### 10.367.1 Detailed Description

[UserInformation](#).

[Table 9-16](#) USER INFORMATION ITEM FIELDS

TODO what is the goal of :

[Table 9-20](#) USER INFORMATION ITEM FIELDS

## 10.367.2 Constructor & Destructor Documentation

### 10.367.2.1 UserInformation() [1/2]

```
gdcm::network::UserInformation::UserInformation ( )
```

### 10.367.2.2 ~UserInformation()

```
gdcm::network::UserInformation::~~UserInformation ( )
```

### 10.367.2.3 UserInformation() [2/2]

```
gdcm::network::UserInformation::UserInformation (
    const UserInformation & ) [delete]
```

## 10.367.3 Member Function Documentation

### 10.367.3.1 AddRoleSelectionSub()

```
void gdcm::network::UserInformation::AddRoleSelectionSub (
    RoleSelectionSub const & r )
```

### 10.367.3.2 AddSOPClassExtendedNegociationSub()

```
void gdcm::network::UserInformation::AddSOPClassExtendedNegociationSub (
    SOPClassExtendedNegociationSub const & s )
```

### 10.367.3.3 GetMaximumLengthSub() [1/2]

```
MaximumLengthSub & gdcm::network::UserInformation::GetMaximumLengthSub ( ) [inline]
```

### 10.367.3.4 GetMaximumLengthSub() [2/2]

```
const MaximumLengthSub & gdcm::network::UserInformation::GetMaximumLengthSub ( ) const [inline]
```

### 10.367.3.5 operator=()

```
UserInformation & gdcm::network::UserInformation::operator= (
    const UserInformation & )
```

### 10.367.3.6 Print()

```
void gdcmm::network::UserInformation::Print (
    std::ostream & os ) const
```

### 10.367.3.7 Read()

```
std::istream & gdcmm::network::UserInformation::Read (
    std::istream & is )
```

### 10.367.3.8 Size()

```
size_t gdcmm::network::UserInformation::Size ( ) const
```

### 10.367.3.9 Write()

```
const std::ostream & gdcmm::network::UserInformation::Write (
    std::ostream & os ) const
```

The documentation for this class was generated from the following file:

- [gdcmmUserInformation.h](#)

## 10.368 gdcmm::UUIDGenerator Class Reference

Class for generating unique UUID.

```
#include <gdcmmUUIDGenerator.h>
```

### Public Member Functions

- const char \* [Generate](#) ()

### Static Public Member Functions

- static bool [IsValid](#) (const char \*uid)  
*Find out if the string is a valid UUID or not.*

### 10.368.1 Detailed Description

Class for generating unique UUID.

generate DCE 1.1 uid

### 10.368.2 Member Function Documentation

#### 10.368.2.1 Generate()

```
const char * gdcm::UUIDGenerator::Generate ( )
```

Return the generated uuid NOT THREAD SAFE

#### 10.368.2.2 IsValid()

```
static bool gdcm::UUIDGenerator::IsValid (
    const char * uid ) [static]
```

Find out if the string is a valid UUID or not.

The documentation for this class was generated from the following file:

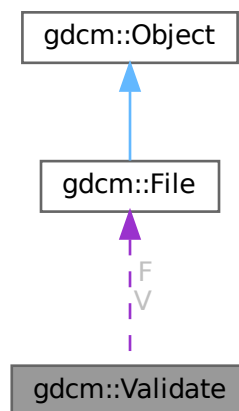
- [gdcmUUIDGenerator.h](#)

## 10.369 gdcm::Validate Class Reference

[Validate](#) class.

```
#include <gdcmValidate.h>
```

Collaboration diagram for gdcm::Validate:



## Public Member Functions

- [Validate](#) ()
- [~Validate](#) ()
- const [File](#) & [GetValidatedFile](#) ()
- void [SetFile](#) ([File](#) const &f)
- void [Validation](#) ()

## Protected Attributes

- const [File](#) \* [F](#)
- [File](#) [V](#)

## 10.369.1 Detailed Description

[Validate](#) class.

## 10.369.2 Constructor & Destructor Documentation

### 10.369.2.1 [Validate](#)()

```
gdcm::Validate::Validate ( )
```

### 10.369.2.2 [~Validate](#)()

```
gdcm::Validate::~~Validate ( )
```

## 10.369.3 Member Function Documentation

### 10.369.3.1 [GetValidatedFile](#)()

```
const File & gdcm::Validate::GetValidatedFile ( ) [inline]
```

### 10.369.3.2 [SetFile](#)()

```
void gdcm::Validate::SetFile (
    File const & f ) [inline]
```

### 10.369.3.3 [Validation](#)()

```
void gdcm::Validate::Validation ( )
```



## 10.369.4 Member Data Documentation

### 10.369.4.1 F

```
const File* gdcm::Validate::F [protected]
```

### 10.369.4.2 V

```
File gdcm::Validate::V [protected]
```

The documentation for this class was generated from the following file:

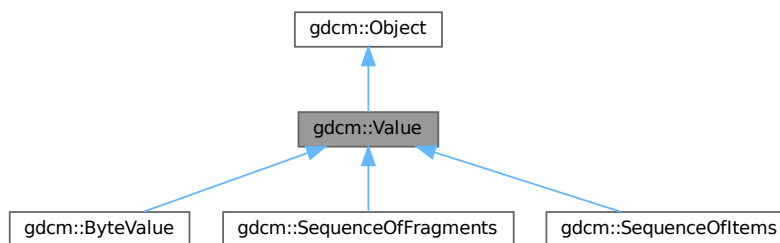
- [gdcmValidate.h](#)

## 10.370 gdcm::Value Class Reference

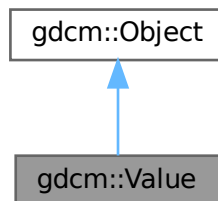
Class to represent the value of a Data [Element](#).

```
#include <gdcmValue.h>
```

Inheritance diagram for gdcm::Value:



Collaboration diagram for gdcm::Value:



**Public Member Functions**

- [Value](#) ()=default
- [~Value](#) () override=default
- virtual void [Clear](#) ()=0
- virtual [VL GetLength](#) () const =0
- virtual bool [operator==](#) (const [Value](#) &val) const =0
- virtual void [SetLength](#) ([VL](#) l)=0

**Public Member Functions inherited from [gdcm::Object](#)**

- [Object](#) ()
- [Object](#) (const [Object](#) &)
- *Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

**Protected Member Functions**

- virtual void [SetLengthOnly](#) ([VL](#) l)

**Protected Member Functions inherited from [gdcm::Object](#)**

- void [Register](#) ()
- void [UnRegister](#) ()

**Friends**

- class [DataElement](#)

**10.370.1 Detailed Description**

Class to represent the value of a Data [Element](#).

**Note**

VALUE: A component of a [Value](#) Field. A [Value](#) Field may consist of one or more of these components.

**10.370.2 Constructor & Destructor Documentation****10.370.2.1 Value()**

```
gdcm::Value::Value ( ) [default]
```

### 10.370.2.2 ~Value()

```
gdcm::Value::~~Value ( ) [override], [default]
```

## 10.370.3 Member Function Documentation

### 10.370.3.1 Clear()

```
virtual void gdcm::Value::Clear ( ) [pure virtual]
```

Implemented in [gdcm::ByteValue](#), [gdcm::SequenceOfFragments](#), and [gdcm::SequenceOfItems](#).

### 10.370.3.2 GetLength()

```
virtual VL gdcm::Value::GetLength ( ) const [pure virtual]
```

Implemented in [gdcm::ByteValue](#), [gdcm::SequenceOfFragments](#), and [gdcm::SequenceOfItems](#).

Referenced by [gdcm::DataSet::InsertDataElement\(\)](#), and [gdcm::DataElement::SetValue\(\)](#).

### 10.370.3.3 operator==( )

```
virtual bool gdcm::Value::operator== (
    const Value & val ) const [pure virtual]
```

Implemented in [gdcm::ByteValue](#), [gdcm::SequenceOfFragments](#), and [gdcm::SequenceOfItems](#).

### 10.370.3.4 SetLength()

```
virtual void gdcm::Value::SetLength (
    VL l ) [pure virtual]
```

Implemented in [gdcm::SequenceOfFragments](#), [gdcm::SequenceOfItems](#), and [gdcm::ByteValue](#).

### 10.370.3.5 SetLengthOnly()

```
virtual void gdcm::Value::SetLengthOnly (
    VL l ) [protected], [virtual]
```

Reimplemented in [gdcm::ByteValue](#).

## 10.370.4 Friends And Related Symbol Documentation

### 10.370.4.1 DataElement

```
friend class DataElement [friend]
```

The documentation for this class was generated from the following file:

- [gdcmValue.h](#)

## 10.371 gdcm::ValueIO< TDE, TSwap, TType > Class Template Reference

Class to dispatch template calls.

```
#include <gdcmValueIO.h>
```

### Static Public Member Functions

- static std::istream & [Read](#) (std::istream &is, [Value](#) &v, bool readvalues)
- static const std::ostream & [Write](#) (std::ostream &os, const [Value](#) &v)

### 10.371.1 Detailed Description

```
template<typename TDE, typename TSwap, typename TType = uint8_t>  
class gdcm::ValueIO< TDE, TSwap, TType >
```

Class to dispatch template calls.

## 10.371.2 Member Function Documentation

### 10.371.2.1 Read()

```
template<typename TDE , typename TSwap , typename TType = uint8_t>  
static std::istream & gdcm::ValueIO< TDE, TSwap, TType >::Read (  
    std::istream & is,  
    Value & v,  
    bool readvalues ) [static]
```

### 10.371.2.2 Write()

```
template<typename TDE , typename TSwap , typename TType = uint8_t>
static const std::ostream & gdcm::ValueIO< TDE, TSwap, TType >::Write (
    std::ostream & os,
    const Value & v ) [static]
```

The documentation for this class was generated from the following file:

- [gdcmValueIO.h](#)

## 10.372 gdcm::MrProtocol::Vector3 Struct Reference

```
#include <gdcmMrProtocol.h>
```

### Public Attributes

- double [dCor](#)
- double [dSag](#)
- double [dTra](#)

### 10.372.1 Member Data Documentation

#### 10.372.1.1 dCor

```
double gdcm::MrProtocol::Vector3::dCor
```

#### 10.372.1.2 dSag

```
double gdcm::MrProtocol::Vector3::dSag
```

#### 10.372.1.3 dTra

```
double gdcm::MrProtocol::Vector3::dTra
```

The documentation for this struct was generated from the following file:

- [gdcmMrProtocol.h](#)

## 10.373 gdcm::Version Class Reference

major/minor and build version

```
#include <gdcmVersion.h>
```

### Public Member Functions

- [Version](#) ()=default
- [~Version](#) ()=default
- void [Print](#) (std::ostream &os=std::cout) const

### Static Public Member Functions

- static int [GetBuildVersion](#) ()
- static int [GetMajorVersion](#) ()
- static int [GetMinorVersion](#) ()
- static const char \* [GetVersion](#) ()

### Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [Version](#) &v)

### 10.373.1 Detailed Description

major/minor and build version

### 10.373.2 Constructor & Destructor Documentation

#### 10.373.2.1 Version()

```
gdcm::Version::Version ( ) [default]
```

#### 10.373.2.2 ~Version()

```
gdcm::Version::~~Version ( ) [default]
```

### 10.373.3 Member Function Documentation

#### 10.373.3.1 GetBuildVersion()

```
static int gdcm::Version::GetBuildVersion ( ) [static]
```

### 10.373.3.2 GetMajorVersion()

```
static int gdcm::Version::GetMajorVersion ( ) [static]
```

### 10.373.3.3 GetMinorVersion()

```
static int gdcm::Version::GetMinorVersion ( ) [static]
```

### 10.373.3.4 GetVersion()

```
static const char * gdcm::Version::GetVersion ( ) [static]
```

### 10.373.3.5 Print()

```
void gdcm::Version::Print (
    std::ostream & os = std::cout ) const
```

## 10.373.4 Friends And Related Symbol Documentation

### 10.373.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const Version & v ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmVersion.h](#)

## 10.374 gdcm::VL Class Reference

[Value](#) Length.

```
#include <gdcmVL.h>
```

### Public Types

- typedef uint32\_t [Type](#)

## Public Member Functions

- [VL](#) (uint32\_t vl=0)
- [VL GetLength](#) () const
- bool [IsOdd](#) () const  
*Return whether or not the [VL](#) is odd or not.*
- bool [IsUndefined](#) () const
- [operator uint32\\_t](#) () const
- [VL](#) & [operator++](#) ()
- [VL](#) [operator++](#) (int)
- [VL](#) & [operator+=](#) ([VL](#) const &vl)  
*+= operator*
- template<typename TSwap >  
std::istream & [Read](#) (std::istream &is)
- template<typename TSwap >  
std::istream & [Read16](#) (std::istream &is)
- void [SetToUndefined](#) ()
- template<typename TSwap >  
const std::ostream & [Write](#) (std::ostream &os) const
- template<typename TSwap >  
const std::ostream & [Write16](#) (std::ostream &os) const

## Static Public Member Functions

- static uint16\_t [GetVL16Max](#) ()
- static uint32\_t [GetVL32Max](#) ()

## Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [VL](#) &vl)

### 10.374.1 Detailed Description

[Value](#) Length.

#### Warning

this is a 4bytes value ! Do not try to use it for 2bytes value length

#### Examples

[BasicImageAnonymizer.cs](#), [DecompressImage.cs](#), [ReadAndDumpDICOMDIR2.cxx](#), and [rle2img.cxx](#).

### 10.374.2 Member Typedef Documentation

#### 10.374.2.1 Type

```
typedef uint32_t gdcm::VL::Type
```



## 10.374.3 Constructor & Destructor Documentation

### 10.374.3.1 VL()

```
gdcm::VL::VL (
    uint32_t vl = 0 ) [inline]
```

## 10.374.4 Member Function Documentation

### 10.374.4.1 GetLength()

```
VL gdcm::VL::GetLength ( ) const [inline]
```

#### Examples

[ReadAndDumpDICOMDIR2.cxx](#).

Referenced by [gdcm::FileMetaInformation::GetFullLength\(\)](#), [gdcm::DataSet::GetLength\(\)](#), and [gdcm::Item::Write\(\)](#).

### 10.374.4.2 GetVL16Max()

```
static uint16_t gdcm::VL::GetVL16Max ( ) [inline], [static]
```

### 10.374.4.3 GetVL32Max()

```
static uint32_t gdcm::VL::GetVL32Max ( ) [inline], [static]
```

### 10.374.4.4 IsOdd()

```
bool gdcm::VL::IsOdd ( ) const [inline]
```

Return whether or not the [VL](#) is odd or not.

### 10.374.4.5 IsUndefined()

```
bool gdcm::VL::IsUndefined ( ) const [inline]
```

### 10.374.4.6 operator uint32\_t()

```
gdcm::VL::operator uint32_t ( ) const [inline]
```

**10.374.4.7 operator++()** [1/2]

```
VL & gdcmm::VL::operator++ ( ) [inline]
```

**10.374.4.8 operator++()** [2/2]

```
VL gdcmm::VL::operator++ (
    int ) [inline]
```

**10.374.4.9 operator+=( )**

```
VL & gdcmm::VL::operator+= (
    VL const & vl ) [inline]
```

**+= operator**

**10.374.4.10 Read()**

```
template<typename TSwap >
std::istream & gdcmm::VL::Read (
    std::istream & is ) [inline]
```

**10.374.4.11 Read16()**

```
template<typename TSwap >
std::istream & gdcmm::VL::Read16 (
    std::istream & is ) [inline]
```

**10.374.4.12 SetToUndefined()**

```
void gdcmm::VL::SetToUndefined ( ) [inline]
```

**10.374.4.13 Write()**

```
template<typename TSwap >
const std::ostream & gdcmm::VL::Write (
    std::ostream & os ) const [inline]
```

Referenced by [gdcmm::Fragment::Write\(\)](#), [gdcmm::Item::Write\(\)](#), [gdcmm::SequenceOfFragments::Write\(\)](#), and [gdcmm::SequenceOfItems::Write\(\)](#)

#### 10.374.4.14 Write16()

```
template<typename TSwap >
const std::ostream & gdcm::VL::Write16 (
    std::ostream & os ) const [inline]
```

### 10.374.5 Friends And Related Symbol Documentation

#### 10.374.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & os,
    const VL & vl ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmVL.h](#)

## 10.375 gdcm::VM Class Reference

**Value** Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n.

```
#include <gdcmVM.h>
```

### Public Types

- enum **VMType** {
  - VM0** = 0 ,
  - VM1** = 1 ,
  - VM2** = 2 ,
  - VM3** = 4 ,
  - VM4** = 8 ,
  - VM5** = 16 ,
  - VM6** = 32 ,
  - VM8** = 64 ,
  - VM9** = 128 ,
  - VM10** = 256 ,
  - VM12** = 512 ,
  - VM16** = 1024 ,
  - VM18** = 2048 ,
  - VM24** = 4096 ,
  - VM28** = 8192 ,
  - VM32** = 16384 ,
  - VM35** = 32768 ,
  - VM99** = 65536 ,
  - VM256** = 131072 ,

```

VM1_2 = VM1 | VM2 ,
VM1_3 = VM1 | VM2 | VM3 ,
VM1_4 = VM1 | VM2 | VM3 | VM4 ,
VM1_5 = VM1 | VM2 | VM3 | VM4 | VM5 ,
VM1_8 = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 ,
VM1_32 = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 ,
VM1_99 = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 ,
VM1_n = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256 ,
VM2_2n = VM2 | VM4 | VM6 | VM8 | VM16 | VM24 | VM32 | VM256 ,
VM2_n = VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256 ,
VM3_4 = VM3 | VM4 ,
VM3_3n = VM3 | VM6 | VM9 | VM24 | VM99 | VM256 ,
VM3_n = VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256 ,
VM4_4n = VM4 | VM16 | VM24 | VM32 | VM256 ,
VM6_6n = VM6 | VM12 | VM18 | VM24 ,
VM6_n = VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256 ,
VM7_7n ,
VM30_30n ,
VM47_47n ,
VM_END = VM1_n + 1 }

```

### Public Member Functions

- [VM](#) ([VMType](#) type=[VM0](#))
- bool [Compatible](#) ([VM](#) const &vm) const
- unsigned int [GetLength](#) () const
- [operator VMType](#) () const

### Static Public Member Functions

- static size\_t [GetNumberOfElementsFromArray](#) (const char \*array, size\_t length)
- static const char \* [GetVMString](#) ([VMType](#) vm)
- static [VMType](#) [GetVMType](#) (const char \*vm)
- static [VMType](#) [GetVMTypeFromLength](#) (size\_t length, unsigned int size)
- static bool [IsValid](#) (int vm1, [VMType](#) vm2)

### Static Protected Member Functions

- static unsigned int [GetIndex](#) ([VMType](#) vm)

### Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [VM](#) &vm)

## 10.375.1 Detailed Description

**Value** Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n.

Some private dict define some more: 4-4n 1-4 1-5 256 9 3-4

even more:

7-7n 10 18 12 35 47\_47n 30\_30n 28

6-6n

## 10.375.2 Member Enumeration Documentation

### 10.375.2.1 VMType

enum `gdcmm::VM::VMType`

#### Enumerator

VM0	
VM1	
VM2	
VM3	
VM4	
VM5	
VM6	
VM8	
VM9	
VM10	
VM12	
VM16	
VM18	
VM24	
VM28	
VM32	
VM35	
VM99	
VM256	
VM1_2	
VM1_3	
VM1_4	
VM1_5	
VM1_8	
VM1_32	
VM1_99	
VM1_n	
VM2_2n	

## Enumerator

VM2_n	
VM3_4	
VM3_3n	
VM3_n	
VM4_4n	
VM6_6n	
VM6_n	
VM7_7n	
VM30_30n	
VM47_47n	
VM_END	

### 10.375.3 Constructor & Destructor Documentation

#### 10.375.3.1 VM()

```
gdcmm::VM::VM (
    VMType type = VM0 ) [inline]
```

### 10.375.4 Member Function Documentation

#### 10.375.4.1 Compatible()

```
bool gdcmm::VM::Compatible (
    VM const & vm ) const
```

WARNING: Implementation deficiency The Compatible function is poorly implemented, the reference vm should be coming from the dictionary, while the passed in value is the value guess from the file.

#### 10.375.4.2 GetIndex()

```
static unsigned int gdcmm::VM::GetIndex (
    VMType vm ) [static], [protected]
```

#### 10.375.4.3 GetLength()

```
unsigned int gdcmm::VM::GetLength ( ) const
```

References [gdcmm::operator<<\(\)](#).

**10.375.4.4 GetNumberOfElementsFromArray()**

```
static size_t gdcm::VM::GetNumberOfElementsFromArray (
    const char * array,
    size_t length ) [static]
```

**10.375.4.5 GetVMString()**

```
static const char * gdcm::VM::GetVMString (
    VMType vm ) [static]
```

Return the string as written in the official DICOM dict from a custom enum type

**10.375.4.6 GetVMType()**

```
static VMType gdcm::VM::GetVMType (
    const char * vm ) [static]
```

**10.375.4.7 GetVMTypeFromLength()**

```
static VMType gdcm::VM::GetVMTypeFromLength (
    size_t length,
    unsigned int size ) [static]
```

**10.375.4.8 IsValid()**

```
static bool gdcm::VM::IsValid (
    int vm1,
    VMType vm2 ) [static]
```

Check if vm1 is valid compare to vm2, i.e vm1 is element of vm2 vm1 is typically deduce from counting in a ValueField

**10.375.4.9 operator VMType()**

```
gdcm::VM::operator VMType ( ) const [inline]
```

**10.375.5 Friends And Related Symbol Documentation****10.375.5.1 operator<<**

```
std::ostream & operator<< (
    std::ostream & os,
    const VM & vm ) [friend]
```

The documentation for this class was generated from the following file:

- [gdcmVM.h](#)

## 10.376 gdcm::VMToLength< T > Struct Template Reference

The documentation for this struct was generated from the following file:

- [gdcmVM.h](#)

## 10.377 gdcm::VR Class Reference

VR class.

```
#include <gdcmVR.h>
```

### Public Types

- enum [VRType](#) : long long {  
    [INVALID](#) = 0 ,  
    [AE](#) = 1 ,  
    [AS](#) = 2 ,  
    [AT](#) = 4 ,  
    [CS](#) = 8 ,  
    [DA](#) = 16 ,  
    [DS](#) = 32 ,  
    [DT](#) = 64 ,  
    [FD](#) = 128 ,  
    [FL](#) = 256 ,  
    [IS](#) = 512 ,  
    [LO](#) = 1024 ,  
    [LT](#) = 2048 ,  
    [OB](#) = 4096 ,  
    [OD](#) = 134217728 ,  
    [OF](#) = 8192 ,  
    [OL](#) = 268435456 ,  
    [OV](#) = 2147483648 ,  
    [OW](#) = 16384 ,  
    [PN](#) = 32768 ,  
    [SH](#) = 65536 ,  
    [SL](#) = 131072 ,  
    [SQ](#) = 262144 ,  
    [SS](#) = 524288 ,  
    [ST](#) = 1048576 ,  
    [SV](#) = 4294967296 ,  
    [TM](#) = 2097152 ,  
    [UC](#) = 536870912 ,  
    [UI](#) = 4194304 ,  
    [UL](#) = 8388608 ,  
    [UN](#) = 16777216 ,  
    [UR](#) = 1073741824 ,  
    [US](#) = 33554432 ,  
    [UT](#) = 67108864 ,



```

UV = 8589934592 ,
OB_OW = OB | OW ,
US_SS = US | SS ,
US_SS_OW = US | SS | OW ,
US_OW = US | OW ,
VL16 = AE | AS | AT | CS | DA | DS | DT | FD | FL | IS | LO | LT | PN | SH | SL | SS | ST | TM | UI | UL | US ,
VL32 = OB | OW | OD | OF | OL | OV | SQ | SV | UC | UN | UR | UT | UV ,
VRASCII = AE | AS | CS | DA | DS | DT | IS | LO | LT | PN | SH | ST | TM | UC | UI | UR | UT ,
VRBINARY = AT | FL | FD | OB | OD | OF | OL | OV | OW | SL | SQ | SS | SV | UL | UN | US | UV ,
VR_VM1 = AS | LT | ST | UT | SQ | OF | OL | OV | OD | OW | OB | UN ,
VRALL = VRASCII | VRBINARY ,
VR_END = UV+1 }

```

### Public Member Functions

- [VR](#) (VRType vr=INVALID)
- bool [Compatible](#) (VR const &vr) const
- int [GetLength](#) () const
- unsigned int [GetSize](#) () const
- unsigned int [GetSizeof](#) () const
- bool [IsDual](#) () const
- bool [IsVRFile](#) () const
- [operator VRType](#) () const
- std::istream & [Read](#) (std::istream &is)
- const std::ostream & [Write](#) (std::ostream &os) const

### Static Public Member Functions

- static bool [CanDisplay](#) (VRType vr)
- static uint32\_t [GetLength](#) (VRType vr)
- static const char \* [GetVRString](#) (VRType vr)
- static const char \* [GetVRStringFromFile](#) (VRType vr)
- static VRType [GetVRType](#) (const char \*vr)
- static VRType [GetVRTypeFromFile](#) (const char \*vr)
- static bool [IsASCII](#) (VRType vr)
- static bool [IsASCII2](#) (VRType vr)
- static bool [IsBinary](#) (VRType vr)
- static bool [IsBinary2](#) (VRType vr)
- static bool [IsSwap](#) (const char \*vr)
- static bool [IsValid](#) (const char \*vr)
- static bool [IsValid](#) (const char \*vr1, VRType vr2)

### Friends

- std::ostream & [operator<<](#) (std::ostream &os, const VR &vr)

### 10.377.1 Detailed Description

[VR](#) class.

This is adapted from DICOM standard The biggest difference is the INVALID [VR](#) and the composite one that differ from standard (more like an addition) This allow us to represent all the possible case express in the DICOMV3 dict

#### Note

VALUE REPRESENTATION ([VR](#)) Specifies the data type and format of the Value(s) contained in the [Value](#) Field of a Data [Element](#). VALUE REPRESENTATION FIELD: The field where the [Value](#) Representation of a Data [Element](#) is stored in the encoding of a Data [Element](#) structure with explicit [VR](#).

#### Examples

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), and [SimplePrint.cs](#).

### 10.377.2 Member Enumeration Documentation

#### 10.377.2.1 VRType

```
enum gdcm::VR::VRType : long long
```

#### Enumerator

INVALID	
AE	
AS	
AT	
CS	
DA	
DS	
DT	
FD	
FL	
IS	
LO	
LT	
OB	
OD	
OF	
OL	
OV	
OW	
PN	
SH	
SL	
SQ	
SS	

## Enumerator

ST	
SV	
TM	
UC	
UI	
UL	
UN	
UR	
US	
UT	
UV	
OB_OW	
US_SS	
US_SS_OW	
US_OW	
VL16	
VL32	
VRASCII	
VRBINARY	
VR_VM1	
VRALL	
VR_END	

## Examples

[Cleaner.cs](#), [NewSequence.cs](#), and [SimplePrint.cs](#).

### 10.377.3 Constructor & Destructor Documentation

#### 10.377.3.1 VR()

```
gdcm::VR::VR (
    VRType vr = INVALID ) [inline]
```

### 10.377.4 Member Function Documentation

#### 10.377.4.1 CanDisplay()

```
static bool gdcm::VR::CanDisplay (
    VRType vr ) [static]
```

#### 10.377.4.2 Compatible()

```
bool gdcm::VR::Compatible (
    VR const & vr ) const
```

##### Examples

[SimplePrint.cs](#).

#### 10.377.4.3 GetLength() [1/2]

```
int gdcm::VR::GetLength ( ) const [inline]
```

#### 10.377.4.4 GetLength() [2/2]

```
static uint32_t gdcm::VR::GetLength (
    VRType vr ) [inline], [static]
```

#### 10.377.4.5 GetSize()

```
unsigned int gdcm::VR::GetSize ( ) const [inline]
```

References [AE](#), [AS](#), [AT](#), [CS](#), [DA](#), [DS](#), [DT](#), [FD](#), [FL](#), [INVALID](#), [IS](#), [LT](#), [OB](#), [OB\\_OW](#), [OD](#), [OF](#), [OL](#), [OV](#), [OW](#), [PN](#), [SH](#), [SL](#), [SQ](#), [SS](#), [ST](#), [SV](#), [TM](#), [UC](#), [UL](#), [UN](#), [UR](#), [US](#), [US\\_OW](#), [US\\_SS](#), [US\\_SS\\_OW](#), [UT](#), [UV](#), [VL16](#), [VL32](#), [VR\\_END](#), [VR\\_VM1](#), [VRALL](#), [VRASCII](#), [VRBINARY](#), and [VRTypeTemplateCase](#).

#### 10.377.4.6 GetSizeof()

```
unsigned int gdcm::VR::GetSizeof ( ) const
```

#### 10.377.4.7 GetVRString()

```
static const char * gdcm::VR::GetVRString (
    VRType vr ) [static]
```

#### 10.377.4.8 GetVRStringFromFile()

```
static const char * gdcm::VR::GetVRStringFromFile (
    VRType vr ) [static]
```

#### 10.377.4.9 GetVRType()

```
static VRType gdcm::VR::GetVRType (
    const char * vr ) [static]
```

#### 10.377.4.10 GetVRTypeFromFile()

```
static VRType gdcm::VR::GetVRTypeFromFile (
    const char * vr ) [static]
```

#### 10.377.4.11 IsASCII()

```
static bool gdcm::VR::IsASCII (
    VRType vr ) [static]
```

#### 10.377.4.12 IsASCII2()

```
static bool gdcm::VR::IsASCII2 (
    VRType vr ) [static]
```

#### 10.377.4.13 IsBinary()

```
static bool gdcm::VR::IsBinary (
    VRType vr ) [static]
```

#### 10.377.4.14 IsBinary2()

```
static bool gdcm::VR::IsBinary2 (
    VRType vr ) [static]
```

#### 10.377.4.15 IsDual()

```
bool gdcm::VR::IsDual ( ) const
```

#### 10.377.4.16 IsSwap()

```
static bool gdcm::VR::IsSwap (
    const char * vr ) [static]
```

**10.377.4.17 IsValid() [1/2]**

```
static bool gdcm::VR::IsValid (
    const char * vr ) [static]
```

**10.377.4.18 IsValid() [2/2]**

```
static bool gdcm::VR::IsValid (
    const char * vr1,
    VRType vr2 ) [static]
```

**10.377.4.19 IsVRFile()**

```
bool gdcm::VR::IsVRFile ( ) const
```

Referenced by [gdcm::DataElement::SetVR\(\)](#).

**10.377.4.20 operator VRType()**

```
gdcm::VR::operator VRType ( ) const [inline]
```

**10.377.4.21 Read()**

```
std::istream & gdcm::VR::Read (
    std::istream & is ) [inline]
```

References [gdcmDebugMacro](#).

**10.377.4.22 Write()**

```
const std::ostream & gdcm::VR::Write (
    std::ostream & os ) const [inline]
```

References [gdcmAssertAlwaysMacro](#).

**10.377.5 Friends And Related Symbol Documentation****10.377.5.1 operator<<**

```
std::ostream & operator<< (
    std::ostream & os,
    const VR & vr ) [friend]
```

The documentation for this class was generated from the following file:

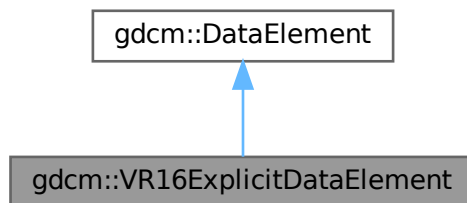
- [gdcmVR.h](#)

## 10.378 gdcm::VR16ExplicitDataElement Class Reference

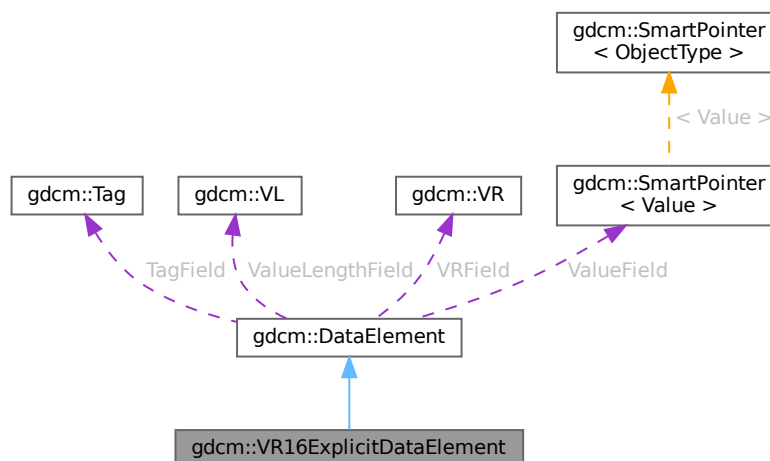
Class to read/write a [DataElement](#) as Explicit Data [Element](#).

```
#include <gdcmVR16ExplicitDataElement.h>
```

Inheritance diagram for gdcm::VR16ExplicitDataElement:



Collaboration diagram for gdcm::VR16ExplicitDataElement:



### Public Member Functions

- [VL GetLength](#) () const
- template<typename TSwap >  
std::istream & [Read](#) (std::istream &is)

- `template<typename TSwap >`  
`std::istream & ReadPreValue (std::istream &is)`
- `template<typename TSwap >`  
`std::istream & ReadValue (std::istream &is, bool readvalues=true)`
- `template<typename TSwap >`  
`std::istream & ReadWithLength (std::istream &is, VL &length)`

## Public Member Functions inherited from `gdcm::DataElement`

- `DataElement (const DataElement &_val)`
- `DataElement (const Tag &t=Tag(0), const VL &vl=0, const VR &vr=VR::INVALID)`
- `void Clear ()`  
*Clear Data Element (make Value empty and invalidate Tag & VR)*
- `void Empty ()`  
*Make Data Element empty (no Value)*
- `const ByteValue * GetByteValue () const`
- `template<typename TDE >`  
`VL GetLength () const`
- `SequenceOfFragments * GetSequenceOfFragments ()`
- `const SequenceOfFragments * GetSequenceOfFragments () const`
- `Tag & GetTag ()`
- `const Tag & GetTag () const`  
*Get Tag.*
- `Value & GetValue ()`
- `Value const & GetValue () const`  
*Set/Get Value (bytes array, SQ of items, SQ of fragments):*
- `SmartPointer< SequenceOfItems > GetValueAsSQ () const`
- `VL & GetVL ()`
- `const VL & GetVL () const`  
*Get VL.*
- `VR const & GetVR () const`
- `bool IsEmpty () const`  
*Check if Data Element is empty.*
- `bool IsUndefinedLength () const`  
*return if Value Length if of undefined length*
- `bool operator< (const DataElement &de) const`
- `DataElement & operator= (const DataElement &)=default`
- `bool operator== (const DataElement &de) const`
- `template<typename TDE , typename TSwap >`  
`std::istream & Read (std::istream &is)`
- `template<typename TDE , typename TSwap >`  
`std::istream & ReadOrSkip (std::istream &is, std::set< Tag > const &skiptags)`
- `template<typename TDE , typename TSwap >`  
`std::istream & ReadPreValue (std::istream &is, std::set< Tag > const &skiptags)`
- `template<typename TDE , typename TSwap >`  
`std::istream & ReadValue (std::istream &is, std::set< Tag > const &skiptags)`
- `template<typename TDE , typename TSwap >`  
`std::istream & ReadValueWithLength (std::istream &is, VL &length, std::set< Tag > const &skiptags)`



- template<typename TDE , typename TSwap >  
std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)
- void [SetByteValue](#) (const char \*array, [VL](#) length)
- void [SetTag](#) (const [Tag](#) &t)
- void [SetValue](#) ([Value](#) const &vl)
- void [SetVL](#) (const [VL](#) &vl)
- void [SetVLToUndefined](#) ()
- void [SetVR](#) ([VR](#) const &vr)
- template<typename TDE , typename TSwap >  
const std::ostream & [Write](#) (std::ostream &os) const

### Additional Inherited Members

### Protected Types inherited from [gdcm::DataElement](#)

- typedef [SmartPointer](#)< [Value](#) > [ValuePtr](#)

### Protected Member Functions inherited from [gdcm::DataElement](#)

- void [SetValueFieldLength](#) ([VL](#) vl, bool readvalues)

### Protected Attributes inherited from [gdcm::DataElement](#)

- [Tag](#) TagField
- [ValuePtr](#) ValueField
- [VL](#) ValueLengthField
- [VR](#) VRField

## 10.378.1 Detailed Description

Class to read/write a [DataElement](#) as Explicit Data [Element](#).

### Note

This class support 16 bits when finding an unknown [VR](#): For instance: Siemens\_CT\_Sensation64\_has\_VR\_RT.↔  
dcm

## 10.378.2 Member Function Documentation

### 10.378.2.1 GetLength()

```
VL gdcm::VR16ExplicitDataElement::GetLength ( ) const
```

### 10.378.2.2 Read()

```
template<typename TSwap >
std::istream & gdcM::VR16ExplicitDataElement::Read (
    std::istream & is )
```

### 10.378.2.3 ReadPreValue()

```
template<typename TSwap >
std::istream & gdcM::VR16ExplicitDataElement::ReadPreValue (
    std::istream & is )
```

### 10.378.2.4 ReadValue()

```
template<typename TSwap >
std::istream & gdcM::VR16ExplicitDataElement::ReadValue (
    std::istream & is,
    bool readvalues = true )
```

### 10.378.2.5 ReadWithLength()

```
template<typename TSwap >
std::istream & gdcM::VR16ExplicitDataElement::ReadWithLength (
    std::istream & is,
    VL & length )
```

The documentation for this class was generated from the following file:

- [gdcMVR16ExplicitDataElement.h](#)

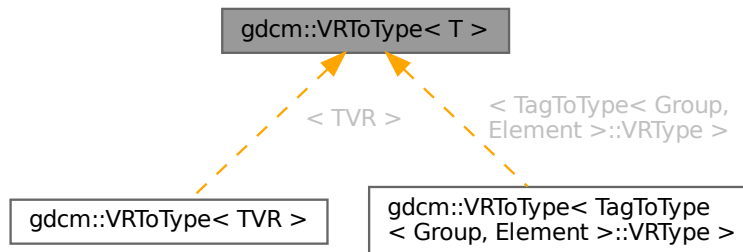
## 10.379 gdcM::VRToEncoding< T > Struct Template Reference

The documentation for this struct was generated from the following file:

- [gdcMVR.h](#)

## 10.380 gdcm::VRToType< T > Struct Template Reference

Inheritance diagram for gdcm::VRToType< T >:



### 10.380.1 Detailed Description

```
template<long long T>
struct gdcm::VRToType< T >
```

Examples

[DumpGEMSMovieGroup.cxx](#).

The documentation for this struct was generated from the following file:

- [gdcmVR.h](#)

## 10.381 gdcm::VRVLSize< T > Class Template Reference

The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

## 10.382 gdcm::VRVLSize< 0 > Class Reference

```
#include <gdcmAttribute.h>
```

### Static Public Member Functions

- static uint16\_t [Read](#) (std::istream &\_is)
- static void [Write](#) (std::ostream &os)

## 10.382.1 Member Function Documentation

### 10.382.1.1 Read()

```
static uint16_t gdcm::VRVLSIZE< 0 >::Read (  
    std::istream & _is ) [inline], [static]
```

### 10.382.1.2 Write()

```
static void gdcm::VRVLSIZE< 0 >::Write (  
    std::ostream & os ) [inline], [static]
```

The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

## 10.383 gdcm::VRVLSIZE< 1 > Class Reference

```
#include <gdcmAttribute.h>
```

### Static Public Member Functions

- static uint32\_t [Read](#) (std::istream &\_is)
- static void [Write](#) (std::ostream &os)

## 10.383.1 Member Function Documentation

### 10.383.1.1 Read()

```
static uint32_t gdcm::VRVLSIZE< 1 >::Read (  
    std::istream & _is ) [inline], [static]
```

### 10.383.1.2 Write()

```
static void gdcm::VRVLSize< 1 >::Write (
    std::ostream & os ) [inline], [static]
```

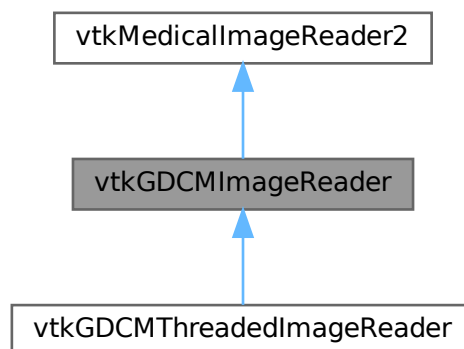
The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

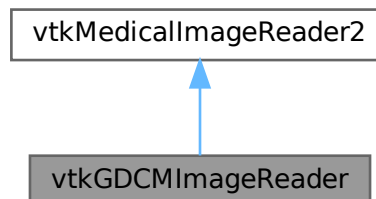
## 10.384 vtkGDCMImageReader Class Reference

```
#include <vtkGDCMImageReader.h>
```

Inheritance diagram for vtkGDCMImageReader:



Collaboration diagram for vtkGDCMImageReader:



## Public Member Functions

- virtual int [CanReadFile](#) (const char \*fname)
- virtual const char \* [GetDescriptiveName](#) ()
- virtual const char \* [GetFileExtensions](#) ()
- vtkImageData \* [GetIconImage](#) ()
- vtkImageData \* [GetOverlay](#) (int i)
- virtual void [PrintSelf](#) (ostream &os, vtkIndent indent)
- virtual void [SetCurve](#) (vtkPolyData \*pd)
- virtual void [SetFileNames](#) (vtkStringArray \*)
- virtual void [SetMedicalImageProperties](#) (vtkMedicalImageProperties \*pd)
- [vtkBooleanMacro](#) ([ApplyLookupTable](#), int)
- int [vtkBooleanMacro](#) ([ApplyYBRToRGB](#), int)
- [vtkBooleanMacro](#) ([LoadIconImage](#), int)
- [vtkBooleanMacro](#) ([LoadOverlays](#), int)
- [vtkBooleanMacro](#) ([LossyFlag](#), int)
- [vtkGetMacro](#) ([ApplyLookupTable](#), int)
- [vtkGetMacro](#) ([ApplyYBRToRGB](#), int) [vtkSetMacro](#)([ApplyYBRToRGB](#)
- [vtkGetMacro](#) ([ImageFormat](#), int)
- [vtkGetMacro](#) ([LoadIconImage](#), int)
- [vtkGetMacro](#) ([LoadOverlays](#), int)
- [vtkGetMacro](#) ([LossyFlag](#), int)
- [vtkGetMacro](#) ([NumberOfIconImages](#), int)
- [vtkGetMacro](#) ([NumberOfOverlays](#), int)
- [vtkGetMacro](#) ([PlanarConfiguration](#), int)
- [vtkGetMacro](#) ([Scale](#), double)
- [vtkGetMacro](#) ([Shift](#), double)
- [vtkGetObjectMacro](#) ([Curve](#), vtkPolyData)
- [vtkGetObjectMacro](#) ([DirectionCosines](#), vtkMatrix4x4)
- [vtkGetObjectMacro](#) ([FileNames](#), vtkStringArray)
- [vtkGetObjectMacro](#) ([MedicalImageProperties](#), vtkMedicalImageProperties)
- [vtkGetVector3Macro](#) ([ImagePositionPatient](#), double)
- [vtkGetVector6Macro](#) ([ImageOrientationPatient](#), double)
- [vtkSetMacro](#) ([ApplyLookupTable](#), int)
- [vtkSetMacro](#) ([LoadIconImage](#), int)
- [vtkSetMacro](#) ([LoadOverlays](#), int)
- [vtkSetMacro](#) ([LossyFlag](#), int)
- [vtkTypeMacro](#) ([vtkGDCMImageReader](#), vtkMedicalImageReader2)

## Static Public Member Functions

- static [vtkGDCMImageReader](#) \* [New](#) ()

**Protected Member Functions**

- [vtkGDCMImageReader](#) ()
- [~vtkGDCMImageReader](#) ()
- void [ExecuteData](#) (vtkDataObject \*out)
- void [ExecuteInformation](#) ()
- void [FillMedicalImageInformation](#) (const [gdcmm::ImageReader](#) &reader)
- int [LoadSingleFile](#) (const char \*filename, char \*pointer, unsigned long &outlen)
- int [RequestDataCompat](#) ()
- int [RequestInformationCompat](#) ()
- void [SetFilePattern](#) (const char \*)
- void [SetFilePrefix](#) (const char \*)
- [vtkGetStringMacro](#) (FilePattern)
- [vtkGetStringMacro](#) (FilePrefix)
- [vtkSetVector6Macro](#) (ImageOrientationPatient, double)

**Protected Attributes**

- int [ApplyInverseVideo](#)
- int [ApplyLookupTable](#)
- int [ApplyPlanarConfiguration](#)
- int [ApplyShiftScale](#)
- int [ApplyYBRToRGB](#)
- vtkPolyData \* [Curve](#)
- vtkMatrix4x4 \* [DirectionCosines](#)
- vtkStringArray \* [FileNames](#)
- int [ForceRescale](#)
- int [IconDataScalarType](#)
- int [IconImageDataExtent](#) [6]
- int [IconNumberOfScalarComponents](#)
- int [ImageFormat](#)
- double [ImageOrientationPatient](#) [6]
- double [ImagePositionPatient](#) [3]
- int [LoadIconImage](#)
- int [LoadOverlays](#)
- int [LossyFlag](#)
- vtkMedicalImageProperties \* [MedicalImageProperties](#)
- int [NumberOfIconImages](#)
- int [NumberOfOverlays](#)
- int [PlanarConfiguration](#)
- double [Scale](#)
- double [Shift](#)

**10.384.1 Detailed Description****Examples**

[AWTMedical3.java](#), [Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [HelloActiviz.cs](#), [HelloActiviz2.cs](#), [HelloActiviz3.cs](#), [HelloActiviz4.cs](#), [HelloActiviz5.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld.java](#), [MIPViewer.java](#), [MPRViewer.java](#), [MPRViewer2.java](#), [MagnifyFile.cxx](#), [MetalImageMD5Activiz.cs](#), [ReadSeriesIntoVTK.java](#), [RefCounting.cs](#), [gdcmmorthoplanes.cxx](#), [gdcmmreslice.cxx](#), [gdcmmtexture.cxx](#), [gdcmmvolume.cxx](#), [offscreenimage.cxx](#), and [reslicesphere.cxx](#).

## 10.384.2 Constructor & Destructor Documentation

### 10.384.2.1 vtkGDCMImageReader()

```
vtkGDCMImageReader::vtkGDCMImageReader ( ) [protected]
```

#### Examples

[HelloActiviz2.cs](#).

### 10.384.2.2 ~vtkGDCMImageReader()

```
vtkGDCMImageReader::~vtkGDCMImageReader ( ) [protected]
```

## 10.384.3 Member Function Documentation

### 10.384.3.1 CanReadFile()

```
virtual int vtkGDCMImageReader::CanReadFile (
    const char * fname ) [virtual]
```

#### Examples

[AWTMedical3.java](#), and [MetaImageMD5Activiz.cs](#).

### 10.384.3.2 ExecuteData()

```
void vtkGDCMImageReader::ExecuteData (
    vtkDataObject * out ) [protected]
```

### 10.384.3.3 ExecuteInformation()

```
void vtkGDCMImageReader::ExecuteInformation ( ) [protected]
```

### 10.384.3.4 FillMedicalImageInformation()

```
void vtkGDCMImageReader::FillMedicalImageInformation (
    const gdcmm::ImageReader & reader ) [protected]
```



### 10.384.3.5 GetDescriptiveName()

```
virtual const char * vtkGDCMImageReader::GetDescriptiveName ( ) [inline], [virtual]
```

### 10.384.3.6 GetFileExtensions()

```
virtual const char * vtkGDCMImageReader::GetFileExtensions ( ) [inline], [virtual]
```

### 10.384.3.7 GetIconImage()

```
vtkImageData * vtkGDCMImageReader::GetIconImage ( )
```

### 10.384.3.8 GetOverlay()

```
vtkImageData * vtkGDCMImageReader::GetOverlay (
    int i )
```

### 10.384.3.9 LoadSingleFile()

```
int vtkGDCMImageReader::LoadSingleFile (
    const char * filename,
    char * pointer,
    unsigned long & outlen ) [protected]
```

### 10.384.3.10 New()

```
static vtkGDCMImageReader * vtkGDCMImageReader::New ( ) [static]
```

#### Examples

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [HelloActiviz.cs](#), [HelloActiviz3.cs](#), [HelloActiviz4.cs](#), [HelloActiviz5.cs](#), [HelloVTKWorld.cs](#), [MagnifyFile.cxx](#), [MetalImageMD5Activiz.cs](#), [RefCounting.cs](#), [gdcmothoplanes.cxx](#), [gdcmlreslice.cxx](#), [gdcmltexture.cxx](#), [gdcmlvolume.cxx](#), [offscreenimage.cxx](#), and [reslicesphere.cxx](#).

### 10.384.3.11 PrintSelf()

```
virtual void vtkGDCMImageReader::PrintSelf (
    ostream & os,
    vtkIndent indent ) [virtual]
```

Reimplemented in [vtkGDCMThreadedImageReader](#).

**10.384.3.12 RequestDataCompat()**

```
int vtkGDCMImageReader::RequestDataCompat ( ) [protected]
```

**10.384.3.13 RequestInformationCompat()**

```
int vtkGDCMImageReader::RequestInformationCompat ( ) [protected]
```

**10.384.3.14 SetCurve()**

```
virtual void vtkGDCMImageReader::SetCurve (
    vtkPolyData * pd ) [virtual]
```

**10.384.3.15 SetFileNames()**

```
virtual void vtkGDCMImageReader::SetFileNames (
    vtkStringArray * ) [virtual]
```

**Examples**

[AWTMedical3.java](#), [HelloActiviz3.cs](#), [HelloActiviz4.cs](#), [HelloActiviz5.cs](#), [MIPViewer.java](#), [MPRViewer.java](#), [MPRViewer2.java](#), [ReadSeriesIntoVTK.java](#), and [gdcmonthoplanes.cxx](#).

**10.384.3.16 SetFilePattern()**

```
void vtkGDCMImageReader::SetFilePattern (
    const char * ) [inline], [protected]
```

**10.384.3.17 SetFilePrefix()**

```
void vtkGDCMImageReader::SetFilePrefix (
    const char * ) [inline], [protected]
```

**10.384.3.18 SetMedicalImageProperties()**

```
virtual void vtkGDCMImageReader::SetMedicalImageProperties (
    vtkMedicalImageProperties * pd ) [virtual]
```

**10.384.3.19 vtkBooleanMacro()** [1/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
    ApplyLookupTable ,
    int )
```

**10.384.3.20 vtkBooleanMacro()** [2/5]

```
int vtkGDCMImageReader::vtkBooleanMacro (
    ApplyYBRToRGB ,
    int )
```

**10.384.3.21 vtkBooleanMacro()** [3/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
    LoadIconImage ,
    int )
```

**10.384.3.22 vtkBooleanMacro()** [4/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
    LoadOverlays ,
    int )
```

**10.384.3.23 vtkBooleanMacro()** [5/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
    LossyFlag ,
    int )
```

**10.384.3.24 vtkGetMacro()** [1/11]

```
vtkGDCMImageReader::vtkGetMacro (
    ApplyLookupTable ,
    int )
```

**10.384.3.25 vtkGetMacro()** [2/11]

```
vtkGDCMImageReader::vtkGetMacro (
    ApplyYBRToRGB ,
    int )
```

**10.384.3.26 vtkGetMacro()** [3/11]

```
vtkGDCMImageReader::vtkGetMacro (
    ImageFormat ,
    int )
```

**10.384.3.27 vtkGetMacro()** [4/11]

```
vtkGDCMImageReader::vtkGetMacro (
    LoadIconImage ,
    int )
```

**10.384.3.28 vtkGetMacro()** [5/11]

```
vtkGDCMImageReader::vtkGetMacro (
    LoadOverlays ,
    int )
```

**10.384.3.29 vtkGetMacro()** [6/11]

```
vtkGDCMImageReader::vtkGetMacro (
    LossyFlag ,
    int )
```

**10.384.3.30 vtkGetMacro()** [7/11]

```
vtkGDCMImageReader::vtkGetMacro (
    NumberOfIconImages ,
    int )
```

**10.384.3.31 vtkGetMacro()** [8/11]

```
vtkGDCMImageReader::vtkGetMacro (
    NumberOfOverlays ,
    int )
```

**10.384.3.32 vtkGetMacro()** [9/11]

```
vtkGDCMImageReader::vtkGetMacro (
    PlanarConfiguration ,
    int )
```

**10.384.3.33 vtkGetMacro()** [10/11]

```
vtkGDCMImageReader::vtkGetMacro (
    Scale ,
    double )
```

**10.384.3.34 vtkGetMacro()** [11/11]

```
vtkGDCMImageReader::vtkGetMacro (
    Shift ,
    double )
```

**10.384.3.35 vtkGetObjectMacro()** [1/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
    Curve ,
    vtkPolyData )
```

**10.384.3.36 vtkGetObjectMacro()** [2/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
    DirectionCosines ,
    vtkMatrix4x4 )
```

**10.384.3.37 vtkGetObjectMacro()** [3/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
    FileNames ,
    vtkStringArray )
```

**10.384.3.38 vtkGetObjectMacro()** [4/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
    MedicalImageProperties ,
    vtkMedicalImageProperties )
```

**10.384.3.39 vtkGetStringMacro()** [1/2]

```
vtkGDCMImageReader::vtkGetStringMacro (
    FilePattern ) [protected]
```

**10.384.3.40 vtkGetStringMacro() [2/2]**

```
vtkGDCMImageReader::vtkGetStringMacro (
    FilePrefix ) [protected]
```

**10.384.3.41 vtkGetVector3Macro()**

```
vtkGDCMImageReader::vtkGetVector3Macro (
    ImagePositionPatient ,
    double )
```

**10.384.3.42 vtkGetVector6Macro()**

```
vtkGDCMImageReader::vtkGetVector6Macro (
    ImageOrientationPatient ,
    double )
```

**10.384.3.43 vtkSetMacro() [1/4]**

```
vtkGDCMImageReader::vtkSetMacro (
    ApplyLookupTable ,
    int )
```

**10.384.3.44 vtkSetMacro() [2/4]**

```
vtkGDCMImageReader::vtkSetMacro (
    LoadIconImage ,
    int )
```

**10.384.3.45 vtkSetMacro() [3/4]**

```
vtkGDCMImageReader::vtkSetMacro (
    LoadOverlays ,
    int )
```

**10.384.3.46 vtkSetMacro() [4/4]**

```
vtkGDCMImageReader::vtkSetMacro (
    LossyFlag ,
    int )
```

#### 10.384.3.47 vtkSetVector6Macro()

```
vtkGDCMImageReader::vtkSetVector6Macro (
    ImageOrientationPatient ,
    double ) [protected]
```

#### 10.384.3.48 vtkTypeMacro()

```
vtkGDCMImageReader::vtkTypeMacro (
    vtkGDCMImageReader ,
    vtkMedicalImageReader2 )
```

### 10.384.4 Member Data Documentation

#### 10.384.4.1 ApplyInverseVideo

```
int vtkGDCMImageReader::ApplyInverseVideo [protected]
```

#### 10.384.4.2 ApplyLookupTable

```
int vtkGDCMImageReader::ApplyLookupTable [protected]
```

#### 10.384.4.3 ApplyPlanarConfiguration

```
int vtkGDCMImageReader::ApplyPlanarConfiguration [protected]
```

#### 10.384.4.4 ApplyShiftScale

```
int vtkGDCMImageReader::ApplyShiftScale [protected]
```

#### 10.384.4.5 ApplyYBRToRGB

```
int vtkGDCMImageReader::ApplyYBRToRGB [protected]
```

#### 10.384.4.6 Curve

```
vtkPolyData* vtkGDCMImageReader::Curve [protected]
```

**10.384.4.7 DirectionCosines**

`vtkMatrix4x4* vtkGDCMImageReader::DirectionCosines` [protected]

**10.384.4.8 FileNames**

`vtkStringArray* vtkGDCMImageReader::FileNames` [protected]

**10.384.4.9 ForceRescale**

`int vtkGDCMImageReader::ForceRescale` [protected]

**10.384.4.10 IconDataScalarType**

`int vtkGDCMImageReader::IconDataScalarType` [protected]

**10.384.4.11 IconImageDataExtent**

`int vtkGDCMImageReader::IconImageDataExtent[6]` [protected]

**10.384.4.12 IconNumberOfScalarComponents**

`int vtkGDCMImageReader::IconNumberOfScalarComponents` [protected]

**10.384.4.13 ImageFormat**

`int vtkGDCMImageReader::ImageFormat` [protected]

**10.384.4.14 ImageOrientationPatient**

`double vtkGDCMImageReader::ImageOrientationPatient[6]` [protected]

**10.384.4.15 ImagePositionPatient**

`double vtkGDCMImageReader::ImagePositionPatient[3]` [protected]

**10.384.4.16 LoadIconImage**

`int vtkGDCMImageReader::LoadIconImage` [protected]



#### 10.384.4.17 LoadOverlays

```
int vtkGDCMImageReader::LoadOverlays [protected]
```

#### 10.384.4.18 LossyFlag

```
int vtkGDCMImageReader::LossyFlag [protected]
```

#### 10.384.4.19 MedicalImageProperties

```
vtkMedicalImageProperties* vtkGDCMImageReader::MedicalImageProperties [protected]
```

#### 10.384.4.20 NumberOfIconImages

```
int vtkGDCMImageReader::NumberOfIconImages [protected]
```

#### 10.384.4.21 NumberOfOverlays

```
int vtkGDCMImageReader::NumberOfOverlays [protected]
```

#### 10.384.4.22 PlanarConfiguration

```
int vtkGDCMImageReader::PlanarConfiguration [protected]
```

#### 10.384.4.23 Scale

```
double vtkGDCMImageReader::Scale [protected]
```

#### 10.384.4.24 Shift

```
double vtkGDCMImageReader::Shift [protected]
```

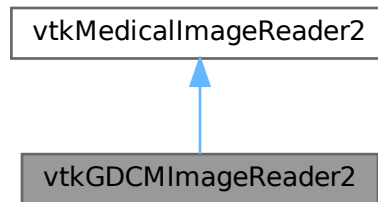
The documentation for this class was generated from the following file:

- [vtkGDCMImageReader.h](#)

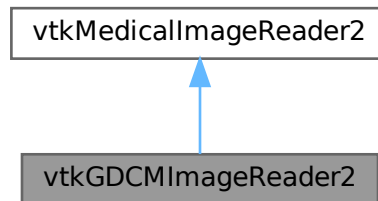
## 10.385 vtkGDCMImageReader2 Class Reference

```
#include <vtkGDCMImageReader2.h>
```

Inheritance diagram for vtkGDCMImageReader2:



Collaboration diagram for vtkGDCMImageReader2:



### Public Member Functions

- virtual int [CanReadFile](#) (const char \*fname)
- virtual const char \* [GetDescriptiveName](#) ()
- virtual const char \* [GetFileExtensions](#) ()
- vtkImageData \* [GetIconImage](#) ()
- vtkAlgorithmOutput \* [GetIconImagePort](#) ()
- vtkImageData \* [GetOverlay](#) (int i)
- vtkAlgorithmOutput \* [GetOverlayPort](#) (int index)
- virtual void [PrintSelf](#) (ostream &os, vtkIndent indent)
- virtual void [SetCurve](#) (vtkPolyData \*pd)
- virtual void [SetMedicalImageProperties](#) (vtkMedicalImageProperties \*pd)
- [vtkBooleanMacro](#) ([ApplyLookupTable](#), int)

- [int vtkBooleanMacro \(ApplyYBRToRGB, int\)](#)
- [vtkBooleanMacro \(LoadIconImage, int\)](#)
- [vtkBooleanMacro \(LoadOverlays, int\)](#)
- [vtkBooleanMacro \(LossyFlag, int\)](#)
- [vtkGetMacro \(ApplyLookupTable, int\)](#)
- [vtkGetMacro \(ApplyYBRToRGB, int\) vtkSetMacro \(ApplyYBRToRGB](#)
- [vtkGetMacro \(ImageFormat, int\)](#)
- [vtkGetMacro \(LoadIconImage, int\)](#)
- [vtkGetMacro \(LoadOverlays, int\)](#)
- [vtkGetMacro \(LossyFlag, int\)](#)
- [vtkGetMacro \(NumberOfIconImages, int\)](#)
- [vtkGetMacro \(NumberOfOverlays, int\)](#)
- [vtkGetMacro \(PlanarConfiguration, int\)](#)
- [vtkGetMacro \(Scale, double\)](#)
- [vtkGetMacro \(Shift, double\)](#)
- [vtkGetObjectMacro \(Curve, vtkPolyData\)](#)
- [vtkGetObjectMacro \(DirectionCosines, vtkMatrix4x4\)](#)
- [vtkGetVector3Macro \(ImagePositionPatient, double\)](#)
- [vtkGetVector6Macro \(ImageOrientationPatient, double\)](#)
- [vtkSetMacro \(ApplyLookupTable, int\)](#)
- [vtkSetMacro \(LoadIconImage, int\)](#)
- [vtkSetMacro \(LoadOverlays, int\)](#)
- [vtkSetMacro \(LossyFlag, int\)](#)
- [vtkTypeMacro \(vtkGDCMImageReader2, vtkMedicalImageReader2\)](#)

### Static Public Member Functions

- static [vtkGDCMImageReader2 \\* New \(\)](#)

### Protected Member Functions

- [vtkGDCMImageReader2 \(\)](#)
- [~vtkGDCMImageReader2 \(\)](#)
- void [FillMedicalImageInformation](#) (const [gdcml::ImageReader](#) &reader)
- int [LoadSingleFile](#) (const char \*filename, char \*pointer, unsigned long &outlen)
- int [ProcessRequest](#) (vtkInformation \*request, vtkInformationVector \*\*inputVector, vtkInformationVector \*output↵  
Vector)
- int [RequestData](#) (vtkInformation \*request, vtkInformationVector \*\*inputVector, vtkInformationVector \*output↵  
Vector)
- int [RequestDataCompat](#) ()
- int [RequestInformation](#) (vtkInformation \*request, vtkInformationVector \*\*inputVector, vtkInformationVector  
\*outputVector)
- int [RequestInformationCompat](#) ()
- void [SetFilePattern](#) (const char \*)
- void [SetFilePrefix](#) (const char \*)
- [vtkGetStringMacro](#) (FilePattern)
- [vtkGetStringMacro](#) (FilePrefix)
- [vtkSetVector6Macro](#) (ImageOrientationPatient, double)

## Protected Attributes

- int [ApplyInverseVideo](#)
- int [ApplyLookupTable](#)
- int [ApplyPlanarConfiguration](#)
- int [ApplyShiftScale](#)
- int [ApplyYBRToRGB](#)
- vtkPolyData \* [Curve](#)
- vtkMatrix4x4 \* [DirectionCosines](#)
- int [ForceRescale](#)
- int [IconDataScalarType](#)
- int [IconImageDataExtent](#) [6]
- int [IconNumberOfScalarComponents](#)
- int [ImageFormat](#)
- double [ImageOrientationPatient](#) [6]
- double [ImagePositionPatient](#) [3]
- int [LoadIconImage](#)
- int [LoadOverlays](#)
- int [LossyFlag](#)
- int [NumberOfIconImages](#)
- int [NumberOfOverlays](#)
- int [PlanarConfiguration](#)
- double [Scale](#)
- double [Shift](#)

## 10.385.1 Detailed Description

### Examples

[Compute3DSpacing.cxx](#).

## 10.385.2 Constructor & Destructor Documentation

### 10.385.2.1 vtkGDCMImageReader2()

```
vtkGDCMImageReader2::vtkGDCMImageReader2 ( ) [protected]
```

### 10.385.2.2 ~vtkGDCMImageReader2()

```
vtkGDCMImageReader2::~~vtkGDCMImageReader2 ( ) [protected]
```

## 10.385.3 Member Function Documentation

### 10.385.3.1 CanReadFile()

```
virtual int vtkGDCMImageReader2::CanReadFile (
    const char * fname ) [virtual]
```

### 10.385.3.2 FillMedicalImageInformation()

```
void vtkGDCMImageReader2::FillMedicalImageInformation (
    const gdcm::ImageReader & reader ) [protected]
```

### 10.385.3.3 GetDescriptiveName()

```
virtual const char * vtkGDCMImageReader2::GetDescriptiveName ( ) [inline], [virtual]
```

### 10.385.3.4 GetFileExtensions()

```
virtual const char * vtkGDCMImageReader2::GetFileExtensions ( ) [inline], [virtual]
```

### 10.385.3.5 GetIconImage()

```
vtkImageData * vtkGDCMImageReader2::GetIconImage ( )
```

### 10.385.3.6 GetIconImagePort()

```
vtkAlgorithmOutput * vtkGDCMImageReader2::GetIconImagePort ( )
```

### 10.385.3.7 GetOverlay()

```
vtkImageData * vtkGDCMImageReader2::GetOverlay (
    int i )
```

### 10.385.3.8 GetOverlayPort()

```
vtkAlgorithmOutput * vtkGDCMImageReader2::GetOverlayPort (
    int index )
```

### 10.385.3.9 LoadSingleFile()

```
int vtkGDCMImageReader2::LoadSingleFile (
    const char * filename,
    char * pointer,
    unsigned long & outlen ) [protected]
```

#### 10.385.3.10 New()

```
static vtkGDCMImageReader2 * vtkGDCMImageReader2::New ( ) [static]
```

#### Examples

[Compute3DSpacing.cxx](#).

#### 10.385.3.11 PrintSelf()

```
virtual void vtkGDCMImageReader2::PrintSelf (
    ostream & os,
    vtkIndent indent ) [virtual]
```

#### 10.385.3.12 ProcessRequest()

```
int vtkGDCMImageReader2::ProcessRequest (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector ) [protected]
```

#### 10.385.3.13 RequestData()

```
int vtkGDCMImageReader2::RequestData (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector ) [protected]
```

#### 10.385.3.14 RequestDataCompat()

```
int vtkGDCMImageReader2::RequestDataCompat ( ) [protected]
```

#### 10.385.3.15 RequestInformation()

```
int vtkGDCMImageReader2::RequestInformation (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector ) [protected]
```

#### 10.385.3.16 RequestInformationCompat()

```
int vtkGDCMImageReader2::RequestInformationCompat ( ) [protected]
```

**10.385.3.17 SetCurve()**

```
virtual void vtkGDCMImageReader2::SetCurve (
    vtkPolyData * pd ) [virtual]
```

**10.385.3.18 SetFilePattern()**

```
void vtkGDCMImageReader2::SetFilePattern (
    const char * ) [inline], [protected]
```

**10.385.3.19 SetFilePrefix()**

```
void vtkGDCMImageReader2::SetFilePrefix (
    const char * ) [inline], [protected]
```

**10.385.3.20 SetMedicalImageProperties()**

```
virtual void vtkGDCMImageReader2::SetMedicalImageProperties (
    vtkMedicalImageProperties * pd ) [virtual]
```

**10.385.3.21 vtkBooleanMacro() [1/5]**

```
vtkGDCMImageReader2::vtkBooleanMacro (
    ApplyLookupTable ,
    int )
```

**10.385.3.22 vtkBooleanMacro() [2/5]**

```
int vtkGDCMImageReader2::vtkBooleanMacro (
    ApplyYBRToRGB ,
    int )
```

**10.385.3.23 vtkBooleanMacro() [3/5]**

```
vtkGDCMImageReader2::vtkBooleanMacro (
    LoadIconImage ,
    int )
```

**10.385.3.24 vtkBooleanMacro() [4/5]**

```
vtkGDCMImageReader2::vtkBooleanMacro (
    LoadOverlays ,
    int )
```

**10.385.3.25 vtkBooleanMacro()** [5/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (
    LossyFlag ,
    int )
```

**10.385.3.26 vtkGetMacro()** [1/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    ApplyLookupTable ,
    int )
```

**10.385.3.27 vtkGetMacro()** [2/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    ApplyYBRToRGB ,
    int )
```

**10.385.3.28 vtkGetMacro()** [3/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    ImageFormat ,
    int )
```

**10.385.3.29 vtkGetMacro()** [4/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    LoadIconImage ,
    int )
```

**10.385.3.30 vtkGetMacro()** [5/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    LoadOverlays ,
    int )
```

**10.385.3.31 vtkGetMacro()** [6/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    LossyFlag ,
    int )
```



**10.385.3.32 vtkGetMacro()** [7/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    NumberOfIconImages ,
    int )
```

**10.385.3.33 vtkGetMacro()** [8/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    NumberOfOverlays ,
    int )
```

**10.385.3.34 vtkGetMacro()** [9/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    PlanarConfiguration ,
    int )
```

**10.385.3.35 vtkGetMacro()** [10/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    Scale ,
    double )
```

**10.385.3.36 vtkGetMacro()** [11/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    Shift ,
    double )
```

**10.385.3.37 vtkGetObjectMacro()** [1/2]

```
vtkGDCMImageReader2::vtkGetObjectMacro (
    Curve ,
    vtkPolyData )
```

**10.385.3.38 vtkGetObjectMacro()** [2/2]

```
vtkGDCMImageReader2::vtkGetObjectMacro (
    DirectionCosines ,
    vtkMatrix4x4 )
```

**10.385.3.39 vtkGetStringMacro() [1/2]**

```
vtkGDCMImageReader2::vtkGetStringMacro (
    FilePattern ) [protected]
```

**10.385.3.40 vtkGetStringMacro() [2/2]**

```
vtkGDCMImageReader2::vtkGetStringMacro (
    FilePrefix ) [protected]
```

**10.385.3.41 vtkGetVector3Macro()**

```
vtkGDCMImageReader2::vtkGetVector3Macro (
    ImagePositionPatient ,
    double )
```

**10.385.3.42 vtkGetVector6Macro()**

```
vtkGDCMImageReader2::vtkGetVector6Macro (
    ImageOrientationPatient ,
    double )
```

**10.385.3.43 vtkSetMacro() [1/4]**

```
vtkGDCMImageReader2::vtkSetMacro (
    ApplyLookupTable ,
    int )
```

**10.385.3.44 vtkSetMacro() [2/4]**

```
vtkGDCMImageReader2::vtkSetMacro (
    LoadIconImage ,
    int )
```

**10.385.3.45 vtkSetMacro() [3/4]**

```
vtkGDCMImageReader2::vtkSetMacro (
    LoadOverlays ,
    int )
```

**10.385.3.46 vtkSetMacro() [4/4]**

```
vtkGDCMImageReader2::vtkSetMacro (
    LossyFlag ,
    int )
```

**10.385.3.47 vtkSetVector6Macro()**

```
vtkGDCMImageReader2::vtkSetVector6Macro (
    ImageOrientationPatient ,
    double ) [protected]
```

**10.385.3.48 vtkTypeMacro()**

```
vtkGDCMImageReader2::vtkTypeMacro (
    vtkGDCMImageReader2 ,
    vtkMedicalImageReader2 )
```

**10.385.4 Member Data Documentation****10.385.4.1 ApplyInverseVideo**

```
int vtkGDCMImageReader2::ApplyInverseVideo [protected]
```

**10.385.4.2 ApplyLookupTable**

```
int vtkGDCMImageReader2::ApplyLookupTable [protected]
```

**10.385.4.3 ApplyPlanarConfiguration**

```
int vtkGDCMImageReader2::ApplyPlanarConfiguration [protected]
```

**10.385.4.4 ApplyShiftScale**

```
int vtkGDCMImageReader2::ApplyShiftScale [protected]
```

**10.385.4.5 ApplyYBRToRGB**

```
int vtkGDCMImageReader2::ApplyYBRToRGB [protected]
```

**10.385.4.6 Curve**

`vtkPolyData* vtkGDCMImageReader2::Curve` [protected]

**10.385.4.7 DirectionCosines**

`vtkMatrix4x4* vtkGDCMImageReader2::DirectionCosines` [protected]

**10.385.4.8 ForceRescale**

`int vtkGDCMImageReader2::ForceRescale` [protected]

**10.385.4.9 IconDataScalarType**

`int vtkGDCMImageReader2::IconDataScalarType` [protected]

**10.385.4.10 IconImageDataExtent**

`int vtkGDCMImageReader2::IconImageDataExtent[6]` [protected]

**10.385.4.11 IconNumberOfScalarComponents**

`int vtkGDCMImageReader2::IconNumberOfScalarComponents` [protected]

**10.385.4.12 ImageFormat**

`int vtkGDCMImageReader2::ImageFormat` [protected]

**10.385.4.13 ImageOrientationPatient**

`double vtkGDCMImageReader2::ImageOrientationPatient[6]` [protected]

**10.385.4.14 ImagePositionPatient**

`double vtkGDCMImageReader2::ImagePositionPatient[3]` [protected]

**10.385.4.15 LoadIconImage**

`int vtkGDCMImageReader2::LoadIconImage` [protected]

#### 10.385.4.16 LoadOverlays

```
int vtkGDCMImageReader2::LoadOverlays [protected]
```

#### 10.385.4.17 LossyFlag

```
int vtkGDCMImageReader2::LossyFlag [protected]
```

#### 10.385.4.18 NumberOfIconImages

```
int vtkGDCMImageReader2::NumberOfIconImages [protected]
```

#### 10.385.4.19 NumberOfOverlays

```
int vtkGDCMImageReader2::NumberOfOverlays [protected]
```

#### 10.385.4.20 PlanarConfiguration

```
int vtkGDCMImageReader2::PlanarConfiguration [protected]
```

#### 10.385.4.21 Scale

```
double vtkGDCMImageReader2::Scale [protected]
```

#### 10.385.4.22 Shift

```
double vtkGDCMImageReader2::Shift [protected]
```

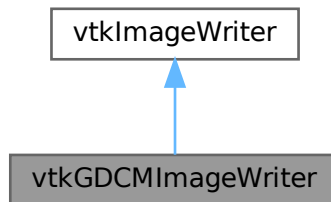
The documentation for this class was generated from the following file:

- [vtkGDCMImageReader2.h](#)

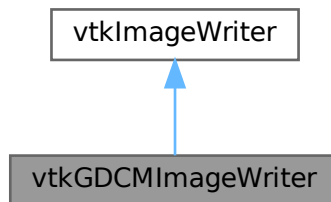
## 10.386 vtkGDCMImageWriter Class Reference

```
#include <vtkGDCMImageWriter.h>
```

Inheritance diagram for vtkGDCMImageWriter:



Collaboration diagram for vtkGDCMImageWriter:



### Public Types

- enum [CompressionTypes](#) {  
    [NO\\_COMPRESSION](#) = 0 ,  
    [JPEG\\_COMPRESSION](#) ,  
    [JPEG2000\\_COMPRESSION](#) ,  
    [JPEGLS\\_COMPRESSION](#) ,  
    [RLE\\_COMPRESSION](#) }

**Public Member Functions**

- virtual const char \* [GetDescriptiveName](#) ()
- virtual const char \* [GetFileExtensions](#) ()
- virtual void [PrintSelf](#) (ostream &os, vtkIndent indent)
- virtual void [SetDirectionCosines](#) (vtkMatrix4x4 \*matrix)
- virtual void [SetDirectionCosinesFromImageOrientationPatient](#) (const double dircos[6])
- virtual void [SetFileNames](#) (vtkStringArray \*)
- virtual void [SetMedicalImageProperties](#) (vtkMedicalImageProperties \*)
- [vtkBooleanMacro](#) (FileLowerLeft, int)
- [vtkBooleanMacro](#) (LossyFlag, int)
- [vtkGetMacro](#) (CompressionType, int)
- [vtkGetMacro](#) (FileLowerLeft, int)
- [vtkGetMacro](#) (ImageFormat, int)
- [vtkGetMacro](#) (LossyFlag, int)
- [vtkGetMacro](#) (PlanarConfiguration, int)
- [vtkGetMacro](#) (Scale, double)
- [vtkGetMacro](#) (Shift, double)
- [vtkGetObjectMacro](#) (DirectionCosines, vtkMatrix4x4)
- [vtkGetObjectMacro](#) (FileNames, vtkStringArray)
- [vtkGetObjectMacro](#) (MedicalImageProperties, vtkMedicalImageProperties)
- [vtkGetStringMacro](#) (SeriesUID)
- [vtkGetStringMacro](#) (StudyUID)
- [vtkSetMacro](#) (CompressionType, int)
- [vtkSetMacro](#) (FileLowerLeft, int)
- [vtkSetMacro](#) (ImageFormat, int)
- [vtkSetMacro](#) (LossyFlag, int)
- [vtkSetMacro](#) (PlanarConfiguration, int)
- [vtkSetMacro](#) (Scale, double)
- [vtkSetMacro](#) (Shift, double)
- [vtkSetStringMacro](#) (SeriesUID)
- [vtkSetStringMacro](#) (StudyUID)
- [vtkTypeMacro](#) (vtkGDCMImageWriter, vtkImageWriter)
- virtual void [Write](#) ()

**Static Public Member Functions**

- static [vtkGDCMImageWriter](#) \* [New](#) ()

**Protected Member Functions**

- [vtkGDCMImageWriter](#) ()
- [~vtkGDCMImageWriter](#) ()
- virtual char \* [GetFileName](#) ()
- int [WriteGDCMData](#) (vtkImageData \*data, int timeStep)
- void [WriteSlice](#) (vtkImageData \*data)

## 10.386.1 Detailed Description

### Examples

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [CreateFakePET.cxx](#), [CreateFakeRTDOSE.cxx](#), [HelloActiviz.cs](#), [HelloActiviz2.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld.java](#), [HelloVTKWorld2.cs](#), [MagnifyFile.cxx](#), [RefCounting.cs](#), and [gdcmmorthoplanes.cxx](#).

## 10.386.2 Member Enumeration Documentation

### 10.386.2.1 CompressionTypes

```
enum vtkGDCMImageWriter::CompressionTypes
```

#### Enumerator

NO_COMPRESSION	
JPEG_COMPRESSION	
JPEG2000_COMPRESSION	
JPEGLS_COMPRESSION	
RLE_COMPRESSION	

## 10.386.3 Constructor & Destructor Documentation

### 10.386.3.1 vtkGDCMImageWriter()

```
vtkGDCMImageWriter::vtkGDCMImageWriter ( ) [protected]
```

### 10.386.3.2 ~vtkGDCMImageWriter()

```
vtkGDCMImageWriter::~~vtkGDCMImageWriter ( ) [protected]
```

## 10.386.4 Member Function Documentation

### 10.386.4.1 GetDescriptiveName()

```
virtual const char * vtkGDCMImageWriter::GetDescriptiveName ( ) [inline], [virtual]
```

### 10.386.4.2 GetFileExtensions()

```
virtual const char * vtkGDCMImageWriter::GetFileExtensions ( ) [inline], [virtual]
```



### 10.386.4.3 GetFileName()

```
virtual char * vtkGDCMImageWriter::GetFileName ( ) [protected], [virtual]
```

### 10.386.4.4 New()

```
static vtkGDCMImageWriter * vtkGDCMImageWriter::New ( ) [static]
```

#### Examples

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [CreateFakePET.cxx](#), [CreateFakeRTDOSE.cxx](#), [HelloActiviz.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld2.cs](#), [MagnifyFile.cxx](#), [RefCounting.cs](#), and [gdcmorphoplanes.cxx](#).

### 10.386.4.5 PrintSelf()

```
virtual void vtkGDCMImageWriter::PrintSelf (
    ostream & os,
    vtkIndent indent ) [virtual]
```

### 10.386.4.6 SetDirectionCosines()

```
virtual void vtkGDCMImageWriter::SetDirectionCosines (
    vtkMatrix4x4 * matrix ) [virtual]
```

#### Examples

[Convert16BitsTo8Bits.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [HelloActiviz2.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld.java](#), [MagnifyFile.cxx](#), and [gdcmorphoplanes.cxx](#).

### 10.386.4.7 SetDirectionCosinesFromImageOrientationPatient()

```
virtual void vtkGDCMImageWriter::SetDirectionCosinesFromImageOrientationPatient (
    const double dircos[6] ) [virtual]
```

### 10.386.4.8 SetFileNames()

```
virtual void vtkGDCMImageWriter::SetFileNames (
    vtkStringArray * ) [virtual]
```

#### Examples

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

#### 10.386.4.9 SetMedicalImageProperties()

```
virtual void vtkGDCMImageWriter::SetMedicalImageProperties (
    vtkMedicalImageProperties * ) [virtual]
```

##### Examples

[Convert16BitsTo8Bits.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [HelloActiviz.cs](#), [HelloActiviz2.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld.java](#), [MagnifyFile.cxx](#), and [gdcmorphoplanes.cxx](#).

#### 10.386.4.10 vtkBooleanMacro() [1/2]

```
vtkGDCMImageWriter::vtkBooleanMacro (
    FileLowerLeft ,
    int )
```

#### 10.386.4.11 vtkBooleanMacro() [2/2]

```
vtkGDCMImageWriter::vtkBooleanMacro (
    LossyFlag ,
    int )
```

#### 10.386.4.12 vtkGetMacro() [1/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    CompressionType ,
    int )
```

#### 10.386.4.13 vtkGetMacro() [2/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    FileLowerLeft ,
    int )
```

#### 10.386.4.14 vtkGetMacro() [3/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    ImageFormat ,
    int )
```

**10.386.4.15 vtkGetMacro()** [4/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    LossyFlag ,
    int )
```

**10.386.4.16 vtkGetMacro()** [5/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    PlanarConfiguration ,
    int )
```

**10.386.4.17 vtkGetMacro()** [6/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    Scale ,
    double )
```

**10.386.4.18 vtkGetMacro()** [7/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    Shift ,
    double )
```

**10.386.4.19 vtkGetObjectMacro()** [1/3]

```
vtkGDCMImageWriter::vtkGetObjectMacro (
    DirectionCosines ,
    vtkMatrix4x4 )
```

**10.386.4.20 vtkGetObjectMacro()** [2/3]

```
vtkGDCMImageWriter::vtkGetObjectMacro (
    FileNames ,
    vtkStringArray )
```

**10.386.4.21 vtkGetObjectMacro()** [3/3]

```
vtkGDCMImageWriter::vtkGetObjectMacro (
    MedicalImageProperties ,
    vtkMedicalImageProperties )
```

**10.386.4.22 vtkGetStringMacro() [1/2]**

```
vtkGDCMImageWriter::vtkGetStringMacro (
    SeriesUID )
```

**10.386.4.23 vtkGetStringMacro() [2/2]**

```
vtkGDCMImageWriter::vtkGetStringMacro (
    StudyUID )
```

**10.386.4.24 vtkSetMacro() [1/7]**

```
vtkGDCMImageWriter::vtkSetMacro (
    CompressionType ,
    int )
```

**10.386.4.25 vtkSetMacro() [2/7]**

```
vtkGDCMImageWriter::vtkSetMacro (
    FileLowerLeft ,
    int )
```

**10.386.4.26 vtkSetMacro() [3/7]**

```
vtkGDCMImageWriter::vtkSetMacro (
    ImageFormat ,
    int )
```

**10.386.4.27 vtkSetMacro() [4/7]**

```
vtkGDCMImageWriter::vtkSetMacro (
    LossyFlag ,
    int )
```

**10.386.4.28 vtkSetMacro() [5/7]**

```
vtkGDCMImageWriter::vtkSetMacro (
    PlanarConfiguration ,
    int )
```

**10.386.4.29 vtkSetMacro() [6/7]**

```
vtkGDCMImageWriter::vtkSetMacro (
    Scale ,
    double )
```

**10.386.4.30 vtkSetMacro() [7/7]**

```
vtkGDCMImageWriter::vtkSetMacro (
    Shift ,
    double )
```

**10.386.4.31 vtkSetStringMacro() [1/2]**

```
vtkGDCMImageWriter::vtkSetStringMacro (
    SeriesUID )
```

**10.386.4.32 vtkSetStringMacro() [2/2]**

```
vtkGDCMImageWriter::vtkSetStringMacro (
    StudyUID )
```

**10.386.4.33 vtkTypeMacro()**

```
vtkGDCMImageWriter::vtkTypeMacro (
    vtkGDCMImageWriter ,
    vtkImageWriter )
```

**10.386.4.34 Write()**

```
virtual void vtkGDCMImageWriter::Write ( ) [virtual]
```

**Examples**

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [CreateFakePET.cxx](#), [CreateFakeRTDOSE.cxx](#), [HelloActiviz.cs](#), [HelloActiviz2.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld.java](#), [HelloVTKWorld2.cs](#), [MagnifyFile.cxx](#), and [gdcmorthoplanes.cxx](#).

**10.386.4.35 WriteGDCMData()**

```
int vtkGDCMImageWriter::WriteGDCMData (
    vtkImageData * data,
    int timeStep ) [protected]
```

### 10.386.4.36 WriteSlice()

```
void vtkGDCMImageWriter::WriteSlice (
    vtkImageData * data ) [protected]
```

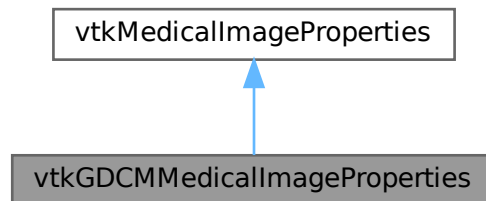
The documentation for this class was generated from the following file:

- [vtkGDCMImageWriter.h](#)

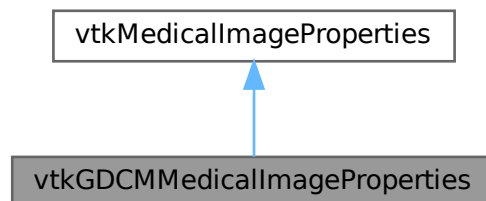
## 10.387 vtkGDCMMedicalImageProperties Class Reference

```
#include <vtkGDCMMedicalImageProperties.h>
```

Inheritance diagram for vtkGDCMMedicalImageProperties:



Collaboration diagram for vtkGDCMMedicalImageProperties:



**Public Member Functions**

- virtual void [Clear](#) ()
- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkTypeMacro](#) ([vtkGDCMMedicalImageProperties](#), vtkMedicalImageProperties)

**Static Public Member Functions**

- static [vtkGDCMMedicalImageProperties](#) \* [New](#) ()

**Protected Member Functions**

- [vtkGDCMMedicalImageProperties](#) ()
- [~vtkGDCMMedicalImageProperties](#) ()
- [gdcmm::File](#) const & [GetFile](#) (unsigned int t)
- void [PushBackFile](#) ([gdcmm::File](#) const &f)

**Friends**

- class [vtkGDCMImageReader](#)
- class [vtkGDCMImageReader2](#)
- class [vtkGDCMImageWriter](#)

**10.387.1 Constructor & Destructor Documentation****10.387.1.1 [vtkGDCMMedicalImageProperties\(\)](#)**

```
vtkGDCMMedicalImageProperties::vtkGDCMMedicalImageProperties ( ) [protected]
```

**10.387.1.2 [~vtkGDCMMedicalImageProperties\(\)](#)**

```
vtkGDCMMedicalImageProperties::~~vtkGDCMMedicalImageProperties ( ) [protected]
```

**10.387.2 Member Function Documentation****10.387.2.1 [Clear\(\)](#)**

```
virtual void vtkGDCMMedicalImageProperties::Clear ( ) [virtual]
```

**10.387.2.2 [GetFile\(\)](#)**

```
gdcmm::File const & vtkGDCMMedicalImageProperties::GetFile (
    unsigned int t ) [protected]
```

### 10.387.2.3 New()

```
static vtkGDCMMedicalImageProperties * vtkGDCMMedicalImageProperties::New ( ) [static]
```

### 10.387.2.4 PrintSelf()

```
void vtkGDCMMedicalImageProperties::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

### 10.387.2.5 PushBackFile()

```
void vtkGDCMMedicalImageProperties::PushBackFile (
    gdcmm::File const & f ) [protected]
```

### 10.387.2.6 vtkTypeMacro()

```
vtkGDCMMedicalImageProperties::vtkTypeMacro (
    vtkGDCMMedicalImageProperties ,
    vtkMedicalImageProperties )
```

## 10.387.3 Friends And Related Symbol Documentation

### 10.387.3.1 vtkGDCMImageReader

```
friend class vtkGDCMImageReader [friend]
```

### 10.387.3.2 vtkGDCMImageReader2

```
friend class vtkGDCMImageReader2 [friend]
```

### 10.387.3.3 vtkGDCMImageWriter

```
friend class vtkGDCMImageWriter [friend]
```

The documentation for this class was generated from the following file:

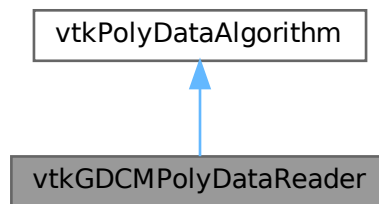
- [vtkGDCMMedicalImageProperties.h](#)



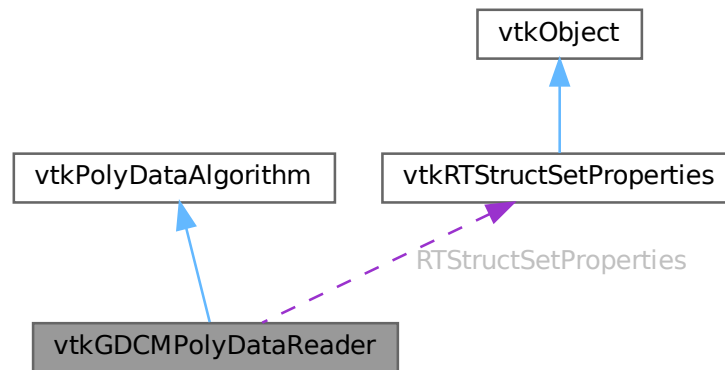
## 10.388 vtkGDCMPolyDataReader Class Reference

```
#include <vtkGDCMPolyDataReader.h>
```

Inheritance diagram for vtkGDCMPolyDataReader:



Collaboration diagram for vtkGDCMPolyDataReader:



### Public Member Functions

- virtual void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkGetObjectMacro](#) ([MedicalImageProperties](#), [vtkMedicalImageProperties](#))
- [vtkGetObjectMacro](#) ([RTStructSetProperties](#), [vtkRTStructSetProperties](#))
- [vtkGetStringMacro](#) ([FileName](#))
- [vtkSetStringMacro](#) ([FileName](#))
- [vtkTypeMacro](#) ([vtkGDCMPolyDataReader](#), [vtkPolyDataAlgorithm](#))

## Static Public Member Functions

- static [vtkGDCMPolyDataReader](#) \* [New](#) ()

## Protected Member Functions

- [vtkGDCMPolyDataReader](#) ()
- [~vtkGDCMPolyDataReader](#) ()
- void [FillMedicalImageInformation](#) (const [gdcm::Reader](#) &reader)
- int [RequestData](#) (vtkInformation \*, vtkInformationVector \*\*, vtkInformationVector \*)
- int [RequestData\\_HemodynamicWaveformStorage](#) ([gdcm::Reader](#) const &reader, vtkInformationVector \*outputVector)
- int [RequestData\\_RTStructureSetStorage](#) ([gdcm::Reader](#) const &reader, vtkInformationVector \*outputVector)
- int [RequestInformation](#) (vtkInformation \*vtkNotUsed(request), vtkInformationVector \*\*vtkNotUsed(inputVector), vtkInformationVector \*outputVector)
- int [RequestInformation\\_HemodynamicWaveformStorage](#) ([gdcm::Reader](#) const &reader)
- int [RequestInformation\\_RTStructureSetStorage](#) ([gdcm::Reader](#) const &reader)

## Protected Attributes

- char \* [FileName](#)
- vtkMedicalImageProperties \* [MedicalImageProperties](#)
- [vtkRTStructSetProperties](#) \* [RTStructSetProperties](#)

## 10.388.1 Detailed Description

### Examples

[GenerateRTSTRUCT.cxx](#), [gdcmscene.cxx](#), and [rtstructapp.cxx](#).

## 10.388.2 Constructor & Destructor Documentation

### 10.388.2.1 [vtkGDCMPolyDataReader](#)()

```
vtkGDCMPolyDataReader::vtkGDCMPolyDataReader ( ) [protected]
```

### 10.388.2.2 [~vtkGDCMPolyDataReader](#)()

```
vtkGDCMPolyDataReader::~~vtkGDCMPolyDataReader ( ) [protected]
```

## 10.388.3 Member Function Documentation

### 10.388.3.1 [FillMedicalImageInformation](#)()

```
void vtkGDCMPolyDataReader::FillMedicalImageInformation (
    const gdcm::Reader & reader ) [protected]
```

**10.388.3.2 New()**

```
static vtkGDCMPolyDataReader * vtkGDCMPolyDataReader::New ( ) [static]
```

**Examples**

[GenerateRTSTRUCT.cxx](#), [gdcmscene.cxx](#), and [rtstructapp.cxx](#).

**10.388.3.3 PrintSelf()**

```
virtual void vtkGDCMPolyDataReader::PrintSelf (
    ostream & os,
    vtkIndent indent ) [virtual]
```

**10.388.3.4 RequestData()**

```
int vtkGDCMPolyDataReader::RequestData (
    vtkInformation * ,
    vtkInformationVector ** ,
    vtkInformationVector * ) [protected]
```

**10.388.3.5 RequestData\_HemodynamicWaveformStorage()**

```
int vtkGDCMPolyDataReader::RequestData_HemodynamicWaveformStorage (
    gdcmm::Reader const & reader,
    vtkInformationVector * outputVector ) [protected]
```

**10.388.3.6 RequestData\_RTStructureSetStorage()**

```
int vtkGDCMPolyDataReader::RequestData_RTStructureSetStorage (
    gdcmm::Reader const & reader,
    vtkInformationVector * outputVector ) [protected]
```

**10.388.3.7 RequestInformation()**

```
int vtkGDCMPolyDataReader::RequestInformation (
    vtkInformation * vtkNotUsedrequest,
    vtkInformationVector ** vtkNotUsedinputVector,
    vtkInformationVector * outputVector ) [protected]
```

**10.388.3.8 RequestInformation\_HemodynamicWaveformStorage()**

```
int vtkGDCMPolyDataReader::RequestInformation_HemodynamicWaveformStorage (
    gdcmm::Reader const & reader ) [protected]
```

### 10.388.3.9 RequestInformation\_RTStructureSetStorage()

```
int vtkGDCMPolyDataReader::RequestInformation_RTStructureSetStorage (
    gdcM::Reader const & reader ) [protected]
```

### 10.388.3.10 vtkGetObjectMacro() [1/2]

```
vtkGDCMPolyDataReader::vtkGetObjectMacro (
    MedicalImageProperties ,
    vtkMedicalImageProperties )
```

### 10.388.3.11 vtkGetObjectMacro() [2/2]

```
vtkGDCMPolyDataReader::vtkGetObjectMacro (
    RTStructSetProperties ,
    vtkRTStructSetProperties )
```

### 10.388.3.12 vtkGetStringMacro()

```
vtkGDCMPolyDataReader::vtkGetStringMacro (
    FileName )
```

### 10.388.3.13 vtkSetStringMacro()

```
vtkGDCMPolyDataReader::vtkSetStringMacro (
    FileName )
```

### 10.388.3.14 vtkTypeMacro()

```
vtkGDCMPolyDataReader::vtkTypeMacro (
    vtkGDCMPolyDataReader ,
    vtkPolyDataAlgorithm )
```

## 10.388.4 Member Data Documentation

### 10.388.4.1 FileName

```
char* vtkGDCMPolyDataReader::FileName [protected]
```

### 10.388.4.2 MedicalImageProperties

```
vtkMedicalImageProperties* vtkGDCMPolyDataReader::MedicalImageProperties [protected]
```

### 10.388.4.3 RTStructSetProperties

```
vtkRTStructSetProperties* vtkGDCMPolyDataReader::RTStructSetProperties [protected]
```

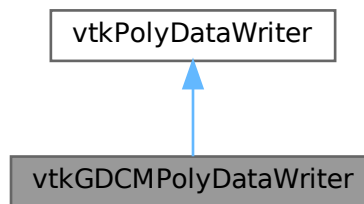
The documentation for this class was generated from the following file:

- [vtkGDCMPolyDataReader.h](#)

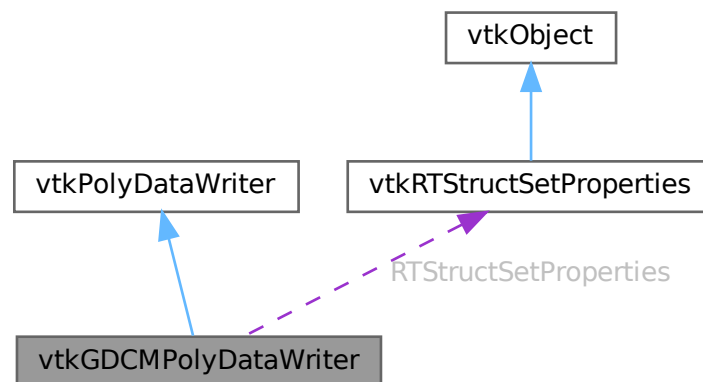
## 10.389 vtkGDCMPolyDataWriter Class Reference

```
#include <vtkGDCMPolyDataWriter.h>
```

Inheritance diagram for vtkGDCMPolyDataWriter:



Collaboration diagram for vtkGDCMPolyDataWriter:



## Public Member Functions

- void [InitializeRTStructSet](#) (vtkStdString inDirectory, vtkStdString inStructLabel, vtkStdString inStructName, vtkStringArray \*inROINames, vtkStringArray \*inROIAlgorithmName, vtkStringArray \*inROIType)
- virtual void [PrintSelf](#) (ostream &os, vtkIndent indent)
- virtual void [SetMedicalImageProperties](#) (vtkMedicalImageProperties \*pd)
- void [SetNumberOfInputPorts](#) (int n)
- virtual void [SetRTStructSetProperties](#) (vtkRTStructSetProperties \*pd)
- [vtkTypeMacro](#) (vtkGDCMPolyDataWriter, vtkPolyDataWriter)

## Static Public Member Functions

- static [vtkGDCMPolyDataWriter \\* New](#) ()

## Protected Member Functions

- [vtkGDCMPolyDataWriter](#) ()
- [~vtkGDCMPolyDataWriter](#) ()
- void [WriteData](#) ()
- void [WriteRTSTRUCTData](#) (gdcmm::File &file, int num)
- void [WriteRTSTRUCTInfo](#) (gdcmm::File &file)

## Protected Attributes

- vtkMedicalImageProperties \* [MedicalImageProperties](#)
- [vtkRTStructSetProperties](#) \* [RTStructSetProperties](#)

## 10.389.1 Detailed Description

### Examples

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

## 10.389.2 Constructor & Destructor Documentation

### 10.389.2.1 vtkGDCMPolyDataWriter()

```
vtkGDCMPolyDataWriter::vtkGDCMPolyDataWriter ( ) [protected]
```

### 10.389.2.2 ~vtkGDCMPolyDataWriter()

```
vtkGDCMPolyDataWriter::~~vtkGDCMPolyDataWriter ( ) [protected]
```

## 10.389.3 Member Function Documentation

### 10.389.3.1 InitializeRTStructSet()

```
void vtkGDCMPolyDataWriter::InitializeRTStructSet (
    vtkStdString inDirectory,
    vtkStdString inStructLabel,
    vtkStdString inStructName,
    vtkStringArray * inROINames,
    vtkStringArray * inROIAlgorithmName,
    vtkStringArray * inROIType )
```

#### Examples

[GenerateRTSTRUCT.cxx](#).

### 10.389.3.2 New()

```
static vtkGDCMPolyDataWriter * vtkGDCMPolyDataWriter::New ( ) [static]
```

#### Examples

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

### 10.389.3.3 PrintSelf()

```
virtual void vtkGDCMPolyDataWriter::PrintSelf (
    ostream & os,
    vtkIndent indent ) [virtual]
```

### 10.389.3.4 SetMedicalImageProperties()

```
virtual void vtkGDCMPolyDataWriter::SetMedicalImageProperties (
    vtkMedicalImageProperties * pd ) [virtual]
```

#### Examples

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

### 10.389.3.5 SetNumberOfInputPorts()

```
void vtkGDCMPolyDataWriter::SetNumberOfInputPorts (
    int n )
```

#### Examples

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

### 10.389.3.6 SetRTStructSetProperties()

```
virtual void vtkGDCMPolyDataWriter::SetRTStructSetProperties (
    vtkRTStructSetProperties * pd ) [virtual]
```

#### Examples

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

### 10.389.3.7 vtkTypeMacro()

```
vtkGDCMPolyDataWriter::vtkTypeMacro (
    vtkGDCMPolyDataWriter ,
    vtkPolyDataWriter )
```

### 10.389.3.8 WriteData()

```
void vtkGDCMPolyDataWriter::WriteData ( ) [protected]
```

### 10.389.3.9 WriteRTSTRUCTData()

```
void vtkGDCMPolyDataWriter::WriteRTSTRUCTData (
    gdcM::File & file,
    int num ) [protected]
```

### 10.389.3.10 WriteRTSTRUCTInfo()

```
void vtkGDCMPolyDataWriter::WriteRTSTRUCTInfo (
    gdcM::File & file ) [protected]
```

## 10.389.4 Member Data Documentation

### 10.389.4.1 MedicalImageProperties

```
vtkMedicalImageProperties* vtkGDCMPolyDataWriter::MedicalImageProperties [protected]
```

### 10.389.4.2 RTStructSetProperties

```
vtkRTStructSetProperties* vtkGDCMPolyDataWriter::RTStructSetProperties [protected]
```

The documentation for this class was generated from the following file:

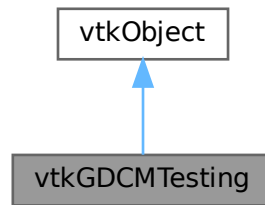
- [vtkGDCMPolyDataWriter.h](#)



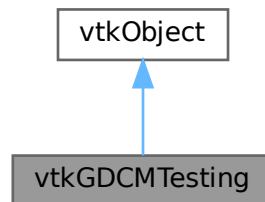
## 10.390 vtkGDCMTesting Class Reference

```
#include <vtkGDCMTesting.h>
```

Inheritance diagram for vtkGDCMTesting:



Collaboration diagram for vtkGDCMTesting:



### Public Types

- typedef const char \*const (\* [MD5MetalImagesType](#))[3]

### Public Member Functions

- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkTypeMacro](#) ([vtkGDCMTesting](#), vtkObject)

### Static Public Member Functions

- static const char \* [GetGDCMDataRoot](#) ()
- static const char \*const \* [GetMD5MetaImage](#) (unsigned int file)
- static const char \* [GetMHDMD5FromFile](#) (const char \*filepath)
- static unsigned int [GetNumberOfMD5MetaImages](#) ()
- static const char \* [GetRAWMD5FromFile](#) (const char \*filepath)
- static const char \* [GetVTKDataRoot](#) ()
- static [vtkGDCMTesting](#) \* [New](#) ()

### Protected Member Functions

- [vtkGDCMTesting](#) ()
- [~vtkGDCMTesting](#) ()

## 10.390.1 Detailed Description

### Examples

[HelloActiviz5.cs](#), [HelloVTKWorld2.cs](#), [MetaImageMD5Activiz.cs](#), [ReadSeriesIntoVTK.java](#), and [RefCounting.cs](#).

## 10.390.2 Member Typedef Documentation

### 10.390.2.1 MD5MetaImagesType

```
typedef const char* const(* vtkGDCMTesting::MD5MetaImagesType) [3]
```

## 10.390.3 Constructor & Destructor Documentation

### 10.390.3.1 vtkGDCMTesting()

```
vtkGDCMTesting::vtkGDCMTesting ( ) [protected]
```

### 10.390.3.2 ~vtkGDCMTesting()

```
vtkGDCMTesting::~~vtkGDCMTesting ( ) [protected]
```

## 10.390.4 Member Function Documentation

### 10.390.4.1 GetGDCMDataRoot()

```
static const char * vtkGDCMTesting::GetGDCMDataRoot ( ) [static]
```

### Examples

[HelloActiviz5.cs](#), and [ReadSeriesIntoVTK.java](#).

#### 10.390.4.2 GetMD5MetaImage()

```
static const char *const * vtkGDCMTesting::GetMD5MetaImage (
    unsigned int file ) [static]
```

#### 10.390.4.3 GetMHDMD5FromFile()

```
static const char * vtkGDCMTesting::GetMHDMD5FromFile (
    const char * filepath ) [static]
```

##### Examples

[MetaImageMD5Activiz.cs](#).

#### 10.390.4.4 GetNumberOfMD5MetaImages()

```
static unsigned int vtkGDCMTesting::GetNumberOfMD5MetaImages ( ) [static]
```

#### 10.390.4.5 GetRAWMD5FromFile()

```
static const char * vtkGDCMTesting::GetRAWMD5FromFile (
    const char * filepath ) [static]
```

##### Examples

[MetaImageMD5Activiz.cs](#).

#### 10.390.4.6 GetVTKDataRoot()

```
static const char * vtkGDCMTesting::GetVTKDataRoot ( ) [static]
```

##### Examples

[HelloActiviz5.cs](#), and [HelloVTKWorld2.cs](#).

#### 10.390.4.7 New()

```
static vtkGDCMTesting * vtkGDCMTesting::New ( ) [static]
```

##### Examples

[RefCounting.cs](#).

#### 10.390.4.8 PrintSelf()

```
void vtkGDCMTesting::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

#### 10.390.4.9 vtkTypeMacro()

```
vtkGDCMTesting::vtkTypeMacro (
    vtkGDCMTesting ,
    vtkObject )
```

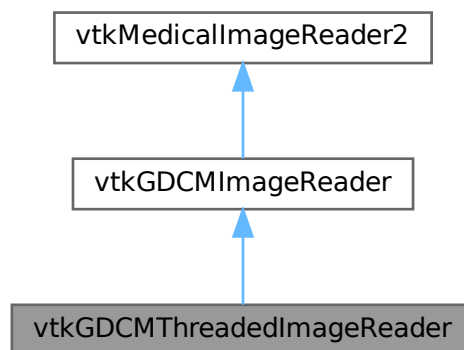
The documentation for this class was generated from the following file:

- [vtkGDCMTesting.h](#)

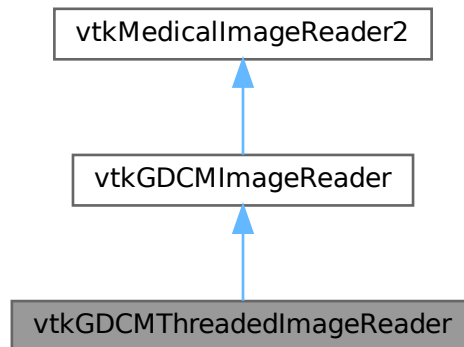
## 10.391 vtkGDCMThreadedImageReader Class Reference

```
#include <vtkGDCMThreadedImageReader.h>
```

Inheritance diagram for vtkGDCMThreadedImageReader:



Collaboration diagram for vtkGDCMThreadedImageReader:



### Public Member Functions

- virtual void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkBooleanMacro](#) (UseShiftScale, int)
- [vtkGetMacro](#) (UseShiftScale, int)
- [vtkSetMacro](#) (Scale, double)
- [vtkSetMacro](#) (Shift, double)
- [vtkSetMacro](#) (UseShiftScale, int)
- [vtkTypeMacro](#) (vtkGDCMThreadedImageReader, vtkGDCMImageReader)

### Public Member Functions inherited from [vtkGDCMImageReader](#)

- virtual int [CanReadFile](#) (const char \*fname)
- virtual const char \* [GetDescriptiveName](#) ()
- virtual const char \* [GetFileExtensions](#) ()
- vtkImageData \* [GetIconImage](#) ()
- vtkImageData \* [GetOverlay](#) (int i)
- virtual void [SetCurve](#) (vtkPolyData \*pd)
- virtual void [SetFileNames](#) (vtkStringArray \*)
- virtual void [SetMedicalImageProperties](#) (vtkMedicalImageProperties \*pd)
- [vtkBooleanMacro](#) (ApplyLookupTable, int)
- int [vtkBooleanMacro](#) (ApplyYBRToRGB, int)
- [vtkBooleanMacro](#) (LoadIconImage, int)
- [vtkBooleanMacro](#) (LoadOverlays, int)
- [vtkBooleanMacro](#) (LossyFlag, int)
- [vtkGetMacro](#) (ApplyLookupTable, int)
- [vtkGetMacro](#) (ApplyYBRToRGB, int) [vtkSetMacro](#) (ApplyYBRToRGB, int)
- [vtkGetMacro](#) (ImageFormat, int)
- [vtkGetMacro](#) (LoadIconImage, int)

- [vtkGetMacro](#) ([LoadOverlays](#), int)
- [vtkGetMacro](#) ([LossyFlag](#), int)
- [vtkGetMacro](#) ([NumberOfIconImages](#), int)
- [vtkGetMacro](#) ([NumberOfOverlays](#), int)
- [vtkGetMacro](#) ([PlanarConfiguration](#), int)
- [vtkGetMacro](#) ([Scale](#), double)
- [vtkGetMacro](#) ([Shift](#), double)
- [vtkGetObjectMacro](#) ([Curve](#), [vtkPolyData](#))
- [vtkGetObjectMacro](#) ([DirectionCosines](#), [vtkMatrix4x4](#))
- [vtkGetObjectMacro](#) ([FileNames](#), [vtkStringArray](#))
- [vtkGetObjectMacro](#) ([MedicalImageProperties](#), [vtkMedicalImageProperties](#))
- [vtkGetVector3Macro](#) ([ImagePositionPatient](#), double)
- [vtkGetVector6Macro](#) ([ImageOrientationPatient](#), double)
- [vtkSetMacro](#) ([ApplyLookupTable](#), int)
- [vtkSetMacro](#) ([LoadIconImage](#), int)
- [vtkSetMacro](#) ([LoadOverlays](#), int)
- [vtkSetMacro](#) ([LossyFlag](#), int)
- [vtkTypeMacro](#) ([vtkGDCMImageReader](#), [vtkMedicalImageReader2](#))

### Static Public Member Functions

- static [vtkGDCMThreadedImageReader](#) \* [New](#) ()

### Static Public Member Functions inherited from [vtkGDCMImageReader](#)

- static [vtkGDCMImageReader](#) \* [New](#) ()

### Protected Member Functions

- [vtkGDCMThreadedImageReader](#) ()
- [~vtkGDCMThreadedImageReader](#) ()
- void [ExecuteData](#) ([vtkDataObject](#) \*out)
- void [ExecuteInformation](#) ()
- void [ReadFiles](#) (unsigned int nfiles, const char \*filenames[])
- void [RequestDataCompat](#) ()

### Protected Member Functions inherited from [vtkGDCMImageReader](#)

- [vtkGDCMImageReader](#) ()
- [~vtkGDCMImageReader](#) ()
- void [ExecuteData](#) ([vtkDataObject](#) \*out)
- void [ExecuteInformation](#) ()
- void [FillMedicalImageInformation](#) (const [gdcm::ImageReader](#) &reader)
- int [LoadSingleFile](#) (const char \*filename, char \*pointer, unsigned long &outlen)
- int [RequestDataCompat](#) ()
- int [RequestInformationCompat](#) ()
- void [SetFilePattern](#) (const char \*)
- void [SetFilePrefix](#) (const char \*)
- [vtkGetStringMacro](#) ([FilePattern](#))
- [vtkGetStringMacro](#) ([FilePrefix](#))
- [vtkSetVector6Macro](#) ([ImageOrientationPatient](#), double)

## Additional Inherited Members

### Protected Attributes inherited from [vtkGDCMImageReader](#)

- int [ApplyInverseVideo](#)
- int [ApplyLookupTable](#)
- int [ApplyPlanarConfiguration](#)
- int [ApplyShiftScale](#)
- int [ApplyYBRToRGB](#)
- vtkPolyData \* [Curve](#)
- vtkMatrix4x4 \* [DirectionCosines](#)
- vtkStringArray \* [FileNames](#)
- int [ForceRescale](#)
- int [IconDataScalarType](#)
- int [IconImageDataExtent](#) [6]
- int [IconNumberOfScalarComponents](#)
- int [ImageFormat](#)
- double [ImageOrientationPatient](#) [6]
- double [ImagePositionPatient](#) [3]
- int [LoadIconImage](#)
- int [LoadOverlays](#)
- int [LossyFlag](#)
- vtkMedicalImageProperties \* [MedicalImageProperties](#)
- int [NumberOfIconImages](#)
- int [NumberOfOverlays](#)
- int [PlanarConfiguration](#)
- double [Scale](#)
- double [Shift](#)

## 10.391.1 Constructor & Destructor Documentation

### 10.391.1.1 [vtkGDCMThreadedImageReader\(\)](#)

```
vtkGDCMThreadedImageReader::vtkGDCMThreadedImageReader ( ) [protected]
```

### 10.391.1.2 [~vtkGDCMThreadedImageReader\(\)](#)

```
vtkGDCMThreadedImageReader::~~vtkGDCMThreadedImageReader ( ) [protected]
```

## 10.391.2 Member Function Documentation

### 10.391.2.1 [ExecuteData\(\)](#)

```
void vtkGDCMThreadedImageReader::ExecuteData (
    vtkDataObject * out ) [protected]
```

### 10.391.2.2 ExecuteInformation()

```
void vtkGDCMThreadedImageReader::ExecuteInformation ( ) [protected]
```

### 10.391.2.3 New()

```
static vtkGDCMThreadedImageReader * vtkGDCMThreadedImageReader::New ( ) [static]
```

### 10.391.2.4 PrintSelf()

```
virtual void vtkGDCMThreadedImageReader::PrintSelf (
    ostream & os,
    vtkIndent indent ) [virtual]
```

Reimplemented from [vtkGDCMImageReader](#).

### 10.391.2.5 ReadFiles()

```
void vtkGDCMThreadedImageReader::ReadFiles (
    unsigned int nfiles,
    const char * filenames[] ) [protected]
```

### 10.391.2.6 RequestDataCompat()

```
void vtkGDCMThreadedImageReader::RequestDataCompat ( ) [protected]
```

### 10.391.2.7 vtkBooleanMacro()

```
vtkGDCMThreadedImageReader::vtkBooleanMacro (
    UseShiftScale ,
    int )
```

### 10.391.2.8 vtkGetMacro()

```
vtkGDCMThreadedImageReader::vtkGetMacro (
    UseShiftScale ,
    int )
```

### 10.391.2.9 vtkSetMacro() [1/3]

```
vtkGDCMThreadedImageReader::vtkSetMacro (
    Scale ,
    double )
```



**10.391.2.10 vtkSetMacro() [2/3]**

```
vtkGDCMThreadedImageReader::vtkSetMacro (
    Shift ,
    double )
```

**10.391.2.11 vtkSetMacro() [3/3]**

```
vtkGDCMThreadedImageReader::vtkSetMacro (
    UseShiftScale ,
    int )
```

**10.391.2.12 vtkTypeMacro()**

```
vtkGDCMThreadedImageReader::vtkTypeMacro (
    vtkGDCMThreadedImageReader ,
    vtkGDCMImageReader )
```

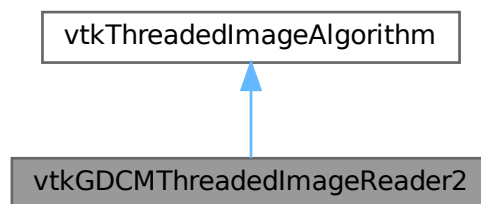
The documentation for this class was generated from the following file:

- [vtkGDCMThreadedImageReader.h](#)

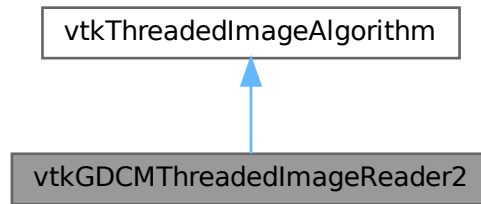
**10.392 vtkGDCMThreadedImageReader2 Class Reference**

```
#include <vtkGDCMThreadedImageReader2.h>
```

Inheritance diagram for vtkGDCMThreadedImageReader2:



Collaboration diagram for vtkGDCMThreadedImageReader2:



### Public Member Functions

- virtual const char \* [GetFileName](#) (int i=0)
- virtual void [PrintSelf](#) (ostream &os, vtkIndent indent)
- virtual void [SetFileName](#) (const char \*filename)
- virtual void [SetFileNames](#) (vtkStringArray \*)
- int [SplitExtent](#) (int splitExt[6], int startExt[6], int num, int total)
- [vtkBooleanMacro](#) (FileLowerLeft, int)
- [vtkBooleanMacro](#) (LoadOverlays, int)
- [vtkBooleanMacro](#) (UseShiftScale, int)
- [vtkGetMacro](#) (DataScalarType, int)
- [vtkGetMacro](#) (FileLowerLeft, int)
- [vtkGetMacro](#) (LoadOverlays, int)
- [vtkGetMacro](#) (NumberOfOverlays, int)
- [vtkGetMacro](#) (NumberOfScalarComponents, int)
- [vtkGetMacro](#) (Scale, double)
- [vtkGetMacro](#) (Shift, double)
- [vtkGetMacro](#) (UseShiftScale, int)
- [vtkGetObjectMacro](#) (FileNames, vtkStringArray)
- [vtkGetVector3Macro](#) (DataOrigin, double)
- [vtkGetVector3Macro](#) (DataSpacing, double)
- [vtkGetVector6Macro](#) (DataExtent, int)
- [vtkSetMacro](#) (DataScalarType, int)
- [vtkSetMacro](#) (FileLowerLeft, int)
- [vtkSetMacro](#) (LoadOverlays, int)
- [vtkSetMacro](#) (NumberOfScalarComponents, int)
- [vtkSetMacro](#) (Scale, double)
- [vtkSetMacro](#) (Shift, double)
- [vtkSetMacro](#) (UseShiftScale, int)
- [vtkSetVector3Macro](#) (DataOrigin, double)
- [vtkSetVector3Macro](#) (DataSpacing, double)
- [vtkSetVector6Macro](#) (DataExtent, int)
- [vtkTypeMacro](#) (vtkGDCMThreadedImageReader2, vtkThreadedImageAlgorithm)

## Static Public Member Functions

- static [vtkGDCMThreadedImageReader2](#) \* [New](#) ()

## Protected Member Functions

- [vtkGDCMThreadedImageReader2](#) ()
- [~vtkGDCMThreadedImageReader2](#) ()
- int [RequestInformation](#) (vtkInformation \*request, vtkInformationVector \*\*inputVector, vtkInformationVector \*outputVector)
- void [ThreadedRequestData](#) (vtkInformation \*request, vtkInformationVector \*\*inputVector, vtkInformationVector \*outputVector, vtkImageData \*\*\*inData, vtkImageData \*\*outData, int outExt[6], int id)

## 10.392.1 Constructor & Destructor Documentation

### 10.392.1.1 vtkGDCMThreadedImageReader2()

```
vtkGDCMThreadedImageReader2::vtkGDCMThreadedImageReader2 ( ) [protected]
```

### 10.392.1.2 ~vtkGDCMThreadedImageReader2()

```
vtkGDCMThreadedImageReader2::~vtkGDCMThreadedImageReader2 ( ) [protected]
```

## 10.392.2 Member Function Documentation

### 10.392.2.1 GetFileName()

```
virtual const char * vtkGDCMThreadedImageReader2::GetFileName (
    int i = 0 ) [virtual]
```

### 10.392.2.2 New()

```
static vtkGDCMThreadedImageReader2 * vtkGDCMThreadedImageReader2::New ( ) [static]
```

### 10.392.2.3 PrintSelf()

```
virtual void vtkGDCMThreadedImageReader2::PrintSelf (
    ostream & os,
    vtkIndent indent ) [virtual]
```

#### 10.392.2.4 RequestInformation()

```
int vtkGDCMThreadedImageReader2::RequestInformation (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector ) [protected]
```

#### 10.392.2.5 SetFileName()

```
virtual void vtkGDCMThreadedImageReader2::SetFileName (
    const char * filename ) [virtual]
```

#### 10.392.2.6 SetFileNames()

```
virtual void vtkGDCMThreadedImageReader2::SetFileNames (
    vtkStringArray * ) [virtual]
```

#### 10.392.2.7 SplitExtent()

```
int vtkGDCMThreadedImageReader2::SplitExtent (
    int splitExt[6],
    int startExt[6],
    int num,
    int total )
```

#### 10.392.2.8 ThreadedRequestData()

```
void vtkGDCMThreadedImageReader2::ThreadedRequestData (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector,
    vtkImageData *** inData,
    vtkImageData ** outData,
    int outExt[6],
    int id ) [protected]
```

#### 10.392.2.9 vtkBooleanMacro() [1/3]

```
vtkGDCMThreadedImageReader2::vtkBooleanMacro (
    FileLowerLeft ,
    int )
```

**10.392.2.10 vtkBooleanMacro()** [2/3]

```
vtkGDCMThreadedImageReader2::vtkBooleanMacro (
    LoadOverlays ,
    int )
```

**10.392.2.11 vtkBooleanMacro()** [3/3]

```
vtkGDCMThreadedImageReader2::vtkBooleanMacro (
    UseShiftScale ,
    int )
```

**10.392.2.12 vtkGetMacro()** [1/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    DataScalarType ,
    int )
```

**10.392.2.13 vtkGetMacro()** [2/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    FileLowerLeft ,
    int )
```

**10.392.2.14 vtkGetMacro()** [3/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    LoadOverlays ,
    int )
```

**10.392.2.15 vtkGetMacro()** [4/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    NumberOfOverlays ,
    int )
```

**10.392.2.16 vtkGetMacro()** [5/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    NumberOfScalarComponents ,
    int )
```

**10.392.2.17 vtkGetMacro()** [6/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    Scale ,
    double )
```

**10.392.2.18 vtkGetMacro()** [7/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    Shift ,
    double )
```

**10.392.2.19 vtkGetMacro()** [8/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    UseShiftScale ,
    int )
```

**10.392.2.20 vtkGetObjectMacro()**

```
vtkGDCMThreadedImageReader2::vtkGetObjectMacro (
    FileNames ,
    vtkStringArray )
```

**10.392.2.21 vtkGetVector3Macro()** [1/2]

```
vtkGDCMThreadedImageReader2::vtkGetVector3Macro (
    DataOrigin ,
    double )
```

**10.392.2.22 vtkGetVector3Macro()** [2/2]

```
vtkGDCMThreadedImageReader2::vtkGetVector3Macro (
    DataSpacing ,
    double )
```

**10.392.2.23 vtkGetVector6Macro()**

```
vtkGDCMThreadedImageReader2::vtkGetVector6Macro (
    DataExtent ,
    int )
```

**10.392.2.24 vtkSetMacro()** [1/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    DataScalarType ,
    int )
```

**10.392.2.25 vtkSetMacro()** [2/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    FileLowerLeft ,
    int )
```

**10.392.2.26 vtkSetMacro()** [3/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    LoadOverlays ,
    int )
```

**10.392.2.27 vtkSetMacro()** [4/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    NumberOfScalarComponents ,
    int )
```

**10.392.2.28 vtkSetMacro()** [5/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    Scale ,
    double )
```

**10.392.2.29 vtkSetMacro()** [6/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    Shift ,
    double )
```

**10.392.2.30 vtkSetMacro()** [7/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    UseShiftScale ,
    int )
```

**10.392.2.31 vtkSetVector3Macro()** [1/2]

```
vtkGDCMThreadedImageReader2::vtkSetVector3Macro (
    DataOrigin ,
    double )
```

**10.392.2.32 vtkSetVector3Macro()** [2/2]

```
vtkGDCMThreadedImageReader2::vtkSetVector3Macro (
    DataSpacing ,
    double )
```

**10.392.2.33 vtkSetVector6Macro()**

```
vtkGDCMThreadedImageReader2::vtkSetVector6Macro (
    DataExtent ,
    int )
```

**10.392.2.34 vtkTypeMacro()**

```
vtkGDCMThreadedImageReader2::vtkTypeMacro (
    vtkGDCMThreadedImageReader2 ,
    vtkThreadedImageAlgorithm )
```

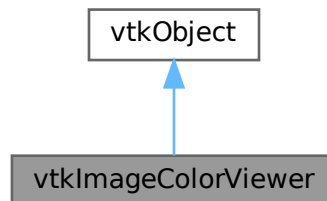
The documentation for this class was generated from the following file:

- [vtkGDCMThreadedImageReader2.h](#)

**10.393 vtkImageColorViewer Class Reference**

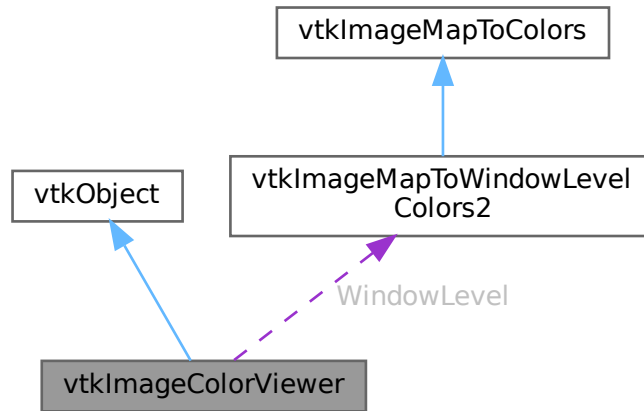
```
#include <vtkImageColorViewer.h>
```

Inheritance diagram for vtkImageColorViewer:





Collaboration diagram for vtkImageColorViewer:



## Public Types

- enum {  
[SLICE\\_ORIENTATION\\_YZ](#) = 0 ,  
[SLICE\\_ORIENTATION\\_XZ](#) = 1 ,  
[SLICE\\_ORIENTATION\\_XY](#) = 2 }

## Public Member Functions

- virtual void [AddInput](#) (vtkImageData \*input)
- virtual void [AddInputConnection](#) (vtkAlgorithmOutput \*input)
- virtual double [GetColorLevel](#) ()
- virtual double [GetColorWindow](#) ()
- virtual vtkImageData \* [GetInput](#) ()
- virtual int [GetOffScreenRendering](#) ()
- double [GetOverlayVisibility](#) ()
- virtual int \* [GetPosition](#) ()
- virtual int \* [GetSize](#) ()
- virtual int [GetSliceMax](#) ()
- virtual int [GetSliceMin](#) ()
- virtual int \* [GetSliceRange](#) ()
- virtual void [GetSliceRange](#) (int &min, int &max)
- virtual void [GetSliceRange](#) (int range[2])
- virtual const char \* [GetWindowName](#) ()
- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- virtual void [Render](#) (void)
- virtual void [SetColorLevel](#) (double s)

- virtual void [SetColorWindow](#) (double s)
- virtual void [SetDisplayId](#) (void \*a)
- virtual void [SetInput](#) (vtkImageData \*in)
- virtual void [SetInputConnection](#) (vtkAlgorithmOutput \*input)
- virtual void [SetOffScreenRendering](#) (int)
- void [SetOverlayVisibility](#) (double vis)
- virtual void [SetParentId](#) (void \*a)
- virtual void [SetPosition](#) (int a, int b)
- virtual void [SetPosition](#) (int a[2])
- virtual void [SetRenderer](#) (vtkRenderer \*arg)
- virtual void [SetRenderWindow](#) (vtkRenderWindow \*arg)
- virtual void [SetSize](#) (int a, int b)
- virtual void [SetSize](#) (int a[2])
- virtual void [SetSlice](#) (int s)
- virtual void [SetSliceOrientation](#) (int orientation)
- virtual void [SetSliceOrientationToXY](#) ()
- virtual void [SetSliceOrientationToXZ](#) ()
- virtual void [SetSliceOrientationToYZ](#) ()
- virtual void [SetupInteractor](#) (vtkRenderWindowInteractor \*)
- virtual void [SetWindowId](#) (void \*a)
- virtual void [UpdateDisplayExtent](#) ()
- [vtkBooleanMacro](#) (OffScreenRendering, int)
- [vtkGetMacro](#) (Slice, int)
- [vtkGetMacro](#) (SliceOrientation, int)
- [vtkGetObjectMacro](#) (ImageActor, vtkImageActor)
- [vtkGetObjectMacro](#) (InteractorStyle, vtkInteractorStyleImage)
- [vtkGetObjectMacro](#) (Renderer, vtkRenderer)
- [vtkGetObjectMacro](#) (RenderWindow, vtkRenderWindow)
- [vtkGetObjectMacro](#) (WindowLevel, vtkImageMapToWindowLevelColors2)
- [vtkTypeMacro](#) (vtkImageColorViewer, vtkObject)

### Static Public Member Functions

- static [vtkImageColorViewer \\* New](#) ()

### Protected Member Functions

- [vtkImageColorViewer](#) ()
- [~vtkImageColorViewer](#) ()
- virtual void [InstallPipeline](#) ()
- virtual void [UnInstallPipeline](#) ()
- virtual void [UpdateOrientation](#) ()

**Protected Attributes**

- int [FirstRender](#)
- vtkImageActor \* [ImageActor](#)
- vtkRenderWindowInteractor \* [Interactor](#)
- vtkInteractorStyleImage \* [InteractorStyle](#)
- vtkImageActor \* [OverlayImageActor](#)
- vtkRenderer \* [Renderer](#)
- vtkRenderWindow \* [RenderWindow](#)
- int [Slice](#)
- int [SliceOrientation](#)
- [vtkImageMapToWindowLevelColors2](#) \* [WindowLevel](#)

**Friends**

- class [vtkImageColorViewerCallback](#)

**10.393.1 Detailed Description****Examples**

[gdcmrptionplan.cxx](#), and [gdcmrtpplan.cxx](#).

**10.393.2 Member Enumeration Documentation****10.393.2.1 anonymous enum**

anonymous enum

**Enumerator**

SLICE_ORIENTATION_YZ	
SLICE_ORIENTATION_XZ	
SLICE_ORIENTATION_XY	

**10.393.3 Constructor & Destructor Documentation****10.393.3.1 vtkImageColorViewer()**

`vtkImageColorViewer::vtkImageColorViewer ( )` [protected]

**10.393.3.2 ~vtkImageColorViewer()**

`vtkImageColorViewer::~~vtkImageColorViewer ( )` [protected]

## 10.393.4 Member Function Documentation

### 10.393.4.1 AddInput()

```
virtual void vtkImageColorViewer::AddInput (
    vtkImageData * input ) [virtual]
```

### 10.393.4.2 AddInputConnection()

```
virtual void vtkImageColorViewer::AddInputConnection (
    vtkAlgorithmOutput * input ) [virtual]
```

### 10.393.4.3 GetColorLevel()

```
virtual double vtkImageColorViewer::GetColorLevel ( ) [virtual]
```

### 10.393.4.4 GetColorWindow()

```
virtual double vtkImageColorViewer::GetColorWindow ( ) [virtual]
```

### 10.393.4.5 GetInput()

```
virtual vtkImageData * vtkImageColorViewer::GetInput ( ) [virtual]
```

### 10.393.4.6 GetOffScreenRendering()

```
virtual int vtkImageColorViewer::GetOffScreenRendering ( ) [virtual]
```

### 10.393.4.7 GetOverlayVisibility()

```
double vtkImageColorViewer::GetOverlayVisibility ( )
```

### 10.393.4.8 GetPosition()

```
virtual int * vtkImageColorViewer::GetPosition ( ) [virtual]
```

### 10.393.4.9 GetSize()

```
virtual int * vtkImageColorViewer::GetSize ( ) [virtual]
```

**10.393.4.10 GetSliceMax()**

```
virtual int vtkImageColorViewer::GetSliceMax ( ) [virtual]
```

**10.393.4.11 GetSliceMin()**

```
virtual int vtkImageColorViewer::GetSliceMin ( ) [virtual]
```

**10.393.4.12 GetSliceRange() [1/3]**

```
virtual int * vtkImageColorViewer::GetSliceRange ( ) [virtual]
```

**10.393.4.13 GetSliceRange() [2/3]**

```
virtual void vtkImageColorViewer::GetSliceRange (
    int & min,
    int & max ) [virtual]
```

**10.393.4.14 GetSliceRange() [3/3]**

```
virtual void vtkImageColorViewer::GetSliceRange (
    int range[2] ) [inline], [virtual]
```

**10.393.4.15 GetWindowName()**

```
virtual const char * vtkImageColorViewer::GetWindowName ( ) [virtual]
```

**10.393.4.16 InstallPipeline()**

```
virtual void vtkImageColorViewer::InstallPipeline ( ) [protected], [virtual]
```

**10.393.4.17 New()**

```
static vtkImageColorViewer * vtkImageColorViewer::New ( ) [static]
```

**Examples**

[gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

**10.393.4.18 PrintSelf()**

```
void vtkImageColorViewer::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

**10.393.4.19 Render()**

```
virtual void vtkImageColorViewer::Render (
    void ) [virtual]
```

**Examples**

[gdcmrptionplan.cxx](#), and [gdcmrtpplan.cxx](#).

**10.393.4.20 SetColorLevel()**

```
virtual void vtkImageColorViewer::SetColorLevel (
    double s ) [virtual]
```

**10.393.4.21 SetColorWindow()**

```
virtual void vtkImageColorViewer::SetColorWindow (
    double s ) [virtual]
```

**10.393.4.22 SetDisplayId()**

```
virtual void vtkImageColorViewer::SetDisplayId (
    void * a ) [virtual]
```

**10.393.4.23 SetInput()**

```
virtual void vtkImageColorViewer::SetInput (
    vtkImageData * in ) [virtual]
```

**Examples**

[gdcmrptionplan.cxx](#), and [gdcmrtpplan.cxx](#).

**10.393.4.24 SetInputConnection()**

```
virtual void vtkImageColorViewer::SetInputConnection (
    vtkAlgorithmOutput * input ) [virtual]
```

**10.393.4.25 SetOffScreenRendering()**

```
virtual void vtkImageColorViewer::SetOffScreenRendering (
    int ) [virtual]
```

**10.393.4.26 SetOverlayVisibility()**

```
void vtkImageColorViewer::SetOverlayVisibility (
    double vis )
```

**10.393.4.27 SetParentId()**

```
virtual void vtkImageColorViewer::SetParentId (
    void * a ) [virtual]
```

**10.393.4.28 SetPosition() [1/2]**

```
virtual void vtkImageColorViewer::SetPosition (
    int a,
    int b ) [virtual]
```

**10.393.4.29 SetPosition() [2/2]**

```
virtual void vtkImageColorViewer::SetPosition (
    int a[2] ) [inline], [virtual]
```

References [SetPosition\(\)](#).

Referenced by [SetPosition\(\)](#).

**10.393.4.30 SetRenderer()**

```
virtual void vtkImageColorViewer::SetRenderer (
    vtkRenderer * arg ) [virtual]
```

**10.393.4.31 SetRenderWindow()**

```
virtual void vtkImageColorViewer::SetRenderWindow (
    vtkRenderWindow * arg ) [virtual]
```

**10.393.4.32 SetSize()** [1/2]

```
virtual void vtkImageColorViewer::SetSize (
    int a,
    int b ) [virtual]
```

**Examples**

[gdcmrptionplan.cxx](#), and [gdcmrtpian.cxx](#).

**10.393.4.33 SetSize()** [2/2]

```
virtual void vtkImageColorViewer::SetSize (
    int a[2] ) [inline], [virtual]
```

References [SetSize\(\)](#).

Referenced by [SetSize\(\)](#).

**10.393.4.34 SetSlice()**

```
virtual void vtkImageColorViewer::SetSlice (
    int s ) [virtual]
```

**10.393.4.35 SetSliceOrientation()**

```
virtual void vtkImageColorViewer::SetSliceOrientation (
    int orientation ) [virtual]
```

**10.393.4.36 SetSliceOrientationToXY()**

```
virtual void vtkImageColorViewer::SetSliceOrientationToXY ( ) [inline], [virtual]
```

References [SLICE\\_ORIENTATION\\_XY](#).

**10.393.4.37 SetSliceOrientationToXZ()**

```
virtual void vtkImageColorViewer::SetSliceOrientationToXZ ( ) [inline], [virtual]
```

References [SLICE\\_ORIENTATION\\_XZ](#).



**10.393.4.38 SetSliceOrientationToYZ()**

```
virtual void vtkImageColorViewer::SetSliceOrientationToYZ ( ) [inline], [virtual]
```

References [SLICE\\_ORIENTATION\\_YZ](#).

**10.393.4.39 SetupInteractor()**

```
virtual void vtkImageColorViewer::SetupInteractor (
    vtkRenderWindowInteractor * ) [virtual]
```

**Examples**

[gdcmrtonplan.cxx](#), and [gdcmrtpplan.cxx](#).

**10.393.4.40 SetWindowId()**

```
virtual void vtkImageColorViewer::SetWindowId (
    void * a ) [virtual]
```

**10.393.4.41 UnInstallPipeline()**

```
virtual void vtkImageColorViewer::UnInstallPipeline ( ) [protected], [virtual]
```

**10.393.4.42 UpdateDisplayExtent()**

```
virtual void vtkImageColorViewer::UpdateDisplayExtent ( ) [virtual]
```

**10.393.4.43 UpdateOrientation()**

```
virtual void vtkImageColorViewer::UpdateOrientation ( ) [protected], [virtual]
```

**10.393.4.44 vtkBooleanMacro()**

```
vtkImageColorViewer::vtkBooleanMacro (
    OffScreenRendering ,
    int )
```

**10.393.4.45 vtkGetMacro() [1/2]**

```
vtkImageColorViewer::vtkGetMacro (
    Slice ,
    int )
```

**10.393.4.46 vtkGetMacro() [2/2]**

```
vtkImageColorViewer::vtkGetMacro (
    SliceOrientation ,
    int )
```

**10.393.4.47 vtkGetObjectMacro() [1/5]**

```
vtkImageColorViewer::vtkGetObjectMacro (
    ImageActor ,
    vtkImageActor )
```

**10.393.4.48 vtkGetObjectMacro() [2/5]**

```
vtkImageColorViewer::vtkGetObjectMacro (
    InteractorStyle ,
    vtkInteractorStyleImage )
```

**10.393.4.49 vtkGetObjectMacro() [3/5]**

```
vtkImageColorViewer::vtkGetObjectMacro (
    Renderer ,
    vtkRenderer )
```

**10.393.4.50 vtkGetObjectMacro() [4/5]**

```
vtkImageColorViewer::vtkGetObjectMacro (
    RenderWindow ,
    vtkRenderWindow )
```

**10.393.4.51 vtkGetObjectMacro() [5/5]**

```
vtkImageColorViewer::vtkGetObjectMacro (
    WindowLevel ,
    vtkImageMapToWindowLevelColors2 )
```

#### 10.393.4.52 vtkTypeMacro()

```
vtkImageColorViewer::vtkTypeMacro (
    vtkImageColorViewer ,
    vtkObject )
```

### 10.393.5 Friends And Related Symbol Documentation

#### 10.393.5.1 vtkImageColorViewerCallback

```
friend class vtkImageColorViewerCallback [friend]
```

### 10.393.6 Member Data Documentation

#### 10.393.6.1 FirstRender

```
int vtkImageColorViewer::FirstRender [protected]
```

#### 10.393.6.2 ImageActor

```
vtkImageActor* vtkImageColorViewer::ImageActor [protected]
```

#### 10.393.6.3 Interactor

```
vtkRenderWindowInteractor* vtkImageColorViewer::Interactor [protected]
```

#### 10.393.6.4 InteractorStyle

```
vtkInteractorStyleImage* vtkImageColorViewer::InteractorStyle [protected]
```

#### 10.393.6.5 OverlayImageActor

```
vtkImageActor* vtkImageColorViewer::OverlayImageActor [protected]
```

#### 10.393.6.6 Renderer

```
vtkRenderer* vtkImageColorViewer::Renderer [protected]
```

### 10.393.6.7 RenderWindow

```
vtkRenderWindow* vtkImageColorViewer::RenderWindow [protected]
```

### 10.393.6.8 Slice

```
int vtkImageColorViewer::Slice [protected]
```

### 10.393.6.9 SliceOrientation

```
int vtkImageColorViewer::SliceOrientation [protected]
```

### 10.393.6.10 WindowLevel

```
vtkImageMapToWindowLevelColors2* vtkImageColorViewer::WindowLevel [protected]
```

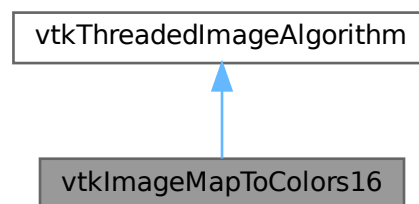
The documentation for this class was generated from the following file:

- [vtkImageColorViewer.h](#)

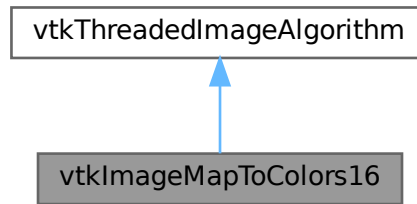
## 10.394 vtkImageMapToColors16 Class Reference

```
#include <vtkImageMapToColors16.h>
```

Inheritance diagram for vtkImageMapToColors16:



Collaboration diagram for vtkImageMapToColors16:



### Public Member Functions

- virtual unsigned long [GetMTime](#) ()
- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- virtual void [SetLookupTable](#) (vtkScalarsToColors \*)
- void [SetOutputFormatToLuminance](#) ()
- void [SetOutputFormatToLuminanceAlpha](#) ()
- void [SetOutputFormatToRGB](#) ()
- void [SetOutputFormatToRGBA](#) ()
- [vtkBooleanMacro](#) ([PassAlphaToOutput](#), int)
- [vtkGetMacro](#) ([ActiveComponent](#), int)
- [vtkGetMacro](#) ([OutputFormat](#), int)
- [vtkGetMacro](#) ([PassAlphaToOutput](#), int)
- [vtkGetObjectMacro](#) ([LookupTable](#), vtkScalarsToColors)
- [vtkSetMacro](#) ([ActiveComponent](#), int)
- [vtkSetMacro](#) ([OutputFormat](#), int)
- [vtkSetMacro](#) ([PassAlphaToOutput](#), int)
- [vtkTypeMacro](#) (vtkImageMapToColors16, vtkThreadedImageAlgorithm)

### Static Public Member Functions

- static [vtkImageMapToColors16 \\* New](#) ()

### Protected Member Functions

- [vtkImageMapToColors16](#) ()
- [~vtkImageMapToColors16](#) ()
- virtual int [RequestData](#) (vtkInformation \*request, vtkInformationVector \*\*inputVector, vtkInformationVector \*outputVector)
- virtual int [RequestInformation](#) (vtkInformation \*, vtkInformationVector \*\*, vtkInformationVector \*)
- void [ThreadedRequestData](#) (vtkInformation \*request, vtkInformationVector \*\*inputVector, vtkInformationVector \*outputVector, vtkImageData \*\*\*inData, vtkImageData \*\*outData, int extent[6], int id)

**Protected Attributes**

- int [ActiveComponent](#)
- int [DataWasPassed](#)
- vtkScalarsToColors \* [LookupTable](#)
- int [OutputFormat](#)
- int [PassAlphaToOutput](#)

**10.394.1 Constructor & Destructor Documentation****10.394.1.1 vtkImageMapToColors16()**

```
vtkImageMapToColors16::vtkImageMapToColors16 ( ) [protected]
```

**10.394.1.2 ~vtkImageMapToColors16()**

```
vtkImageMapToColors16::~~vtkImageMapToColors16 ( ) [protected]
```

**10.394.2 Member Function Documentation****10.394.2.1 GetMTime()**

```
virtual unsigned long vtkImageMapToColors16::GetMTime ( ) [virtual]
```

**10.394.2.2 New()**

```
static vtkImageMapToColors16 * vtkImageMapToColors16::New ( ) [static]
```

**10.394.2.3 PrintSelf()**

```
void vtkImageMapToColors16::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

**10.394.2.4 RequestData()**

```
virtual int vtkImageMapToColors16::RequestData (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector ) [protected], [virtual]
```

#### 10.394.2.5 RequestInformation()

```
virtual int vtkImageMapToColors16::RequestInformation (
    vtkInformation * ,
    vtkInformationVector ** ,
    vtkInformationVector * ) [protected], [virtual]
```

#### 10.394.2.6 SetLookupTable()

```
virtual void vtkImageMapToColors16::SetLookupTable (
    vtkScalarsToColors * ) [virtual]
```

#### 10.394.2.7 SetOutputFormatToLuminance()

```
void vtkImageMapToColors16::SetOutputFormatToLuminance ( ) [inline]
```

#### 10.394.2.8 SetOutputFormatToLuminanceAlpha()

```
void vtkImageMapToColors16::SetOutputFormatToLuminanceAlpha ( ) [inline]
```

#### 10.394.2.9 SetOutputFormatToRGB()

```
void vtkImageMapToColors16::SetOutputFormatToRGB ( ) [inline]
```

#### 10.394.2.10 SetOutputFormatToRGBA()

```
void vtkImageMapToColors16::SetOutputFormatToRGBA ( ) [inline]
```

#### 10.394.2.11 ThreadedRequestData()

```
void vtkImageMapToColors16::ThreadedRequestData (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector,
    vtkImageData *** inData,
    vtkImageData ** outData,
    int extent[6],
    int id ) [protected]
```

**10.394.2.12 vtkBooleanMacro()**

```
vtkImageMapToColors16::vtkBooleanMacro (
    PassAlphaToOutput ,
    int )
```

**10.394.2.13 vtkGetMacro() [1/3]**

```
vtkImageMapToColors16::vtkGetMacro (
    ActiveComponent ,
    int )
```

**10.394.2.14 vtkGetMacro() [2/3]**

```
vtkImageMapToColors16::vtkGetMacro (
    OutputFormat ,
    int )
```

**10.394.2.15 vtkGetMacro() [3/3]**

```
vtkImageMapToColors16::vtkGetMacro (
    PassAlphaToOutput ,
    int )
```

**10.394.2.16 vtkGetObjectMacro()**

```
vtkImageMapToColors16::vtkGetObjectMacro (
    LookupTable ,
    vtkScalarsToColors )
```

**10.394.2.17 vtkSetMacro() [1/3]**

```
vtkImageMapToColors16::vtkSetMacro (
    ActiveComponent ,
    int )
```

**10.394.2.18 vtkSetMacro() [2/3]**

```
vtkImageMapToColors16::vtkSetMacro (
    OutputFormat ,
    int )
```



#### 10.394.2.19 vtkSetMacro() [3/3]

```
vtkImageMapToColors16::vtkSetMacro (
    PassAlphaToOutput ,
    int )
```

#### 10.394.2.20 vtkTypeMacro()

```
vtkImageMapToColors16::vtkTypeMacro (
    vtkImageMapToColors16 ,
    vtkThreadedImageAlgorithm )
```

### 10.394.3 Member Data Documentation

#### 10.394.3.1 ActiveComponent

```
int vtkImageMapToColors16::ActiveComponent [protected]
```

#### 10.394.3.2 DataWasPassed

```
int vtkImageMapToColors16::DataWasPassed [protected]
```

#### 10.394.3.3 LookupTable

```
vtkScalarsToColors* vtkImageMapToColors16::LookupTable [protected]
```

#### 10.394.3.4 OutputFormat

```
int vtkImageMapToColors16::OutputFormat [protected]
```

#### 10.394.3.5 PassAlphaToOutput

```
int vtkImageMapToColors16::PassAlphaToOutput [protected]
```

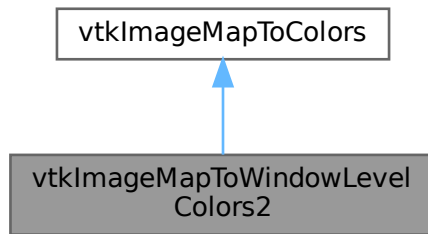
The documentation for this class was generated from the following file:

- [vtkImageMapToColors16.h](#)

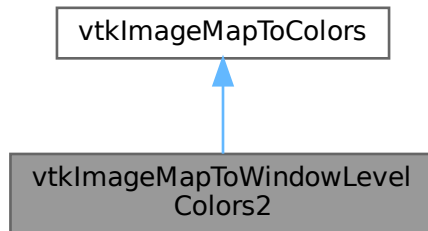
## 10.395 vtkImageMapToWindowLevelColors2 Class Reference

```
#include <vtkImageMapToWindowLevelColors2.h>
```

Inheritance diagram for vtkImageMapToWindowLevelColors2:



Collaboration diagram for vtkImageMapToWindowLevelColors2:



### Public Member Functions

- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkGetMacro](#) (Level, double)
- [vtkGetMacro](#) (Window, double)
- [vtkSetMacro](#) (Level, double)
- [vtkSetMacro](#) (Window, double)
- [vtkTypeMacro](#) (vtkImageMapToWindowLevelColors2, vtkImageMapToColors)

### Static Public Member Functions

- static [vtkImageMapToWindowLevelColors2](#) \* [New](#) ()

**Protected Member Functions**

- [vtkImageMapToWindowLevelColors2](#) ()
- [~vtkImageMapToWindowLevelColors2](#) ()
- virtual int [RequestData](#) (vtkInformation \*request, vtkInformationVector \*\*inputVector, vtkInformationVector \*outputVector)
- virtual int [RequestInformation](#) (vtkInformation \*, vtkInformationVector \*\*, vtkInformationVector \*)
- void [ThreadedRequestData](#) (vtkInformation \*request, vtkInformationVector \*\*inputVector, vtkInformationVector \*outputVector, vtkImageData \*\*\*inData, vtkImageData \*\*outData, int extent[6], int id)

**Protected Attributes**

- double [Level](#)
- double [Window](#)

**10.395.1 Constructor & Destructor Documentation****10.395.1.1 vtkImageMapToWindowLevelColors2()**

```
vtkImageMapToWindowLevelColors2::vtkImageMapToWindowLevelColors2 ( ) [protected]
```

**10.395.1.2 ~vtkImageMapToWindowLevelColors2()**

```
vtkImageMapToWindowLevelColors2::~~vtkImageMapToWindowLevelColors2 ( ) [protected]
```

**10.395.2 Member Function Documentation****10.395.2.1 New()**

```
static vtkImageMapToWindowLevelColors2 * vtkImageMapToWindowLevelColors2::New ( ) [static]
```

**10.395.2.2 PrintSelf()**

```
void vtkImageMapToWindowLevelColors2::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

**10.395.2.3 RequestData()**

```
virtual int vtkImageMapToWindowLevelColors2::RequestData (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector ) [protected], [virtual]
```

#### 10.395.2.4 RequestInformation()

```
virtual int vtkImageMapToWindowLevelColors2::RequestInformation (
    vtkInformation * ,
    vtkInformationVector ** ,
    vtkInformationVector * ) [protected], [virtual]
```

#### 10.395.2.5 ThreadedRequestData()

```
void vtkImageMapToWindowLevelColors2::ThreadedRequestData (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector,
    vtkImageData *** inData,
    vtkImageData ** outData,
    int extent[6],
    int id ) [protected]
```

#### 10.395.2.6 vtkGetMacro() [1/2]

```
vtkImageMapToWindowLevelColors2::vtkGetMacro (
    Level ,
    double )
```

#### 10.395.2.7 vtkGetMacro() [2/2]

```
vtkImageMapToWindowLevelColors2::vtkGetMacro (
    Window ,
    double )
```

#### 10.395.2.8 vtkSetMacro() [1/2]

```
vtkImageMapToWindowLevelColors2::vtkSetMacro (
    Level ,
    double )
```

#### 10.395.2.9 vtkSetMacro() [2/2]

```
vtkImageMapToWindowLevelColors2::vtkSetMacro (
    Window ,
    double )
```

### 10.395.2.10 vtkTypeMacro()

```
vtkImageMapToWindowLevelColors2::vtkTypeMacro (
    vtkImageMapToWindowLevelColors2 ,
    vtkImageMapToColors )
```

## 10.395.3 Member Data Documentation

### 10.395.3.1 Level

```
double vtkImageMapToWindowLevelColors2::Level [protected]
```

### 10.395.3.2 Window

```
double vtkImageMapToWindowLevelColors2::Window [protected]
```

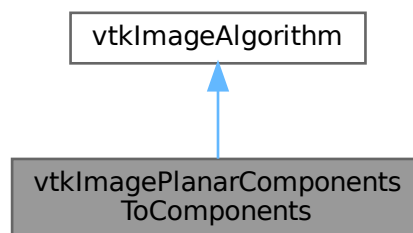
The documentation for this class was generated from the following file:

- [vtkImageMapToWindowLevelColors2.h](#)

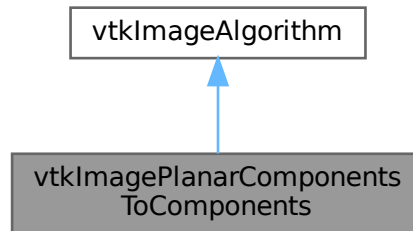
## 10.396 vtkImagePlanarComponentsToComponents Class Reference

```
#include <vtkImagePlanarComponentsToComponents.h>
```

Inheritance diagram for vtkImagePlanarComponentsToComponents:



Collaboration diagram for vtkImagePlanarComponentsToComponents:



### Public Member Functions

- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkTypeMacro](#) (vtkImagePlanarComponentsToComponents, vtkImageAlgorithm)

### Static Public Member Functions

- static [vtkImagePlanarComponentsToComponents \\* New](#) ()

### Protected Member Functions

- [vtkImagePlanarComponentsToComponents](#) ()
- [~vtkImagePlanarComponentsToComponents](#) ()
- virtual int [RequestData](#) (vtkInformation \*, vtkInformationVector \*\*, vtkInformationVector \*)

## 10.396.1 Constructor & Destructor Documentation

### 10.396.1.1 vtkImagePlanarComponentsToComponents()

```
vtkImagePlanarComponentsToComponents::vtkImagePlanarComponentsToComponents ( ) [protected]
```

### 10.396.1.2 ~vtkImagePlanarComponentsToComponents()

```
vtkImagePlanarComponentsToComponents::~~vtkImagePlanarComponentsToComponents ( ) [inline], [protected]
```

## 10.396.2 Member Function Documentation

### 10.396.2.1 New()

```
static vtkImagePlanarComponentsToComponents * vtkImagePlanarComponentsToComponents::New ( ) [static]
```

### 10.396.2.2 PrintSelf()

```
void vtkImagePlanarComponentsToComponents::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

### 10.396.2.3 RequestData()

```
virtual int vtkImagePlanarComponentsToComponents::RequestData (
    vtkInformation * ,
    vtkInformationVector ** ,
    vtkInformationVector * ) [protected], [virtual]
```

### 10.396.2.4 vtkTypeMacro()

```
vtkImagePlanarComponentsToComponents::vtkTypeMacro (
    vtkImagePlanarComponentsToComponents ,
    vtkImageAlgorithm )
```

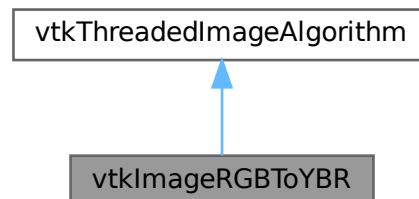
The documentation for this class was generated from the following file:

- [vtkImagePlanarComponentsToComponents.h](#)

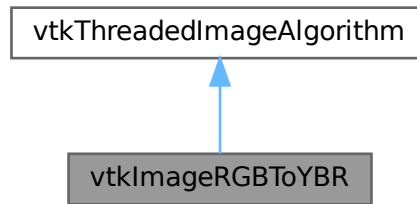
## 10.397 vtkImageRGBToYBR Class Reference

```
#include <vtkImageRGBToYBR.h>
```

Inheritance diagram for vtkImageRGBToYBR:



Collaboration diagram for vtkImageRGBToYBR:



### Public Member Functions

- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkTypeMacro](#) ([vtkImageRGBToYBR](#), vtkThreadedImageAlgorithm)

### Static Public Member Functions

- static [vtkImageRGBToYBR](#) \* [New](#) ()

### Protected Member Functions

- [vtkImageRGBToYBR](#) ()
- [~vtkImageRGBToYBR](#) ()
- void [ThreadedExecute](#) (vtkImageData \*inData, vtkImageData \*outData, int ext[6], int id)

## 10.397.1 Constructor & Destructor Documentation

### 10.397.1.1 vtkImageRGBToYBR()

```
vtkImageRGBToYBR::vtkImageRGBToYBR ( ) [protected]
```

### 10.397.1.2 ~vtkImageRGBToYBR()

```
vtkImageRGBToYBR::~~vtkImageRGBToYBR ( ) [inline], [protected]
```



## 10.397.2 Member Function Documentation

### 10.397.2.1 New()

```
static vtkImageRGBToYBR * vtkImageRGBToYBR::New ( ) [static]
```

### 10.397.2.2 PrintSelf()

```
void vtkImageRGBToYBR::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

### 10.397.2.3 ThreadedExecute()

```
void vtkImageRGBToYBR::ThreadedExecute (
    vtkImageData * inData,
    vtkImageData * outData,
    int ext[6],
    int id ) [protected]
```

### 10.397.2.4 vtkTypeMacro()

```
vtkImageRGBToYBR::vtkTypeMacro (
    vtkImageRGBToYBR ,
    vtkThreadedImageAlgorithm )
```

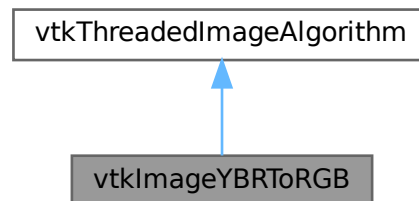
The documentation for this class was generated from the following file:

- [vtkImageRGBToYBR.h](#)

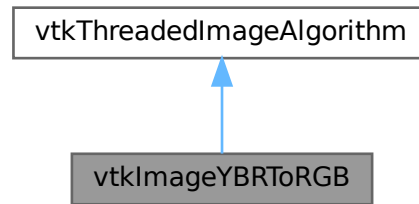
## 10.398 vtkImageYBRToRGB Class Reference

```
#include <vtkImageYBRToRGB.h>
```

Inheritance diagram for vtkImageYBRToRGB:



Collaboration diagram for vtkImageYBRToRGB:



### Public Member Functions

- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkTypeMacro](#) ([vtkImageYBRToRGB](#), vtkThreadedImageAlgorithm)

### Static Public Member Functions

- static [vtkImageYBRToRGB](#) \* [New](#) ()

### Protected Member Functions

- [vtkImageYBRToRGB](#) ()
- [~vtkImageYBRToRGB](#) ()
- void [ThreadedExecute](#) (vtkImageData \*inData, vtkImageData \*outData, int ext[6], int id)

## 10.398.1 Constructor & Destructor Documentation

### 10.398.1.1 vtkImageYBRToRGB()

```
vtkImageYBRToRGB::vtkImageYBRToRGB ( ) [protected]
```

### 10.398.1.2 ~vtkImageYBRToRGB()

```
vtkImageYBRToRGB::~~vtkImageYBRToRGB ( ) [inline], [protected]
```

## 10.398.2 Member Function Documentation

### 10.398.2.1 New()

```
static vtkImageYBRToRGB * vtkImageYBRToRGB::New ( ) [static]
```

### 10.398.2.2 PrintSelf()

```
void vtkImageYBRToRGB::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

### 10.398.2.3 ThreadedExecute()

```
void vtkImageYBRToRGB::ThreadedExecute (
    vtkImageData * inData,
    vtkImageData * outData,
    int ext[6],
    int id ) [protected]
```

### 10.398.2.4 vtkTypeMacro()

```
vtkImageYBRToRGB::vtkTypeMacro (
    vtkImageYBRToRGB ,
    vtkThreadedImageAlgorithm )
```

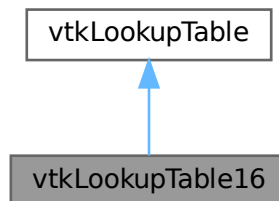
The documentation for this class was generated from the following file:

- [vtkImageYBRToRGB.h](#)

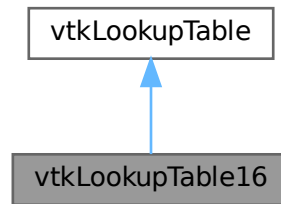
## 10.399 vtkLookupTable16 Class Reference

```
#include <vtkLookupTable16.h>
```

Inheritance diagram for vtkLookupTable16:



Collaboration diagram for vtkLookupTable16:



### Public Member Functions

- void [Build](#) ()
- unsigned short \* [GetPointer](#) (const vtkIdType id)
- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- void [SetNumberOfTableValues](#) (vtkIdType number)
- [vtkTypeMacro](#) ([vtkLookupTable16](#), vtkLookupTable)
- unsigned char \* [WritePointer](#) (const vtkIdType id, const int number)

### Static Public Member Functions

- static [vtkLookupTable16](#) \* [New](#) ()

### Protected Member Functions

- [vtkLookupTable16](#) (int size=256, int ext=256)
- [~vtkLookupTable16](#) ()
- void [MapScalarsThroughTable2](#) (void \*input, unsigned char \*output, int inputDataType, int numberOfValues, int inputIncrement, int outputFormat)

### Protected Attributes

- vtkUnsignedShortArray \* [Table16](#)

## 10.399.1 Constructor & Destructor Documentation

### 10.399.1.1 vtkLookupTable16()

```

vtkLookupTable16::vtkLookupTable16 (
    int size = 256,
    int ext = 256 ) [protected]
  
```

### 10.399.1.2 ~vtkLookupTable16()

```
vtkLookupTable16::~vtkLookupTable16 ( ) [protected]
```

## 10.399.2 Member Function Documentation

### 10.399.2.1 Build()

```
void vtkLookupTable16::Build ( )
```

### 10.399.2.2 GetPointer()

```
unsigned short * vtkLookupTable16::GetPointer (
    const vtkIdType id ) [inline]
```

### 10.399.2.3 MapScalarsThroughTable2()

```
void vtkLookupTable16::MapScalarsThroughTable2 (
    void * input,
    unsigned char * output,
    int inputDataType,
    int numberOfValues,
    int inputIncrement,
    int outputFormat ) [protected]
```

### 10.399.2.4 New()

```
static vtkLookupTable16 * vtkLookupTable16::New ( ) [static]
```

### 10.399.2.5 PrintSelf()

```
void vtkLookupTable16::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

### 10.399.2.6 SetNumberOfTableValues()

```
void vtkLookupTable16::SetNumberOfTableValues (
    vtkIdType number )
```

### 10.399.2.7 vtkTypeMacro()

```
vtkLookupTable16::vtkTypeMacro (
    vtkLookupTable16 ,
    vtkLookupTable )
```

### 10.399.2.8 WritePointer()

```
unsigned char * vtkLookupTable16::WritePointer (
    const vtkIdType id,
    const int number ) [inline]
```

References [Table16](#).

## 10.399.3 Member Data Documentation

### 10.399.3.1 Table16

```
vtkUnsignedShortArray* vtkLookupTable16::Table16 [protected]
```

Referenced by [WritePointer\(\)](#).

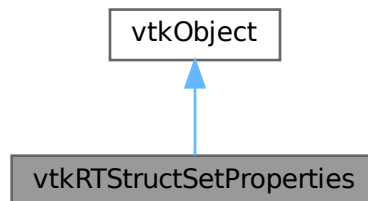
The documentation for this class was generated from the following file:

- [vtkLookupTable16.h](#)

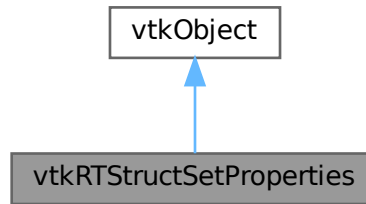
## 10.400 vtkRTStructSetProperties Class Reference

```
#include <vtkRTStructSetProperties.h>
```

Inheritance diagram for vtkRTStructSetProperties:



Collaboration diagram for vtkRTStructSetProperties:



### Public Member Functions

- void [AddContourReferencedFrameOfReference](#) (vtkIdType pdnum, const char \*classuid, const char \*instanceuid)
- void [AddReferencedFrameOfReference](#) (const char \*classuid, const char \*instanceuid)
- void [AddStructureSetROI](#) (int roinumber, const char \*refframerefid, const char \*roiname, const char \*ROIGenerationAlgorithm, const char \*ROIDescription=0)
- void [AddStructureSetROIObservation](#) (int refnumber, int observationnumber, const char \*rtroiinterpretedtype, const char \*roiinterpreter, const char \*roiobservationlabel=0)
- virtual void [Clear](#) ()
- virtual void [DeepCopy](#) (vtkRTStructSetProperties \*p)
- const char \* [GetContourReferencedFrameOfReferenceClassUID](#) (vtkIdType pdnum, vtkIdType id)
- const char \* [GetContourReferencedFrameOfReferenceInstanceUID](#) (vtkIdType pdnum, vtkIdType id)
- vtkIdType [GetNumberOfContourReferencedFrameOfReferences](#) ()
- vtkIdType [GetNumberOfContourReferencedFrameOfReferences](#) (vtkIdType pdnum)
- vtkIdType [GetNumberOfReferencedFrameOfReferences](#) ()
- vtkIdType [GetNumberOfStructureSetROIs](#) ()
- const char \* [GetReferencedFrameOfReferenceClassUID](#) (vtkIdType id)
- const char \* [GetReferencedFrameOfReferenceInstanceUID](#) (vtkIdType id)
- int [GetStructureSetObservationNumber](#) (vtkIdType id)
- const char \* [GetStructureSetROIDescription](#) (vtkIdType id)
- const char \* [GetStructureSetROIGenerationAlgorithm](#) (vtkIdType)
- const char \* [GetStructureSetROIName](#) (vtkIdType)
- int [GetStructureSetROINumber](#) (vtkIdType id)
- const char \* [GetStructureSetROIObservationLabel](#) (vtkIdType id)
- const char \* [GetStructureSetROIRefFrameRefUID](#) (vtkIdType)
- const char \* [GetStructureSetRTROIInterpretedType](#) (vtkIdType id)
- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkGetStringMacro](#) (ReferenceFrameOfReferenceUID)
- [vtkGetStringMacro](#) (ReferenceSeriesInstanceUID)
- [vtkGetStringMacro](#) (SeriesInstanceUID)
- [vtkGetStringMacro](#) (SOPInstanceUID)
- [vtkGetStringMacro](#) (StructureSetDate)
- [vtkGetStringMacro](#) (StructureSetLabel)
- [vtkGetStringMacro](#) (StructureSetName)

- [vtkGetStringMacro](#) ([StructureSetTime](#))
- [vtkGetStringMacro](#) ([StudyInstanceUID](#))
- [vtkSetStringMacro](#) ([ReferenceFrameOfReferenceUID](#))
- [vtkSetStringMacro](#) ([ReferenceSeriesInstanceUID](#))
- [vtkSetStringMacro](#) ([SeriesInstanceUID](#))
- [vtkSetStringMacro](#) ([SOPInstanceUID](#))
- [vtkSetStringMacro](#) ([StructureSetDate](#))
- [vtkSetStringMacro](#) ([StructureSetLabel](#))
- [vtkSetStringMacro](#) ([StructureSetName](#))
- [vtkSetStringMacro](#) ([StructureSetTime](#))
- [vtkSetStringMacro](#) ([StudyInstanceUID](#))
- [vtkTypeMacro](#) ([vtkRTStructSetProperties](#), [vtkObject](#))

### Static Public Member Functions

- static [vtkRTStructSetProperties](#) \* [New](#) ()

### Protected Member Functions

- [vtkRTStructSetProperties](#) ()
- [~vtkRTStructSetProperties](#) ()

### Protected Attributes

- [vtkRTStructSetPropertiesInternals](#) \* [Internals](#)
- char \* [ReferenceFrameOfReferenceUID](#)
- char \* [ReferenceSeriesInstanceUID](#)
- char \* [SeriesInstanceUID](#)
- char \* [SOPInstanceUID](#)
- char \* [StructureSetDate](#)
- char \* [StructureSetLabel](#)
- char \* [StructureSetName](#)
- char \* [StructureSetTime](#)
- char \* [StudyInstanceUID](#)

## 10.400.1 Detailed Description

### Examples

[GenerateRTSTRUCT.cxx](#).

## 10.400.2 Constructor & Destructor Documentation

### 10.400.2.1 [vtkRTStructSetProperties\(\)](#)

```
vtkRTStructSetProperties::vtkRTStructSetProperties ( ) [protected]
```



### 10.400.2.2 ~vtkRTStructSetProperties()

```
vtkRTStructSetProperties::~vtkRTStructSetProperties ( ) [protected]
```

## 10.400.3 Member Function Documentation

### 10.400.3.1 AddContourReferencedFrameOfReference()

```
void vtkRTStructSetProperties::AddContourReferencedFrameOfReference (
    vtkIdType pdnum,
    const char * classuid,
    const char * instanceuid )
```

### 10.400.3.2 AddReferencedFrameOfReference()

```
void vtkRTStructSetProperties::AddReferencedFrameOfReference (
    const char * classuid,
    const char * instanceuid )
```

### 10.400.3.3 AddStructureSetROI()

```
void vtkRTStructSetProperties::AddStructureSetROI (
    int roinumber,
    const char * refframerefuid,
    const char * roiname,
    const char * ROIGenerationAlgorithm,
    const char * ROIDescription = 0 )
```

### 10.400.3.4 AddStructureSetROIObservation()

```
void vtkRTStructSetProperties::AddStructureSetROIObservation (
    int refnumber,
    int observationnumber,
    const char * rtroiinterpretedtype,
    const char * roiinterpreter,
    const char * roiobservationlabel = 0 )
```

### 10.400.3.5 Clear()

```
virtual void vtkRTStructSetProperties::Clear ( ) [virtual]
```

#### 10.400.3.6 DeepCopy()

```
virtual void vtkRTStructSetProperties::DeepCopy (
    vtkRTStructSetProperties * p ) [virtual]
```

#### 10.400.3.7 GetContourReferencedFrameOfReferenceClassUID()

```
const char * vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceClassUID (
    vtkIdType pdnum,
    vtkIdType id )
```

#### 10.400.3.8 GetContourReferencedFrameOfReferenceInstanceUID()

```
const char * vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceInstanceUID (
    vtkIdType pdnum,
    vtkIdType id )
```

#### 10.400.3.9 GetNumberOfContourReferencedFrameOfReferences() [1/2]

```
vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences ( )
```

#### 10.400.3.10 GetNumberOfContourReferencedFrameOfReferences() [2/2]

```
vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences (
    vtkIdType pdnum )
```

#### 10.400.3.11 GetNumberOfReferencedFrameOfReferences()

```
vtkIdType vtkRTStructSetProperties::GetNumberOfReferencedFrameOfReferences ( )
```

#### 10.400.3.12 GetNumberOfStructureSetROIs()

```
vtkIdType vtkRTStructSetProperties::GetNumberOfStructureSetROIs ( )
```

#### 10.400.3.13 GetReferencedFrameOfReferenceClassUID()

```
const char * vtkRTStructSetProperties::GetReferencedFrameOfReferenceClassUID (
    vtkIdType id )
```

**10.400.3.14 GetReferencedFrameOfReferenceInstanceUID()**

```
const char * vtkRTStructSetProperties::GetReferencedFrameOfReferenceInstanceUID (
    vtkIdType id )
```

**10.400.3.15 GetStructureSetObservationNumber()**

```
int vtkRTStructSetProperties::GetStructureSetObservationNumber (
    vtkIdType id )
```

**10.400.3.16 GetStructureSetROIDescription()**

```
const char * vtkRTStructSetProperties::GetStructureSetROIDescription (
    vtkIdType id )
```

**10.400.3.17 GetStructureSetROIGenerationAlgorithm()**

```
const char * vtkRTStructSetProperties::GetStructureSetROIGenerationAlgorithm (
    vtkIdType )
```

**10.400.3.18 GetStructureSetROIName()**

```
const char * vtkRTStructSetProperties::GetStructureSetROIName (
    vtkIdType )
```

**10.400.3.19 GetStructureSetROINumber()**

```
int vtkRTStructSetProperties::GetStructureSetROINumber (
    vtkIdType id )
```

**10.400.3.20 GetStructureSetROIObservationLabel()**

```
const char * vtkRTStructSetProperties::GetStructureSetROIObservationLabel (
    vtkIdType id )
```

**10.400.3.21 GetStructureSetROIRefFrameRefUID()**

```
const char * vtkRTStructSetProperties::GetStructureSetROIRefFrameRefUID (
    vtkIdType )
```

**10.400.3.22 GetStructureSetRTROIInterpretedType()**

```
const char * vtkRTStructSetProperties::GetStructureSetRTROIInterpretedType (
    vtkIdType id )
```

**10.400.3.23 New()**

```
static vtkRTStructSetProperties * vtkRTStructSetProperties::New ( ) [static]
```

**Examples**

[GenerateRTSTRUCT.cxx](#).

**10.400.3.24 PrintSelf()**

```
void vtkRTStructSetProperties::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

**10.400.3.25 vtkGetStringMacro() [1/9]**

```
vtkRTStructSetProperties::vtkGetStringMacro (
    ReferenceFrameOfReferenceUID )
```

**10.400.3.26 vtkGetStringMacro() [2/9]**

```
vtkRTStructSetProperties::vtkGetStringMacro (
    ReferenceSeriesInstanceUID )
```

**10.400.3.27 vtkGetStringMacro() [3/9]**

```
vtkRTStructSetProperties::vtkGetStringMacro (
    SeriesInstanceUID )
```

**10.400.3.28 vtkGetStringMacro() [4/9]**

```
vtkRTStructSetProperties::vtkGetStringMacro (
    SOPInstanceUID )
```

**10.400.3.29 vtkGetStringMacro()** [5/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
    StructureSetDate )
```

**10.400.3.30 vtkGetStringMacro()** [6/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
    StructureSetLabel )
```

**10.400.3.31 vtkGetStringMacro()** [7/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
    StructureSetName )
```

**10.400.3.32 vtkGetStringMacro()** [8/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
    StructureSetTime )
```

**10.400.3.33 vtkGetStringMacro()** [9/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
    StudyInstanceUID )
```

**10.400.3.34 vtkSetStringMacro()** [1/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    ReferenceFrameOfReferenceUID )
```

**10.400.3.35 vtkSetStringMacro()** [2/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    ReferenceSeriesInstanceUID )
```

**10.400.3.36 vtkSetStringMacro()** [3/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    SeriesInstanceUID )
```

**10.400.3.37 vtkSetStringMacro()** [4/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    SOPInstanceUID )
```

**10.400.3.38 vtkSetStringMacro()** [5/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    StructureSetDate )
```

**10.400.3.39 vtkSetStringMacro()** [6/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    StructureSetLabel )
```

**10.400.3.40 vtkSetStringMacro()** [7/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    StructureSetName )
```

**10.400.3.41 vtkSetStringMacro()** [8/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    StructureSetTime )
```

**10.400.3.42 vtkSetStringMacro()** [9/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    StudyInstanceUID )
```

**10.400.3.43 vtkTypeMacro()**

```
vtkRTStructSetProperties::vtkTypeMacro (
    vtkRTStructSetProperties ,
    vtkObject )
```

**10.400.4 Member Data Documentation****10.400.4.1 Internals**

```
vtkRTStructSetPropertiesInternals* vtkRTStructSetProperties::Internals [protected]
```

#### 10.400.4.2 ReferenceFrameOfReferenceUID

```
char* vtkRTStructSetProperties::ReferenceFrameOfReferenceUID [protected]
```

#### 10.400.4.3 ReferenceSeriesInstanceUID

```
char* vtkRTStructSetProperties::ReferenceSeriesInstanceUID [protected]
```

#### 10.400.4.4 SeriesInstanceUID

```
char* vtkRTStructSetProperties::SeriesInstanceUID [protected]
```

#### 10.400.4.5 SOPInstanceUID

```
char* vtkRTStructSetProperties::SOPInstanceUID [protected]
```

#### 10.400.4.6 StructureSetDate

```
char* vtkRTStructSetProperties::StructureSetDate [protected]
```

#### 10.400.4.7 StructureSetLabel

```
char* vtkRTStructSetProperties::StructureSetLabel [protected]
```

#### 10.400.4.8 StructureSetName

```
char* vtkRTStructSetProperties::StructureSetName [protected]
```

#### 10.400.4.9 StructureSetTime

```
char* vtkRTStructSetProperties::StructureSetTime [protected]
```

#### 10.400.4.10 StudyInstanceUID

```
char* vtkRTStructSetProperties::StudyInstanceUID [protected]
```

The documentation for this class was generated from the following file:

- [vtkRTStructSetProperties.h](#)

## 10.401 gdcm::Waveform Class Reference

[Waveform](#) class.

```
#include <gdcmWaveform.h>
```

### Public Member Functions

- [Waveform](#) ()=default

### 10.401.1 Detailed Description

[Waveform](#) class.

### 10.401.2 Constructor & Destructor Documentation

#### 10.401.2.1 Waveform()

```
gdcm::Waveform::Waveform ( ) [default]
```

The documentation for this class was generated from the following file:

- [gdcmWaveform.h](#)

## 10.402 gdcm::WLMFindQuery Class Reference

PatientRootQuery.

```
#include <gdcmWLMFindQuery.h>
```





## Public Member Functions inherited from [gdcm::BaseQuery](#)

- [~BaseQuery](#) () override
- void [AddQueryDataSet](#) (const [DataSet](#) &ds)
- [DataSet](#) & [GetQueryDataSet](#) ()
- [DataSet](#) const & [GetQueryDataSet](#) () const  
*Set/Get the internal representation of the query as a [DataSet](#).*
- std::string [GetSOPInstanceUID](#) () const
- void [Print](#) (std::ostream &os) const override
- void [SetSearchParameter](#) (const std::string &inKeyword, const std::string &inValue)
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const std::string &inValue)
- void [SetSOPInstanceUID](#) (const std::string &iSopInstanceUID)
- const std::ostream & [WriteHelpFile](#) (std::ostream &os)
- bool [WriteQuery](#) (const std::string &inFileName)

## Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)  
*Special requirement for copy/cstor, assignment operator.*
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

## Protected Member Functions

- [DataSet](#) [GetValidDataSet](#) () const

## Protected Member Functions inherited from [gdcm::BaseRootQuery](#)

- [BaseRootQuery](#) ()

## Protected Member Functions inherited from [gdcm::BaseQuery](#)

- [BaseQuery](#) ()
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const [DictEntry](#) &inDictEntry, const std::string &inValue)
- bool [ValidDataSet](#) (const [DataSet](#) &dataSetToValid, const [DataSet](#) &dataSetReference) const

## Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

## Friends

- class [QueryFactory](#)

## Additional Inherited Members

### Static Public Member Functions inherited from [gdcm::BaseRootQuery](#)

- static [QueryBase](#) \* [Construct](#) ([ERootType](#) inRootType, [EQueryLevel](#) ql) level)
- static int [GetQueryLevelFromString](#) (const char \*str)
- static const char \* [GetQueryLevelString](#) ([EQueryLevel](#) ql)

### Protected Attributes inherited from [gdcm::BaseRootQuery](#)

- std::string [mHelpDescription](#)
- [QueryImage](#) mImage
- [QueryPatient](#) mPatient
- [ERootType](#) mRootType
- [QuerySeries](#) mSeries
- [QueryStudy](#) mStudy

### Protected Attributes inherited from [gdcm::BaseQuery](#)

- [DataSet](#) mDataSet
- std::string [mSopInstanceUID](#)

## 10.402.1 Detailed Description

PatientRootQuery.

contains: the class which will produce a dataset for c-find with patient root

## 10.402.2 Constructor & Destructor Documentation

### 10.402.2.1 WLMFindQuery()

```
gdcm::WLMFindQuery::WLMFindQuery ( )
```

## 10.402.3 Member Function Documentation

### 10.402.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::WLMFindQuery::GetAbstractSyntaxUID ( ) const [override], [virtual]
```

Implements [gdcm::BaseQuery](#).

### 10.402.3.2 GetTagListByLevel()

```
std::vector< Tag > gdcm::WLMFindQuery::GetTagListByLevel (
    const EQueryLevel & inQueryLevel ) [override], [virtual]
```

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implements [gdcm::BaseRootQuery](#).

### 10.402.3.3 GetValidDataSet()

```
DataSet gdcm::WLMFindQuery::GetValidDataSet ( ) const [protected]
```

### 10.402.3.4 InitializeDataSet()

```
void gdcm::WLMFindQuery::InitializeDataSet (
    const EQueryLevel & inQueryLevel ) [override], [virtual]
```

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcm4che

Implements [gdcm::BaseRootQuery](#).

### 10.402.3.5 ValidateQuery()

```
bool gdcm::WLMFindQuery::ValidateQuery (
    bool inStrict = true ) const [override], [virtual]
```

have to be able to ensure that 0x8,0x52 is set (which will be true if InitializeDataSet is called...) that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional) by default, this function checks to see if the query is for finding, which is more permissive than for moving. For moving, only the unique tags are allowed. 10 Jan 2011: adding in the 'strict' mode. according to the standard (at least, how I've read it), only tags for a particular level should be allowed in a particular query (ie, just series level tags in a series level query). However, it seems that dcm4chee doesn't share that interpretation. So, if 'inStrict' is false, then tags from the current level and all higher levels are now considered valid. So, if you're doing a non-strict series-level query, tags from the patient and study level can be passed along as well.

Implements [gdcm::BaseRootQuery](#).

## 10.402.4 Friends And Related Symbol Documentation

### 10.402.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

The documentation for this class was generated from the following file:

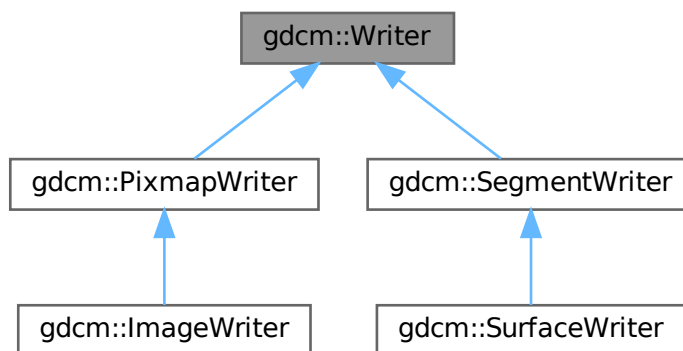
- [gdcmWLMFindQuery.h](#)

## 10.403 gdcm::Writer Class Reference

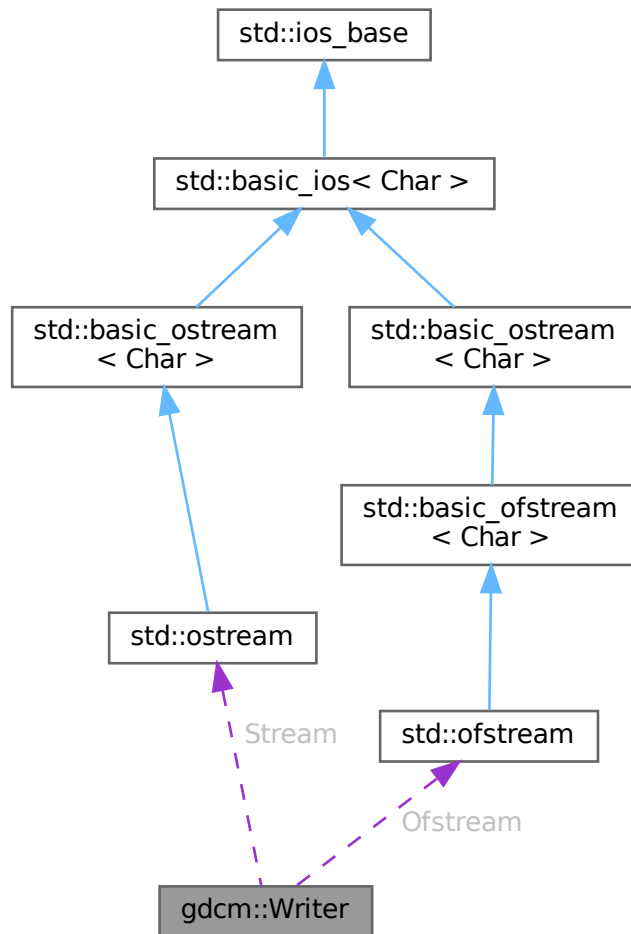
[Writer](#) ala DOM (Document [Object](#) Model)

```
#include <gdcmWriter.h>
```

Inheritance diagram for gdcm::Writer:



Collaboration diagram for `gdcm::Writer`:



## Public Member Functions

- [Writer](#) ()
- virtual [~Writer](#) ()
- void [CheckFileMetaInformationOff](#) ()
- void [CheckFileMetaInformationOn](#) ()
- [File](#) & [GetFile](#) ()
- void [SetCheckFileMetaInformation](#) (bool b)  
*Undocumented function, do not use (= leave default)*
- void [SetFile](#) (const [File](#) &f)  
*Set/Get the DICOM file ([DataSet](#) + Header)*
- void [SetFileName](#) (const char \*filename\_native)

*Set the filename of DICOM file to write:*

- void [SetStream](#) (std::ostream &output\_stream)

*Set user ostream buffer.*

- virtual bool [Write](#) ()

*Main function to tell the writer to write.*

### Protected Member Functions

- bool [GetCheckFileMetaInformation](#) () const
- std::ostream \* [GetStreamPtr](#) () const
- void [SetWriteDataSetOnly](#) (bool b)

### Protected Attributes

- std::ofstream \* [Ofstream](#)
- std::ostream \* [Stream](#)

### Friends

- class [StreamImageWriter](#)

## 10.403.1 Detailed Description

[Writer](#) ala DOM (Document [Object](#) Model)

This class is a non-validating writer, it will only performs well- formedness check only.

Detailed description here To avoid GDCM being yet another broken DICOM lib we try to be user level and avoid writing illegal stuff (odd length, non-zero value for [Item](#) start/end length ...) Therefore you cannot (well unless you are really smart) write DICOM with even length tag. All the checks are consider basics:

- Correct Meta Information Header (see [gdcm::FileMetaInformation](#))
- Zero value for [Item](#) Length (0xfffe, 0xe00d/0xe0dd)
- Even length for any elements
- Alphabetical order for elements (guaranteed by design of internals)
- 32bits [VR](#) will be rewritten with 00

### Warning

[gdcm::Writer](#) cannot write a [DataSet](#) if no SOP Instance UID (0008,0018) is found, unless a [DICOMDIR](#) is being written out

### See also

[Reader DataSet File](#)

### Examples

[BasicAnonymizer.cs](#), [ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [Cleaner.cs](#), [ClinicalTrialAnnotate.cxx](#), [ClinicalTrialIdentificationWorkflow.cs](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [DeriveSeries.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GenerateDICOMDIR.cs](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [ManipulateFile.cs](#), [NewSequence.cs](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReformatFile.cs](#), [StreamImageReaderTest.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

## 10.403.2 Constructor & Destructor Documentation

### 10.403.2.1 Writer()

```
gdcm::Writer::Writer ( )
```

### 10.403.2.2 ~Writer()

```
virtual gdcm::Writer::~~Writer ( ) [virtual]
```

## 10.403.3 Member Function Documentation

### 10.403.3.1 CheckFileMetaInformationOff()

```
void gdcm::Writer::CheckFileMetaInformationOff ( ) [inline]
```

#### Examples

[CreateFakeRTDOSE.cxx](#), [FixBrokenJ2K.cxx](#), and [HelloWorld.cxx](#).

### 10.403.3.2 CheckFileMetaInformationOn()

```
void gdcm::Writer::CheckFileMetaInformationOn ( ) [inline]
```

### 10.403.3.3 GetCheckFileMetaInformation()

```
bool gdcm::Writer::GetCheckFileMetaInformation ( ) const [inline], [protected]
```

### 10.403.3.4 GetFile()

```
File & gdcm::Writer::GetFile ( ) [inline]
```

#### Examples

[CreateJPIPDataSet.cxx](#), [EncapsulateFileInRawData.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GetSubSequenceData.cxx](#), [MpegVideoInfo.cs](#), [QIDO-RS.cxx](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

### 10.403.3.5 GetStreamPtr()

```
std::ostream * gdcm::Writer::GetStreamPtr ( ) const [inline], [protected]
```



### 10.403.3.6 SetCheckFileMetaInformation()

```
void gdcm::Writer::SetCheckFileMetaInformation (
    bool b ) [inline]
```

Undocumented function, do not use (= leave default)

#### Examples

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), and [PatchFile.cxx](#).

### 10.403.3.7 SetFile()

```
void gdcm::Writer::SetFile (
    const File & f ) [inline]
```

Set/Get the DICOM file ([DataSet](#) + Header)

#### Examples

[BasicAnonymizer.cs](#), [BasicImageAnonymizer.cs](#), [ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [Cleaner.cs](#), [ClinicalTrialAnnotate.cxx](#), [ClinicalTrialIdentificationWorkflow.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [CreateFakeRTDOSE.cxx](#), [DecompressImage.cs](#), [DeriveSeries.cxx](#), [DuplicatePCDE.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GenerateDICOMDIR.cs](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [ManipulateFile.cs](#), [MergeTwoFiles.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [ReformatFile.cs](#), [StandardizeFiles.cs](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

### 10.403.3.8 SetFileName()

```
void gdcm::Writer::SetFileName (
    const char * filename_native )
```

Set the filename of DICOM file to write:

#### Examples

[BasicAnonymizer.cs](#), [BasicImageAnonymizer.cs](#), [ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [Cleaner.cs](#), [ClinicalTrialAnnotate.cxx](#), [ClinicalTrialIdentificationWorkflow.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [DecompressImage.cs](#), [DeriveSeries.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GenerateDICOMDIR.cs](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [ManipulateFile.cs](#), [MergeTwoFiles.cxx](#), [MpegVideoInfo.cs](#), [NewSequence.cs](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReformatFile.cs](#), [StandardizeFiles.cs](#), [TemplateEmptyImage.cxx](#), [csa2img.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

### 10.403.3.9 SetStream()

```
void gdcm::Writer::SetStream (
    std::ostream & output_stream ) [inline]
```

Set user ostream buffer.

### 10.403.3.10 SetWriteDataSetOnly()

```
void gdcm::Writer::SetWriteDataSetOnly (
    bool b ) [inline], [protected]
```

### 10.403.3.11 Write()

```
virtual bool gdcm::Writer::Write ( ) [virtual]
```

Main function to tell the writer to write.

Reimplemented in [gdcm::ImageWriter](#), [gdcm::PixmapWriter](#), [gdcm::SegmentWriter](#), and [gdcm::SurfaceWriter](#).

#### Examples

[BasicAnonymizer.cs](#), [ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [Cleaner.cs](#), [ClinicalTrialAnnotate.cxx](#), [ClinicalTrialIdentificationWorkflow.cs](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [DeriveSeries.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GenerateDICOMDIR.cs](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [ManipulateFile.cs](#), [NewSequence.cs](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReformatFile.cs](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

## 10.403.4 Friends And Related Symbol Documentation

### 10.403.4.1 StreamImageWriter

```
friend class StreamImageWriter [friend]
```

## 10.403.5 Member Data Documentation

### 10.403.5.1 Ofstream

```
std::ofstream* gdcm::Writer::Ofstream [protected]
```

### 10.403.5.2 Stream

```
std::ostream* gdcm::Writer::Stream [protected]
```

The documentation for this class was generated from the following file:

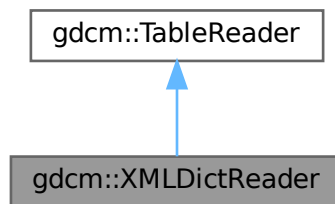
- [gdcmWriter.h](#)

## 10.404 gdcm::XMLDictReader Class Reference

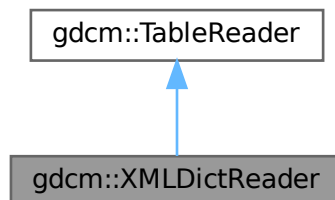
Class for representing a [XMLDictReader](#).

```
#include <gdcmXMLDictReader.h>
```

Inheritance diagram for gdcm::XMLDictReader:



Collaboration diagram for gdcm::XMLDictReader:



### Public Member Functions

- [XMLDictReader](#) ()
- [~XMLDictReader](#) ()
- void [CharacterDataHandler](#) (const char \*data, int length)
- void [EndElement](#) (const char \*name)
- const [Dict](#) & [GetDict](#) ()
- void [StartElement](#) (const char \*name, const char \*\*atts)

### Public Member Functions inherited from [gdcm::TableReader](#)

- [TableReader](#) (Defs &defs)
- virtual [~TableReader](#) ()=default
- const [Defs](#) & [GetDefs](#) () const
- const char \* [GetFilename](#) ()
- void [HandleIOD](#) (const char \*\*atts)
- void [HandleIODEntry](#) (const char \*\*atts)
- void [HandleMacro](#) (const char \*\*atts)
- void [HandleMacroEntry](#) (const char \*\*atts)
- void [HandleMacroEntryDescription](#) (const char \*\*atts)
- void [HandleModule](#) (const char \*\*atts)
- void [HandleModuleEntry](#) (const char \*\*atts)
- void [HandleModuleEntryDescription](#) (const char \*\*atts)
- void [HandleModuleInclude](#) (const char \*\*atts)
- int [Read](#) ()
- void [SetFilename](#) (const char \*filename)

### Protected Member Functions

- void [HandleDescription](#) (const char \*\*atts)
- void [HandleEntry](#) (const char \*\*atts)

## 10.404.1 Detailed Description

Class for representing a [XMLDictReader](#).

#### Note

bla Will read the DICOMV3.xml file

## 10.404.2 Constructor & Destructor Documentation

### 10.404.2.1 XMLDictReader()

```
gdcm::XMLDictReader::XMLDictReader ( )
```

### 10.404.2.2 ~XMLDictReader()

```
gdcm::XMLDictReader::~XMLDictReader ( ) [inline]
```

## 10.404.3 Member Function Documentation

### 10.404.3.1 CharacterDataHandler()

```
void gdcm::XMLDictReader::CharacterDataHandler (
    const char * data,
    int length ) [virtual]
```

Reimplemented from [gdcm::TableReader](#).

### 10.404.3.2 EndElement()

```
void gdcm::XMLDictReader::EndElement (
    const char * name ) [virtual]
```

Reimplemented from [gdcm::TableReader](#).

### 10.404.3.3 GetDict()

```
const Dict & gdcm::XMLDictReader::GetDict ( ) [inline]
```

### 10.404.3.4 HandleDescription()

```
void gdcm::XMLDictReader::HandleDescription (
    const char ** atts ) [protected]
```

### 10.404.3.5 HandleEntry()

```
void gdcm::XMLDictReader::HandleEntry (
    const char ** atts ) [protected]
```

### 10.404.3.6 StartElement()

```
void gdcm::XMLDictReader::StartElement (
    const char * name,
    const char ** atts ) [virtual]
```

Reimplemented from [gdcm::TableReader](#).

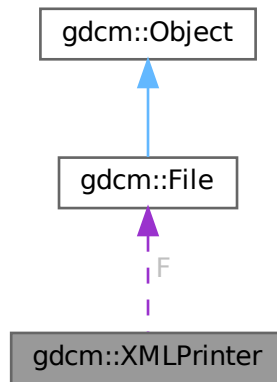
The documentation for this class was generated from the following file:

- [gdcmXMLDictReader.h](#)

## 10.405 gdcm::XMLPrinter Class Reference

```
#include <gdcmXMLPrinter.h>
```

Collaboration diagram for gdcm::XMLPrinter:



### Public Types

- enum `PrintStyles` {  
     `OnlyUUID` = 0 ,  
     `LOADBULKDATA` = 1 }

### Public Member Functions

- `XMLPrinter ()`
- virtual `~XMLPrinter ()`
- `PrintStyles GetPrintStyle ()` const
- virtual void `HandleBulkData` (const char \*uuid, const `TransferSyntax` &ts, const char \*bulkdata, size\_t bulklen)
- void `Print` (std::ostream &os)
- void `PrintDataSet` (const `DataSet` &ds, const `TransferSyntax` &ts, std::ostream &os)
- void `SetFile` (`File` const &f)
- void `SetStyle` (`PrintStyles` ps)

### Protected Member Functions

- VR `PrintDataElement` (std::ostream &os, const `Dicts` &dicts, const `DataSet` &ds, const `DataElement` &de, const `TransferSyntax` &ts)
- void `PrintSQ` (const `SequenceOfItems` \*sqi, const `TransferSyntax` &ts, std::ostream &os)

**Protected Attributes**

- const [File](#) \* [F](#)
- [PrintStyles](#) [PrintStyle](#)

**10.405.1 Member Enumeration Documentation****10.405.1.1 PrintStyles**

```
enum gdcm::XMLPrinter::PrintStyles
```

Enumerator

OnlyUUID	
LOADBULKDATA	

**10.405.2 Constructor & Destructor Documentation****10.405.2.1 XMLPrinter()**

```
gdcm::XMLPrinter::XMLPrinter ( )
```

**10.405.2.2 ~XMLPrinter()**

```
virtual gdcm::XMLPrinter::~~XMLPrinter ( ) [virtual]
```

**10.405.3 Member Function Documentation****10.405.3.1 GetPrintStyle()**

```
PrintStyles gdcm::XMLPrinter::GetPrintStyle ( ) const [inline]
```

**10.405.3.2 HandleBulkData()**

```
virtual void gdcm::XMLPrinter::HandleBulkData (
    const char * uuid,
    const TransferSyntax & ts,
    const char * bulkdata,
    size_t bulklen ) [virtual]
```

Virtual function mechanism to allow application programmer to override the default mechanism for BulkData handling. By default GDCM will simply discard the BulkData and only write the UUID

### 10.405.3.3 Print()

```
void gdcm::XMLPrinter::Print (
    std::ostream & os )
```

### 10.405.3.4 PrintDataElement()

```
VR gdcm::XMLPrinter::PrintDataElement (
    std::ostream & os,
    const Dicts & dicts,
    const DataSet & ds,
    const DataElement & de,
    const TransferSyntax & ts ) [protected]
```

### 10.405.3.5 PrintDataSet()

```
void gdcm::XMLPrinter::PrintDataSet (
    const DataSet & ds,
    const TransferSyntax & ts,
    std::ostream & os )
```

### 10.405.3.6 PrintSQ()

```
void gdcm::XMLPrinter::PrintSQ (
    const SequenceOfItems * sqi,
    const TransferSyntax & ts,
    std::ostream & os ) [protected]
```

### 10.405.3.7 SetFile()

```
void gdcm::XMLPrinter::SetFile (
    File const & f ) [inline]
```

### 10.405.3.8 SetStyle()

```
void gdcm::XMLPrinter::SetStyle (
    PrintStyles ps ) [inline]
```

## 10.405.4 Member Data Documentation

### 10.405.4.1 F

```
const File* gdcm::XMLPrinter::F [protected]
```



#### 10.405.4.2 PrintStyle

`PrintStyle` gdcm::XMLPrinter::PrintStyle [protected]

The documentation for this class was generated from the following file:

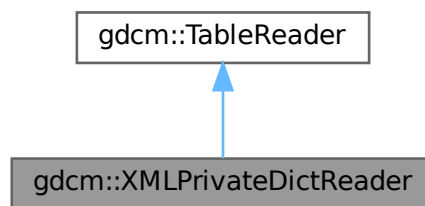
- [gdcmXMLPrinter.h](#)

### 10.406 gdcm::XMLPrivateDictReader Class Reference

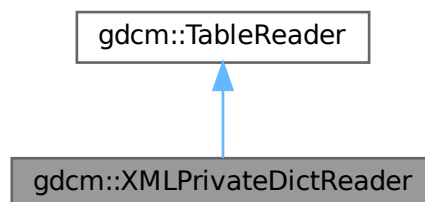
Class for representing a [XMLPrivateDictReader](#).

```
#include <gdcmXMLPrivateDictReader.h>
```

Inheritance diagram for gdcm::XMLPrivateDictReader:



Collaboration diagram for gdcm::XMLPrivateDictReader:



## Public Member Functions

- [XMLPrivateDictReader](#) ()
- [~XMLPrivateDictReader](#) ()
- void [CharacterDataHandler](#) (const char \*data, int length)
- void [EndElement](#) (const char \*name)
- const [PrivateDict](#) & [GetPrivateDict](#) ()
- void [StartElement](#) (const char \*name, const char \*\*atts)

## Public Member Functions inherited from [gdcm::TableReader](#)

- [TableReader](#) (Defs &defs)
- virtual [~TableReader](#) ()=default
- const [Defs](#) & [GetDefs](#) () const
- const char \* [GetFilename](#) ()
- void [HandleIOD](#) (const char \*\*atts)
- void [HandleIODEntry](#) (const char \*\*atts)
- void [HandleMacro](#) (const char \*\*atts)
- void [HandleMacroEntry](#) (const char \*\*atts)
- void [HandleMacroEntryDescription](#) (const char \*\*atts)
- void [HandleModule](#) (const char \*\*atts)
- void [HandleModuleEntry](#) (const char \*\*atts)
- void [HandleModuleEntryDescription](#) (const char \*\*atts)
- void [HandleModuleInclude](#) (const char \*\*atts)
- int [Read](#) ()
- void [SetFilename](#) (const char \*filename)

## Protected Member Functions

- void [HandleDescription](#) (const char \*\*atts)
- void [HandleEntry](#) (const char \*\*atts)

### 10.406.1 Detailed Description

Class for representing a [XMLPrivateDictReader](#).

#### Note

bla Will read the Private.xml file

### 10.406.2 Constructor & Destructor Documentation

#### 10.406.2.1 XMLPrivateDictReader()

```
gdcm::XMLPrivateDictReader::XMLPrivateDictReader ( )
```

### 10.406.2.2 ~XMLPrivateDictReader()

```
gdcm::XMLPrivateDictReader::~XMLPrivateDictReader ( ) [inline]
```

## 10.406.3 Member Function Documentation

### 10.406.3.1 CharacterDataHandler()

```
void gdcm::XMLPrivateDictReader::CharacterDataHandler (
    const char * data,
    int length ) [virtual]
```

Reimplemented from [gdcm::TableReader](#).

### 10.406.3.2 EndElement()

```
void gdcm::XMLPrivateDictReader::EndElement (
    const char * name ) [virtual]
```

Reimplemented from [gdcm::TableReader](#).

### 10.406.3.3 GetPrivateDict()

```
const PrivateDict & gdcm::XMLPrivateDictReader::GetPrivateDict ( ) [inline]
```

### 10.406.3.4 HandleDescription()

```
void gdcm::XMLPrivateDictReader::HandleDescription (
    const char ** atts ) [protected]
```

### 10.406.3.5 HandleEntry()

```
void gdcm::XMLPrivateDictReader::HandleEntry (
    const char ** atts ) [protected]
```

### 10.406.3.6 StartElement()

```
void gdcm::XMLPrivateDictReader::StartElement (
    const char * name,
    const char ** atts ) [virtual]
```

Reimplemented from [gdcm::TableReader](#).

The documentation for this class was generated from the following file:

- [gdcmXMLPrivateDictReader.h](#)



# Chapter 11

## File Documentation

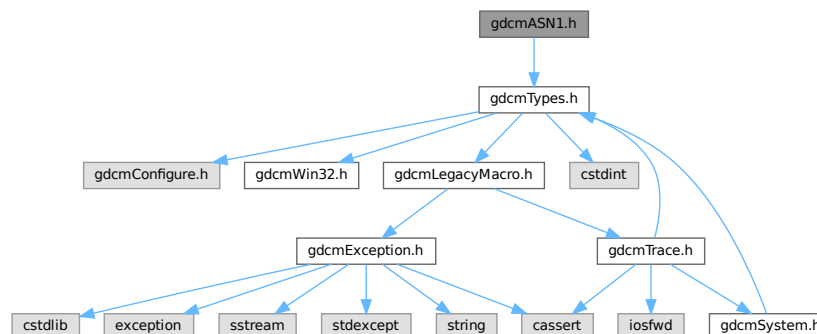
### 11.1 README.txt File Reference

### 11.2 TestsList.txt File Reference

### 11.3 gdcmASN1.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmASN1.h:



### Classes

- class `gdcm::ASN1`  
Class for `ASN1`.

## Namespaces

- namespace [gdcm](#)

## 11.4 gdcmASN1.h

[Go to the documentation of this file.](#)

```

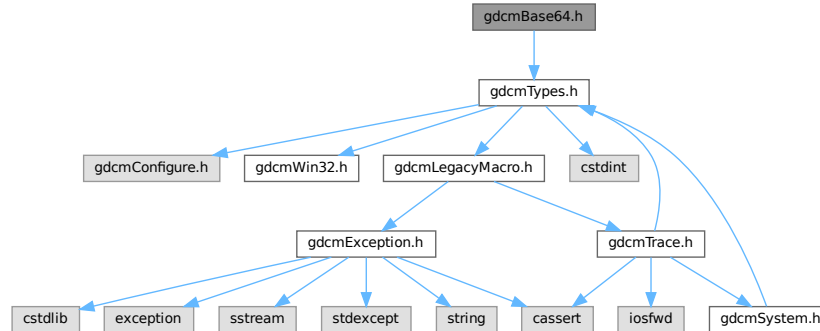
00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMASN1_H
00015 #define GDCMASN1_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021     //-----
00022     class ASN1Internals;
00023     class GDCM_EXPORT ASN1
00024     {
00025     public :
00026         ASN1();
00027         ~ASN1();
00028
00029         static bool ParseDumpFile(const char *filename);
00030
00031         static bool ParseDump(const char *array, size_t length);
00032
00033         ASN1(const ASN1&) = delete;
00034         void operator=(const ASN1&) = delete;
00035     protected:
00036         int TestPBKDF2();
00037
00038     private:
00039         ASN1Internals *Internals;
00040     };
00041 } // end namespace gdcm
00042 //-----
00043 #endif //GDCMASN1_H

```

## 11.5 gdcmBase64.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmBase64.h:



### Classes

- class [gdcm::Base64](#)  
Class for [Base64](#).

### Namespaces

- namespace [gdcm](#)

## 11.6 gdcmBase64.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMBASE64_H
00015 #define GDCMBASE64_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021
00022   class GDCM_EXPORT Base64
00023   {
00024   public:

```

```

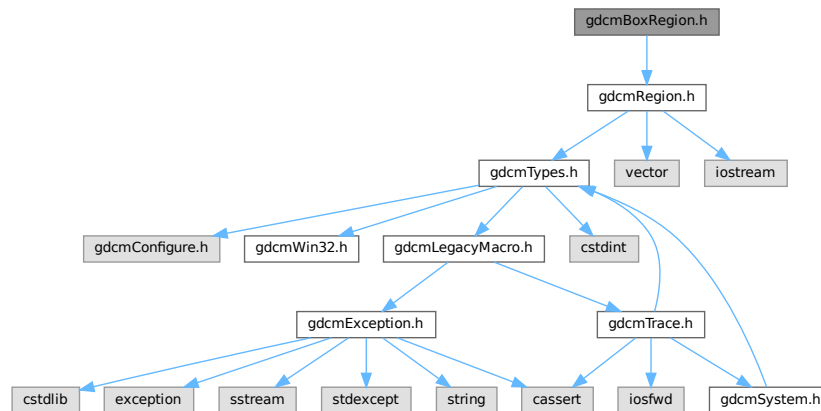
00028
00032  static size_t GetEncodeLength(const char *src, size_t srclen );
00033
00045  static size_t Encode( char *dst, size_t dlen, const char *src, size_t slen );
00046
00050  static size_t GetDecodeLength( const char *src, size_t len );
00051
00062  static size_t Decode( char *dst, size_t dlen, const char *src, size_t slen );
00063
00064  Base64(const Base64&) = delete;
00065  void operator=(const Base64&) = delete;
00066 };
00067
00068 } // end namespace gdcm
00069
00070 #endif // GDCMBASE64_H

```

## 11.7 gdcmBoxRegion.h File Reference

```
#include "gdcmRegion.h"
```

Include dependency graph for gdcmBoxRegion.h:



### Classes

- class `gdcm::BoxRegion`  
Class for manipulation box region.

### Namespaces

- namespace `gdcm`



## 11.8 gdcmBoxRegion.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMBOXREGION_H
00015  #define GDCMBOXREGION_H
00016
00017  #include "gdcmRegion.h"
00018
00019  namespace gdcm
00020  {
00021  class BoxRegionInternals;
00022  //-----
00023  class GDCM_EXPORT BoxRegion : public Region
00024  {
00025  public :
00026    BoxRegion();
00027    ~BoxRegion() override;
00028
00029    void SetDomain(unsigned int xmin, unsigned int xmax,
00030                  unsigned int ymin, unsigned int ymax,
00031                  unsigned int zmin, unsigned int zmax);
00032
00033    unsigned int GetXMin() const;
00034    unsigned int GetXMax() const;
00035    unsigned int GetYMin() const;
00036    unsigned int GetYMax() const;
00037    unsigned int GetZMin() const;
00038    unsigned int GetZMax() const;
00039
00040    // Satisfy pure virtual parent class
00041    Region *Clone() const override;
00042    bool Empty() const override;
00043    bool IsValid() const override;
00044    size_t Area() const override;
00045    BoxRegion ComputeBoundingBox() override;
00046
00047    void Print(std::ostream &os = std::cout) const override;
00048
00049    static BoxRegion BoundingBox(BoxRegion const & b1, BoxRegion const & b2 );
00050
00051    BoxRegion(const BoxRegion&);
00052    void operator=(const BoxRegion&);
00053  private:
00054    BoxRegionInternals *Internals;
00055  };
00056
00057  } // end namespace gdcm
00058  //-----
00059  #endif //GDCMREGION_H

```

## 11.9 gdcmByteSwap.h File Reference

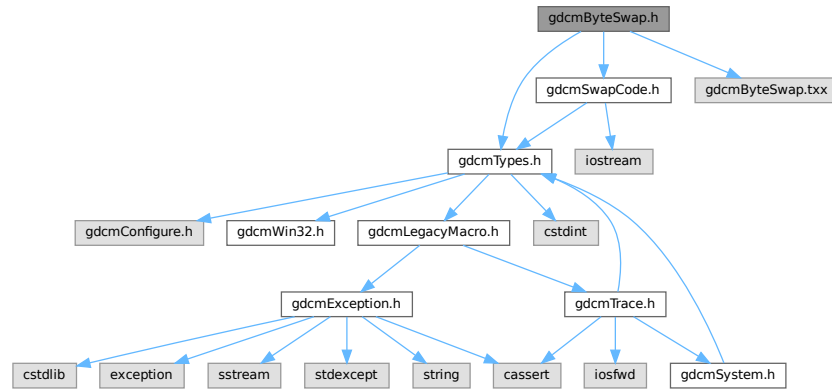
```

#include "gdcmTypes.h"
#include "gdcmSwapCode.h"

```

```
#include "gdcmByteSwap.txx"
```

Include dependency graph for gdcmByteSwap.h:



## Classes

- class [gdcm::ByteSwap< T >](#)  
*ByteSwap.*

## Namespaces

- namespace [gdcm](#)

## 11.10 gdcmByteSwap.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMBYTESWAP_H
00015 #define GDCMBYTESWAP_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmSwapCode.h"
00019
00020 namespace gdcm
00021 {
00022
00023   template<class T>
00024   class ByteSwap
00025   {

```

```

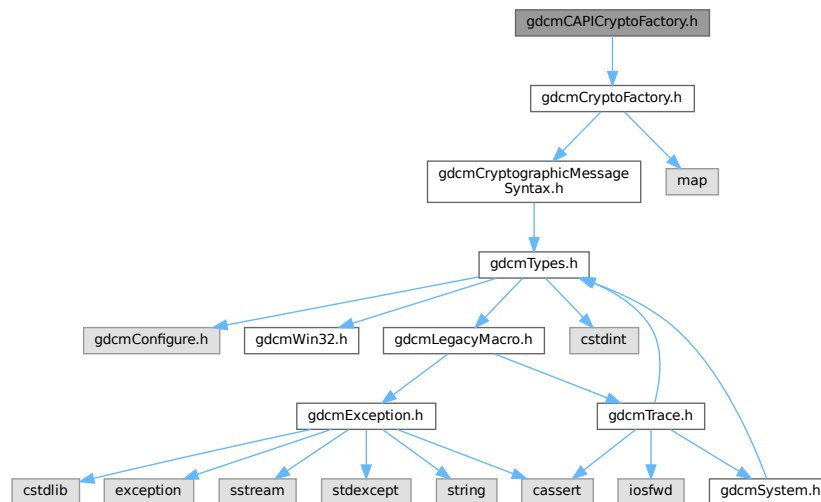
00032 public:
00033     static bool SystemIsBigEndian ();
00034     static bool SystemIsLittleEndian ();
00035
00036     static void Swap(T &p);
00037     static void SwapFromSwapCodeIntoSystem(T &p, SwapCode const &sc);
00038     static void SwapRange(T *p, unsigned int num);
00039     static void SwapRangeFromSwapCodeIntoSystem(T *p, SwapCode const &sc,
00040         std::streamoff num);
00041
00042 protected:
00043     // ByteSwap() {}
00044     // ~ByteSwap() {}
00045
00046 private:
00047
00048 };
00049
00050 } // end namespace gdcm
00051
00052 #include "gdcmByteSwap.txx"
00053
00054 #endif //GDCMBYTESWAP_H

```

## 11.11 gdcmCAPICryptoFactory.h File Reference

#include "gdcmCryptoFactory.h"

Include dependency graph for gdcmCAPICryptoFactory.h:



### Classes

- class `gdcm::CAPICryptoFactory`

### Namespaces

- namespace `gdcm`

## 11.12 gdcmCAPICryptoFactory.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMCAPICRYPTOFACTORY_H
00015 #define GDCMCAPICRYPTOFACTORY_H
00016
00017 #include "gdcmCryptoFactory.h"
00018
00019 namespace gdcm
00020 {
00021
00022   class GDCM_EXPORT CAPICryptoFactory : public CryptoFactory
00023   {
00024   public:
00025     CAPICryptoFactory(CryptoLib id);
00026     CryptographicMessageSyntax* CreateCMSProvider();
00027
00028   private:
00029     CAPICryptoFactory() {}
00030   };
00031
00032 } // end namespace gdcm
00033
00034 #endif //GDCMCAPICRYPTOFACTORY_H

```

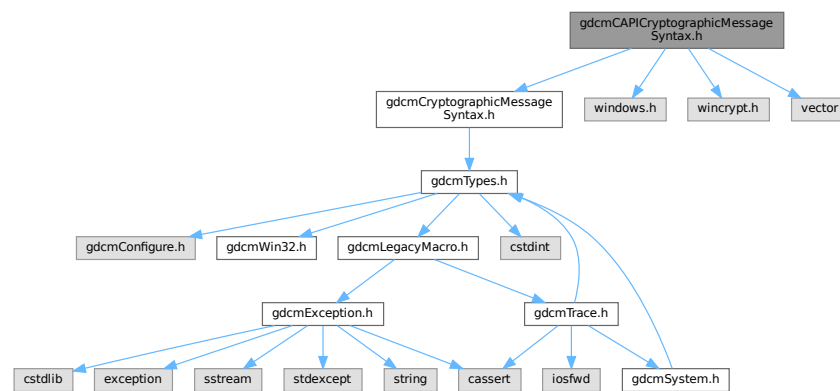
## 11.13 gdcmCAPICryptographicMessageSyntax.h File Reference

```

#include "gdcmCryptographicMessageSyntax.h"
#include <windows.h>
#include <wincrypt.h>
#include <vector>

```

Include dependency graph for gdcmCAPICryptographicMessageSyntax.h:



**Classes**

- class [gdcm::CAPICryptographicMessageSyntax](#)

**Namespaces**

- namespace [gdcm](#)

**11.14 gdcmCAPICryptographicMessageSyntax.h**

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMCAPICRYPTOGRAPHICMESSAGESYNTAX_H
00015 #define GDCMCAPICRYPTOGRAPHICMESSAGESYNTAX_H
00016
00017 #include "gdcmCryptographicMessageSyntax.h"
00018 #include <windows.h>
00019 #include <wincrypt.h>
00020 #include <vector>
00021
00022 namespace gdcm
00023 {
00024
00025 class GDCM_EXPORT CAPICryptographicMessageSyntax : public CryptographicMessageSyntax
00026 {
00027 public:
00028     CAPICryptographicMessageSyntax();
00029     ~CAPICryptographicMessageSyntax();
00030
00031     // X.509
00032     bool ParseCertificateFile( const char *filename );
00033     bool ParseKeyFile( const char *filename );
00034
00035     // PBE
00036     bool SetPassword(const char * pass, size_t passLen);
00037
00038     void SetCipherType(CipherTypes type);
00039
00040     CipherTypes GetCipherType() const;
00041
00042     bool Encrypt(char *output, size_t &outlen, const char *array, size_t len) const;
00043     bool Decrypt(char *output, size_t &outlen, const char *array, size_t len) const;
00044
00045     bool GetInitialized() const
00046     {
00047         return initialized;
00048     }
00049
00050 private:
00051     bool Initialize();
00052     static ALG_ID GetAlgIdByObjId(const char * pszObjId);
00053     static const char *GetCipherObjId() const;
00054     static void ReverseBytes(unsigned char* data, DWORD len);
00055     static bool LoadFile(const char * filename, unsigned char* & buffer, DWORD & bufLen);
00056
00057 private:
00058     bool initialized;
00059     HCRYPTPROV hProv;

```

```

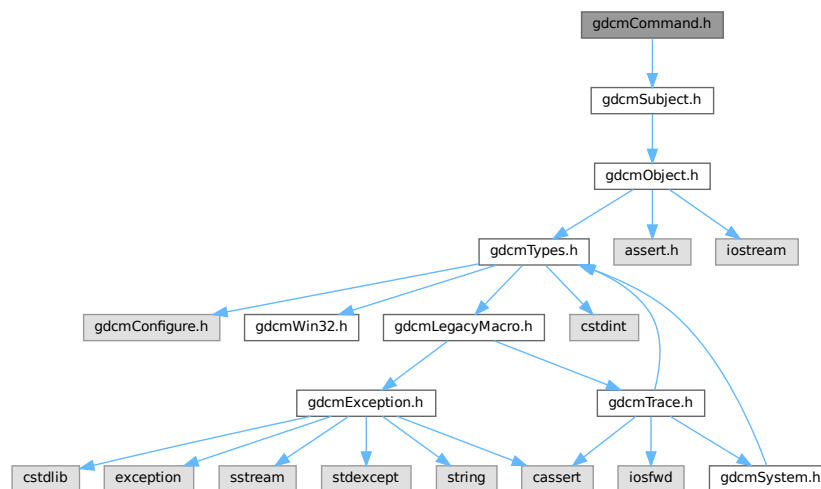
00062     std::vector<PCCERT_CONTEXT> certifList;
00063     HCRYPTKEY hRsaPrivK;
00064     CipherTypes cipherType;
00065 };
00066
00067 } // end namespace gdcM
00068
00069 #endif //GDCMCAPICRYPTOGRAPHICMESSAGESYNTAX_H

```

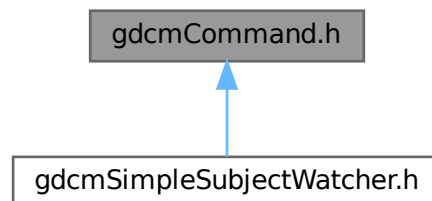
## 11.15 gdcMCommand.h File Reference

```
#include "gdcMSubject.h"
```

Include dependency graph for gdcMCommand.h:



This graph shows which files directly or indirectly include this file:



**Classes**

- class `gdcm::Command`  
*Command superclass for callback/observer methods.*
- class `gdcm::MemberCommand< T >`  
*Command subclass that calls a pointer to a member function.*
- class `gdcm::SimpleMemberCommand< T >`  
*Command subclass that calls a pointer to a member function.*

**Namespaces**

- namespace `gdcm`

**11.16 gdcmCommand.h**

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 #ifndef GDCMCOMMAND_H
00015 #define GDCMCOMMAND_H
00016
00017 #include "gdcmSubject.h"
00018
00019 namespace gdcm
00020 {
00021     class Event;
00022
00027     class GDCM_EXPORT Command : public Subject
00028     {
00029     public :
00030         Command(const Command&) = delete;
00031         void operator=(const Command&) = delete;
00032
00033         virtual void Execute(Subject *caller, const Event & event ) = 0;
00034
00040         virtual void Execute(const Subject *caller, const Event & event ) = 0;
00041
00042     protected:
00043         Command();
00044         ~Command() override;
00045     };
00046
00054     template <class T>
00055     class MemberCommand : public Command
00056     {
00057     public:
00058
00060         typedef void (T::*TMemberFunctionPointer)(Subject*, const Event &);
00061         typedef void (T::*TConstMemberFunctionPointer)(const Subject*,
00062                                                         const Event &);
00063
00065         typedef MemberCommand      Self;
00066         //typedef SmartPointer<Self>  Pointer;
00067
00068         MemberCommand(const Self&) = delete;

```

```

00069 void operator=(const Self&) = delete;
00070
00072 static SmartPointer<MemberCommand> New()
00073 {
00074     return new MemberCommand;
00075 }
00076
00078 //gdcmTypeMacro(MemberCommand,Command);
00079
00082 void SetCallbackFunction(T* object,
00083                         TMemberFunctionPointer memberFunction)
00084 {
00085     m_This = object;
00086     m_MemberFunction = memberFunction;
00087 }
00088 void SetCallbackFunction(T* object,
00089                         TConstMemberFunctionPointer memberFunction)
00090 {
00091     m_This = object;
00092     m_ConstMemberFunction = memberFunction;
00093 }
00094
00096 void Execute(Subject *caller, const Event & event ) override
00097 {
00098     if( m_MemberFunction )
00099     {
00100         ((*m_This).*(m_MemberFunction))(caller, event);
00101     }
00102 }
00103
00105 void Execute( const Subject *caller, const Event & event ) override
00106 {
00107     if( m_ConstMemberFunction )
00108     {
00109         ((*m_This).*(m_ConstMemberFunction))(caller, event);
00110     }
00111 }
00112
00113 protected:
00114
00115     T* m_This;
00116     TMemberFunctionPointer m_MemberFunction;
00117     TConstMemberFunctionPointer m_ConstMemberFunction;
00118     MemberCommand():m_MemberFunction(nullptr),m_ConstMemberFunction(nullptr) {}
00119     ~MemberCommand() override= default;
00120
00121 };
00122
00129 template <typename T>
00130 class SimpleMemberCommand : public Command
00131 {
00132 public:
00133
00135     typedef void (T::*TMemberFunctionPointer)();
00136
00138     typedef SimpleMemberCommand Self;
00139     //typedef SmartPointer<Self> Pointer;
00140
00141     SimpleMemberCommand(const Self&) = delete;
00142     void operator=(const Self&) = delete;
00143
00145     //gdcmTypeMacro(SimpleMemberCommand,Command);
00146
00148     static SmartPointer<SimpleMemberCommand> New()
00149     {
00150         return new SimpleMemberCommand;
00151     }
00152
00154     void SetCallbackFunction(T* object,
00155                             TMemberFunctionPointer memberFunction)
00156     {
00157         m_This = object;
00158         m_MemberFunction = memberFunction;
00159     }
00160
00162     void Execute(Subject *,const Event & ) override
00163     {
00164         if( m_MemberFunction )
00165         {
00166             ((*m_This).*(m_MemberFunction))();
00167         }

```



```

00168     }
00169     void Execute(const Subject *,const Event & ) override
00170     {
00171         if( m_MemberFunction )
00172         {
00173             ((*m_This).*(m_MemberFunction)) ();
00174         }
00175     }
00176
00177 protected:
00178     T* m_This;
00179     TMemberFunctionPointer m_MemberFunction;
00180     SimpleMemberCommand():m_This(nullptr),m_MemberFunction(nullptr) {}
00181     ~SimpleMemberCommand() override = default;
00182 };
00183
00184 } // end namespace gdcrypto
00185 //-----
00186 #endif //GDCMCOMMAND_H

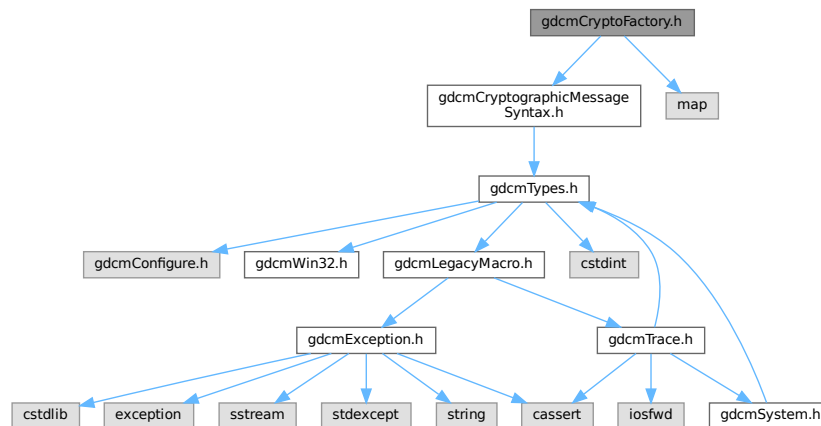
```

## 11.17 gdcryptoFactory.h File Reference

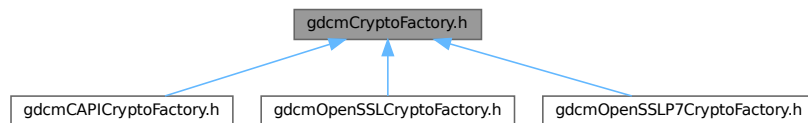
```
#include "gdcryptographicMessageSyntax.h"
```

```
#include <map>
```

Include dependency graph for gdcryptoFactory.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::CryptoFactory](#)  
Class to do handle the crypto factory.

## Namespaces

- namespace [gdcm](#)

## 11.18 gdcmCryptoFactory.h

[Go to the documentation of this file.](#)

```

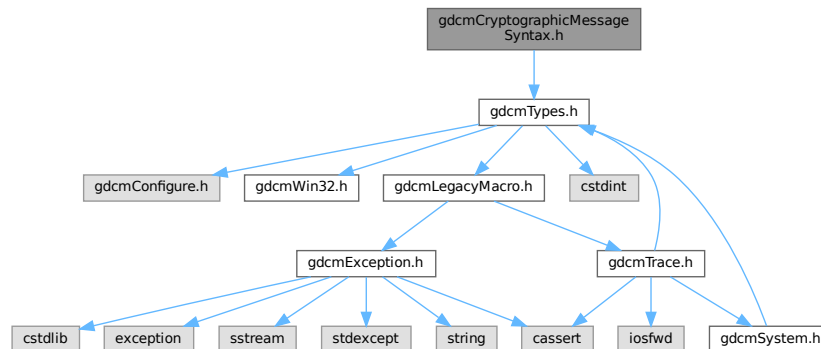
00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMCRYPTOFACTORY_H
00015 #define GDCMCRYPTOFACTORY_H
00016
00017 #include "gdcmCryptographicMessageSyntax.h"
00018 #include <map>
00019
00020 namespace gdcm
00021 {
00022
00035 class GDCM_EXPORT CryptoFactory
00036 {
00037 public:
00038     enum CryptoLib {DEFAULT = 0, OPENSSSL = 1, CAPI = 2, OPENSSSL7 = 3};
00039
00040     virtual CryptographicMessageSyntax* CreateCMSProvider() = 0;
00041     static CryptoFactory* GetFactoryInstance(CryptoLib id = DEFAULT);
00042
00043 protected:
00044     CryptoFactory(CryptoLib id)
00045     {
00046         AddLib(id, this);
00047     }
00048
00049 private:
00050     static std::map<CryptoLib, CryptoFactory*> getInstanceMap()
00051     {
00052         static std::map<CryptoLib, CryptoFactory*> libs;
00053         return libs;
00054     }
00055
00056     static void AddLib(CryptoLib id, CryptoFactory* f)
00057     {
00058         if (getInstanceMap().insert(std::pair<CryptoLib, CryptoFactory*>(id, f)).second == false)
00059         {
00060             gdcmErrorMacro( "Library already registered under id " << (int)id );
00061         }
00062     }
00063
00064 protected:
00065     CryptoFactory()= default;
00066     ~CryptoFactory()= default;
00067 };
00068
00069 } // end namespace gdcm
00070
00071 #endif // GDCMCRYPTOFACTORY_H

```

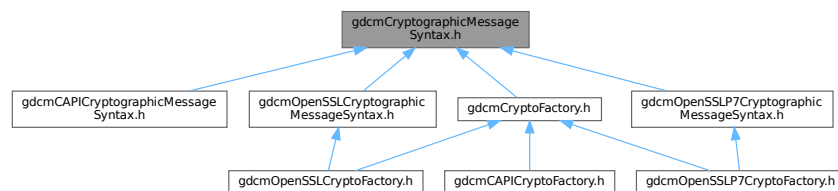
## 11.19 gdcmCryptographicMessageSyntax.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmCryptographicMessageSyntax.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [gdcm::CryptographicMessageSyntax](#)

### Namespaces

- namespace [gdcm](#)

## 11.20 gdcmCryptographicMessageSyntax.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004

```

```

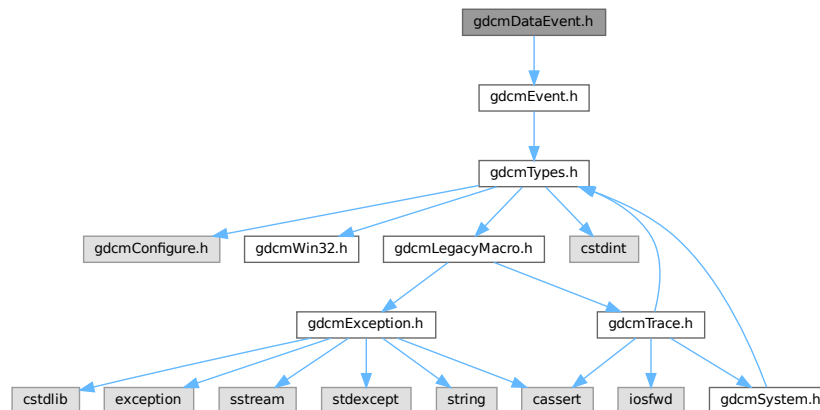
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMCRYPTOGRAPHICMESSAGESYNTAX_H
00015 #define GDCMCRYPTOGRAPHICMESSAGESYNTAX_H
00016
00017 #include "gdcmtypes.h"
00018
00019 namespace gdcms
00020 {
00021
00022 class GDCM_EXPORT CryptographicMessageSyntax
00023 {
00024 public:
00025     CryptographicMessageSyntax() = default;
00026
00027     virtual ~CryptographicMessageSyntax() = default;
00028     CryptographicMessageSyntax(const CryptographicMessageSyntax&) = delete;
00029     void operator=(const CryptographicMessageSyntax&) = delete;
00030
00031     typedef enum {
00032         DES3_CIPHER, // Triple DES
00033         AES128_CIPHER, // CBC AES
00034         AES192_CIPHER, // ' '
00035         AES256_CIPHER // ' '
00036     } CipherTypes;
00037
00038     // X.509
00039     virtual bool ParseCertificateFile( const char *filename ) = 0;
00040     virtual bool ParseKeyFile( const char *filename ) = 0;
00041
00042     // PBE
00043     virtual bool SetPassword(const char * pass, size_t passLen) = 0;
00044
00045     virtual bool Encrypt(char *output, size_t &outlen, const char *array, size_t len) const = 0;
00046     virtual bool Decrypt(char *output, size_t &outlen, const char *array, size_t len) const = 0;
00047
00048     virtual void SetCipherType(CipherTypes type) = 0;
00049
00050     virtual CipherTypes GetCipherType() const = 0;
00051 };
00052
00053 } // end namespace gdcms
00054
00055 #endif //GDCMCRYPTOGRAPHICMESSAGESYNTAX_H

```

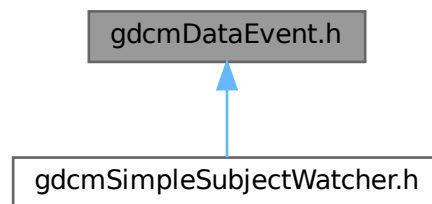
## 11.21 gdcmDataEvent.h File Reference

```
#include "gdcmEvent.h"
```

Include dependency graph for gdcmDataEvent.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [gdcm::DataEvent](#)  
*DataEvent.*

### Namespaces

- namespace [gdcm](#)

## 11.22 gdcmDataEvent.h

[Go to the documentation of this file.](#)

```

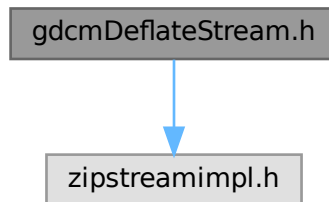
00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMDATAEVENT_H
00015 #define GDCMDATAEVENT_H
00016
00017 #include "gdcmEvent.h"
00018
00019 namespace gdcm
00020 {
00021
00022   class DataEvent : public AnyEvent
00023   {
00024   public:
00025     typedef DataEvent Self;
00026     typedef AnyEvent Superclass;
00027     DataEvent(const char *bytes = nullptr, size_t len = 0):Bytes(bytes),Length(len) {}
00028     ~DataEvent() override = default;
00029     DataEvent(const Self&s) : AnyEvent(s){};
00030     void operator=(const Self&) = delete;
00031
00032     const char * GetEventName() const override { return "DataEvent"; }
00033     bool CheckEvent(const ::gdcm::Event* e) const override
00034     { return (dynamic_cast<const Self*>(e) == nullptr ? false : true) ; }
00035     ::gdcm::Event* MakeObject() const override
00036     { return new Self; }
00037
00038     void SetData(const char *bytes, size_t len) {
00039       Bytes = bytes;
00040       Length = len;
00041     }
00042     size_t GetDataLength() const { return Length; }
00043     const char *GetData() const { return Bytes; }
00044
00045     //std::string GetValueAsString() const { return; }
00046
00047   private:
00048     const char *Bytes;
00049     size_t Length;
00050   };
00051
00052 } // end namespace gdcm
00053
00054 #endif //GDCMDATAEVENT_H

```

## 11.23 gdcmDeflateStream.h File Reference

```
#include "zipstreamimpl.h"
```

Include dependency graph for gdcmDeflateStream.h:



## 11.24 gdcmDeflateStream.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMDEFLATESTREAM_H
00015 #define GDCMDEFLATESTREAM_H
00016
00017 #include "zipstreamimpl.h"
00018
00019 #endif //GDCMDEFLATESTREAM_H
  
```

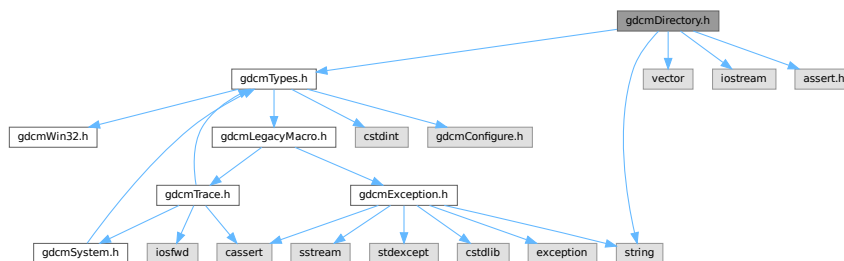
## 11.25 gdcmDirectory.h File Reference

```

#include "gdcmTypes.h"
#include <string>
#include <vector>
#include <iostream>
  
```

```
#include <assert.h>
```

Include dependency graph for `gdcmDirectory.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::Directory`  
*Class for manipulation directories.*

## Namespaces

- namespace `gdcm`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Directory &d)`

## 11.26 gdcmDirectory.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.

```



```

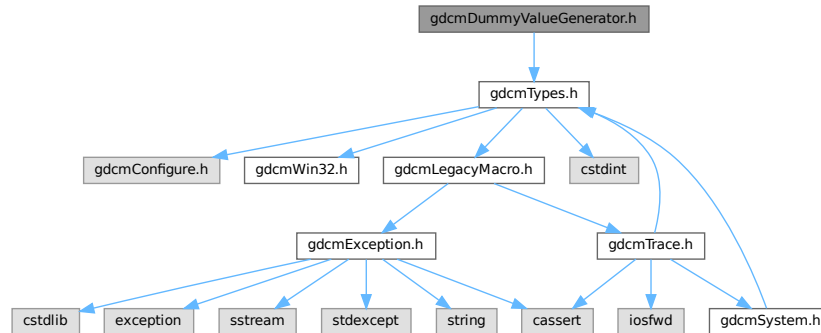
00012
00013 =====*/
00014 #ifndef GDCMDIRECTORY_H
00015 #define GDCMDIRECTORY_H
00016
00017 #include "gdcmTypes.h"
00018
00019 #include <string>
00020 #include <vector>
00021 #include <iostream>
00022 #include <assert.h>
00023
00024 namespace gdcm
00025 {
00041 //-----
00042 class GDCM_EXPORT Directory
00043 {
00044     friend std::ostream& operator<<(std::ostream &_os, const Directory &d);
00045 public :
00046     Directory() = default;
00047     ~Directory() = default;
00048     typedef std::string FilenameType;
00049     typedef std::vector<FilenameType> FilenamesType;
00050
00052     void Print(std::ostream &os = std::cout) const;
00053
00055     FilenameType const &GetToplevel() const { return Toplevel; }
00056
00058     FilenamesType const &GetFilenames() const {
00059         assert( !(Toplevel.empty()) && "Need to call Explore first" );
00060         return Filenames; }
00061
00063     FilenamesType const &GetDirectories() const { return Directories; }
00064
00067     unsigned int Load(FilenameType const &name, bool recursive = false);
00068
00069     // \todo later: GLOB
00070     // The glob() function searches for all the pathnames matching pattern according to
00071     // the rules used by the shell (see glob(7)). No tilde expansion or parameter
00072     // substitution is done; if you want these, use wordexp(3).
00073     // int Glob(...);
00074
00075 protected:
00077     unsigned int Explore(FilenameType const &name, bool recursive);
00078
00079 private :
00081     FilenamesType Filenames;
00082     FilenamesType Directories;
00083
00085     FilenameType Toplevel;
00086 };
00087 //-----
00088 inline std::ostream& operator<<(std::ostream &os, const Directory &d)
00089 {
00090     d.Print( os );
00091     return os;
00092 }
00093
00094 } // end namespace gdcm
00095 //-----
00096 #endif //GDCMDIRECTORY_H

```

## 11.27 gdcmDummyValueGenerator.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmDummyValueGenerator.h:



### Classes

- class [gdcm::DummyValueGenerator](#)  
Class for generating dummy value.

### Namespaces

- namespace [gdcm](#)

## 11.28 gdcmDummyValueGenerator.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMDUMMYVALUEGENERATOR_H
00015 #define GDCMDUMMYVALUEGENERATOR_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021
00022   class GDCM_EXPORT DummyValueGenerator
00023   {
00024
00025   
```

```

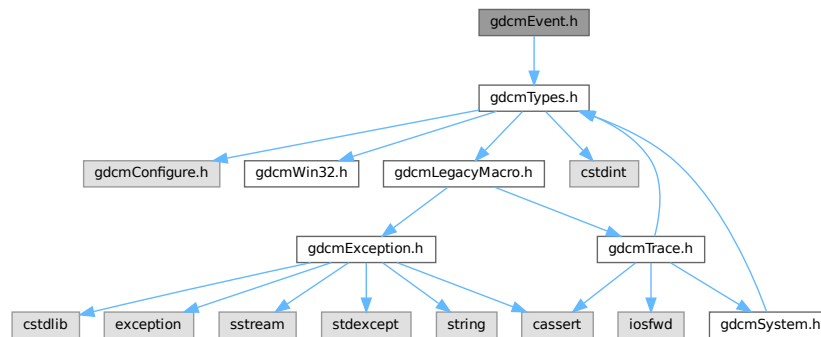
00028 public:
00029
00035     static const char* Generate(const char *input);
00036
00037 private:
00038 };
00039
00040
00041 } // end namespace gdcm
00042
00043 #endif //GDCMDUMMYVALUEGENERATOR_H

```

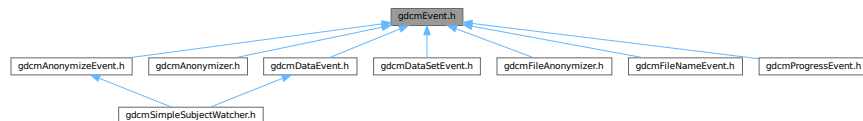
## 11.29 gdcmEvent.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmEvent.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class `gdcm::AbortEvent`
- class `gdcm::AnyEvent`
- class `gdcm::EndEvent`
- class `gdcm::Event`  
*superclass for callback/observer methods*
- class `gdcm::ExitEvent`
- class `gdcm::InitializeEvent`

- class [gdcm::IterationEvent](#)
- class [gdcm::ModifiedEvent](#)
- class [gdcm::NoEvent](#)
- class [gdcm::StartEvent](#)
- class [gdcm::UserEvent](#)

## Namespaces

- namespace [gdcm](#)

## Macros

- `#define gdcmEventMacro(classname, super)`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Event &e)`  
*Generic inserter operator for [Event](#) and its subclasses.*

## 11.29.1 Macro Definition Documentation

### 11.29.1.1 [gdcmEventMacro](#)

```
#define gdcmEventMacro(
    classname,
    super )
```

#### Value:

```
\
class classname : public super { \
public: \
    typedef classname Self; \
    typedef super Superclass; \
    classname() {} \
    virtual ~classname() override = default; \
    virtual const char * GetEventName() const override { return #classname; } \
    virtual bool CheckEvent(const ::gdcm::Event* e) const override \
    { return dynamic_cast<const Self*>(e) ? true : false; } \
    virtual ::gdcm::Event* MakeObject() const override \
    { return new Self; } \
    classname(const Self&s) : super(s){} \
private: \
    void operator=(const Self&); \
}
```

## 11.30 gdcmEvent.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMEVENT_H
00015  #define GDCMEVENT_H
00016
00017  #include "gdcmTypes.h"
00018
00019  namespace gdcm
00020  {
00021  //-----
00026  class GDCM_EXPORT Event
00027  {
00028  public :
00029      Event();
00030      virtual ~Event();
00031      Event(const Event&);
00032      void operator=(const Event&) = delete;
00033
00036      virtual Event* MakeObject() const = 0;
00037
00041      virtual void Print(std::ostream& os) const;
00042
00044      virtual const char * GetEventName() const = 0;
00045
00047      virtual bool CheckEvent(const Event*) const = 0;
00048  };
00049
00050
00052  inline std::ostream& operator<<(std::ostream& os, const Event &e)
00053  {
00054      e.Print(os);
00055      return os;
00056  }
00057
00058  /*
00059   * Macro for creating new Events
00060   */
00061  #define gdcmEventMacro( classname , super ) \
00062  \
00063  class classname : public super { \
00064  public: \
00065      typedef classname Self; \
00066      typedef super Superclass; \
00067      classname() {} \
00068      virtual ~classname() override = default; \
00069      virtual const char * GetEventName() const override { return #classname; } \
00070      virtual bool CheckEvent(const ::gdcm::Event* e) const override \
00071      { return dynamic_cast<const Self*>(e) ? true : false; } \
00072      virtual ::gdcm::Event* MakeObject() const override \
00073      { return new Self; } \
00074      classname(const Self&s) : super(s){} \
00075  private: \
00076      void operator=(const Self&); \
00077  }
00078
00082  gdcmEventMacro( NoEvent , Event );
00083  gdcmEventMacro( AnyEvent , Event );
00084  gdcmEventMacro( StartEvent , AnyEvent );
00085  gdcmEventMacro( EndEvent , AnyEvent );
00086  //gdcmEventMacro( ProgressEvent , AnyEvent );
00087  gdcmEventMacro( ExitEvent , AnyEvent );
00088  gdcmEventMacro( AbortEvent , AnyEvent );
00089  gdcmEventMacro( ModifiedEvent , AnyEvent );
00090  gdcmEventMacro( InitializeEvent , AnyEvent );
00091  gdcmEventMacro( IterationEvent , AnyEvent );

```

```

00092 //gdcmEventMacro( AnonymizeEvent      , AnyEvent );
00093 gdcmEventMacro( UserEvent                , AnyEvent );
00094
00095
00096 } // end namespace gdcm
00097 //-----
00098 #endif //GDCMEVENT_H

```

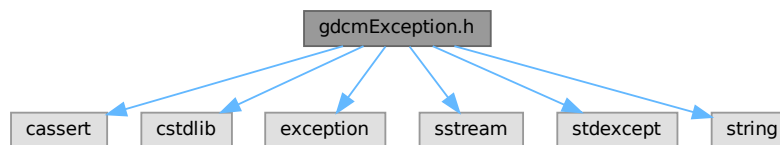
### 11.31 gdcmException.h File Reference

```

#include <cassert>
#include <cstdlib>
#include <exception>
#include <sstream>
#include <stdexcept>
#include <string>

```

Include dependency graph for gdcmException.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class `gdcm::Exception`  
*Exception.*

#### Namespaces

- namespace `gdcm`

## 11.32 gdcmException.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMEXCEPTION_H
00015  #define GDCMEXCEPTION_H
00016
00017  #include <cassert>
00018  #include <cstdlib> // NULL
00019  #include <exception>
00020  #include <sstream> // ostringstream
00021  #include <stdexcept> // logic_error
00022  #include <string>
00023
00024  // Disable clang warning "dynamic exception specifications are deprecated".
00025  // We need to be C++03 and C++11 compatible, and if we remove the 'throw()'
00026  // specifier we'll get an error in C++03 by not matching the superclass.
00027  #if defined(__clang__) && defined(__has_warning)
00028  # if __has_warning("-Wdeprecated")
00029  # pragma clang diagnostic push
00030  # pragma clang diagnostic ignored "-Wdeprecated"
00031  # endif
00032  #endif
00033
00034  namespace gdcm
00035  {
00036
00043  class Exception : public std::exception
00044  {
00049      typedef std::logic_error StringHolder;
00050
00052      static StringHolder CreateWhat(const char* const desc,
00053                                     const char* const file,
00054                                     const unsigned int lineNumber,
00055                                     const char* const func)
00056      {
00057          assert(desc != nullptr);
00058          assert(file != nullptr);
00059          assert(func != nullptr);
00060          std::ostringstream oswhat;
00061          oswhat << file << ":" << lineNumber << " (" << func << "):\n";
00062          oswhat << desc;
00063          return StringHolder( oswhat.str() );
00064      }
00065
00066  public:
00072      explicit Exception(const char *desc = "None",
00073                         const char *file = __FILE__,
00074                         unsigned int lineNumber = __LINE__,
00075                         // FIXME: __PRETTY_FUNCTION__ is the non-mangled version for __GNUC__ compiler
00076                         const char *func = "" /*__FUNCTION__*/)
00077      :
00078      What( CreateWhat(desc, file, lineNumber, func) ),
00079      Description(desc)
00080      {
00081      }
00082
00083      ~Exception() throw() override {}
00084
00086      const char* what() const throw() override
00087      {
00088          return What.what();
00089      }
00090
00092      const char * GetDescription() const { return Description.what(); }
00093
00094  private:

```

```

00095   StringHolder  What;
00096   StringHolder  Description;
00097 };
00098
00099 } // end namespace gdcm
00100
00101 // Undo warning suppression.
00102 #if defined(__clang__) && defined(__has_warning)
00103 # if __has_warning("-Wdeprecated")
00104 #   pragma clang diagnostic pop
00105 # endif
00106 #endif
00107
00108 #endif

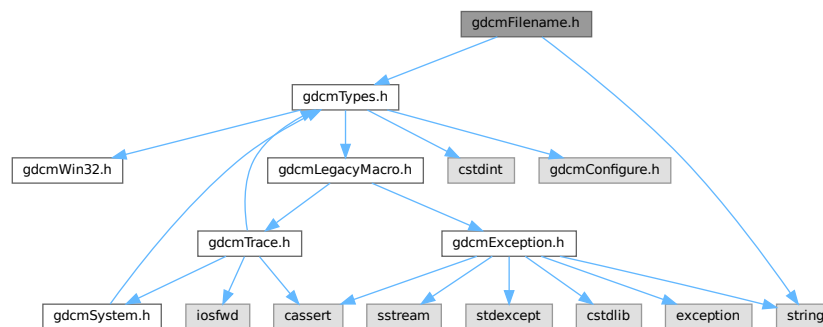
```

## 11.33 gdcmFilename.h File Reference

```
#include "gdcmTypes.h"
```

```
#include <string>
```

Include dependency graph for gdcmFilename.h:



### Classes

- class [gdcm::Filename](#)  
Class to manipulate file name's.

### Namespaces

- namespace [gdcm](#)



## 11.34 gdcmFilename.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMFILENAME_H
00015 #define GDCMFILENAME_H
00016
00017 #include "gdcmTypes.h"
00018
00019 #include <string>
00020
00021 namespace gdcm
00022 {
00023     class GDCM_EXPORT Filename
00024     {
00025     public:
00026         Filename(const char* filename = ""):FileName(filename ? filename : ""),Path(),Conversion() {}
00027
00028         const char *GetFileName() const { return FileName.c_str(); }
00029         const char *GetPath();
00030         const char *GetName();
00031         const char *GetExtension();
00032         const char *ToUnixSlashes();
00033         const char *ToWindowsSlashes();
00034
00035         static const char *Join(const char *path, const char *filename);
00036
00037         bool IsEmpty() const { return FileName.empty(); }
00038
00039         operator const char * () const { return GetFileName(); }
00040
00041         // FIXME: I don't like this function
00042         // It hides the realpath call (maybe useful)
00043         // and it forces file to exist on the disk whereas Filename
00044         // should be independent from file existence.
00045         bool IsIdentical(Filename const &fn) const;
00046
00047         bool EndWith(const char ending[]) const;
00048
00049     private:
00050         std::string FileName;
00051         std::string Path;
00052         std::string Conversion;
00053     };
00054 } // end namespace gdcm
00055
00056 #endif //GDCMFILENAME_H

```

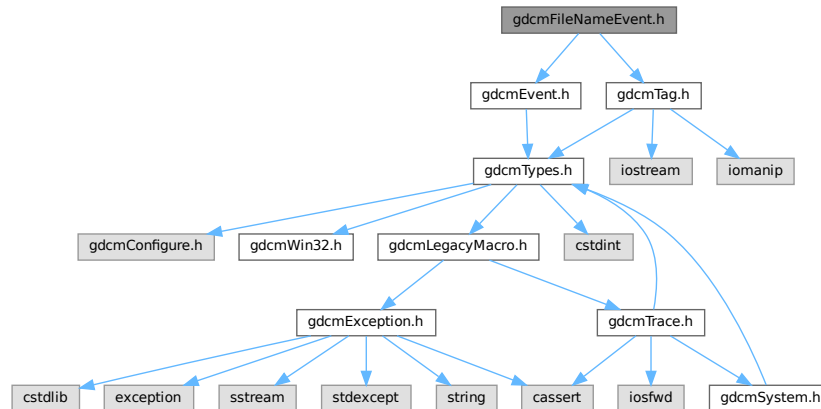
## 11.35 gdcmFileNameEvent.h File Reference

```

#include "gdcmEvent.h"
#include "gdcmTag.h"

```

Include dependency graph for `gdcmFileNameEvent.h`:



## Classes

- class `gdcm::FileNameEvent`  
*FileNameEvent.*

## Namespaces

- namespace `gdcm`

## 11.36 `gdcmFileNameEvent.h`

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMFILENAMEEVENT_H
00015 #define GDCMFILENAMEEVENT_H
00016
00017 #include "gdcmEvent.h"
00018 #include "gdcmTag.h"
00019
00020 namespace gdcm
00021 {
00022
00029   class FileNameEvent : public AnyEvent
00030   {
00031   public:

```

```

00032 typedef FileNameEvent Self;
00033 typedef AnyEvent Superclass;
00034 FileNameEvent(const char *s = ""):m_FileName(s) {}
00035 ~FileNameEvent() override = default;
00036
00037 FileNameEvent(const Self&s) : AnyEvent(s){}
00038 void operator=(const Self&) = delete;
00039
00040
00041 const char * GetEventName() const override { return "FileNameEvent"; }
00042 bool CheckEvent(const ::gdcm::Event* e) const override
00043 { return dynamic_cast<const Self*>(e) ? true : false; }
00044 ::gdcm::Event* MakeObject() const override
00045 { return new Self; }
00046
00047 void SetFileName(const char *f) { m_FileName = f; }
00048 const char *GetFileName() const { return m_FileName.c_str(); }
00049 private:
00050 std::string m_FileName;
00051 };
00052
00053
00054 } // end namespace gdcm
00055
00056 #endif //GDCMFILENAMEEVENT_H

```

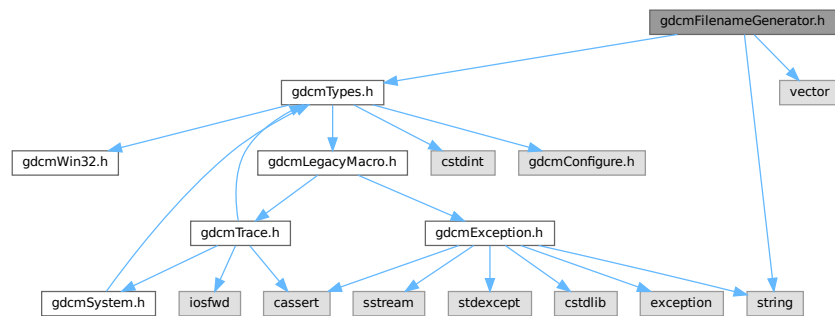
## 11.37 gdcmFilenameGenerator.h File Reference

```
#include "gdcmTypes.h"
```

```
#include <string>
```

```
#include <vector>
```

Include dependency graph for gdcmFilenameGenerator.h:



### Classes

- class [gdcm::FilenameGenerator](#)  
*FilenameGenerator.*

### Namespaces

- namespace [gdcm](#)

## 11.38 gdcmFilenameGenerator.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009       This software is distributed WITHOUT ANY WARRANTY; without even
00010       the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011       PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMFILENAMEGENERATOR_H
00015 #define GDCMFILENAMEGENERATOR_H
00016
00017 #include "gdcmTypes.h"
00018 #include <string>
00019 #include <vector>
00020
00021
00022 namespace gdcm
00023 {
00024
00025     class GDCM_EXPORT FilenameGenerator
00026     {
00027     public:
00028         FilenameGenerator():Pattern(),Prefix(),FileNames() {}
00029         ~FilenameGenerator() = default;
00030         // FIXME: already defines in gdcm::Directory
00031         typedef std::string FilenameType;
00032         typedef std::vector<FilenameType> FileNamesType;
00033         typedef FileNamesType::size_type SizeType;
00034
00035         void SetPattern(const char *pattern) { Pattern = pattern; }
00036         const char *GetPattern() const { return Pattern.c_str(); }
00037
00038         void SetPrefix(const char *prefix) { Prefix = prefix; }
00039         const char *GetPrefix() const { return Prefix.c_str(); }
00040
00041         bool Generate();
00042
00043         void SetNumberOfFileNames(SizeType nfiles);
00044         SizeType GetNumberOfFileNames() const;
00045
00046         const char * GetFilename(SizeType n) const;
00047         FileNamesType const & GetFileNames() const { assert( !Pattern.empty() ); return FileNames; }
00048
00049     private:
00050         FilenameType Pattern;
00051         FilenameType Prefix;
00052         FileNamesType FileNames;
00053     };
00054 } // end namespace gdcm
00055
00056 #endif //GDCMFILENAMEGENERATOR_H

```

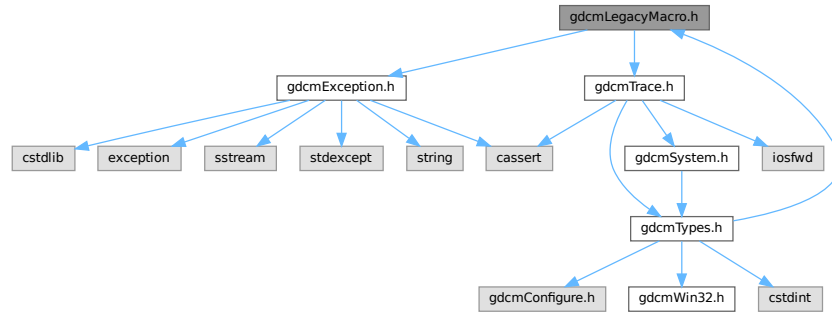
## 11.39 gdcmLegacyMacro.h File Reference

```

#include "gdcmException.h"
#include "gdcmTrace.h"

```

Include dependency graph for gdcmLegacyMacro.h:



This graph shows which files directly or indirectly include this file:



## Macros

- #define `GDCM_LEGACY`(method) method;
- #define `GDCM_LEGACY_BODY`(method, version) `gdcmWarningMacro`(#method " was deprecated for " version " and will be removed in a future version.")
- #define `GDCM_LEGACY_REPLACED_BODY`(method, version, replace) `gdcmWarningMacro`(#method " was deprecated for " version " and will be removed in a future version. Use " #replace " instead.")
- #define `GDCM_NOOP_STATEMENT` static\_assert(true, "")

## 11.39.1 Macro Definition Documentation

### 11.39.1.1 GDCM\_LEGACY

```
#define GDCM_LEGACY(
    method ) method;
```

### 11.39.1.2 GDCM\_LEGACY\_BODY

```
#define GDCM_LEGACY_BODY(
    method,
    version ) gdcmWarningMacro(#method " was deprecated for " version " and will be
removed in a future version.")
```

### 11.39.1.3 GDCM\_LEGACY\_REPLACED\_BODY

```
#define GDCM_LEGACY_REPLACED_BODY(
    method,
    version,
    replace )  gdcmWarningMacro(#method " was deprecated for " version " and will be
removed in a future version.  Use " #replace " instead.")
```

### 11.39.1.4 GDCM\_NOOP\_STATEMENT

```
#define GDCM_NOOP_STATEMENT static_assert(true, "")
```

The `static_assert(true, "")` idiom is commonly employed for C++11 or greater to ensure that it is compile-time only check that can not be part of the binary file. This allows a macro to be used anywhere that a statement is expected, and to enforce consistent use of ; after a macro. The `static_assert` is a `constexpr` that can be used in places where raw statements (i.e. `'do{} while(0)'`) are not allowed (i.e. after class member function definitions).

## 11.40 gdcmLegacyMacro.h

[Go to the documentation of this file.](#)

```
00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 #ifndef GDCMLEGACYMACRO_H
00015 #define GDCMLEGACYMACRO_H
00016
00017 #if !defined(GDCMTYPES_H) && !defined(SWIG)
00018 #error you need to include gdcmTypes.h instead
00019 #endif
00020
00021 #include "gdcmException.h"
00022
00023 //-----
00024 // Setup legacy code policy.
00025
00026 // Define GDCM_LEGACY macro to mark legacy methods where they are
00027 // declared in their class. Example usage:
00028 //
00029 //     @deprecated Replaced by MyOtherMethod() as of GDCM 2.0.
00030 //     GDCM_LEGACY(void MyMethod());
00031 #if defined(GDCM_LEGACY_REMOVE)
00032 # define GDCM_LEGACY(method)
00033 #elif defined(GDCM_LEGACY_SILENT) || defined(SWIG)
00034 // Provide legacy methods with no warnings.
00035 # define GDCM_LEGACY(method) method;
00036 #else
00037 // Setup compile-time warnings for uses of deprecated methods if
00038 // possible on this compiler.
00039 # if defined(__GNUC__) && !defined(__INTEL_COMPILER) && (__GNUC__ > 3 || (__GNUC__ == 3 && __GNUC_MINOR__
    >= 1))
00040 #   define GDCM_LEGACY(method) method __attribute__((deprecated));
00041 # elif defined(_MSC_VER) && _MSC_VER >= 1300
00042 #   define GDCM_LEGACY(method) __declspec(deprecated) method;
00043 # else
```

```

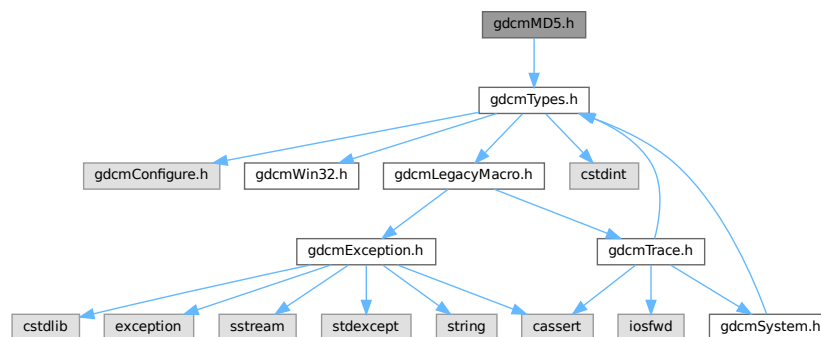
00044 # define GDCM_LEGACY(method) method;
00045 # endif
00046 #endif
00047
00057 # define GDCM_NOOP_STATEMENT static_assert(true, "")
00058
00059 // Macros to create runtime deprecation warning messages in function
00060 // bodies. Example usage:
00061 //
00062 // #if !defined(GDCM_LEGACY_REMOVE)
00063 // void gdcm::MyClass::MyOldMethod()
00064 // {
00065 //     GDCM_LEGACY_BODY(gdcm::MyClass::MyOldMethod, "GDCM 2.0");
00066 // }
00067 // #endif
00068 //
00069 // #if !defined(GDCM_LEGACY_REMOVE)
00070 // void gdcm::MyClass::MyMethod()
00071 // {
00072 //     GDCM_LEGACY_REPLACED_BODY(gdcm::MyClass::MyMethod, "GDCM 2.0",
00073 //                               gdcm::MyClass::MyOtherMethod);
00074 // }
00075 // #endif
00076 #if defined(GDCM_LEGACY_REMOVE) || defined(GDCM_LEGACY_SILENT)
00077 # define GDCM_LEGACY_BODY(method, version)
00078 # define GDCM_LEGACY_REPLACED_BODY(method, version, replace)
00079 #else
00080 # define GDCM_LEGACY_BODY(method, version) \
00081     gdcmWarningMacro(#method " was deprecated for " version " and will be removed in a future version.")
00082 # define GDCM_LEGACY_REPLACED_BODY(method, version, replace) \
00083     gdcmWarningMacro(#method " was deprecated for " version " and will be removed in a future version. Use "
00084                       #replace " instead.")
00085 #endif
00086 #include "gdcmTrace.h"
00087
00088 #endif // GDCMLEGACYMACRO_H

```

## 11.41 gdcmMD5.h File Reference

#include "gdcmTypes.h"

Include dependency graph for gdcmMD5.h:



### Classes

- class [gdcm::MD5](#)  
Class for *MD5*.

## Namespaces

- namespace `gdcm`

## 11.42 gdcmMD5.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMMD5_H
00015 #define GDCMMD5_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021 //-----
00032 class GDCM_EXPORT MD5
00033 {
00034 public :
00035     // Compute md5 from memory pointed by `pointer` of size `buf_len`
00036     static bool Compute(const char *buffer, size_t buf_len, char digest_str[33]);
00037
00038     static bool ComputeFile(const char *filename, char digest_str[33]);
00039 };
00040
00041 } // end namespace gdcm
00042 //-----
00043 #endif //GDCMMD5_H

```

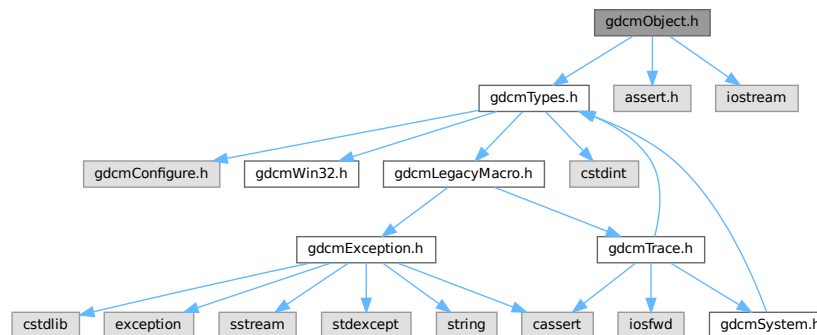
## 11.43 gdcmObject.h File Reference

```

#include "gdcmTypes.h"
#include <assert.h>
#include <iostream>

```

Include dependency graph for `gdcmObject.h`:





This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::Object`  
*Object.*

## Namespaces

- namespace `gdcm`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Object &obj)`

## 11.44 gdcmObject.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 #ifndef GDCMOBJECT_H
00015 #define GDCMOBJECT_H
00016
00017 #include "gdcmTypes.h"
00018
00019 #include <assert.h>
00020 #include <iostream> // grrrr
00021
00022 //namespace std { class ostream; }
00023 namespace gdcm
00024 {
00025
00026     template<class ObjectType> class SmartPointer;
00027
00036     class GDCM_EXPORT Object
00037     {
00038     public:
00039         template <class ObjectType> friend class SmartPointer;
00040         friend std::ostream& operator<<(std::ostream &os, const Object &obj);
00041
00042         Object():ReferenceCount(0) {}
00043
00044         // Implementation note:
00045         // If I move ~Object in the protected section I can prevent people
00046         // from writing:

```

```

00047 // SmartPointer<Object> p = new Object;
00048 // delete p; // due to SmartPointer::operator ObjectType * () const
00049 // but on the other hand one could not define an Object on the stack
00050 // Object obj;
00051 // Furthermore it would not prevent anyone from doing:
00052 // class MyObject : public Object {};
00053 // SmartPointer<MyObject> o = new MyObject;
00054 // delete o; // grrrrrr
00055 virtual ~Object() {
00056     // If your debugger reach here it means you are doing something silly
00057     // like using SmartPointer on object allocated on the stack (vs heap)
00058     assert(ReferenceCount == 0);
00059 }
00060
00061 // http://www.parashift.com/c++-faq-lite/freestore-mgmt.html#faq-16.24
00062 // Set the ref count to 0
00063 // Do NOT copy the reference count !
00064 Object(const Object&):ReferenceCount(0){}
00065 void operator=(const Object&){}
00066
00067 //static Object* New() { return new Object; }
00068
00069 protected:
00070 // For the purpose of the invasive SmartPointer implementation
00071 void Register() {
00072     ReferenceCount++;
00073     assert( ReferenceCount > 0 );
00074 }
00075 void UnRegister() {
00076     assert( ReferenceCount > 0 );
00077     ReferenceCount--;
00078     if(!ReferenceCount)
00079     {
00080         delete this;
00081     }
00082 }
00083
00084 public:
00085 // For dealing with printing of object and polymorphism
00086 virtual void Print(std::ostream &) const {}
00087
00088 private:
00089     long ReferenceCount;
00090 };
00091
00092 //-----
00093 // function do not carry vtable. Thus define in the base class the operator
00094 // and use the member function ->Print() to call the appropriate function
00095 // NOTE: All subclass of Object needs to implement the Print function
00096 inline std::ostream& operator<<(std::ostream &os, const Object &obj)
00097 {
00098     obj.Print(os);
00099     return os;
00100 }
00101
00102 } // end namespace gdcmm
00103
00104 #endif //GDCMOBJECT_H

```

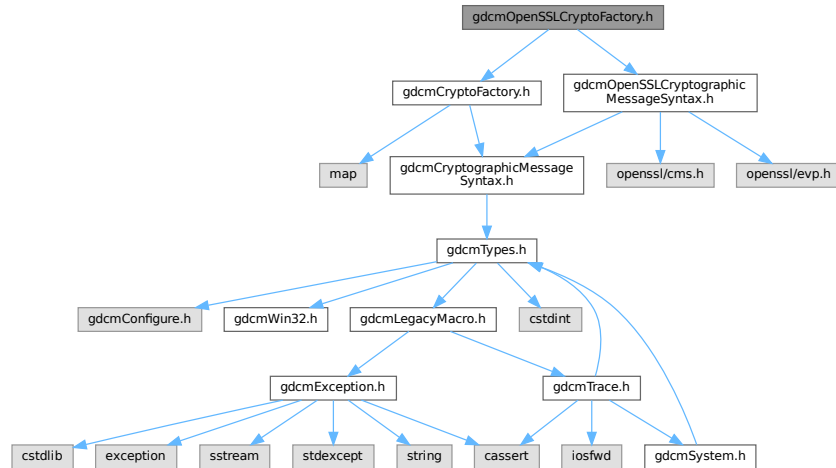
## 11.45 gdcmmOpenSSLCryptoFactory.h File Reference

```

#include "gdcmmCryptoFactory.h"
#include "gdcmmOpenSSLCryptographicMessageSyntax.h"

```

Include dependency graph for gdcmOpenSSLCryptoFactory.h:



## Classes

- class [gdcm::OpenSSLCryptoFactory](#)

## Namespaces

- namespace [gdcm](#)

## 11.46 gdcmOpenSSLCryptoFactory.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMOPENSSLCRYPTOFACTORY_H
00015 #define GDCMOPENSSLCRYPTOFACTORY_H
00016
00017 #include "gdcmCryptoFactory.h"
00018 #include "gdcmOpenSSLCryptographicMessageSyntax.h"
00019
00020 namespace gdcm
00021 {
00022
00023   class GDCM_EXPORT OpenSSLCryptoFactory : public CryptoFactory
00024   {
00025   public:

```

```

00026  OpenSSLCryptoFactory(CryptoLib id) : CryptoFactory(id)
00027  {
00028      gdcmlDebugMacro( "OpenSSL Factory registered." );
00029  }
00030
00031 public:
00032  CryptographicMessageSyntax* CreateCMSProvider()
00033  {
00034      InitOpenSSL();
00035      return new OpenSSLCryptographicMessageSyntax();
00036  }
00037
00038 protected:
00039  void InitOpenSSL();
00040
00041 private:
00042  OpenSSLCryptoFactory() {}
00043 };
00044
00045 } // end namespace gdcml
00046
00047 #endif //GDCMOPENSSLCRYPTOFACTORY_H

```

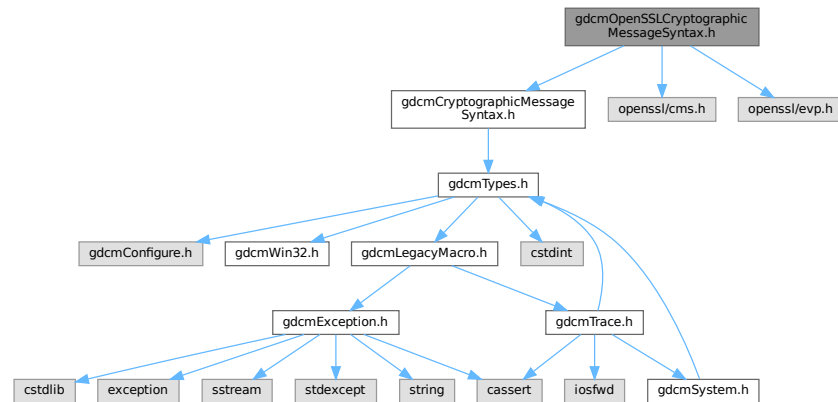
## 11.47 gdcmlOpenSSLCryptographicMessageSyntax.h File Reference

```
#include "gdcmlCryptographicMessageSyntax.h"
```

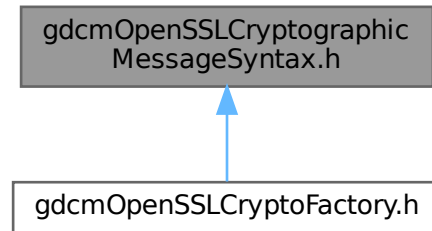
```
#include <openssl/cms.h>
```

```
#include <openssl/evp.h>
```

Include dependency graph for gdcmlOpenSSLCryptographicMessageSyntax.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::OpenSSLCryptographicMessageSyntax](#)

## Namespaces

- namespace [gdcm](#)

## 11.48 gdcmOpenSSLCryptographicMessageSyntax.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMOPENSSLCRYPTOGRAPHICMESSAGESYNTAX_H
00015  #define GDCMOPENSSLCRYPTOGRAPHICMESSAGESYNTAX_H
00016
00017  #include "gdcmCryptographicMessageSyntax.h"
00018  #include <openssl/cms.h>
00019  #include <openssl/evp.h>
00020
00021  namespace gdcm
00022  {
00023
00024  class GDCM_EXPORT OpenSSLCryptographicMessageSyntax : public CryptographicMessageSyntax
00025  {
00026  public:
00027    OpenSSLCryptographicMessageSyntax();
00028    ~OpenSSLCryptographicMessageSyntax();
00029
00030    // X.509
00031    bool ParseCertificateFile( const char *filename );
  
```

```

00032 bool ParseKeyFile( const char *filename );
00033
00034 // PBE
00035 bool SetPassword(const char * pass, size_t passLen);
00036
00037 void SetCipherType(CipherTypes type);
00038 CipherTypes GetCipherType() const;
00042 bool Encrypt(char *output, size_t &outlen, const char *array, size_t len) const;
00044 bool Decrypt(char *output, size_t &outlen, const char *array, size_t len) const;
00045
00046 private:
00047 // #ifdef GDCM_HAVE_CMS_RECIPIENT_PASSWORD
00048 // ::stack_st_X509 *recips;
00049 // #else
00050 STACK_OF(X509) *recips;
00051 // #endif
00052 ::EVP_PKEY *pkey;
00053 const EVP_CIPHER *internalCipherType;
00054 char * password;
00055 size_t passwordLength;
00056 CipherTypes cipherType;
00057
00058 private:
00059 OpenSSLCryptographicMessageSyntax(const OpenSSLCryptographicMessageSyntax&); // Not implemented.
00060 void operator=(const OpenSSLCryptographicMessageSyntax&); // Not implemented.
00061 const EVP_CIPHER *CreateCipher( CryptographicMessageSyntax::CipherTypes ciphertype);
00062
00063 };
00064
00065 } // end namespace gdcmm
00066
00067 #endif //GDCMOPENSSLCRYPTOGRAPHICMESSAGESYNTAX_H

```

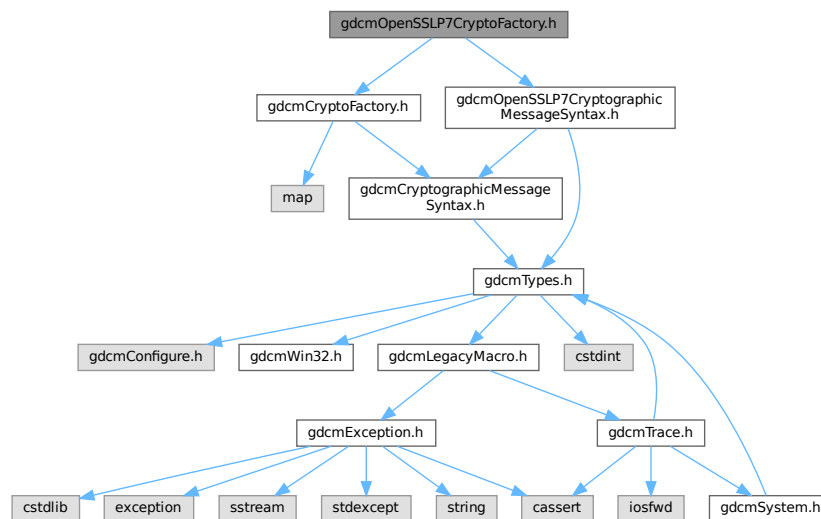
## 11.49 gdcmmOpenSSLP7CryptoFactory.h File Reference

```

#include "gdcmmCryptoFactory.h"
#include "gdcmmOpenSSLP7CryptographicMessageSyntax.h"

```

Include dependency graph for gdcmmOpenSSLP7CryptoFactory.h:



### Classes

- class [gdcmm::OpenSSLP7CryptoFactory](#)

**Namespaces**

- namespace [gdcm](#)

**11.50 gdcmOpenSSL7CryptoFactory.h**

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMOPENSSL7CRYPTOFACTORY_H
00015  #define GDCMOPENSSL7CRYPTOFACTORY_H
00016
00017  #include "gdcmCryptoFactory.h"
00018  #include "gdcmOpenSSL7CryptographicMessageSyntax.h"
00019
00020  namespace gdcm
00021  {
00022  class GDCM_EXPORT OpenSSL7CryptoFactory : public CryptoFactory
00023  {
00024  public:
00025      OpenSSL7CryptoFactory(CryptoLib id) : CryptoFactory(id)
00026      {
00027          gdcmDebugMacro( "OpenSSL (PKCS7) Factory registered." );
00028      }
00029
00030  public:
00031      CryptographicMessageSyntax* CreateCMSProvider()
00032      {
00033          return new OpenSSL7CryptographicMessageSyntax();
00034      }
00035
00036  private:
00037      OpenSSL7CryptoFactory() {}
00038  };
00039  }
00040
00041  #endif //GDCMOPENSSL7CRYPTOFACTORY_H

```

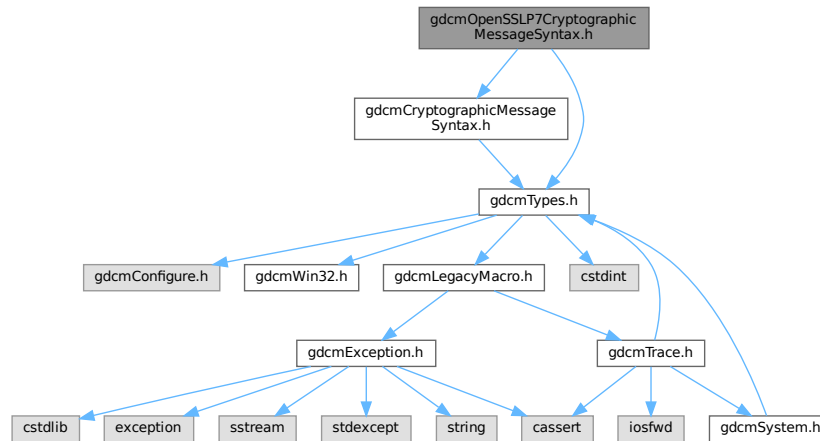
**11.51 gdcmOpenSSL7CryptographicMessageSyntax.h File Reference**

```

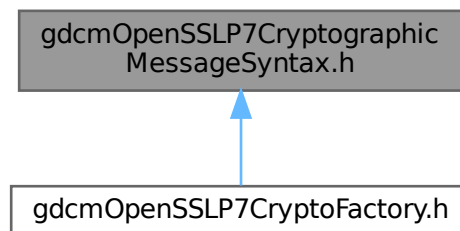
#include "gdcmCryptographicMessageSyntax.h"
#include "gdcmTypes.h"

```

Include dependency graph for `gdcmOpenSSL7CryptographicMessageSyntax.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::OpenSSL7CryptographicMessageSyntax](#)

## Namespaces

- namespace [gdcm](#)



## 11.52 gdcMOpenSSL7CryptographicMessageSyntax.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcM.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMOPENSSL7CRYPTOGRAPHICMESSAGESYNTAX_H
00015 #define GDCMOPENSSL7CRYPTOGRAPHICMESSAGESYNTAX_H
00016
00017 #include "gdcMCryptographicMessageSyntax.h"
00018 #include "gdcMTypes.h"
00019
00020 namespace gdcM
00021 {
00022   class CryptographicMessageSyntaxInternals;
00023   //-----
00024
00025   class GDCM_EXPORT OpenSSL7CryptographicMessageSyntax : public CryptographicMessageSyntax
00026   {
00027   public:
00028     OpenSSL7CryptographicMessageSyntax();
00029     ~OpenSSL7CryptographicMessageSyntax();
00030
00031     // X.509
00032     bool ParseCertificateFile( const char *filename );
00033     bool ParseKeyFile( const char *filename );
00034
00035     // PBE
00036     bool SetPassword(const char * /*pass*/, size_t /*passLen*/)
00037     {
00038       gdcMWarningMacro( "Openssl using PKCS7 does not support Password Based Encryption." );
00039       return false;
00040     }
00041
00042     void SetCipherType(CipherTypes type);
00043     CipherTypes GetCipherType() const;
00044
00045     bool Encrypt(char *output, size_t &outlen, const char *array, size_t len) const;
00046     bool Decrypt(char *output, size_t &outlen, const char *array, size_t len) const;
00047
00048   private:
00049     CryptographicMessageSyntaxInternals *Internals;
00050   private:
00051     OpenSSL7CryptographicMessageSyntax(const OpenSSL7CryptographicMessageSyntax&); // Not implemented.
00052     void operator=(const OpenSSL7CryptographicMessageSyntax&); // Not implemented.
00053   };
00054 } // end namespace gdcM
00055 //-----
00056 #endif //GDCMOPENSSL7CRYPTOGRAPHICMESSAGESYNTAX_H

```

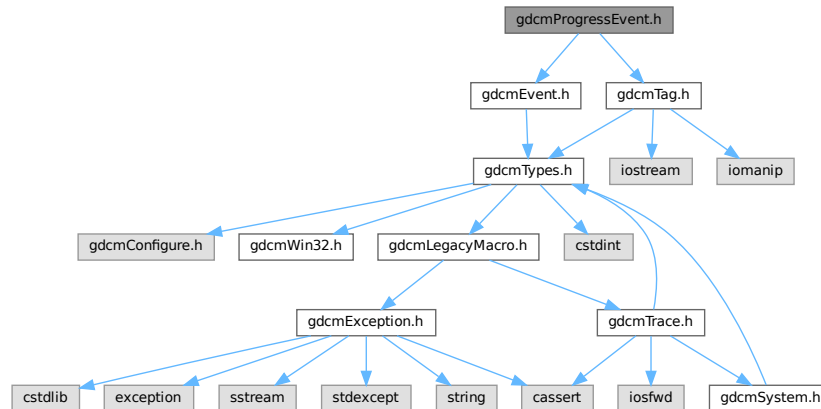
## 11.53 gdcMProgressEvent.h File Reference

```

#include "gdcMEvent.h"
#include "gdcMTag.h"

```

Include dependency graph for `gdcProgressEvent.h`:



## Classes

- class `gdc::ProgressEvent`  
*ProgressEvent.*

## Namespaces

- namespace `gdc`

## 11.54 `gdcProgressEvent.h`

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdc.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMPROGRESSEVENT_H
00015 #define GDCMPROGRESSEVENT_H
00016
00017 #include "gdcEvent.h"
00018 #include "gdcTag.h"
00019
00020 namespace gdc
00021 {
00022
00029   class ProgressEvent : public AnyEvent
00030   {
00031   public:

```

```

00032 typedef ProgressEvent Self;
00033 typedef AnyEvent Superclass;
00034 ProgressEvent(double p = 0):m_Progress(p) {}
00035 ~ProgressEvent() override = default;
00036
00037 ProgressEvent(const Self&s) : AnyEvent(s){};
00038 void operator=(const Self&) = delete;
00039
00040 const char * GetEventName() const override { return "ProgressEvent"; }
00041 bool CheckEvent(const ::gdcm::Event* e) const override
00042 { return dynamic_cast<const Self*>(e) ? true : false; }
00043 ::gdcm::Event* MakeObject() const override
00044 { return new Self; }
00045
00046 void SetProgress(double p) { m_Progress = p; }
00047 double GetProgress() const { return m_Progress; }
00048 private:
00049 double m_Progress;
00050 };
00051
00052
00053 } // end namespace gdcm
00054
00055 #endif //GDCMPROGRESSEVENT_H

```

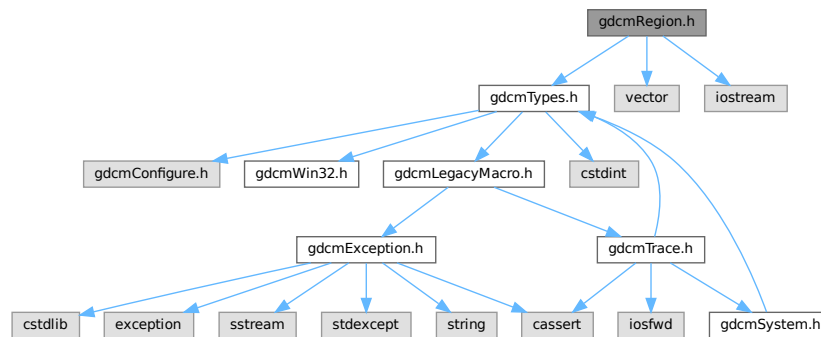
## 11.55 gdcmRegion.h File Reference

```
#include "gdcmTypes.h"
```

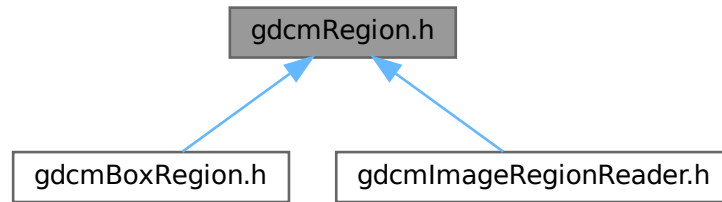
```
#include <vector>
```

```
#include <iostream>
```

Include dependency graph for gdcmRegion.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::Region](#)  
*Class for manipulation region.*

## Namespaces

- namespace [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Region &r)`

## 11.56 gdcmRegion.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012   =====*/
00013
00014 #ifndef GDCMREGION_H
00015 #define GDCMREGION_H
00016
00017 #include "gdcmTypes.h"
00018 #include <vector>
00019 #include <iostream>
00020
00021 namespace gdcm
00022 {
00023   class BoxRegion;
  
```

```

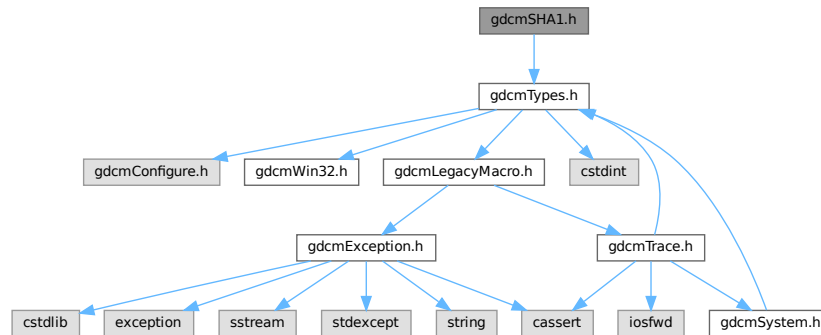
00027 //-----
00028 class GDCM_EXPORT Region
00029 {
00030 public :
00031     Region();
00032     virtual ~Region();
00033
00034     virtual void Print(std::ostream &os = std::cout) const;
00035
00036     virtual bool Empty() const = 0;
00037
00038     virtual bool IsValid() const = 0;
00039
00040     virtual size_t Area() const = 0;
00041
00042     // implementation detail of heterogeneous container in C++
00043     virtual Region *Clone() const = 0;
00044
00045     virtual BoxRegion ComputeBoundingBox() = 0;
00046 private:
00047 };
00048 //-----
00049 inline std::ostream& operator<<(std::ostream &os, const Region&r)
00050 {
00051     r.Print( os );
00052     return os;
00053 }
00054 //-----
00055 #endif //GDCMREGION_H

```

## 11.57 gdcmSHA1.h File Reference

#include "gdcmTypes.h"

Include dependency graph for gdcmSHA1.h:



### Classes

- class [gdcm::SHA1](#)  
Class for *SHA1*.

### Namespaces

- namespace [gdcm](#)

## 11.58 gdcmSHA1.h

[Go to the documentation of this file.](#)

```

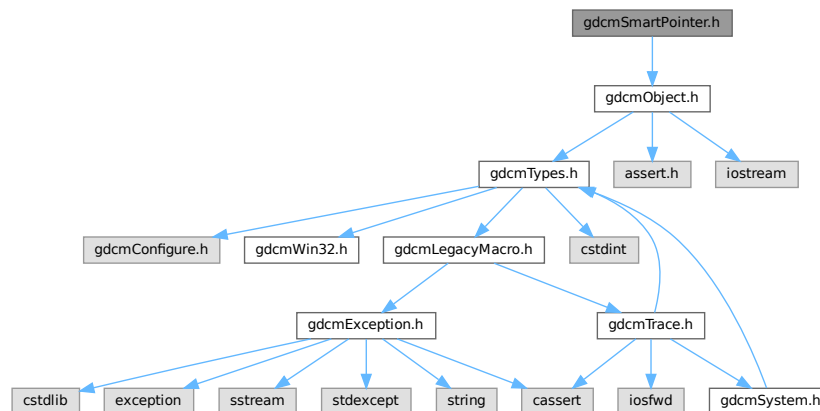
00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMSHA1_H
00015 #define GDCMSHA1_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021 //-----
00022 class SHA1Internals;
00023 class GDCM_EXPORT SHA1
00024 {
00025 public:
00026     SHA1();
00027     ~SHA1();
00028     SHA1(const SHA1&) = delete;
00029     void operator=(const SHA1&) = delete;
00030
00031     static bool Compute(const char *buffer, unsigned long buf_len, char digest_str[20*2+1]);
00032     static bool ComputeFile(const char *filename, char digest_str[20*2+1]);
00033 private:
00034     SHA1Internals *Internals;
00035 };
00036 } // end namespace gdcm
00037 //-----
00038 #endif //GDCMSHA1_H

```

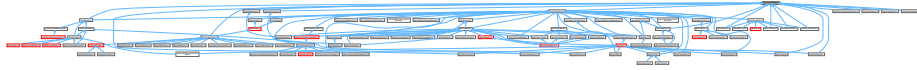
## 11.59 gdcmSmartPointer.h File Reference

```
#include "gdcmObject.h"
```

Include dependency graph for gdcmSmartPointer.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::SmartPointer< ObjectType >`  
*Class for Smart Pointer.*

## Namespaces

- namespace `gdcm`

## 11.60 gdcmSmartPointer.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMSMARTPOINTER_H
00015  #define GDCMSMARTPOINTER_H
00016
00017  #include "gdcmObject.h"
00018
00019  namespace gdcm
00020  {
00038  template<class ObjectType>
00039  class SmartPointer
00040  {
00041  public:
00042      SmartPointer():Pointer(nullptr) {}
00043      SmartPointer(const SmartPointer<ObjectType>& p):Pointer(p.Pointer)
00044      { Register(); }
00045      SmartPointer(ObjectType* p):Pointer(p)
00046      { Register(); }
00047      SmartPointer(ObjectType const & p)
00048      {
00049          Pointer = const_cast<ObjectType*>(&p);
00050          Register();
00051      }
00052      ~SmartPointer() {
00053          UnRegister();
00054          Pointer = nullptr;
00055      }
00056
00058      ObjectType *operator -> () const
00059      { return Pointer; }
00060
00061      ObjectType& operator * () const
00062      {
00063          assert( Pointer );
00064          return *Pointer;

```

```

00065     }
00066
00067     operator ObjectType * () const
00068     { return Pointer; }
00069
00070
00071     SmartPointer &operator = (SmartPointer const &r)
00072     { return operator = (r.Pointer); }
00073
00074
00075     SmartPointer &operator = (ObjectType *r)
00076     {
00077         // http://www.parashift.com/c++-faq-lite/freestore-mgmt.html#faq-16.22
00078         // DO NOT CHANGE THE ORDER OF THESE STATEMENTS!
00079         // (This order properly handles self-assignment)
00080         // (This order also properly handles recursion, e.g., if a ObjectType contains
00081         SmartPointer<ObjectType>s)
00082         if( Pointer != r )
00083         {
00084             ObjectType* old = Pointer;
00085             Pointer = r;
00086             Register();
00087             if ( old ) { old->UnRegister(); }
00088         }
00089         return *this;
00090     }
00091
00092     SmartPointer &operator = (ObjectType const &r)
00093     {
00094         ObjectType* tmp = const_cast<ObjectType*>(&r);
00095         return operator = (tmp);
00096     }
00097
00098     ObjectType *GetPointer() const
00099     { return Pointer; }
00100
00101 private:
00102     void Register()
00103     {
00104         if(Pointer) Pointer->Register();
00105     }
00106
00107     void UnRegister()
00108     {
00109         if(Pointer) Pointer->UnRegister();
00110     }
00111
00112     ObjectType* Pointer;
00113 };
00114
00115 // end namespace gdcmm
00116 #endif //GDCMSMARTPOINTER_H

```

## 11.61 gdcmmStaticAssert.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- struct [gdcmm::static\\_assert\\_test< x >](#)
- struct [gdcmm::STATIC\\_ASSERTION\\_FAILURE< true >](#)



## Namespaces

- namespace [gdcm](#)

## Macros

- #define [GDCM\\_DO\\_JOIN](#)(X, Y) [GDCM\\_DO\\_JOIN2](#)(X,Y)
- #define [GDCM\\_DO\\_JOIN2](#)(X, Y) X##Y
- #define [GDCM\\_JOIN](#)(X, Y) [GDCM\\_DO\\_JOIN](#)( X, Y )
- #define [GDCM\\_STATIC\\_ASSERT](#)(B)

*The [GDCM\\_JOIN](#) + **LINE** is needed to create a uniq identifier.*

## 11.61.1 Macro Definition Documentation

### 11.61.1.1 GDCM\_DO\_JOIN

```
#define GDCM_DO_JOIN(  
    X,  
    Y ) GDCM\_DO\_JOIN2 (X, Y)
```

### 11.61.1.2 GDCM\_DO\_JOIN2

```
#define GDCM_DO_JOIN2(  
    X,  
    Y ) X##Y
```

### 11.61.1.3 GDCM\_JOIN

```
#define GDCM_JOIN(  
    X,  
    Y ) GDCM\_DO\_JOIN( X, Y )
```

### 11.61.1.4 GDCM\_STATIC\_ASSERT

```
#define GDCM_STATIC_ASSERT(  
    B )
```

#### Value:

```
typedef ::gdcm::static_assert_test<\n    sizeof( :gdcm::STATIC_ASSERTION_FAILURE< (bool) ( B ) > )>\  
    GDCM_JOIN(gdcm_static_assert_typedef_, __LINE__)
```

The [GDCM\\_JOIN](#) + **LINE** is needed to create a uniq identifier.

## 11.62 gdcmStaticAssert.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMSTATICASSERT_H
00015  #define GDCMSTATICASSERT_H
00016
00017
00018  // the following was shamelessly borrowed from BOOST static assert:
00019  namespace gdcm
00020  {
00021      template <bool x>
00022      struct STATIC_ASSERTION_FAILURE;
00023
00024      template <>
00025      struct STATIC_ASSERTION_FAILURE<true> { enum { value = 1 }; };
00026
00027      template <int x>
00028      struct static_assert_test {};
00029  }
00030
00031  #define GDCM_JOIN( X, Y ) GDCM_DO_JOIN( X, Y )
00032  #define GDCM_DO_JOIN( X, Y ) GDCM_DO_JOIN2(X,Y)
00033  #define GDCM_DO_JOIN2( X, Y ) X##Y
00034
00035  #define GDCM_STATIC_ASSERT( B ) \
00036      typedef ::gdcm::static_assert_test<\
00037          sizeof(::gdcm::STATIC_ASSERTION_FAILURE< (bool) ( B ) >)>\
00038          GDCM_JOIN(gdcm_static_assert_typedef_, __LINE__)\
00039
00040
00041
00042  /* Example of use:
00043   *
00044   * template <class T>
00045   * struct must_not_be_instantiated
00046   * {
00047   * // this will be triggered if this type is instantiated
00048   * GDCM_STATIC_ASSERT(sizeof(T) == 0);
00049   * };
00050   *
00051   */
00052  #endif // GDCMSTATICASSERT_H

```

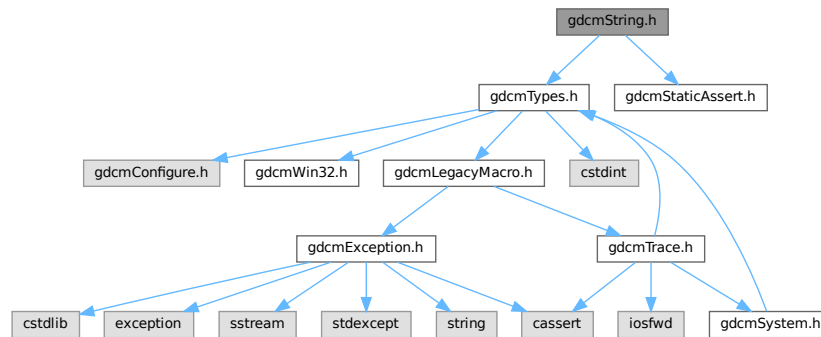
## 11.63 gdcmString.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmStaticAssert.h"

```

Include dependency graph for gdcmString.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::String< TDelimiter, TMaxLength, TPadChar >`  
*String*.

## Namespaces

- namespace `gdcm`

## Functions

- template<char TDelimiter, unsigned int TMaxLength, char TPadChar>  
`std::istream & gdcm::operator>> (std::istream &is, String< TDelimiter, TMaxLength, TPadChar > &ms)`

## 11.64 gdcmString.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
  
```

```

00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMSTRING_H
00015 #define GDCMSTRING_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmStaticAssert.h"
00019
00020 namespace gdcm
00021 {
00022
00023 template <char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
00031 class /*GDCM_EXPORT*/ String : public std::string /* PLEASE do not export me */
00032 {
00033 // UI wants \0 for pad character, while ASCII ones wants space char... do not allow anything else
00034 GDCM_STATIC_ASSERT( TPadChar == ' ' || TPadChar == 0 );
00035
00036 public:
00037 // typedef are not inherited:
00038 typedef std::string::value_type value_type;
00039 typedef std::string::pointer pointer;
00040 typedef std::string::reference reference;
00041 typedef std::string::const_reference const_reference;
00042 typedef std::string::size_type size_type;
00043 typedef std::string::difference_type difference_type;
00044 typedef std::string::iterator iterator;
00045 typedef std::string::const_iterator const_iterator;
00046 typedef std::string::reverse_iterator reverse_iterator;
00047 typedef std::string::const_reverse_iterator const_reverse_iterator;
00048
00050 String(): std::string() {}
00051 String(const value_type* s): std::string(s)
00052 {
00053 if( size() % 2 )
00054 {
00055 push_back( TPadChar );
00056 }
00057 }
00058 String(const value_type* s, size_type n): std::string(s, n)
00059 {
00060 // We are being passed a const char* pointer, so s[n] == 0 (guaranteed!)
00061 if( n % 2 )
00062 {
00063 push_back( TPadChar );
00064 }
00065 }
00066 String(const std::string& s, size_type pos=0, size_type n=npow):
00067 std::string(s, pos, n)
00068 {
00069 // FIXME: some users might already have padded the string 's' with a trailing \0...
00070 if( size() % 2 )
00071 {
00072 push_back( TPadChar );
00073 }
00074 }
00075
00077 operator const char *() const { return this->c_str(); }
00078
00080 bool IsValid() const {
00081 // Check Length:
00082 size_type l = size();
00083 if( l > TMaxLength ) return false;
00084 return true;
00085 }
00086
00087 gdcm::String<TDelimiter, TMaxLength, TPadChar> Truncate() const {
00088 if( !IsValid() ) return *this;
00089 std::string str = *this; // copy
00090 str.resize( TMaxLength );
00091 return str;
00092 }
00093
00096 std::string Trim() const {
00097 std::string str = *this; // copy
00098 std::string::size_type pos1 = str.find_first_not_of(' ');
00099 std::string::size_type pos2 = str.find_last_not_of(' ');

```

```

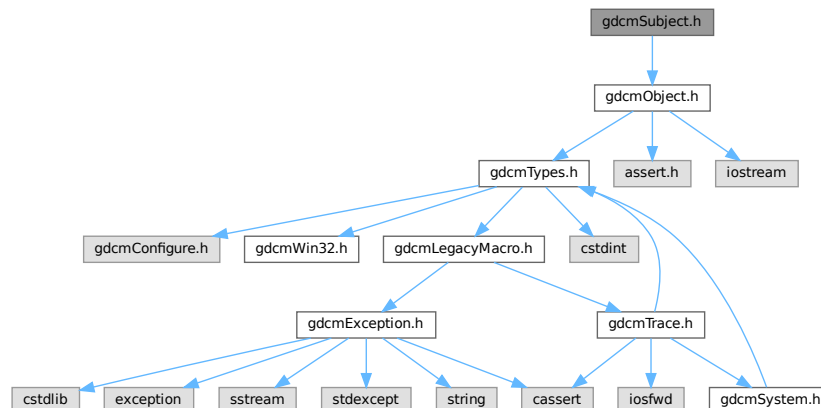
00100     str = str.substr( (pos1 == std::string::npos) ? 0 : pos1,
00101                     (pos2 == std::string::npos) ? (str.size() - 1) : (pos2 - pos1 + 1));
00102     return str;
00103 }
00104
00105 static std::string Trim(const char *input) {
00106     if( !input ) return "";
00107     std::string str = input;
00108     std::string::size_type pos1 = str.find_first_not_of(' ');
00109     std::string::size_type pos2 = str.find_last_not_of(' ');
00110     str = str.substr( (pos1 == std::string::npos) ? 0 : pos1,
00111                     (pos2 == std::string::npos) ? (str.size() - 1) : (pos2 - pos1 + 1));
00112     return str;
00113 }
00114 };
00115 template <char TDelimiter, unsigned int TMaxLength, char TPadChar>
00116 inline std::istream& operator>(std::istream &is, String<TDelimiter,TMaxLength,TPadChar> &ms)
00117 {
00118     if(is)
00119     {
00120         std::getline(is, ms, TDelimiter);
00121         // no such thing as std::get where the delim char would be left, so I need to manually add it back...
00122         // hopefully this is the right thing to do (no overhead)
00123         if( !is.eof() ) is.putback( TDelimiter );
00124     }
00125     return is;
00126 }
00127 //template <char TDelimiter = EOF, unsigned int TMaxLength = 64, char TPadChar = ' '>
00128 //String String::Trim() const
00129 //{
00130 //    String s;
00131 //    return s;
00132 //}
00133 } // end namespace gdcm
00134
00135 #endif //GDCMSTRING_H

```

## 11.65 gdcmSubject.h File Reference

#include "gdcmObject.h"

Include dependency graph for gdcmSubject.h:





```

00069
00070 private:
00071     SubjectInternals *Internals;
00072 private:
00073 };
00074
00075 } // end namespace gdcm
00076
00077 #endif //GDCMSUBJECT_H

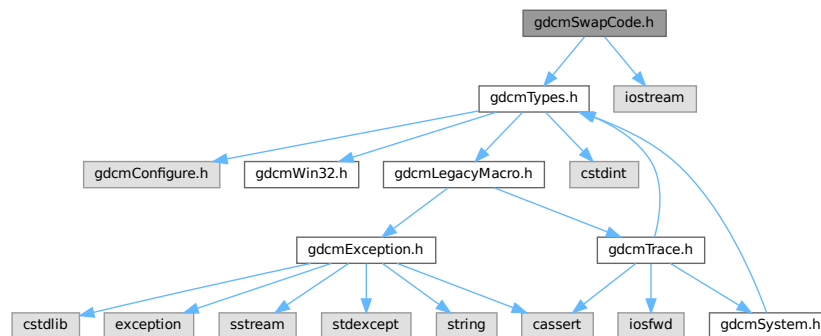
```

## 11.67 gdcmSwapCode.h File Reference

```
#include "gdcmTypes.h"
```

```
#include <iostream>
```

Include dependency graph for gdcmSwapCode.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class `gdcm::SwapCode`  
*SwapCode* representation.

### Namespaces

- namespace `gdcm`

### Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const SwapCode &sc)`

## 11.68 gdcmSwapCode.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMSWAPCODE_H
00015  #define GDCMSWAPCODE_H
00016
00017  #include "gdcmTypes.h"
00018  #include <iostream>
00019
00020  namespace gdcm
00021  {
00022
00026  class GDCM_EXPORT SwapCode
00027  {
00028  public:
00029      typedef enum {
00030          Unknown          = 0,
00031          LittleEndian     = 1234,
00032          BigEndian        = 4321,
00033          BadLittleEndian  = 3412,
00034          BadBigEndian     = 2143
00035      } SwapCodeType;
00036
00037      operator SwapCodeType() const { return SwapCodeValue; }
00038      SwapCodeType sc = Unknown; SwapCodeValue(sc) { }
00039      static const char* GetSwapCodeString(SwapCode const & sc);
00040
00041      friend std::ostream& operator<<(std::ostream& os, const SwapCode& sc);
00042  protected:
00043      static int GetIndex(SwapCode const & sc);
00044
00045  private:
00046      SwapCodeType SwapCodeValue;
00047  };
00048  //-----
00049  inline std::ostream& operator<<(std::ostream& os, const SwapCode& sc)
00050  {
00051      os << SwapCode::GetSwapCodeString(sc);
00052      return os;
00053  }
00054
00055  } // end namespace gdcm
00056
00057  #endif //GDCMSWAPCODE_H

```

## 11.69 gdcmSwapper.h File Reference

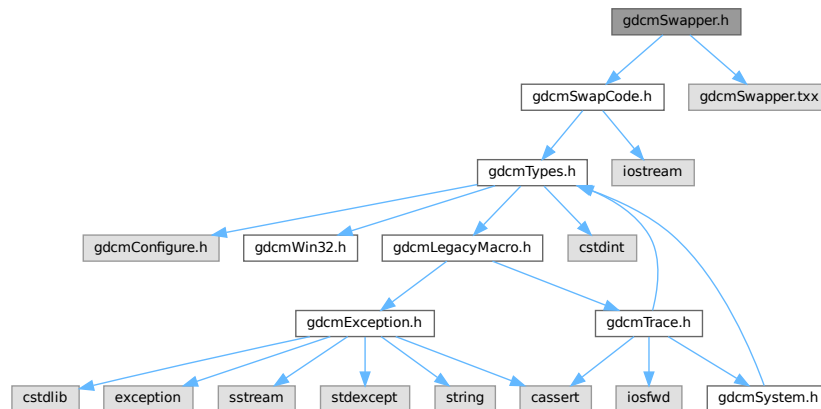
```

#include "gdcmSwapCode.h"
#include "gdcmSwapper.txx"

```



Include dependency graph for gdcmSwapper.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::SwapperDoOp](#)
- class [gdcm::SwapperNoOp](#)

## Namespaces

- namespace [gdcm](#)

## 11.70 gdcmSwapper.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMSWAPPER_H

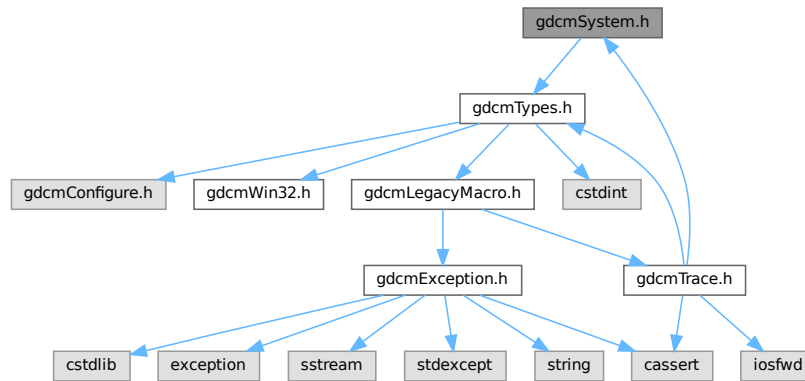
```

```
00015 #define GDCMSWAPPER_H
00016
00017 #include "gdcmSwapCode.h"
00018
00019 namespace gdcm
00020 {
00021
00022
00023 #ifdef GDCM_WORDS_BIGENDIAN
00024 class SwapperDoOp
00025 {
00026 public:
00027     template <typename T> static T Swap(T val) {return val;}
00028     template <typename T> static void SwapArray(T *, size_t ) {}
00029 };
00030
00031 class SwapperNoOp
00032 {
00033 public:
00034     template <typename T> static T Swap(T val);
00035     template <typename T>
00036     static void SwapArray(T *array, size_t n)
00037     {
00038         // TODO: need to unroll loop:
00039         for(size_t i = 0; i < n; ++i)
00040         {
00041             array[i] = Swap<T>(array[i]);
00042         }
00043     }
00044 };
00045 #else
00046 class SwapperNoOp
00047 {
00048 public:
00049     template <typename T> static T Swap(T val) {return val;}
00050     template <typename T> static void SwapArray(T *, size_t ) {}
00051 };
00052
00053 class SwapperDoOp
00054 {
00055 public:
00056     template <typename T> static T Swap(T val);
00057     template <typename T>
00058     static void SwapArray(T *array, size_t n)
00059     {
00060         // TODO: need to unroll loop:
00061         for(size_t i = 0; i < n; ++i)
00062         {
00063             array[i] = Swap<T>(array[i]);
00064         }
00065     }
00066 };
00067 #endif
00068
00069
00070 } // end namespace gdcm
00071
00072 #include "gdcmSwapper.txx"
00073
00074 #endif //GDCMSWAPPER_H
```

## 11.71 gdcmSystem.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmSystem.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [gdcm::System](#)  
*Class to do system operation.*

### Namespaces

- namespace [gdcm](#)

## 11.72 gdcmSystem.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
  
```

```

00011     PURPOSE.  See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMSYSTEM_H
00015 #define GDCMSYSTEM_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021
00022 class GDCM_EXPORT System
00023 {
00024 public:
00025     static bool MakeDirectory(const char *path);
00026     static bool FileExists(const char* filename);
00027     static bool FileIsDirectory(const char* name);
00028     static bool FileIsSymlink(const char* name);
00029     static bool RemoveFile(const char* source);
00030     static bool DeleteDirectory(const char *source);
00031
00032     static std::wstring ConvertToUNC(const char *utf8path);
00033
00034     static const char *GetLastError();
00035
00036     static size_t FileSize(const char* filename);
00037
00038     static time_t FileTime(const char* filename);
00039
00040     static const char *GetCurrentProcessFileName();
00041
00042     static const char *GetCurrentModuleFileName();
00043
00044     static const char *GetCurrentResourcesDirectory();
00045
00046     // TODO some system calls
00047     // Chdir
00048     // copy a file
00049
00050     static bool GetHostName(char hostname[255]);
00051
00052     // In the following the size '22' is explicitly listed. You need to pass in
00053     // at least 22bytes of array. If the string is an output it will be
00054     // automatically padded ( array[21] == 0 ) for you.
00055     // Those functions: GetCurrentDateTime / FormatDateTime / ParseDateTime do
00056     // not return the %YZZ part of the DT structure as defined in DICOM PS 3.5 -
00057     // 2008 In this case it is simple to split the date[22] into a DA and TM
00058     // structure
00059
00060     static bool GetCurrentDateTime(char date[22]);
00061
00062     static bool FormatDateTime(char date[22], time_t t, long milliseconds = 0);
00063
00064     static bool ParseDateTime(time_t &timep, const char date[22]);
00065
00066     static bool ParseDateTime(time_t &timep, long &milliseconds, const char date[22]);
00067
00068     static const char *GetTimezoneOffsetFromUTC();
00069
00070     static size_t EncodeBytes(char *out, const unsigned char *data, int size);
00071
00072     static int StrCaseCmp(const char *s1, const char *s2);
00073     static int StrNCaseCmp(const char *s1, const char *s2, size_t n);
00074
00075     static const char * GetCWD();
00076
00077     static char *StrTokR(char *ptr, const char *sep, char **end);
00078
00079     static char *StrSep(char **stringp, const char *delim);
00080
00081     static const char *GetLocaleCharset();
00082
00083     /*
00084     static void SetArgv0(const char *);
00085     static const char* GetArgv0();
00086     */
00087
00088 protected:
00089     static bool GetPermissions(const char* file, unsigned short& mode);
00090     static bool SetPermissions(const char* file, unsigned short mode);
00091

```

```

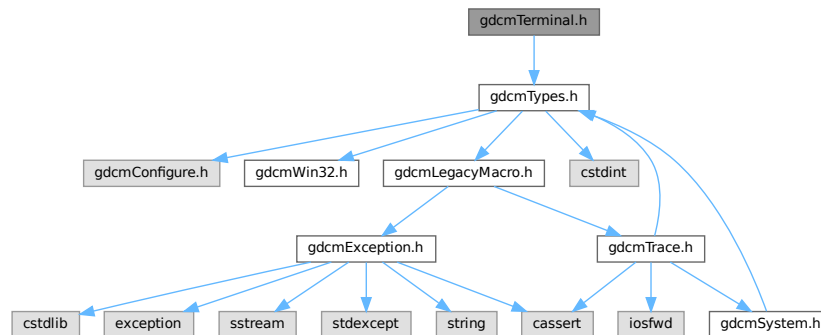
00146 private:
00147 };
00148
00149 } // end namespace gdcm
00150
00151 #endif //GDCMSYSTEM_H

```

## 11.73 gdcmTerminal.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmTerminal.h:



### Namespaces

- namespace `gdcm`
- namespace `gdcm::terminal`

*Class for Terminal.*

### Enumerations

- enum `gdcm::terminal::Attribute` {  
`gdcm::terminal::reset` = 0 ,  
`gdcm::terminal::bright` = 1 ,  
`gdcm::terminal::dim` = 2 ,  
`gdcm::terminal::underline` = 3 ,  
`gdcm::terminal::blink` = 5 ,  
`gdcm::terminal::reverse` = 7 ,  
`gdcm::terminal::hidden` = 8 }
- enum `gdcm::terminal::Color` {  
`gdcm::terminal::black` = 0 ,  
`gdcm::terminal::red` ,  
`gdcm::terminal::green` ,  
`gdcm::terminal::yellow` ,  
`gdcm::terminal::blue` ,  
`gdcm::terminal::magenta` ,  
`gdcm::terminal::cyan` ,  
`gdcm::terminal::white` }

- enum `gdcm::terminal::Mode` {  
`gdcm::terminal::CONSOLE` = 0 ,  
`gdcm::terminal::VT100` }

## Functions

- `GDCM_EXPORT` `std::string` `gdcm::terminal::setAttribute` (`Attribute` att)
- `GDCM_EXPORT` `std::string` `gdcm::terminal::setbgcolor` (`Color` c)
- `GDCM_EXPORT` `std::string` `gdcm::terminal::setfgcolor` (`Color` c)
- `GDCM_EXPORT` `void` `gdcm::terminal::setmode` (`Mode` m)

## 11.74 gdcmTerminal.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMTERMINAL_H
00015 #define GDCMTERMINAL_H
00016
00017 #include "gdcmTypes.h"
00018
00019
00020 namespace gdcm
00021 {
00022 //-----
00023 namespace terminal
00024 {
00025     typedef enum
00026     {
00027         CONSOLE = 0,
00028         VT100
00029     } Mode;
00030     typedef enum
00031     {
00032         black = 0,
00033         red,
00034         green,
00035         yellow, // brown ??
00036         blue,
00037         magenta,
00038         cyan,
00039         white
00040     } Color;
00041     typedef enum
00042     {
00043         reset = 0,
00044         bright = 1, // bold
00045         dim = 2,
00046         underline = 3,
00047         blink = 5,
00048         reverse = 7,
00049         hidden = 8
00050     } Attribute;
00051     GDCM_EXPORT std::string setAttribute( Attribute att );
00052     GDCM_EXPORT std::string setfgcolor( Color c );
00053     GDCM_EXPORT std::string setbgcolor( Color c );

```

```

00061  GDCM_EXPORT void setmode( Mode m);
00062  }
00063
00064  } // end namespace gdcm
00065  //-----
00066  #endif //GDCMTERMINAL_H

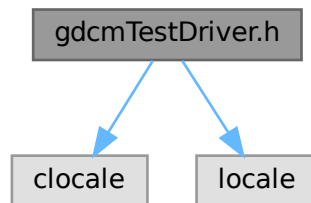
```

## 11.75 gdcmTestDriver.h File Reference

```
#include <clocale>
```

```
#include <locale>
```

Include dependency graph for gdcmTestDriver.h:



## 11.76 gdcmTestDriver.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  // This header is included by all the C++ test drivers in GDCM.
00015  #ifndef GDCMTESTDRIVER_H
00016  #define GDCMTESTDRIVER_H
00017
00018  // CREATE_TEST_SOURCELIST supports the flag EXTRA_INCLUDE but only one per call.
00019  // So there is no way to specify we want to include two files... instead
00020  // gather the #include in a single file and include that one...
00021  #include <clocale> // C setlocale()
00022  #include <locale> // C++ locale
00023
00024  #endif // GDCMTESTDRIVER_H

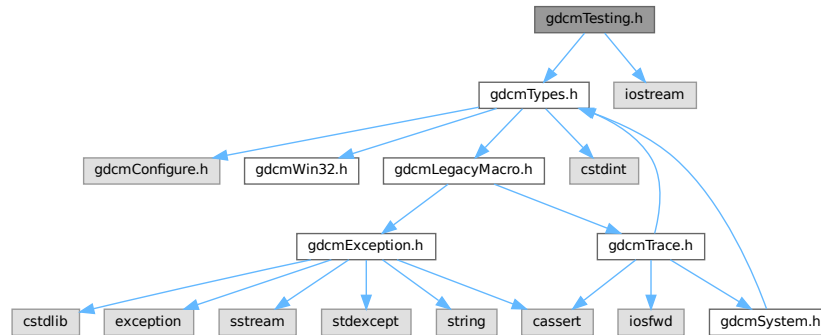
```

## 11.77 gdcmTesting.h File Reference

```
#include "gdcmTypes.h"
```

```
#include <iostream>
```

Include dependency graph for gdcmTesting.h:



### Classes

- class [gdcm::Testing](#)  
*class for testing*

### Namespaces

- namespace [gdcm](#)

## 11.78 gdcmTesting.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMTESTING_H
00015 #define GDCMTESTING_H
00016
00017 #include "gdcmTypes.h"
00018
00019 #include <iostream>
00020
00021 namespace gdcm

```



```

00022 {
00030 //-----
00031 class GDCM_EXPORT Testing
00032 {
00033 public :
00034     Testing() = default;
00035     ~Testing() = default;
00036
00042     static bool ComputeMD5(const char *buffer, size_t buf_len,
00043         char digest_str[33]);
00044     static bool ComputeFileMD5(const char *filename, char digest_str[33]);
00045
00047     void Print(std::ostream &os = std::cout);
00048
00050     static const char * const * GetFileNames();
00051     static unsigned int GetNumberOfFileNames();
00052     static const char * GetFileName(unsigned int file);
00053
00055     typedef const char* const (*MediaStorageDataFilesType) [2];
00056     static MediaStorageDataFilesType GetMediaStorageDataFiles();
00057     static unsigned int GetNumberOfMediaStorageDataFiles();
00058     static const char * const * GetMediaStorageDataFile(unsigned int file);
00059     static const char * GetMediaStorageFromFile(const char *filepath);
00060
00063     typedef const char* const (*MD5DataImagesType) [2];
00064     static MD5DataImagesType GetMD5DataImages();
00065     static unsigned int GetNumberOfMD5DataImages();
00066     static const char * const * GetMD5DataImage(unsigned int file);
00067     static const char * GetMD5FromFile(const char *filepath);
00068
00071     static const char * GetMD5FromBrokenFile(const char *filepath);
00072
00075     static std::streamoff GetStreamOffsetFromFile(const char *filepath);
00076
00080     static std::streamoff GetSelectedTagsOffsetFromFile(const char *filepath);
00081
00085     static std::streamoff GetSelectedPrivateGroupOffsetFromFile(const char *filepath);
00086
00091     static int GetLossyFlagFromFile(const char *filepath);
00092
00094     static const char * GetDataRoot();
00095
00097     static const char * GetDataExtraRoot();
00098
00100     static const char * GetPixelSpacingDataRoot();
00101
00104     static const char * GetTempDirectory(const char * subdir = nullptr);
00105
00107     static const wchar_t * GetTempDirectoryW(const wchar_t * subdir = nullptr);
00108
00110     static const char * GetTempFilename(const char *filename, const char * subdir = nullptr);
00111
00113     static const wchar_t* GetTempFilenameW(const wchar_t *filename, const wchar_t* subdir = nullptr);
00114
00115     static const char *GetSourceDirectory();
00116 };
00117 } // end namespace gdcm
00118 //-----
00119 #endif //GDCMTESTING_H

```

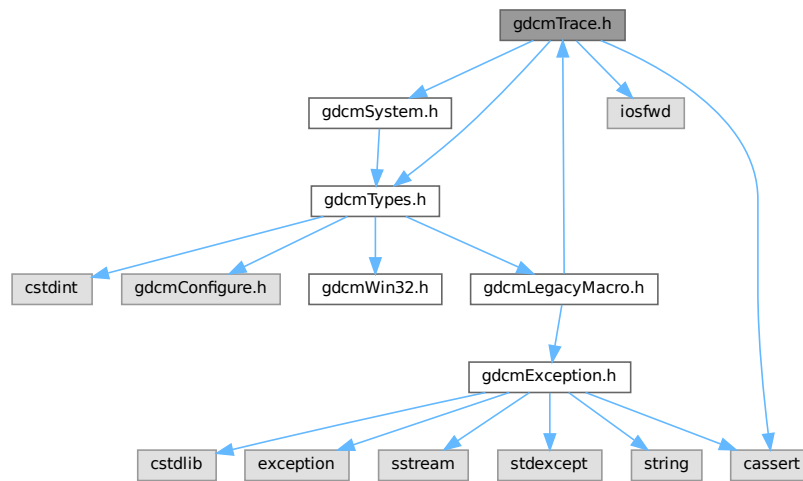
## 11.79 gdcmTrace.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmSystem.h"
#include <iosfwd>
#include <cassert>

```

Include dependency graph for `gdcmTrace.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::Trace`  
*Trace.*

## Namespaces

- namespace `gdcm`

## Macros

- `#define GDCM_FUNCTION "<unknown>"`
- `#define gdcmAssertAlwaysMacro(arg) gdcmAssertMacro(arg)`  
*AssertAlways.*
- `#define gdcmAssertMacro(arg)`  
*Assert.*
- `#define gdcmDebugMacro(msg)`  
*Debug.*
- `#define gdcmErrorMacro(msg)`  
*Error this is pretty bad, more than just warning It could mean lost of data, something not handle...*
- `#define gdcmWarningMacro(msg)`  
*Warning.*

## 11.79.1 Macro Definition Documentation

### 11.79.1.1 GDCM\_FUNCTION

```
#define GDCM_FUNCTION "<unknown>"
```

### 11.79.1.2 gdcmAssertAlwaysMacro

```
#define gdcmAssertAlwaysMacro(  
    arg ) gdcmAssertMacro(arg)
```

AssertAlways.

Parameters

<i>arg</i>	argument to test An easy solution to pass also a message is to do: <code>gdcmAssertMacro( "my message" &amp;&amp; 2 &lt; 3 )</code>
------------	---

### 11.79.1.3 gdcmAssertMacro

```
#define gdcmAssertMacro(  
    arg )
```

Value:

```
{  
    if( !(arg) )  
    {  
        std::ostringstream osmacro;  
        osmacro << "Assert: In " __FILE__ ", line " << __LINE__  
            << ", function " << GDCM_FUNCTION  
            << "\n\n";  
        std::ostream &_os = gdcm::Trace::GetErrorStream();  
        _os << osmacro.str() << std::endl;  
        assert ( arg );  
    }  
}  
GDCM_NOOP_STATEMENT
```

Assert.

Parameters

<i>arg</i>	argument to test An easy solution to pass also a message is to do: <code>gdcmAssertMacro( "my message" &amp;&amp; 2 &lt; 3 )</code>
------------	---

### 11.79.1.4 gdcmDebugMacro

```
#define gdcmDebugMacro(  
    msg )
```

Value:

```

{
    if( gdcm::Trace::GetDebugFlag() )
    {
        std::ostringstream osmacro;
        osmacro << "Debug: In " __FILE__ ", line " << __LINE__
            << ", function " << GDCM_FUNCTION << '\n'
            << "Last system error was: "
            << gdcm::System::GetLastSystemError() << '\n' << msg;
        std::ostream &_os = gdcm::Trace::GetDebugStream();
        _os << osmacro.str() << "\n\n" << std::endl;
    }
}
GDCM_NOOP_STATEMENT

```

Debug.

#### Parameters

<i>msg</i>	message part
------------	--------------

### 11.79.1.5 gdcmErrorMacro

```

#define gdcmErrorMacro(
    msg )

```

#### Value:

```

{
    if( gdcm::Trace::GetErrorFlag() )
    {
        std::ostringstream osmacro;
        osmacro << "Error: In " __FILE__ ", line " << __LINE__
            << ", function " << GDCM_FUNCTION << '\n'
            << msg << "\n\n";
        std::ostream &_os = gdcm::Trace::GetErrorStream();
        _os << osmacro.str() << std::endl;
    }
}
GDCM_NOOP_STATEMENT

```

Error this is pretty bad, more than just warning It could mean lost of data, something not handle...

#### Parameters

<i>msg</i>	second message part
------------	---------------------

### 11.79.1.6 gdcmWarningMacro

```

#define gdcmWarningMacro(
    msg )

```

#### Value:

```

{
    if( gdcm::Trace::GetWarningFlag() )
    {
        std::ostringstream osmacro;
        osmacro << "Warning: In " __FILE__ ", line " << __LINE__
            << ", function " << GDCM_FUNCTION << "\n"
            << msg << "\n\n";
    }
}

```

```

    std::ostream &_os = gdcm::Trace::GetWarningStream();
    _os << osmacro.str() << std::endl;
  }
}
GDCM_NOOP_STATEMENT

```

```

  \
  \
  \

```

Warning.

#### Parameters

<i>msg</i>	message part
------------	--------------

## 11.80 gdcmTrace.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMTRACE_H
00015 #define GDCMTRACE_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmSystem.h"
00019
00020 #include <iosfwd>
00021 #include <cassert>
00022
00023 namespace gdcm
00024 {
00025
00041 class GDCM_EXPORT Trace
00042 {
00043 public:
00044   Trace();
00045   ~Trace();
00046
00049   static void SetStream(std::ostream &os);
00050   static std::ostream &GetStream();
00051
00053   static void SetDebugStream(std::ostream &os);
00054   static std::ostream &GetDebugStream();
00055
00057   static void SetWarningStream(std::ostream &os);
00058   static std::ostream &GetWarningStream();
00059
00061   static void SetErrorStream(std::ostream &os);
00062   static std::ostream &GetErrorStream();
00063
00066   static void SetStreamToFile( const char *filename );
00067
00069   static void SetDebug(bool debug);
00070   static void DebugOn();
00071   static void DebugOff();
00072   static bool GetDebugFlag();
00073
00075   static void SetWarning(bool debug);
00076   static void WarningOn();
00077   static void WarningOff();
00078   static bool GetWarningFlag();
00079

```

```

00081     static void SetError(bool debug);
00082     static void ErrorOn();
00083     static void ErrorOff();
00084     static bool GetErrorFlag();
00085
00086 protected:
00087 private:
00088 };
00089
00090 // Here we define function this is the only way to be able to pass
00091 // stuff with indirection like:
00092 // gdcDebug( "my message:" « i « '\t' );
00093 // You cannot use function unless you use vnsprintf ...
00094
00095 // __FUNCTION is not always defined by preprocessor
00096 // In c++ we should use __PRETTY_FUNCTION__ instead...
00097 #ifdef GDCM_CXX_HAS_FUNCTION
00098 // Handle particular case for GNU C++ which also defines __PRETTY_FUNCTION__
00099 // which is a lot nice in C++
00100 #ifdef __BORLANDC__
00101 #   define __FUNCTION__ __FUNC__
00102 #endif
00103 #ifdef __GNUC__
00104 #   define GDCM_FUNCTION __PRETTY_FUNCTION__
00105 #else
00106 #   define GDCM_FUNCTION __FUNCTION__
00107 #endif //__GNUC__
00108 #else
00109 #   define GDCM_FUNCTION "<unknown>"
00110 #endif //GDCM_CXX_HAS_FUNCTION
00111
00112 #if defined(NDEBUG) && !defined(GDCM_ALWAYS_TRACE_MACRO)
00113 #define gdcDebugMacro(msg) GDCM_NOOP_STATEMENT
00114 #else
00115 #define gdcDebugMacro(msg)
00116 {
00117     if( gdc::Trace::GetDebugFlag() )
00118     {
00119         std::ostringstream osmacro;
00120         osmacro « "Debug: In " __FILE__ ", line " « __LINE__
00121             « ", function " « GDCM_FUNCTION « '\n'
00122             « "Last system error was: "
00123             « gdc::System::GetLastSystemError() « '\n' « msg;
00124         std::ostream &_os = gdc::Trace::GetDebugStream();
00125         _os « osmacro.str() « "\n\n" « std::endl;
00126     }
00127 }
00128 GDCM_NOOP_STATEMENT
00129 #endif //NDEBUG
00130
00131 #if defined(NDEBUG) && !defined(GDCM_ALWAYS_TRACE_MACRO)
00132 #define gdcWarningMacro(msg) GDCM_NOOP_STATEMENT
00133 #else
00134 #define gdcWarningMacro(msg)
00135 {
00136     if( gdc::Trace::GetWarningFlag() )
00137     {
00138         std::ostringstream osmacro;
00139         osmacro « "Warning: In " __FILE__ ", line " « __LINE__
00140             « ", function " « GDCM_FUNCTION « "\n"
00141             « msg « "\n\n";
00142         std::ostream &_os = gdc::Trace::GetWarningStream();
00143         _os « osmacro.str() « std::endl;
00144     }
00145 }
00146 GDCM_NOOP_STATEMENT
00147 #endif //NDEBUG
00148
00149 #if defined(NDEBUG) && !defined(GDCM_ALWAYS_TRACE_MACRO)
00150 #define gdcErrorMacro(msg) GDCM_NOOP_STATEMENT
00151 #else
00152 #define gdcErrorMacro(msg)
00153 {
00154     if( gdc::Trace::GetErrorFlag() )
00155     {
00156         std::ostringstream osmacro;
00157         osmacro « "Error: In " __FILE__ ", line " « __LINE__
00158             « ", function " « GDCM_FUNCTION « '\n'
00159             « msg « "\n\n";
00160         std::ostream &_os = gdc::Trace::GetErrorStream();
00161         _os « osmacro.str() « std::endl;
00162     }
00163 }

```

```

00175     }
00176 }
00177 GDCM_NOOP_STATEMENT
00178 #endif //NDEBUG
00179
00186 #if defined(NDEBUG) && !defined(GDCM_ALWAYS_TRACE_MACRO)
00187 #define gdcmAssertMacro(arg) GDCM_NOOP_STATEMENT
00188 #else
00189 #define gdcmAssertMacro(arg)
00190 {
00191     if( !(arg) )
00192     {
00193         std::ostringstream osmacro;
00194         osmacro << "Assert: In " __FILE__ ", line " << __LINE__
00195             << ", function " << GDCM_FUNCTION
00196             << "\n\n";
00197         std::ostream &_os = gdcm::Trace::GetErrorStream();
00198         _os << osmacro.str() << std::endl;
00199         assert ( arg );
00200     }
00201 }
00202 GDCM_NOOP_STATEMENT
00203 #endif //NDEBUG
00204
00211 #if defined(NDEBUG)
00212 // User asked for release compilation, but still need to report
00213 // if grave issue.
00214 #define gdcmAssertAlwaysMacro(arg) \
00215 {
00216     if( !(arg) )
00217     {
00218         std::ostringstream osmacro;
00219         osmacro << "Assert: In " __FILE__ ", line " << __LINE__
00220             << ", function " << GDCM_FUNCTION
00221             << "\n\n";
00222         throw osmacro.str();
00223     }
00224 }
00225 GDCM_NOOP_STATEMENT
00226 #else
00227 // Simply reproduce gdcmAssertMacro behavior:
00228 #define gdcmAssertAlwaysMacro(arg) gdcmAssertMacro(arg)
00229 #endif //NDEBUG
00230
00231 } // end namespace gdcm
00232 //-----
00233 #endif //GDCMTRACE_H

```

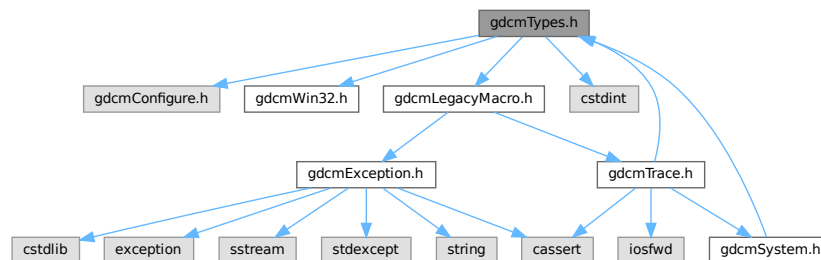
## 11.81 gdcmTypes.h File Reference

```

#include "gdcmConfigure.h"
#include "gdcmWin32.h"
#include "gdcmLegacyMacro.h"
#include <stdint>

```

Include dependency graph for gdcmTypes.h:



## 11.82 gdcmTypes.h

[Go to the documentation of this file.](#)

```

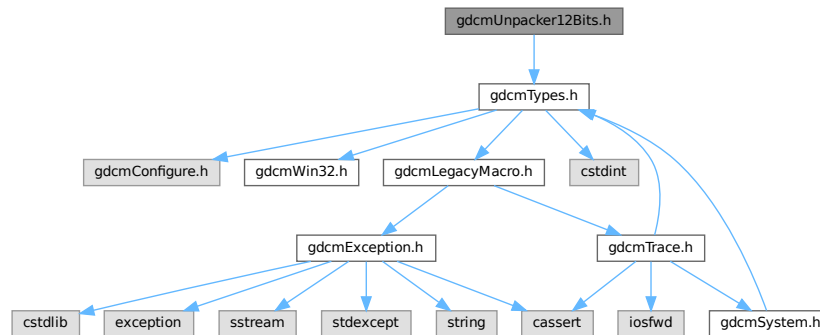
00001 /*=====
00002
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMTYPES_H
00015 #define GDCMTYPES_H
00016
00017 #include "gdcmConfigure.h"
00018 #include "gdcmWin32.h"
00019 #include "gdcmLegacyMacro.h"
00020
00021 //-----
00022 #include <stdint>
00023
00024 //-----
00025 #endif //GDCMTYPES_H

```

## 11.83 gdcmUnpacker12Bits.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmUnpacker12Bits.h:



### Classes

- class [gdcm::Unpacker12Bits](#)  
*Pack/Unpack 12 bits pixel into 16bits.*

### Namespaces

- namespace [gdcm](#)



## 11.84 gdcmUnpacker12Bits.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMUNPACKER12BITS_H
00015 #define GDCMUNPACKER12BITS_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00034   class GDCM_EXPORT Unpacker12Bits
00035   {
00036   public:
00040     static bool Pack(char *out, const char *in, size_t n);
00041
00045     static bool Unpack(char *out, const char *in, size_t n);
00046   };
00047
00048 } // end namespace gdcm
00049
00050 #endif //GDCMUNPACKER12BITS_H

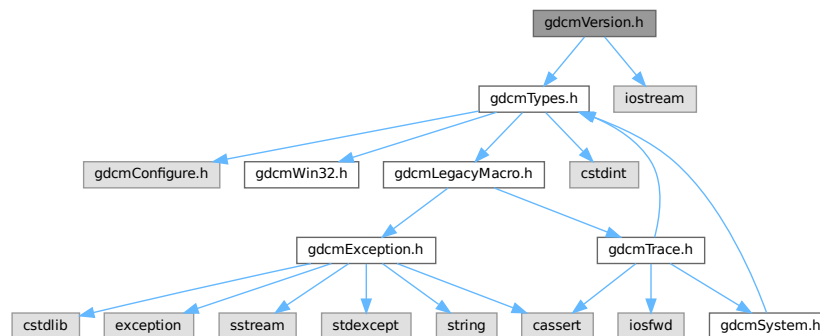
```

## 11.85 gdcmVersion.h File Reference

```
#include "gdcmTypes.h"
```

```
#include <iostream>
```

Include dependency graph for gdcmVersion.h:



### Classes

- class [gdcm::Version](#)  
major/minor and build version

## Namespaces

- namespace `gdcm`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Version &v)`

## 11.86 gdcmVersion.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMVERSION_H
00015 #define GDCMVERSION_H
00016
00017 #include "gdcmTypes.h"
00018 #include <iostream>
00019
00020 namespace gdcm
00021 {
00022     //-----
00023     class GDCM_EXPORT Version
00024     {
00025     public:
00026         friend std::ostream& operator<<(std::ostream &os, const Version &v);
00027         static const char *GetVersion();
00028         static int GetMajorVersion();
00029         static int GetMinorVersion();
00030         static int GetBuildVersion();
00031
00032         void Print(std::ostream &os = std::cout) const;
00033
00034     protected:
00035         Version() = default;
00036         ~Version() = default;
00037     };
00038     //-----
00039     inline std::ostream& operator<<(std::ostream &os, const Version &v)
00040     {
00041         v.Print( os );
00042         return os;
00043     }
00044 } // end namespace gdcm
00045 //-----
00046 #endif //GDCMVERSION_H

```

## 11.87 gdcmWin32.h File Reference

This graph shows which files directly or indirectly include this file:



**Macros**

- `#define GDCM_EXPORT`

**11.87.1 Macro Definition Documentation****11.87.1.1 GDCM\_EXPORT**

```
#define GDCM_EXPORT
```

**11.88 gdcmWin32.h**

[Go to the documentation of this file.](#)

```
00001 /*=====
00002
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014
00015 #ifndef GDCMWIN32_H
00016 #define GDCMWIN32_H
00017
00018 #if !defined(GDCMTYPES_H)
00019 #error you need to include gdcmTypes.h instead
00020 #endif
00021 //-----
00022 // http://gcc.gnu.org/wiki/Visibility
00023 #if defined(_WIN32) && defined(GDCM_BUILD_SHARED_LIBS)
00024 #if (defined(gdcmCommon_EXPORTS) || defined(gdcmDICT_EXPORTS) || defined(gdcmDSED_EXPORTS) ||
00025     defined(gdcmIOD_EXPORTS) || defined(gdcmMSFF_EXPORTS) || defined(gdcmMEXD_EXPORTS) ||
00026     defined(_gdcmSwig_EXPORTS)) || defined(vtkgdcm_EXPORTS)
00027     #define GDCM_EXPORT __declspec( dllexport )
00028 #else
00029     #define GDCM_EXPORT __declspec( dllimport )
00030 #endif
00031 #else
00032 #if __GNUC__ >= 4 && defined(GDCM_BUILD_SHARED_LIBS)
00033     #define GDCM_EXPORT __attribute__ ((visibility ("default")))
00034     #define GDCM_LOCAL __attribute__ ((visibility ("hidden")))
00035 #else
00036     #define GDCM_EXPORT
00037 #endif
00038 #endif
00039 #undef GDCM_EXPORT
00040 #define GDCM_EXPORT
00041 #endif
00042
00043 // In VTK 4.2 vtkWrapPython does not like anything other than VTK_*EXPORT
00044 // [ 86%] Generating vtkGDCMImageReaderPython.cxx
00045 // syntax error
00046 // *** SYNTAX ERROR found in parsing the header file
00047 // /usr/local/src/gdcm2/tags/gdcm-2-0-11/Utilities/VTK/vtkGDCMImageReader.h before line 128***
00048 // make[2]: *** [Utilities/VTK/vtkGDCMImageReaderPython.cxx] Error 1
00049 // make[1]: *** [Utilities/VTK/CMakeFiles/vtkgdcmPythonD.dir/all] Error 2
00050 // make: *** [all] Error 2
00051
00052 #if defined(VTK_MAJOR_VERSION) && ( VTK_MAJOR_VERSION == 4 )
```

```

00052 #undef VTK_EXPORT
00053 #define VTK_EXPORT GDCM_EXPORT
00054 #endif
00055
00056 //-----
00057 //This is needed when compiling in debug mode
00058 #ifdef _MSC_VER
00059 // to allow construct such as: std::numeric_limits<int>::max() we need the following:
00060 // warning C4003: not enough actual parameters for macro 'max'
00061 #ifndef NOMINMAX
00062     #define NOMINMAX
00063 #endif
00064 # pragma warning ( default : 4263 ) /* no override, call convention differs */
00065 // 'identifier' : class 'type' needs to have dll-interface to be used by
00066 // clients of class 'type2'
00067 #pragma warning ( disable : 4251 )
00068 // non dll-interface class 'type' used as base for dll-interface class 'type2'
00069 #pragma warning ( disable : 4275 )
00070 // 'identifier' : identifier was truncated to 'number' characters in the
00071 // debug information
00072 #pragma warning ( disable : 4786 )
00073 // 'identifier' : decorated name length exceeded, name was truncated
00074 #pragma warning ( disable : 4503 )
00075 #endif // _MSC_VER
00076
00077 //-----
00078 #endif //GDCMWIN32_H

```

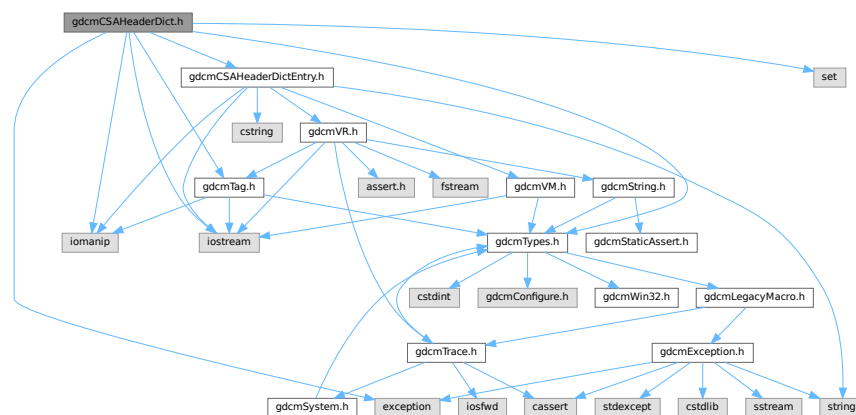
## 11.89 gdcmlCSAHeaderDict.h File Reference

```

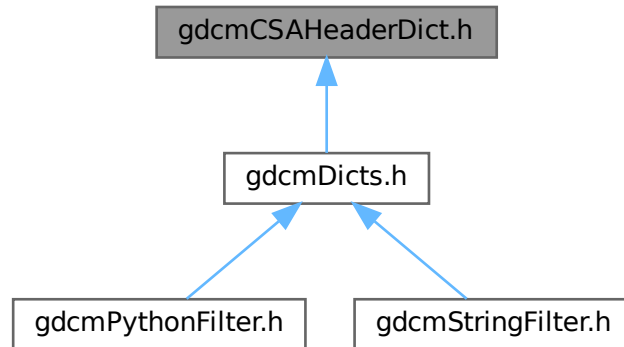
#include "gdcmlTypes.h"
#include "gdcmlTag.h"
#include "gdcmlCSAHeaderDictEntry.h"
#include <iostream>
#include <iomanip>
#include <set>
#include <exception>

```

Include dependency graph for gdcmlCSAHeaderDict.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcml::CSAHeaderDict](#)  
*Class to represent a map of [CSAHeaderDictEntry](#).*
- class [gdcml::CSAHeaderDictException](#)

## Namespaces

- namespace [gdcml](#)

## Functions

- `std::ostream & gdcml::operator<< (std::ostream &os, const CSAHeaderDict &val)`

## 11.90 gdcmlCSAHeaderDict.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMLCSAHEADERDICT_H

```

```

00015 #define GDCMCSAHEADERDICT_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmTag.h"
00019 #include "gdcmCSAHeaderDictEntry.h"
00020
00021 #include <iostream>
00022 #include <iomanip>
00023 #include <set>
00024 #include <exception>
00025
00026 namespace gdcm
00027 {
00028
00029 class GDCM_EXPORT CSAHeaderDictException : public std::exception {};
00030
00031 class GDCM_EXPORT CSAHeaderDict
00032 {
00033 public:
00034     typedef std::set<CSAHeaderDictEntry> MapCSAHeaderDictEntry;
00035     typedef MapCSAHeaderDictEntry::iterator Iterator;
00036     typedef MapCSAHeaderDictEntry::const_iterator ConstIterator;
00037     //static CSAHeaderDictEntry GroupLengthCSAHeaderDictEntry; // = CSAHeaderDictEntry("Group
00038     Length",VR::UL,VM::VMI);
00039
00040     CSAHeaderDict():CSAHeaderDictInternal() {
00041         assert( CSAHeaderDictInternal.empty() );
00042     }
00043     CSAHeaderDict &operator=(const CSAHeaderDict &_val) = delete;
00044     CSAHeaderDict(const CSAHeaderDict &_val) = delete;
00045
00046     friend std::ostream& operator<<(std::ostream& _os, const CSAHeaderDict &_val);
00047
00048     ConstIterator Begin() const { return CSAHeaderDictInternal.begin(); }
00049     ConstIterator End() const { return CSAHeaderDictInternal.end(); }
00050
00051     bool IsEmpty() const { return CSAHeaderDictInternal.empty(); }
00052     void AddCSAHeaderDictEntry(const CSAHeaderDictEntry &de)
00053     {
00054 #ifndef NDEBUG
00055         MapCSAHeaderDictEntry::size_type s = CSAHeaderDictInternal.size();
00056 #endif
00057         CSAHeaderDictInternal.insert( de );
00058         assert( s < CSAHeaderDictInternal.size() );
00059     }
00060
00061     const CSAHeaderDictEntry &GetCSAHeaderDictEntry(const char *name) const
00062     {
00063         MapCSAHeaderDictEntry::const_iterator it = CSAHeaderDictInternal.find( name );
00064         if( it != CSAHeaderDictInternal.end() )
00065         {
00066             return *it;
00067         }
00068         throw CSAHeaderDictException();
00069     }
00070
00071 protected:
00072     friend class Dicts;
00073     void LoadDefault();
00074
00075 private:
00076     MapCSAHeaderDictEntry CSAHeaderDictInternal;
00077 };
00078
00079 //-----
00080 inline std::ostream& operator<<(std::ostream& os, const CSAHeaderDict &val)
00081 {
00082     CSAHeaderDict::MapCSAHeaderDictEntry::const_iterator it = val.CSAHeaderDictInternal.begin();
00083     for(; it != val.CSAHeaderDictInternal.end(); ++it)
00084     {
00085         const CSAHeaderDictEntry &de = *it;
00086         os << de << '\n';
00087     }
00088
00089     return os;
00090 }
00091
00092 // end namespace gdcm
00093
00094
00095
00096
00097

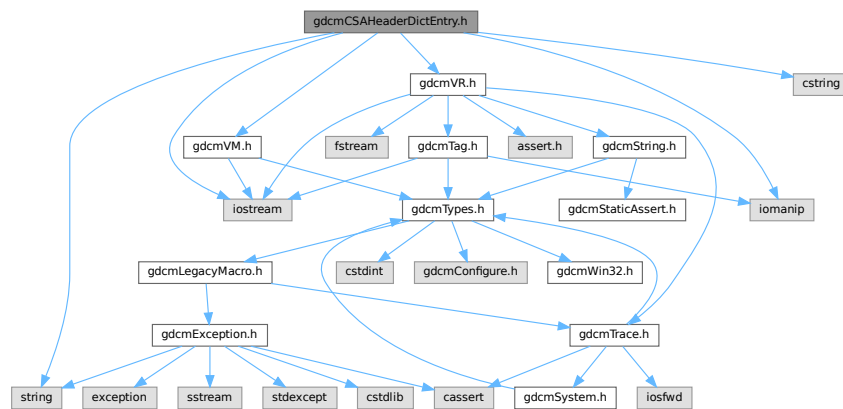
```

```
00098 #endif //GDCMCSAHEADERDICT_H
```

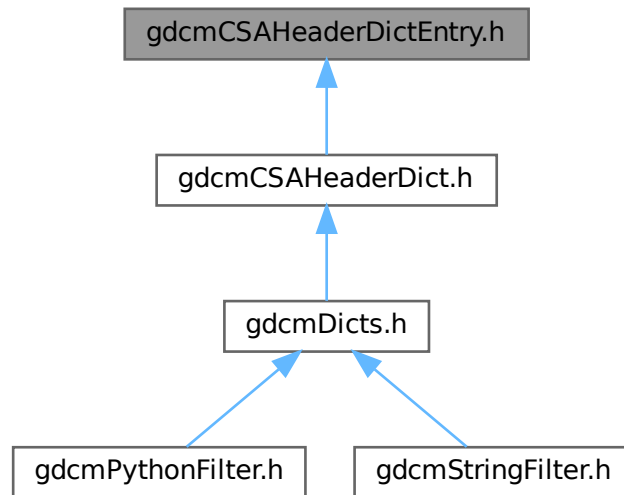
## 11.91 gdcmCSAHeaderDictEntry.h File Reference

```
#include "gdcmVR.h"
#include "gdcmVM.h"
#include <string>
#include <iostream>
#include <iomanip>
#include <cstring>
```

Include dependency graph for gdcmCSAHeaderDictEntry.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcml::CSAHeaderDictEntry](#)  
Class to represent an Entry in the [Dict](#).

## Namespaces

- namespace [gdcml](#)

## Functions

- `std::ostream & gdcml::operator<< (std::ostream &os, const CSAHeaderDictEntry &val)`

## 11.92 gdcmlCSAHeaderDictEntry.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
  
```



```

00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 #ifndef GDCMCSAHEADERDICTENTRY_H
00015 #define GDCMCSAHEADERDICTENTRY_H
00016
00017 #include "gdcmVR.h"
00018 #include "gdcmVM.h"
00019
00020 #include <string>
00021 #include <iostream>
00022 #include <iomanip>
00023
00024 #include <cstring>
00025
00026 namespace gdcm
00027 {
00028 class GDCM_EXPORT CSAHeaderDictEntry
00029 {
00030 public:
00031     CSAHeaderDictEntry(const char *name = "", VR const &vr = VR::INVALID, VM const &vm = VM::VM0, const char
00032 *desc = ""):Name(name),ValueRepresentation(vr),ValueMultiplicity(vm),Description(desc) {
00033     }
00034
00035     friend std::ostream& operator<(std::ostream& _os, const CSAHeaderDictEntry &_val);
00036
00037     const VR &GetVR() const { return ValueRepresentation; }
00038     void SetVR(const VR &vr) { ValueRepresentation = vr; }
00039
00040     const VM &GetVM() const { return ValueMultiplicity; }
00041     void SetVM(VM const &vm) { ValueMultiplicity = vm; }
00042
00043     const char *GetName() const { return Name.c_str(); }
00044     void SetName(const char* name) { Name = name; }
00045
00046     const char *GetDescription() const { return Description.c_str(); }
00047     void SetDescription(const char* desc) { Description = desc; }
00048
00049     bool operator<(const CSAHeaderDictEntry &entry) const
00050     {
00051         return strcmp(GetName(),entry.GetName()) < 0;
00052     }
00053 private:
00054     std::string Name;
00055     VR ValueRepresentation;
00056     VM ValueMultiplicity;
00057     std::string Description;
00058     std::string Type; // TODO
00059 };
00060
00061 //-----
00062 inline std::ostream& operator<(std::ostream& os, const CSAHeaderDictEntry &val)
00063 {
00064     if( val.Name.empty() )
00065     {
00066         os << "[No name]";
00067     }
00068     else
00069     {
00070         os << val.Name;
00071     }
00072     os << "\t" << val.ValueRepresentation << "\t" << val.ValueMultiplicity;
00073     if( !val.Description.empty() )
00074     {
00075         os << "\t" << val.Description;
00076     }
00077     return os;
00078 }
00079
00080 } // end namespace gdcm
00081 #endif //GDCMCSAHEADERDICTENTRY_H

```



## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Dict &val)`
- `std::ostream & gdcm::operator<< (std::ostream &os, const PrivateDict &val)`

## 11.94 gdcmDict.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMDICT_H
00015 #define GDCMDICT_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmTag.h"
00019 #include "gdcmPrivateTag.h"
00020 #include "gdcmDictEntry.h"
00021 #include "gdcmSystem.h"
00022
00023 #include <iostream>
00024 #include <iomanip>
00025 #include <map>
00026
00027 /*
00028  * FIXME / TODO
00029  * I need to seriously rewrite this mess. a class template should work for both a public
00030  * and a private dict
00031  */
00032
00033 namespace gdcm
00034 {
00035   // Data Element Tag
00044   class GDCM_EXPORT Dict
00045   {
00046   public:
00047     typedef std::map<Tag, DictEntry> MapDictEntry;
00048     typedef MapDictEntry::iterator Iterator;
00049     typedef MapDictEntry::const_iterator ConstIterator;
00050     //static DictEntry GroupLengthDictEntry; // = DictEntry("Group Length",VR::UL,VM::VM1);
00051
00052     Dict():DictInternal() {
00053       assert( DictInternal.empty() );
00054     }
00055     Dict &operator=(const Dict &_val) = delete;
00056     Dict(const Dict &_val) = delete;
00057
00058
00059     friend std::ostream& operator<<(std::ostream& _os, const Dict &_val);
00060
00061     ConstIterator Begin() const { return DictInternal.begin(); }
00062     ConstIterator End() const { return DictInternal.end(); }
00063
00064     bool IsEmpty() const { return DictInternal.empty(); }
00065     void AddDictEntry(const Tag &tag, const DictEntry &de)
00066     {
00067     #ifndef NDEBUG
00068       MapDictEntry::size_type s = DictInternal.size();
00069     #endif
00070       DictInternal.insert(
00071         MapDictEntry::value_type(tag, de));
00072       assert( s < DictInternal.size() );
00073     }

```

```

00074
00075     const DictEntry &GetDictEntry(const Tag &tag) const
00076     {
00077         MapDictEntry::const_iterator it =
00078             DictInternal.find(tag);
00079         if (it == DictInternal.end())
00080         {
00081             #ifndef UNKNOWNPUBLICTAG
00082                 // test.acr
00083                 if( tag != Tag(0x28,0x15)
00084                     && tag != Tag(0x28,0x16)
00085                     && tag != Tag(0x28,0x199)
00086                     // gdcmData/TherapysGDCM1.dcm
00087                     && tag != Tag(0x20,0x1)
00088                     // gdcmData/0019004_Baseline_IMG1.dcm
00089                     && tag != Tag(0x8348,0x339)
00090                     && tag != Tag(0xb5e8,0x338)
00091                     // gdcmData/dicomdir_Acusson_WithPrivate_WithSR
00092                     && tag != Tag(0x40,0xa125)
00093                 )
00094                 {
00095                     assert( 0 && "Impossible" );
00096                 }
00097             #endif
00098             it = DictInternal.find( Tag(0xffff,0xffff) );
00099             return it->second;
00100         }
00101         assert( DictInternal.count(tag) == 1 );
00102         return it->second;
00103     }
00104
00106     const char *GetKeywordFromTag(Tag const & tag) const
00107     {
00108         MapDictEntry::const_iterator it =
00109             DictInternal.find(tag);
00110         if (it == DictInternal.end())
00111         {
00112             return nullptr;
00113         }
00114         assert( DictInternal.count(tag) == 1 );
00115         return it->second.GetKeyword();
00116     }
00117
00122     const DictEntry &GetDictEntryByKeyword(const char *keyword, Tag & tag) const
00123     {
00124         MapDictEntry::const_iterator it =
00125             DictInternal.begin();
00126         if( keyword )
00127         {
00128             for(; it != DictInternal.end(); ++it)
00129             {
00130                 if( strcmp( keyword, it->second.GetKeyword() ) == 0 )
00131                 {
00132                     // Found a match !
00133                     tag = it->first;
00134                     break;
00135                 }
00136             }
00137         }
00138         else
00139         {
00140             it = DictInternal.end();
00141         }
00142         if (it == DictInternal.end())
00143         {
00144             tag = Tag(0xffff,0xffff);
00145             it = DictInternal.find( tag );
00146             return it->second;
00147         }
00148         assert( DictInternal.count(tag) == 1 );
00149         return it->second;
00150     }
00151
00155     const DictEntry &GetDictEntryByName(const char *name, Tag & tag) const
00156     {
00157         MapDictEntry::const_iterator it =
00158             DictInternal.begin();
00159         if( name )
00160         {
00161             for(; it != DictInternal.end(); ++it)
00162             {

```

```

00163         if( strcmp( name, it->second.GetName() ) == 0 )
00164         {
00165             // Found a match !
00166             tag = it->first;
00167             break;
00168         }
00169     }
00170 }
00171 else
00172 {
00173     it = DictInternal.end();
00174 }
00175 if (it == DictInternal.end())
00176 {
00177     tag = Tag(0xffff,0xffff);
00178     it = DictInternal.find( tag );
00179     return it->second;
00180 }
00181 assert( DictInternal.count(tag) == 1 );
00182 return it->second;
00183 }
00184
00185 protected:
00186     friend class Dicts;
00187     void LoadDefault();
00188
00189 private:
00190     MapDictEntry DictInternal;
00191 };
00192 //-----
00193 inline std::ostream& operator<(std::ostream& os, const Dict &val)
00194 {
00195     Dict::MapDictEntry::const_iterator it = val.DictInternal.begin();
00196     for(; it != val.DictInternal.end(); ++it)
00197     {
00198         const Tag &t = it->first;
00199         const DictEntry &de = it->second;
00200         os << t << " " << de << '\n';
00201     }
00202
00203     return os;
00204 }
00205
00206 // TODO
00207 // For private dict, element < 0x10 should automatically defined:
00208 // Name = "Private Creator"
00209 // ValueRepresentation = LO
00210 // ValueMultiplicity = 1
00211 // Owner = ""
00212
00216 class GDCM_EXPORT PrivateDict
00217 {
00218     typedef std::map<PrivateTag, DictEntry> MapDictEntry;
00219     friend std::ostream& operator<(std::ostream& os, const PrivateDict &val);
00220 public:
00221     PrivateDict() = default;
00222     ~PrivateDict() = default;
00223     void AddDictEntry(const PrivateTag &tag, const DictEntry &de)
00224     {
00225 #ifndef NDEBUG
00226         MapDictEntry::size_type s = DictInternal.size();
00227 #endif
00228         DictInternal.insert(
00229             MapDictEntry::value_type(tag, de));
00230 // The following code should only be used when manually constructing a Private.xml file by hand
00231 // it will get rid of VR::UN duplicate (ie. if a VR != VR::UN can be found)
00232 #if defined(NDEBUG) && 0
00233         if( s == DictInternal.size() )
00234         {
00235             MapDictEntry::iterator it =
00236                 DictInternal.find(tag);
00237             assert( it != DictInternal.end() );
00238             DictEntry &duplicate = it->second;
00239             assert( de.GetVR() == VR::UN || duplicate.GetVR() == VR::UN );
00240             assert( de.GetVR() != duplicate.GetVR() );
00241             if( duplicate.GetVR() == VR::UN )
00242             {
00243                 assert( de.GetVR() != VR::UN );
00244                 duplicate.SetVR( de.GetVR() );
00245                 duplicate.SetVM( de.GetVM() );
00246                 assert( GetDictEntry(tag).GetVR() != VR::UN );

```

```

00247         assert( GetDictEntry(tag).GetVR() == de.GetVR() );
00248         assert( GetDictEntry(tag).GetVM() == de.GetVM() );
00249     }
00250     return;
00251 }
00252 #endif
00253     assert( s < DictInternal.size() /*&& std::cout << tag << ", " << de << std::endl*/ );
00254 }
00255 bool RemoveDictEntry(const PrivateTag &tag)
00256 {
00257     MapDictEntry::size_type s =
00258         DictInternal.erase(tag);
00259     assert( s == 1 || s == 0 );
00260     return s == 1;
00261 }
00262 bool FindDictEntry(const PrivateTag &tag) const
00263 {
00264     MapDictEntry::const_iterator it =
00265         DictInternal.find(tag);
00266     if (it == DictInternal.end())
00267     {
00268         return false;
00269     }
00270     return true;
00271 }
00272 const DictEntry &GetDictEntry(const PrivateTag &tag) const
00273 {
00274     // if 0x10 -> return Private Creator
00275     MapDictEntry::const_iterator it =
00276         DictInternal.find(tag);
00277     if (it == DictInternal.end())
00278     {
00279         //assert( 0 && "Impossible" );
00280         it = DictInternal.find( PrivateTag(0xffff,0xffff,"GDCM Private Sentinel" ) );
00281         assert( it != DictInternal.end() );
00282         return it->second;
00283     }
00284     assert( DictInternal.count(tag) == 1 );
00285     return it->second;
00286 }
00287 void PrintXML() const
00288 {
00289     MapDictEntry::const_iterator it = DictInternal.begin();
00290     std::cout << "<dict edition=\"2008\">\n";
00291     for(;it != DictInternal.end(); ++it)
00292     {
00293         const PrivateTag &t = it->first;
00294         const DictEntry &de = it->second;
00295         std::cout << "  <entry group=\"" << std::hex << std::setw(4)
00296             << std::setfill('0') << t.GetGroup() << "\" <<
00297             " element=\"" << std::setw(2) << std::setfill('0') << t.GetElement() << "\" <<
00298             " vr=\"" << de.GetVR() << "\" vm=\"" << de.GetVM() << "\" owner=\""
00299             << t.GetOwner();
00300         const char *name = de.GetName();
00301         if( *name == 0 )
00302         {
00303             std::cout << "\"/>\n";
00304         }
00305         else
00306         {
00307             std::cout << "\" name=\"" << de.GetName() << "\"/>\n";
00308         }
00309     }
00310     std::cout << "</dict>\n";
00311 }
00312 bool IsEmpty() const { return DictInternal.empty(); }
00313 protected:
00314     friend class Dicts;
00315     void LoadDefault();
00316 private:
00317     PrivateDict &operator=(const PrivateDict &_val) = delete;
00318     PrivateDict(const PrivateDict &_val) = delete;
00319     MapDictEntry DictInternal;
00320 };
00321 //-----
00322 inline std::ostream& operator<<(std::ostream& os, const PrivateDict &val)

```

```

00330 {
00331     PrivateDict::MapDictEntry::const_iterator it = val.DictInternal.begin();
00332     for(; it != val.DictInternal.end(); ++it)
00333     {
00334         const PrivateTag &t = it->first;
00335         const DictEntry &de = it->second;
00336         os << t << " " << de << '\n';
00337     }
00338
00339     return os;
00340 }
00341
00342 } // end namespace gdcm
00343
00344 #endif //GDCMDICT_H

```

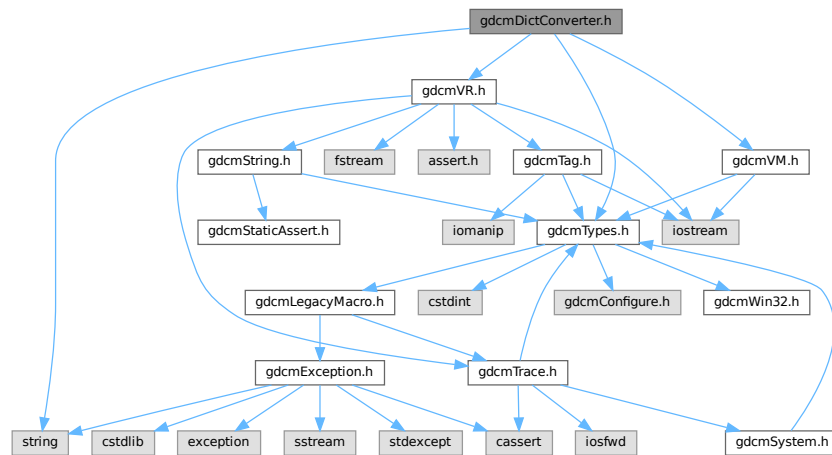
## 11.95 gdcmDictConverter.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmVM.h"
#include <string>

```

Include dependency graph for gdcmDictConverter.h:



### Classes

- class [gdcm::DictConverter](#)  
Class to convert a .dic file into something else:

### Namespaces

- namespace [gdcm](#)

## 11.96 gdcmDictConverter.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014
00015 #ifndef GDCMDICTCONVERTER_H
00016 #define GDCMDICTCONVERTER_H
00017
00018 #include "gdcmTypes.h"
00019 #include "gdcmVR.h"
00020 #include "gdcmVM.h"
00021
00022 #include <string>
00023
00024 namespace gdcm
00025 {
00026
00027   class DictConverterInternal;
00036   class GDCM_EXPORT DictConverter
00037   {
00038   public:
00039     DictConverter();
00040     ~DictConverter();
00041     void SetInputFileName(const char* filename);
00042     const std::string &GetInputFilename() const;
00043     void SetOutputFileName(const char* filename);
00044     const std::string &GetOutputFilename() const;
00045
00046     int GetOutputType() const {
00047       return OutputType;
00048     }
00049     void SetOutputType(int type) {
00050       OutputType = type;
00051     }
00052     const std::string &GetDictName() const;
00053     void SetDictName(const char *name);
00054
00055     void Convert();
00056
00057     // Leaving them public for now. Not really user oriented but may be
00058     // useful
00059     static bool ReadVR(const char *raw, VR::VRType &type);
00060     static bool ReadVM(const char *raw, VM::VMType &type);
00061     static bool Readuint16(const char *raw, uint16_t &ov);
00062
00063     enum OutputTypes {
00064       DICT_DEFAULT = 0,
00065       DICT_DEBUG,
00066       DICT_XML
00067     };
00068
00069   protected:
00070     void WriteHeader();
00071     void WriteFooter();
00072     bool ConvertToXML(const char *raw, std::string &cxx);
00073     bool ConvertToCXX(const char *raw, std::string &cxx);
00074     void AddGroupLength();
00075
00076   private:
00077     DictConverterInternal *Internal;
00078
00079     int OutputType;
00080   };
00081
00082 } // end namespace gdcm
00083
00084 #endif //GDCMDICTCONVERTER_H

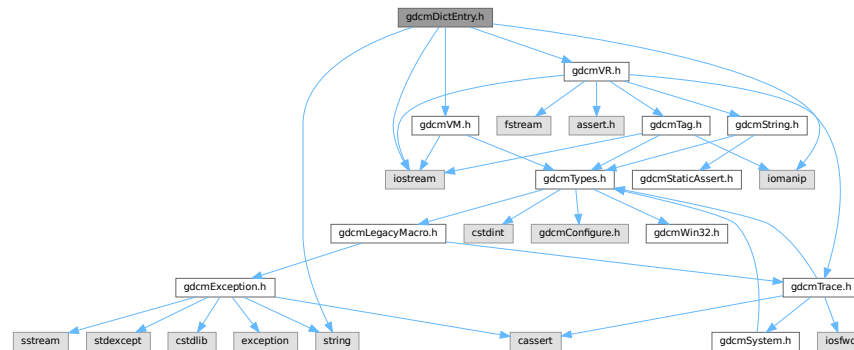
```



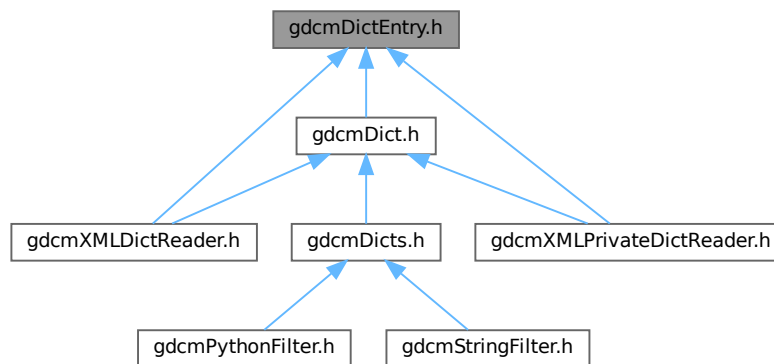
## 11.97 gdcmDictEntry.h File Reference

```
#include "gdcmVR.h"
#include "gdcmVM.h"
#include <string>
#include <iostream>
#include <iomanip>
```

Include dependency graph for gdcmDictEntry.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [gdcm::DictEntry](#)  
Class to represent an Entry in the *Dict*.

### Namespaces

- namespace [gdcm](#)

## Functions

- `std::ostream & gdcmm::operator<< (std::ostream &os, const DictEntry &val)`

## 11.98 gdcmmDictEntry.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMDICTENTRY_H
00015 #define GDCMDICTENTRY_H
00016
00017 #include "gdcmmVR.h"
00018 #include "gdcmmVM.h"
00019
00020 #include <string>
00021 #include <iostream>
00022 #include <iomanip>
00023
00024 namespace gdcmm
00025 {
00026     class GDCM_EXPORT DictEntry
00027     {
00028     public:
00029         DictEntry(const char *name = "", const char *keyword = "", VR const &vr = VR::INVALID, VM const &vm =
VM::VM0, bool ret = false):
00030             Name(name),
00031             Keyword(keyword),
00032             ValueRepresentation(vr),
00033             ValueMultiplicity(vm),
00034             Retired(ret),
00035             GroupXX(false),
00036             ElementXX(false)
00037         {
00038         }
00039     };
00040
00041     friend std::ostream& operator<<(std::ostream& _os, const DictEntry &_val);
00042
00043     const VR &GetVR() const { return ValueRepresentation; }
00044     void SetVR(const VR &vr) { ValueRepresentation = vr; }
00045     // bool IsValid() const { return ValueRepresentation != VR::VR_END; }
00046     // !Name.empty() /*&& ValueRepresentation && ValueMultiplicity*/; }
00047
00048     const VM &GetVM() const { return ValueMultiplicity; }
00049     void SetVM(VM const &vm) { ValueMultiplicity = vm; }
00050
00051     const char *GetName() const { return Name.c_str(); }
00052     void SetName(const char* name) { Name = name; }
00053
00054     const char *GetKeyword() const { return Keyword.c_str(); }
00055     void SetKeyword(const char* keyword) { Keyword = keyword; }
00056
00057     bool GetRetired() const { return Retired; }
00058     void SetRetired(bool retired) { Retired = retired; }
00059
00060     // <entry group="50xx" element="0005" vr="US" vm="1" retired="true" version="3">
00061     void SetGroupXX(bool v) { GroupXX = v; }
00062
00063     // <entry group="0020" element="31xx" vr="CS" vm="1-n" retired="true" version="2">
00064     void SetElementXX(bool v) { ElementXX = v; }
00065
00066     bool IsUnique() const { return ElementXX == false && GroupXX == false; }
00067
00068

```

```

00086 private:
00087     //
00088     friend class Dict;
00089     static bool CheckKeywordAgainstName(const char *name, const char *keyword);
00090
00091 private:
00092     std::string Name;
00093     std::string Keyword;
00094     VR ValueRepresentation;
00095     VM ValueMultiplicity;
00096     bool Retired : 1;
00097     bool GroupXX : 1;
00098     bool ElementXX : 1;
00099 };
00100
00101 #if 0
00102 class GDCM_EXPORT PrivateDictEntry : public DictEntry
00103 {
00104 public:
00105     PrivateDictEntry(const char *name = "", VR::VRType const &vr = VR::INVALID, VM::VMType const &vm =
00106         VM::VM0, bool ret = false, const char *owner = ""):DictEntry(name,vr,vm,ret),Owner(owner) {}
00107     PrivateDictEntry(const char *name, const char *vr, const char *vm):DictEntry(name,vr,vm) {}
00108     const char *GetOwner() const { return Owner.c_str(); }
00109     void SetOwner(const char *owner) { Owner = owner; }
00110
00111 private:
00112     // SIEMENS MED, GEMS_PETD_01 ...
00113     std::string Owner;
00114 };
00115 #endif
00116
00117 //-----
00118 inline std::ostream& operator<<(std::ostream& os, const DictEntry &val)
00119 {
00120     if( val.Name.empty() )
00121     {
00122         os << "[No name]";
00123     }
00124     else
00125     {
00126         os << val.Name;
00127     }
00128     if( val.Keyword.empty() )
00129     {
00130         os << "[No keyword]";
00131     }
00132     else
00133     {
00134         os << val.Keyword;
00135     }
00136     os << "\t" << val.ValueRepresentation << "\t" << val.ValueMultiplicity;
00137     if( val.Retired )
00138     {
00139         os << "\t(RET)";
00140     }
00141     return os;
00142 }
00143
00144 } // end namespace gdcm
00145
00146 #endif //GDCMDICTENTRY_H

```

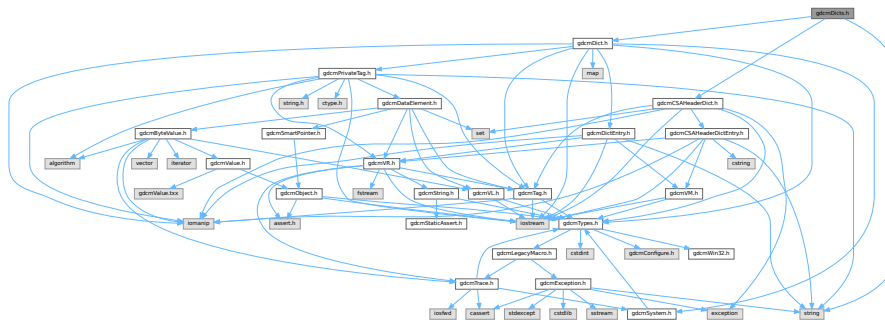
## 11.99 gdcmDicts.h File Reference

```

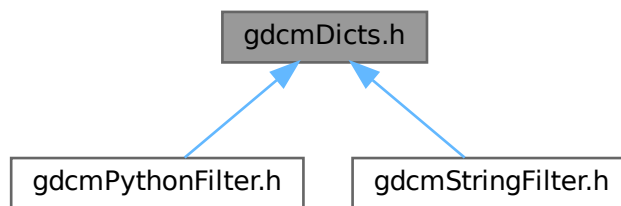
#include "gdcmDict.h"
#include "gdcmCSAHeaderDict.h"
#include <string>

```

Include dependency graph for `gdcmDicts.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::Dicts`  
*Class to manipulate the sum of knowledge (all the dict user load)*

## Namespaces

- namespace `gdcm`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Dicts &d)`

## 11.100 gdcmDicts.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMDICTS_H
00015  #define GDCMDICTS_H
00016
00017  #include "gdcmDict.h"
00018  #include "gdcmCSAHeaderDict.h"
00019
00020  #include <string>
00021
00022  namespace gdcm
00023  {
00024  class GDCM_EXPORT Dicts
00025  {
00026  friend std::ostream& operator<<(std::ostream &_os, const Dicts &d);
00027  public:
00028    Dicts();
00029    ~Dicts();
00030    Dicts &operator=(const Dicts &_val) = delete;
00031    Dicts(const Dicts &_val) = delete;
00032
00033    // DataSet::GetPrivateCreator
00034    const DictEntry &GetDictEntry(const Tag& tag, const char *owner = nullptr) const;
00035
00036    const DictEntry &GetDictEntry(const PrivateTag& tag) const;
00037
00038    //enum PublicTypes {
00039    //  DICOMV3_DICT,
00040    //  ACRNEMA_DICT,
00041    //  NIH_DICT
00042    //};
00043    const Dict &GetPublicDict() const;
00044
00045    const PrivateDict &GetPrivateDict() const;
00046    PrivateDict &GetPrivateDict();
00047
00048    const CSAHeaderDict &GetCSAHeaderDict() const;
00049
00050    bool IsEmpty() const { return GetPublicDict().IsEmpty(); }
00051
00052  protected:
00053    typedef enum {
00054      PHILIPS,
00055      GE,
00056      SIEMENS
00057    } ConstructorType;
00058    static const char *GetConstructorString(ConstructorType type);
00059
00060    friend class Global;
00061    void LoadDefaults();
00062
00063  private:
00064    // Public dict:
00065    Dict PublicDict;
00066
00067    // Private Dicts:
00068    PrivateDict ShadowDict;
00069
00070    CSAHeaderDict CSADict;
00071  };
00072  //-----
00073  inline std::ostream& operator<<(std::ostream &os, const Dicts &d)
00074  {
00075    (void)d;

```

```

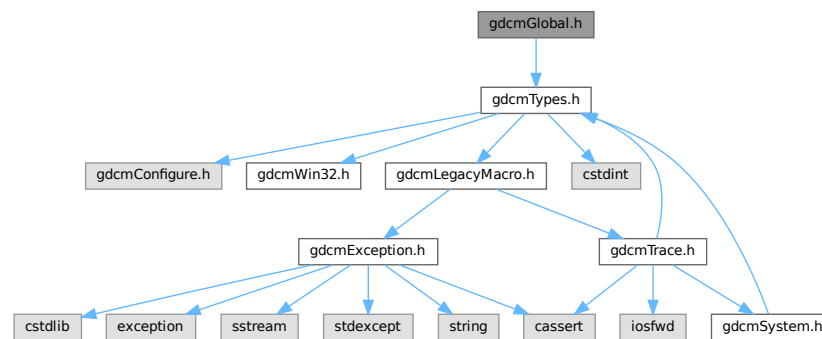
00085     return os;
00086 }
00087
00088
00089 } // end namespace gdcm
00090
00091 #endif //GDCMDICTS_H

```

## 11.101 gdcmGlobal.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmGlobal.h:



### Classes

- class [gdcm::Global](#)  
*Global.*

### Namespaces

- namespace [gdcm](#)

### Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Global &g)`

### Variables

- static [Global](#) [gdcm::GlobalInstance](#)

## 11.102 gdcmGlobal.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 // Implementation detail was shamelessly borrowed from the VTK excellent
00015 // implementation of debug leak manager singleton:
00016 /*=====
00017
00018   Program: Visualization Toolkit
00019   Module: $RCSfile: vtkDebugLeaks.cxx,v $
00020
00021   Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
00022   All rights reserved.
00023   See Copyright.txt or http://www.kitware.com/Copyright.htm for details.
00024
00025   This software is distributed WITHOUT ANY WARRANTY; without even
00026   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00027   PURPOSE. See the above copyright notice for more information.
00028
00029  =====*/
00030 #ifndef GDCMGLOBAL_H
00031 #define GDCMGLOBAL_H
00032
00033 #include "gdcmTypes.h"
00034
00035 namespace gdcm
00036 {
00037   class GlobalInternal;
00038   class Dicts;
00039   class Defs;
00040   class GDCM_EXPORT Global // why expose the symbol I think I only need to expose the instance...
00041   {
00042   friend std::ostream& operator<<(std::ostream &_os, const Global &g);
00043   public:
00044     Global();
00045     ~Global();
00046     Global &operator=(const Global &_val) = delete;
00047     Global(const Global &_val) = delete;
00048
00049     Dicts const &GetDicts() const;
00050     Dicts &GetDicts();
00051
00052     Defs const &GetDefs() const;
00053
00054     static Global &GetInstance();
00055
00056     bool LoadResourcesFiles();
00057
00058     bool Append(const char *path);
00059
00060     bool Prepend(const char *path);
00061
00062   protected:
00063     const char *Locate(const char *resfile) const;
00064
00065   private:
00066     // PIMPL:
00067     // but we could have also directly exposed a Dicts *Internals;
00068     static GlobalInternal *Internals;
00069   };
00070 //-----
00071 inline std::ostream& operator<<(std::ostream &os, const Global &g)
00072 {
00073   (void)g;
00074   return os;
00075 }
00076
00077
00078

```

```

00099 // This instance will show up in any translation unit that uses
00100 // Global or that has a singleton. It will make sure
00101 // Global is initialized before it is used and is the last
00102 // static object destroyed.
00103 static Global GlobalInstance;
00104
00105 } // end namespace gdcm
00106
00107 #endif //GDCMGLOBAL_H

```

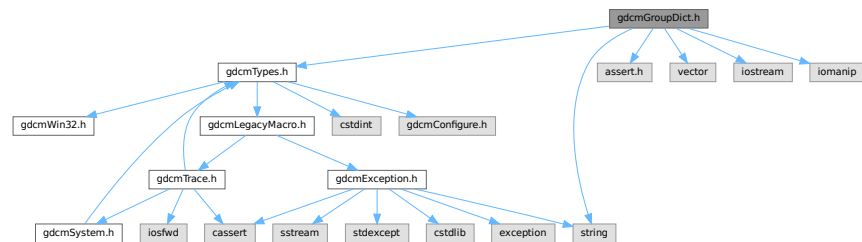
## 11.103 gdcmGroupDict.h File Reference

```

#include "gdcmTypes.h"
#include <assert.h>
#include <vector>
#include <string>
#include <iostream>
#include <iomanip>

```

Include dependency graph for gdcmGroupDict.h:



### Classes

- class [gdcm::GroupDict](#)  
Class to represent the mapping from group number to its abbreviation and name.

### Namespaces

- namespace [gdcm](#)

### Functions

- std::ostream & [gdcm::operator<<](#) (std::ostream &\_os, const [GroupDict](#) &\_val)



## 11.104 gdcmGroupDict.h

[Go to the documentation of this file.](#)

```

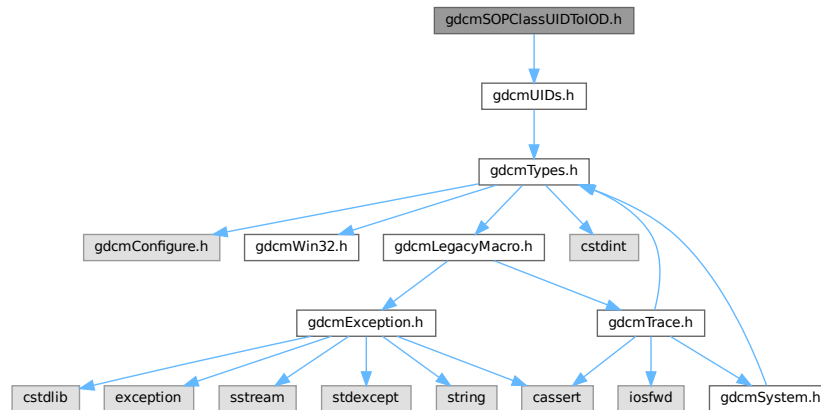
00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014
00015 #ifndef GDCMGROUPDICT_H
00016 #define GDCMGROUPDICT_H
00017
00018 #include "gdcmTypes.h"
00019
00020 #include <assert.h>
00021 #include <vector>
00022 #include <string>
00023 #include <iostream>
00024 #include <iomanip>
00025
00026 namespace gdcm
00027 {
00028     class GDCM_EXPORT GroupDict
00029     {
00030     public:
00031         typedef std::vector<std::string> GroupStringVector;
00032         GroupDict() { FillDefaultGroupName(); }
00033         ~GroupDict() = default;
00034
00035         friend std::ostream& operator<<(std::ostream& _os, const GroupDict &_val);
00036
00037         size_t Size() const
00038         {
00039             assert( Names.size() == Abbreviations.size() );
00040             return Names.size();
00041         }
00042
00043         std::string const &GetAbbreviation(uint16_t num) const;
00044
00045         std::string const &GetName(uint16_t num) const;
00046
00047     protected:
00048         void Add(std::string const &abbreviation, std::string const &name);
00049         void Insert(uint16_t num, std::string const &abbreviation, std::string const &name);
00050     private:
00051         // Generated implementation, see gdcmDefaultGroupNames
00052         void FillDefaultGroupName();
00053
00054         GroupDict &operator=(const GroupDict &_val); // purposely not implemented
00055         GroupDict(const GroupDict &_val); // purposely not implemented
00056
00057         GroupStringVector Abbreviations;
00058         GroupStringVector Names;
00059     };
00060
00061 //-----
00062 inline std::ostream& operator<<(std::ostream& _os, const GroupDict &_val)
00063 {
00064     size_t size = _val.Size();
00065     for(size_t i=0; i<size; ++i)
00066     {
00067         _os << std::hex << std::setw(4) << std::setfill( '0' ) << i << ", "
00068         << _val.GetAbbreviation((uint16_t)i) << ", " << _val.GetName((uint16_t)i) << "\n";
00069     }
00070     return _os;
00071 }
00072
00073 } // end namespace gdcm
00074
00075 #endif //GDCMGROUPDICT_H

```

## 11.105 gdcmSOPClassUIDToIOD.h File Reference

```
#include "gdcmUIDs.h"
```

Include dependency graph for gdcmSOPClassUIDToIOD.h:



### Classes

- class [gdcm::SOPClassUIDToIOD](#)  
Class convert a class SOP Class UID into *IOD*.

### Namespaces

- namespace [gdcm](#)

## 11.106 gdcmSOPClassUIDToIOD.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012  =====*/
00013
00014
00015  #ifndef GDCMSOPCLASSUIDTOIOD_H
00016  #define GDCMSOPCLASSUIDTOIOD_H
00017
00018  #include "gdcmUIDs.h"
00019

```

```

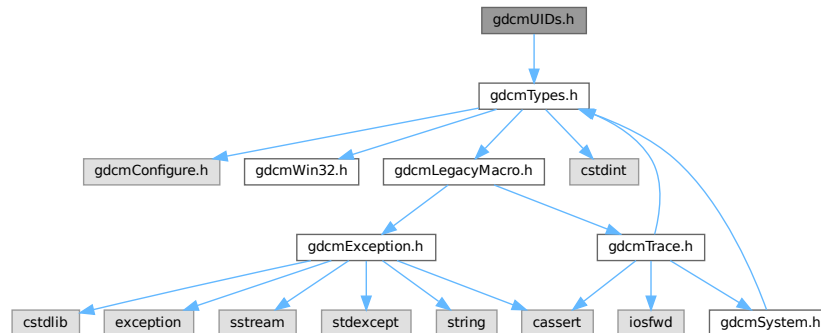
00020 namespace gdcm
00021 {
00022
00028 class GDCM_EXPORT SOPClassUIDToIOD
00029 {
00030 public:
00033     static const char *GetIOD(UIDs const & uid);
00034
00036     static unsigned int GetNumberOfSOPClassToIOD();
00037
00038     typedef const char* const (SOPClassUIDToIODType)[2];
00039     static SOPClassUIDToIODType* GetSOPClassUIDToIODs();
00040
00041     static SOPClassUIDToIODType& GetSOPClassUIDToIOD(unsigned int i);
00042
00043     static const char *GetSOPClassUIDFromIOD(const char *iod);
00044     static const char *GetIODFromSOPClassUID(const char *sopclassuid);
00045 };
00046
00047 } // end namespace gdcm
00048
00049 #endif //GDCMSOPCLASSUIDTOIOD_H

```

## 11.107 gdcmUIDs.h File Reference

#include "gdcmTypes.h"

Include dependency graph for gdcmUIDs.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [gdcm::UIDs](#)  
*all known uids*

## Namespaces

- namespace [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const UIDs &uid)`

## 11.108 gdcmUIDs.h

[Go to the documentation of this file.](#)

```
00001
00002 // GENERATED FILE DO NOT EDIT
00003 // $ xsltproc UIDToC++.xsl Part6.xml > gdcmUIDs.h
00004
00005 /*=====
00006
00007 Program: GDCM (Grassroots DICOM). A DICOM library
00008
00009 Copyright (c) 2006-2011 Mathieu Malaterre
00010 All rights reserved.
00011 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00012
00013 This software is distributed WITHOUT ANY WARRANTY; without even
00014 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00015 PURPOSE. See the above copyright notice for more information.
00016
00017 =====*/
00018
00019 #ifndef GDCMUIDS_H
00020 #define GDCMUIDS_H
00021
00022 #include "gdcmTypes.h"
00023
00024 namespace gdcm
00025 {
00026
00027 class GDCM_EXPORT UIDs
00028 {
00029 public:
00030     typedef enum {
00031         uid_1_2_840_10008_1_1 = 1, // Verification SOP Class
00032         uid_1_2_840_10008_1_2 = 2, // Implicit VR Little Endian: Default Transfer Syntax for DICOM
00033         uid_1_2_840_10008_1_2_1 = 3, // Explicit VR Little Endian
00034         uid_1_2_840_10008_1_2_1_99 = 4, // Deflated Explicit VR Little Endian
00035         uid_1_2_840_10008_1_2_2 = 5, // Explicit VR Big Endian
00036         uid_1_2_840_10008_1_2_4_50 = 6, // JPEG Baseline (Process 1): Default Transfer Syntax for Lossy JPEG 8 Bit
00037         uid_1_2_840_10008_1_2_4_51 = 7, // JPEG Extended (Process 2 & 4): Default Transfer Syntax for Lossy JPEG
00038         uid_1_2_840_10008_1_2_4_52 = 8, // JPEG Extended (Process 3 & 5)
00039         uid_1_2_840_10008_1_2_4_53 = 9, // JPEG Spectral Selection, Non-Hierarchical (Process 6 & 8)
00040         uid_1_2_840_10008_1_2_4_54 = 10, // JPEG Spectral Selection, Non-Hierarchical (Process 7 & 9)
00041         uid_1_2_840_10008_1_2_4_55 = 11, // JPEG Full Progression, Non-Hierarchical (Process 10 & 12)
00042         uid_1_2_840_10008_1_2_4_56 = 12, // JPEG Full Progression, Non-Hierarchical (Process 11 & 13)
00043         uid_1_2_840_10008_1_2_4_57 = 13, // JPEG Lossless, Non-Hierarchical (Process 14)
00044         uid_1_2_840_10008_1_2_4_58 = 14, // JPEG Lossless, Non-Hierarchical (Process 15)
00045         uid_1_2_840_10008_1_2_4_59 = 15, // JPEG Extended, Hierarchical (Process 16 & 18)
00046         uid_1_2_840_10008_1_2_4_60 = 16, // JPEG Extended, Hierarchical (Process 17 & 19)
00047         uid_1_2_840_10008_1_2_4_61 = 17, // JPEG Spectral Selection, Hierarchical (Process 20 & 22)
00048         uid_1_2_840_10008_1_2_4_62 = 18, // JPEG Spectral Selection, Hierarchical (Process 21 & 23)
00049         uid_1_2_840_10008_1_2_4_63 = 19, // JPEG Full Progression, Hierarchical (Process 24 & 26)
00050         uid_1_2_840_10008_1_2_4_64 = 20, // JPEG Full Progression, Hierarchical (Process 25 & 27)
00051         uid_1_2_840_10008_1_2_4_65 = 21, // JPEG Lossless, Hierarchical (Process 28)
00052         uid_1_2_840_10008_1_2_4_66 = 22, // JPEG Lossless, Hierarchical (Process 29)
00053         uid_1_2_840_10008_1_2_4_70 = 23, // JPEG Lossless, Non-Hierarchical, First-Order Prediction (Process 14
00054         [Selection Value 1]): Default Transfer Syntax for Lossless JPEG Image Compression
00055         uid_1_2_840_10008_1_2_4_80 = 24, // JPEG-LS Lossless Image Compression
00056         uid_1_2_840_10008_1_2_4_81 = 25, // JPEG-LS Lossy (Near-Lossless) Image Compression
00057         uid_1_2_840_10008_1_2_4_90 = 26, // JPEG 2000 Image Compression (Lossless Only)
00058     }
00059 }
```

```
00060 uid_1_2_840_10008_1_2_4_91 = 27, // JPEG 2000 Image Compression
00061 uid_1_2_840_10008_1_2_4_92 = 28, // JPEG 2000 Part 2 Multi-component Image Compression (Lossless Only)
00062 uid_1_2_840_10008_1_2_4_93 = 29, // JPEG 2000 Part 2 Multi-component Image Compression
00063 uid_1_2_840_10008_1_2_4_94 = 30, // JPIP Referenced
00064 uid_1_2_840_10008_1_2_4_95 = 31, // JPIP Referenced Deflate
00065 uid_1_2_840_10008_1_2_4_100 = 32, // MPEG2 Main Profile @ Main Level
00066 uid_1_2_840_10008_1_2_5 = 33, // RLE Lossless
00067 uid_1_2_840_10008_1_2_6_1 = 34, // RFC 2557 MIME encapsulation
00068 uid_1_2_840_10008_1_2_6_2 = 35, // XML Encoding
00069 uid_1_2_840_10008_1_3_10 = 36, // Media Storage Directory Storage
00070 uid_1_2_840_10008_1_4_1_1 = 37, // Talairach Brain Atlas Frame of Reference
00071 uid_1_2_840_10008_1_4_1_2 = 38, // SPM2 T1 Frame of Reference
00072 uid_1_2_840_10008_1_4_1_3 = 39, // SPM2 T2 Frame of Reference
00073 uid_1_2_840_10008_1_4_1_4 = 40, // SPM2 PD Frame of Reference
00074 uid_1_2_840_10008_1_4_1_5 = 41, // SPM2 EPI Frame of Reference
00075 uid_1_2_840_10008_1_4_1_6 = 42, // SPM2 FIL T1 Frame of Reference
00076 uid_1_2_840_10008_1_4_1_7 = 43, // SPM2 PET Frame of Reference
00077 uid_1_2_840_10008_1_4_1_8 = 44, // SPM2 TRANSM Frame of Reference
00078 uid_1_2_840_10008_1_4_1_9 = 45, // SPM2 SPECT Frame of Reference
00079 uid_1_2_840_10008_1_4_1_10 = 46, // SPM2 GRAY Frame of Reference
00080 uid_1_2_840_10008_1_4_1_11 = 47, // SPM2 WHITE Frame of Reference
00081 uid_1_2_840_10008_1_4_1_12 = 48, // SPM2 CSF Frame of Reference
00082 uid_1_2_840_10008_1_4_1_13 = 49, // SPM2 BRAINMASK Frame of Reference
00083 uid_1_2_840_10008_1_4_1_14 = 50, // SPM2 AVG305T1 Frame of Reference
00084 uid_1_2_840_10008_1_4_1_15 = 51, // SPM2 AVG152T1 Frame of Reference
00085 uid_1_2_840_10008_1_4_1_16 = 52, // SPM2 AVG152T2 Frame of Reference
00086 uid_1_2_840_10008_1_4_1_17 = 53, // SPM2 AVG152PD Frame of Reference
00087 uid_1_2_840_10008_1_4_1_18 = 54, // SPM2 SINGLESUBJT1 Frame of Reference
00088 uid_1_2_840_10008_1_4_2_1 = 55, // ICBM 452 T1 Frame of Reference
00089 uid_1_2_840_10008_1_4_2_2 = 56, // ICBM Single Subject MRI Frame of Reference
00090 uid_1_2_840_10008_1_9 = 57, // Basic Study Content Notification SOP Class
00091 uid_1_2_840_10008_1_20_1 = 58, // Storage Commitment Push Model SOP Class
00092 uid_1_2_840_10008_1_20_1_1 = 59, // Storage Commitment Push Model SOP Instance
00093 uid_1_2_840_10008_1_20_2 = 60, // Storage Commitment Pull Model SOP Class
00094 uid_1_2_840_10008_1_20_2_1 = 61, // Storage Commitment Pull Model SOP Instance
00095 uid_1_2_840_10008_1_40 = 62, // Procedural Event Logging SOP Class
00096 uid_1_2_840_10008_1_40_1 = 63, // Procedural Event Logging SOP Instance
00097 uid_1_2_840_10008_1_42 = 64, // Substance Administration Logging SOP Class
00098 uid_1_2_840_10008_1_42_1 = 65, // Substance Administration Logging SOP Instance
00099 uid_1_2_840_10008_2_6_1 = 66, // DICOM UID Registry
00100 uid_1_2_840_10008_2_16_4 = 67, // DICOM Controlled Terminology
00101 uid_1_2_840_10008_3_1_1_1 = 68, // DICOM Application Context Name
00102 uid_1_2_840_10008_3_1_2_1_1 = 69, // Detached Patient Management SOP Class
00103 uid_1_2_840_10008_3_1_2_1_4 = 70, // Detached Patient Management Meta SOP Class
00104 uid_1_2_840_10008_3_1_2_2_1 = 71, // Detached Visit Management SOP Class
00105 uid_1_2_840_10008_3_1_2_3_1 = 72, // Detached Study Management SOP Class
00106 uid_1_2_840_10008_3_1_2_3_2 = 73, // Study Component Management SOP Class
00107 uid_1_2_840_10008_3_1_2_3_3 = 74, // Modality Performed Procedure Step SOP Class
00108 uid_1_2_840_10008_3_1_2_3_4 = 75, // Modality Performed Procedure Step Retrieve SOP Class
00109 uid_1_2_840_10008_3_1_2_3_5 = 76, // Modality Performed Procedure Step Notification SOP Class
00110 uid_1_2_840_10008_3_1_2_5_1 = 77, // Detached Results Management SOP Class
00111 uid_1_2_840_10008_3_1_2_5_4 = 78, // Detached Results Management Meta SOP Class
00112 uid_1_2_840_10008_3_1_2_5_5 = 79, // Detached Study Management Meta SOP Class
00113 uid_1_2_840_10008_3_1_2_6_1 = 80, // Detached Interpretation Management SOP Class
00114 uid_1_2_840_10008_4_2 = 81, // Storage Service Class
00115 uid_1_2_840_10008_5_1_1_1 = 82, // Basic Film Session SOP Class
00116 uid_1_2_840_10008_5_1_1_2 = 83, // Basic Film Box SOP Class
00117 uid_1_2_840_10008_5_1_1_4 = 84, // Basic Grayscale Image Box SOP Class
00118 uid_1_2_840_10008_5_1_1_4_1 = 85, // Basic Color Image Box SOP Class
00119 uid_1_2_840_10008_5_1_1_4_2 = 86, // Referenced Image Box SOP Class
00120 uid_1_2_840_10008_5_1_1_9 = 87, // Basic Grayscale Print Management Meta SOP Class
00121 uid_1_2_840_10008_5_1_1_9_1 = 88, // Referenced Grayscale Print Management Meta SOP Class
00122 uid_1_2_840_10008_5_1_1_14 = 89, // Print Job SOP Class
00123 uid_1_2_840_10008_5_1_1_15 = 90, // Basic Annotation Box SOP Class
00124 uid_1_2_840_10008_5_1_1_16 = 91, // Printer SOP Class
00125 uid_1_2_840_10008_5_1_1_16_376 = 92, // Printer Configuration Retrieval SOP Class
00126 uid_1_2_840_10008_5_1_1_17 = 93, // Printer SOP Instance
00127 uid_1_2_840_10008_5_1_1_17_376 = 94, // Printer Configuration Retrieval SOP Instance
00128 uid_1_2_840_10008_5_1_1_18 = 95, // Basic Color Print Management Meta SOP Class
00129 uid_1_2_840_10008_5_1_1_18_1 = 96, // Referenced Color Print Management Meta SOP Class
00130 uid_1_2_840_10008_5_1_1_22 = 97, // VOI LUT Box SOP Class
00131 uid_1_2_840_10008_5_1_1_23 = 98, // Presentation LUT SOP Class
00132 uid_1_2_840_10008_5_1_1_24 = 99, // Image Overlay Box SOP Class
00133 uid_1_2_840_10008_5_1_1_24_1 = 100, // Basic Print Image Overlay Box SOP Class
00134 uid_1_2_840_10008_5_1_1_25 = 101, // Print Queue SOP Instance
00135 uid_1_2_840_10008_5_1_1_26 = 102, // Print Queue Management SOP Class
00136 uid_1_2_840_10008_5_1_1_27 = 103, // Stored Print Storage SOP Class
00137 uid_1_2_840_10008_5_1_1_29 = 104, // Hardcopy Grayscale Image Storage SOP Class
00138 uid_1_2_840_10008_5_1_1_30 = 105, // Hardcopy Color Image Storage SOP Class
00139 uid_1_2_840_10008_5_1_1_31 = 106, // Pull Print Request SOP Class
00140 uid_1_2_840_10008_5_1_1_32 = 107, // Pull Stored Print Management Meta SOP Class
```

```
00141 uid_1_2_840_10008_5_1_1_33 = 108, // Media Creation Management SOP Class UID
00142 uid_1_2_840_10008_5_1_4_1_1_1 = 109, // Computed Radiography Image Storage
00143 uid_1_2_840_10008_5_1_4_1_1_1_1 = 110, // Digital X-Ray Image Storage - For Presentation
00144 uid_1_2_840_10008_5_1_4_1_1_1_1_1 = 111, // Digital X-Ray Image Storage - For Processing
00145 uid_1_2_840_10008_5_1_4_1_1_1_2 = 112, // Digital Mammography X-Ray Image Storage - For Presentation
00146 uid_1_2_840_10008_5_1_4_1_1_1_2_1 = 113, // Digital Mammography X-Ray Image Storage - For Processing
00147 uid_1_2_840_10008_5_1_4_1_1_1_3 = 114, // Digital Intra-oral X-Ray Image Storage - For Presentation
00148 uid_1_2_840_10008_5_1_4_1_1_1_3_1 = 115, // Digital Intra-oral X-Ray Image Storage - For Processing
00149 uid_1_2_840_10008_5_1_4_1_1_2 = 116, // CT Image Storage
00150 uid_1_2_840_10008_5_1_4_1_1_2_1 = 117, // Enhanced CT Image Storage
00151 uid_1_2_840_10008_5_1_4_1_1_3 = 118, // Ultrasound Multi-frame Image Storage
00152 uid_1_2_840_10008_5_1_4_1_1_3_1 = 119, // Ultrasound Multi-frame Image Storage
00153 uid_1_2_840_10008_5_1_4_1_1_4 = 120, // MR Image Storage
00154 uid_1_2_840_10008_5_1_4_1_1_4_1 = 121, // Enhanced MR Image Storage
00155 uid_1_2_840_10008_5_1_4_1_1_4_2 = 122, // MR Spectroscopy Storage
00156 uid_1_2_840_10008_5_1_4_1_1_5 = 123, // Nuclear Medicine Image Storage
00157 uid_1_2_840_10008_5_1_4_1_1_6 = 124, // Ultrasound Image Storage
00158 uid_1_2_840_10008_5_1_4_1_1_6_1 = 125, // Ultrasound Image Storage
00159 uid_1_2_840_10008_5_1_4_1_1_7 = 126, // Secondary Capture Image Storage
00160 uid_1_2_840_10008_5_1_4_1_1_7_1 = 127, // Multi-frame Single Bit Secondary Capture Image Storage
00161 uid_1_2_840_10008_5_1_4_1_1_7_2 = 128, // Multi-frame Grayscale Byte Secondary Capture Image Storage
00162 uid_1_2_840_10008_5_1_4_1_1_7_3 = 129, // Multi-frame Grayscale Word Secondary Capture Image Storage
00163 uid_1_2_840_10008_5_1_4_1_1_7_4 = 130, // Multi-frame True Color Secondary Capture Image Storage
00164 uid_1_2_840_10008_5_1_4_1_1_8 = 131, // Standalone Overlay Storage
00165 uid_1_2_840_10008_5_1_4_1_1_9 = 132, // Standalone Curve Storage
00166 uid_1_2_840_10008_5_1_4_1_1_9_1 = 133, // Waveform Storage - Trial
00167 uid_1_2_840_10008_5_1_4_1_1_9_1_1 = 134, // 12-lead ECG Waveform Storage
00168 uid_1_2_840_10008_5_1_4_1_1_9_1_2 = 135, // General ECG Waveform Storage
00169 uid_1_2_840_10008_5_1_4_1_1_9_1_3 = 136, // Ambulatory ECG Waveform Storage
00170 uid_1_2_840_10008_5_1_4_1_1_9_2_1 = 137, // Hemodynamic Waveform Storage
00171 uid_1_2_840_10008_5_1_4_1_1_9_3_1 = 138, // Cardiac Electrophysiology Waveform Storage
00172 uid_1_2_840_10008_5_1_4_1_1_9_4_1 = 139, // Basic Voice Audio Waveform Storage
00173 uid_1_2_840_10008_5_1_4_1_1_10 = 140, // Standalone Modality LUT Storage
00174 uid_1_2_840_10008_5_1_4_1_1_11 = 141, // Standalone VOI LUT Storage
00175 uid_1_2_840_10008_5_1_4_1_1_11_1 = 142, // Grayscale Softcopy Presentation State Storage SOP Class
00176 uid_1_2_840_10008_5_1_4_1_1_11_2 = 143, // Color Softcopy Presentation State Storage SOP Class
00177 uid_1_2_840_10008_5_1_4_1_1_11_3 = 144, // Pseudo-Color Softcopy Presentation State Storage SOP Class
00178 uid_1_2_840_10008_5_1_4_1_1_11_4 = 145, // Blending Softcopy Presentation State Storage SOP Class
00179 uid_1_2_840_10008_5_1_4_1_1_12_1 = 146, // X-Ray Angiographic Image Storage
00180 uid_1_2_840_10008_5_1_4_1_1_12_1_1 = 147, // Enhanced XA Image Storage
00181 uid_1_2_840_10008_5_1_4_1_1_12_2 = 148, // X-Ray Radiofluoroscopic Image Storage
00182 uid_1_2_840_10008_5_1_4_1_1_12_2_1 = 149, // Enhanced XRF Image Storage
00183 uid_1_2_840_10008_5_1_4_1_1_13_1_1 = 150, // X-Ray 3D Angiographic Image Storage
00184 uid_1_2_840_10008_5_1_4_1_1_13_1_2 = 151, // X-Ray 3D Craniofacial Image Storage
00185 uid_1_2_840_10008_5_1_4_1_1_12_3 = 152, // X-Ray Angiographic Bi-Plane Image Storage
00186 uid_1_2_840_10008_5_1_4_1_1_20 = 153, // Nuclear Medicine Image Storage
00187 uid_1_2_840_10008_5_1_4_1_1_66 = 154, // Raw Data Storage
00188 uid_1_2_840_10008_5_1_4_1_1_66_1 = 155, // Spatial Registration Storage
00189 uid_1_2_840_10008_5_1_4_1_1_66_2 = 156, // Spatial Fiducials Storage
00190 uid_1_2_840_10008_5_1_4_1_1_66_3 = 157, // Deformable Spatial Registration Storage
00191 uid_1_2_840_10008_5_1_4_1_1_66_4 = 158, // Segmentation Storage
00192 uid_1_2_840_10008_5_1_4_1_1_67 = 159, // Real World Value Mapping Storage
00193 uid_1_2_840_10008_5_1_4_1_1_77_1 = 160, // VL Image Storage - Trial
00194 uid_1_2_840_10008_5_1_4_1_1_77_2 = 161, // VL Multi-frame Image Storage - Trial
00195 uid_1_2_840_10008_5_1_4_1_1_77_1_1 = 162, // VL Endoscopic Image Storage
00196 uid_1_2_840_10008_5_1_4_1_1_77_1_1_1 = 163, // Video Endoscopic Image Storage
00197 uid_1_2_840_10008_5_1_4_1_1_77_1_2 = 164, // VL Microscopic Image Storage
00198 uid_1_2_840_10008_5_1_4_1_1_77_1_2_1 = 165, // Video Microscopic Image Storage
00199 uid_1_2_840_10008_5_1_4_1_1_77_1_3 = 166, // VL Slide-Coordinates Microscopic Image Storage
00200 uid_1_2_840_10008_5_1_4_1_1_77_1_4 = 167, // VL Photographic Image Storage
00201 uid_1_2_840_10008_5_1_4_1_1_77_1_4_1 = 168, // Video Photographic Image Storage
00202 uid_1_2_840_10008_5_1_4_1_1_77_1_5_1 = 169, // Ophthalmic Photography 8 Bit Image Storage
00203 uid_1_2_840_10008_5_1_4_1_1_77_1_5_2 = 170, // Ophthalmic Photography 16 Bit Image Storage
00204 uid_1_2_840_10008_5_1_4_1_1_77_1_5_3 = 171, // Stereometric Relationship Storage
00205 uid_1_2_840_10008_5_1_4_1_1_77_1_5_4 = 172, // Ophthalmic Tomography Image Storage
00206 uid_1_2_840_10008_5_1_4_1_1_88_1 = 173, // Text SR Storage - Trial
00207 uid_1_2_840_10008_5_1_4_1_1_88_2 = 174, // Audio SR Storage - Trial
00208 uid_1_2_840_10008_5_1_4_1_1_88_3 = 175, // Detail SR Storage - Trial
00209 uid_1_2_840_10008_5_1_4_1_1_88_4 = 176, // Comprehensive SR Storage - Trial
00210 uid_1_2_840_10008_5_1_4_1_1_88_11 = 177, // Basic Text SR Storage
00211 uid_1_2_840_10008_5_1_4_1_1_88_22 = 178, // Enhanced SR Storage
00212 uid_1_2_840_10008_5_1_4_1_1_88_33 = 179, // Comprehensive SR Storage
00213 uid_1_2_840_10008_5_1_4_1_1_88_40 = 180, // Procedure Log Storage
00214 uid_1_2_840_10008_5_1_4_1_1_88_50 = 181, // Mammography CAD SR Storage
00215 uid_1_2_840_10008_5_1_4_1_1_88_59 = 182, // Key Object Selection Document Storage
00216 uid_1_2_840_10008_5_1_4_1_1_88_65 = 183, // Chest CAD SR Storage
00217 uid_1_2_840_10008_5_1_4_1_1_88_67 = 184, // X-Ray Radiation Dose SR Storage
00218 uid_1_2_840_10008_5_1_4_1_1_104_1 = 185, // Encapsulated PDF Storage
00219 uid_1_2_840_10008_5_1_4_1_1_104_2 = 186, // Encapsulated CDA Storage
00220 uid_1_2_840_10008_5_1_4_1_1_128 = 187, // Positron Emission Tomography Image Storage
00221 uid_1_2_840_10008_5_1_4_1_1_129 = 188, // Standalone PET Curve Storage
```

```
00222 uid_1_2_840_10008_5_1_4_1_1_481_1 = 189, // RT Image Storage
00223 uid_1_2_840_10008_5_1_4_1_1_481_2 = 190, // RT Dose Storage
00224 uid_1_2_840_10008_5_1_4_1_1_481_3 = 191, // RT Structure Set Storage
00225 uid_1_2_840_10008_5_1_4_1_1_481_4 = 192, // RT Beams Treatment Record Storage
00226 uid_1_2_840_10008_5_1_4_1_1_481_5 = 193, // RT Plan Storage
00227 uid_1_2_840_10008_5_1_4_1_1_481_6 = 194, // RT Brachy Treatment Record Storage
00228 uid_1_2_840_10008_5_1_4_1_1_481_7 = 195, // RT Treatment Summary Record Storage
00229 uid_1_2_840_10008_5_1_4_1_1_481_8 = 196, // RT Ion Plan Storage
00230 uid_1_2_840_10008_5_1_4_1_1_481_9 = 197, // RT Ion Beams Treatment Record Storage
00231 uid_1_2_840_10008_5_1_4_1_2_1_1 = 198, // Patient Root Query/Retrieve Information Model - FIND
00232 uid_1_2_840_10008_5_1_4_1_2_1_2 = 199, // Patient Root Query/Retrieve Information Model - MOVE
00233 uid_1_2_840_10008_5_1_4_1_2_1_3 = 200, // Patient Root Query/Retrieve Information Model - GET
00234 uid_1_2_840_10008_5_1_4_1_2_2_1 = 201, // Study Root Query/Retrieve Information Model - FIND
00235 uid_1_2_840_10008_5_1_4_1_2_2_2 = 202, // Study Root Query/Retrieve Information Model - MOVE
00236 uid_1_2_840_10008_5_1_4_1_2_2_3 = 203, // Study Root Query/Retrieve Information Model - GET
00237 uid_1_2_840_10008_5_1_4_1_2_3_1 = 204, // Patient/Study Only Query/Retrieve Information Model - FIND
00238 uid_1_2_840_10008_5_1_4_1_2_3_2 = 205, // Patient/Study Only Query/Retrieve Information Model - MOVE
00239 uid_1_2_840_10008_5_1_4_1_2_3_3 = 206, // Patient/Study Only Query/Retrieve Information Model - GET
00240 uid_1_2_840_10008_5_1_4_31 = 207, // Modality Worklist Information Model - FIND
00241 uid_1_2_840_10008_5_1_4_32_1 = 208, // General Purpose Worklist Information Model - FIND
00242 uid_1_2_840_10008_5_1_4_32_2 = 209, // General Purpose Scheduled Procedure Step SOP Class
00243 uid_1_2_840_10008_5_1_4_32_3 = 210, // General Purpose Performed Procedure Step SOP Class
00244 uid_1_2_840_10008_5_1_4_32 = 211, // General Purpose Worklist Management Meta SOP Class
00245 uid_1_2_840_10008_5_1_4_33 = 212, // Instance Availability Notification SOP Class
00246 uid_1_2_840_10008_5_1_4_34_1 = 213, // RT Beams Delivery Instruction Storage (Supplement 74 Frozen Draft)
00247 uid_1_2_840_10008_5_1_4_34_2 = 214, // RT Conventional Machine Verification (Supplement 74 Frozen Draft)
00248 uid_1_2_840_10008_5_1_4_34_3 = 215, // RT Ion Machine Verification (Supplement 74 Frozen Draft)
00249 uid_1_2_840_10008_5_1_4_34_4 = 216, // Unified Worklist and Procedure Step Service Class
00250 uid_1_2_840_10008_5_1_4_34_4_1 = 217, // Unified Procedure Step - Push SOP Class
00251 uid_1_2_840_10008_5_1_4_34_4_2 = 218, // Unified Procedure Step - Watch SOP Class
00252 uid_1_2_840_10008_5_1_4_34_4_3 = 219, // Unified Procedure Step - Pull SOP Class
00253 uid_1_2_840_10008_5_1_4_34_4_4 = 220, // Unified Procedure Step - Event SOP Class
00254 uid_1_2_840_10008_5_1_4_34_5 = 221, // Unified Worklist and Procedure Step SOP Instance
00255 uid_1_2_840_10008_5_1_4_37_1 = 222, // General Relevant Patient Information Query
00256 uid_1_2_840_10008_5_1_4_37_2 = 223, // Breast Imaging Relevant Patient Information Query
00257 uid_1_2_840_10008_5_1_4_37_3 = 224, // Cardiac Relevant Patient Information Query
00258 uid_1_2_840_10008_5_1_4_38_1 = 225, // Hanging Protocol Storage
00259 uid_1_2_840_10008_5_1_4_38_2 = 226, // Hanging Protocol Information Model - FIND
00260 uid_1_2_840_10008_5_1_4_38_3 = 227, // Hanging Protocol Information Model - MOVE
00261 uid_1_2_840_10008_5_1_4_41 = 228, // Product Characteristics Query SOP Class
00262 uid_1_2_840_10008_5_1_4_42 = 229, // Substance Approval Query SOP Class
00263 uid_1_2_840_10008_15_0_3_1 = 230, // dicomDeviceName
00264 uid_1_2_840_10008_15_0_3_2 = 231, // dicomDescription
00265 uid_1_2_840_10008_15_0_3_3 = 232, // dicomManufacturer
00266 uid_1_2_840_10008_15_0_3_4 = 233, // dicomManufacturerModelName
00267 uid_1_2_840_10008_15_0_3_5 = 234, // dicomSoftwareVersion
00268 uid_1_2_840_10008_15_0_3_6 = 235, // dicomVendorData
00269 uid_1_2_840_10008_15_0_3_7 = 236, // dicomAETitle
00270 uid_1_2_840_10008_15_0_3_8 = 237, // dicomNetworkConnectionReference
00271 uid_1_2_840_10008_15_0_3_9 = 238, // dicomApplicationCluster
00272 uid_1_2_840_10008_15_0_3_10 = 239, // dicomAssociationInitiator
00273 uid_1_2_840_10008_15_0_3_11 = 240, // dicomAssociationAcceptor
00274 uid_1_2_840_10008_15_0_3_12 = 241, // dicomHostname
00275 uid_1_2_840_10008_15_0_3_13 = 242, // dicomPort
00276 uid_1_2_840_10008_15_0_3_14 = 243, // dicomSOPClass
00277 uid_1_2_840_10008_15_0_3_15 = 244, // dicomTransferRole
00278 uid_1_2_840_10008_15_0_3_16 = 245, // dicomTransferSyntax
00279 uid_1_2_840_10008_15_0_3_17 = 246, // dicomPrimaryDeviceType
00280 uid_1_2_840_10008_15_0_3_18 = 247, // dicomRelatedDeviceReference
00281 uid_1_2_840_10008_15_0_3_19 = 248, // dicomPreferredCalledAETitle
00282 uid_1_2_840_10008_15_0_3_20 = 249, // dicomTLSCyphersuite
00283 uid_1_2_840_10008_15_0_3_21 = 250, // dicomAuthorizedNodeCertificateReference
00284 uid_1_2_840_10008_15_0_3_22 = 251, // dicomThisNodeCertificateReference
00285 uid_1_2_840_10008_15_0_3_23 = 252, // dicomInstalled
00286 uid_1_2_840_10008_15_0_3_24 = 253, // dicomStationName
00287 uid_1_2_840_10008_15_0_3_25 = 254, // dicomDeviceSerialNumber
00288 uid_1_2_840_10008_15_0_3_26 = 255, // dicomInstitutionName
00289 uid_1_2_840_10008_15_0_3_27 = 256, // dicomInstitutionAddress
00290 uid_1_2_840_10008_15_0_3_28 = 257, // dicomInstitutionDepartmentName
00291 uid_1_2_840_10008_15_0_3_29 = 258, // dicomIssuerOfPatientID
00292 uid_1_2_840_10008_15_0_3_30 = 259, // dicomPreferredCallingAETitle
00293 uid_1_2_840_10008_15_0_3_31 = 260, // dicomSupportedCharacterSet
00294 uid_1_2_840_10008_15_0_4_1 = 261, // dicomConfigurationRoot
00295 uid_1_2_840_10008_15_0_4_2 = 262, // dicomDevicesRoot
00296 uid_1_2_840_10008_15_0_4_3 = 263, // dicomUniqueAETitlesRegistryRoot
00297 uid_1_2_840_10008_15_0_4_4 = 264, // dicomDevice
00298 uid_1_2_840_10008_15_0_4_5 = 265, // dicomNetworkAE
00299 uid_1_2_840_10008_15_0_4_6 = 266, // dicomNetworkConnection
00300 uid_1_2_840_10008_15_0_4_7 = 267, // dicomUniqueAETitle
00301 uid_1_2_840_10008_15_0_4_8 = 268, // dicomTransferCapability
00302 //
```



```

00303 uid_1_2_840_10008_5_1_4_1_1_77_1_6 = 269, // VL Whole Slide Microscopy
00304 uid_1_2_840_10008_5_1_4_1_1_6_2 = 270, // Enhanced US Volume Storage
00305 uid_1_2_840_10008_5_1_4_1_1_66_5 = 271, // Surface Segmentation Storage
00306 uid_1_2_840_10008_5_1_4_1_1_13_1_3 = 272, // Breast Tomosynthesis Image Storage
00307 uid_1_2_840_10008_5_1_4_1_1_2_2 = 273, // Legacy Converted Enhanced CT
00308 uid_1_2_840_10008_5_1_4_1_1_4_4 = 274, // Legacy Converted Enhanced MR
00309 uid_1_2_840_10008_5_1_4_1_1_128_1 = 275, // Legacy Converted Enhanced PET
00310 uid_1_2_840_10008_1_2_4_101 = 276, // MPEG2 Main Profile High Level
00311 uid_1_2_840_10008_1_2_4_102 = 277, // MPEG-4 AVC/H.264 High Profile Lev. 4.1
00312 uid_1_2_840_10008_1_2_4_103 = 278, // MPEG-4 AVC/H.264 BD-comp High Profile Lev. 4.1
00313
00315 //
00316 // 2019b
00317 //
00318 uid_1_2_840_10008_1_5_2 = 279,
00319 uid_1_2_840_10008_1_5_3 = 280,
00320 uid_1_2_840_10008_1_5_4 = 281,
00321 uid_1_2_840_10008_1_5_5 = 282,
00322 uid_1_2_840_10008_1_5_6 = 283,
00323 uid_1_2_840_10008_1_5_7 = 284,
00324 uid_1_2_840_10008_1_5_8 = 285,
00325 uid_1_2_840_10008_1_20 = 286,
00326 uid_1_2_840_10008_2_16_5 = 287,
00327 uid_1_2_840_10008_2_16_6 = 288,
00328 uid_1_2_840_10008_2_16_7 = 289,
00329 uid_1_2_840_10008_2_16_8 = 290,
00330 uid_1_2_840_10008_2_16_9 = 291,
00331 uid_1_2_840_10008_2_16_10 = 292,
00332 uid_1_2_840_10008_2_16_11 = 293,
00333 uid_1_2_840_10008_2_16_12 = 294,
00334 uid_1_2_840_10008_2_16_13 = 295,
00335 uid_1_2_840_10008_2_16_14 = 296,
00336 uid_1_2_840_10008_5_1_1_40 = 297,
00337 uid_1_2_840_10008_5_1_1_40_1 = 298,
00338 uid_1_2_840_10008_5_1_4_1_1_9_4_2 = 299,
00339 uid_1_2_840_10008_5_1_4_1_1_9_5_1 = 300,
00340 uid_1_2_840_10008_5_1_4_1_1_9_6_1 = 301,
00341 uid_1_2_840_10008_5_1_4_1_1_11_5 = 302,
00342 uid_1_2_840_10008_5_1_4_1_1_11_6 = 303,
00343 uid_1_2_840_10008_1_2_4_104 = 304,
00344 uid_1_2_840_10008_1_2_4_105 = 305,
00345 uid_1_2_840_10008_1_2_4_106 = 306,
00346 uid_1_2_840_10008_1_2_4_107 = 307,
00347 uid_1_2_840_10008_1_2_4_108 = 308,
00348 uid_1_2_840_10008_1_5_1 = 309,
00349 uid_1_2_840_10008_5_1_4_1_1_11_7 = 310,
00350 uid_1_2_840_10008_5_1_4_1_1_11_8 = 311,
00351 uid_1_2_840_10008_5_1_4_1_1_11_9 = 312,
00352 uid_1_2_840_10008_5_1_4_1_1_11_10 = 313,
00353 uid_1_2_840_10008_5_1_4_1_1_11_11 = 314,
00354 uid_1_2_840_10008_5_1_4_1_1_12_77 = 315,
00355 uid_1_2_840_10008_5_1_4_1_1_13_1_4 = 316,
00356 uid_1_2_840_10008_5_1_4_1_1_13_1_5 = 317,
00357 uid_1_2_840_10008_5_1_4_1_1_14_1 = 318,
00358 uid_1_2_840_10008_5_1_4_1_1_14_2 = 319,
00359 uid_1_2_840_10008_5_1_4_1_1_30 = 320,
00360 uid_1_2_840_10008_5_1_4_1_1_40 = 321,
00361 uid_1_2_840_10008_5_1_4_1_1_66_6 = 322,
00362 uid_1_2_840_10008_5_1_4_1_1_68_1 = 323,
00363 uid_1_2_840_10008_5_1_4_1_1_68_2 = 324,
00364 uid_1_2_840_10008_5_1_4_1_1_77_1_5_5 = 325,
00365 uid_1_2_840_10008_5_1_4_1_1_77_1_5_6 = 326,
00366 uid_1_2_840_10008_5_1_4_1_1_77_1_5_7 = 327,
00367 uid_1_2_840_10008_5_1_4_1_1_77_1_5_8 = 328,
00368 uid_1_2_840_10008_5_1_4_1_1_78_1 = 329,
00369 uid_1_2_840_10008_5_1_4_1_1_78_2 = 330,
00370 uid_1_2_840_10008_5_1_4_1_1_78_3 = 331,
00371 uid_1_2_840_10008_5_1_4_1_1_78_4 = 332,
00372 uid_1_2_840_10008_5_1_4_1_1_78_5 = 333,
00373 uid_1_2_840_10008_5_1_4_1_1_78_6 = 334,
00374 uid_1_2_840_10008_5_1_4_1_1_78_7 = 335,
00375 uid_1_2_840_10008_5_1_4_1_1_78_8 = 336,
00376 uid_1_2_840_10008_5_1_4_1_1_79_1 = 337,
00377 uid_1_2_840_10008_5_1_4_1_1_80_1 = 338,
00378 uid_1_2_840_10008_5_1_4_1_1_81_1 = 339,
00379 uid_1_2_840_10008_5_1_4_1_1_82_1 = 340,
00380 uid_1_2_840_10008_5_1_4_1_1_88_34 = 341,
00381 uid_1_2_840_10008_5_1_4_1_1_88_35 = 342,
00382 uid_1_2_840_10008_5_1_4_1_1_88_68 = 343,
00383 uid_1_2_840_10008_5_1_4_1_1_88_69 = 344,
00384 uid_1_2_840_10008_5_1_4_1_1_88_70 = 345,

```



```
00385 uid_1_2_840_10008_5_1_4_1_1_88_71 = 346,
00386 uid_1_2_840_10008_5_1_4_1_1_88_72 = 347,
00387 uid_1_2_840_10008_5_1_4_1_1_88_73 = 348,
00388 uid_1_2_840_10008_5_1_4_1_1_88_74 = 349,
00389 uid_1_2_840_10008_5_1_4_1_1_88_75 = 350,
00390 uid_1_2_840_10008_5_1_4_1_1_90_1 = 351,
00391 uid_1_2_840_10008_5_1_4_1_1_104_3 = 352,
00392 uid_1_2_840_10008_5_1_4_1_1_130 = 353,
00393 uid_1_2_840_10008_5_1_4_1_1_131 = 354,
00394 uid_1_2_840_10008_5_1_4_1_1_200_1 = 355,
00395 uid_1_2_840_10008_5_1_4_1_1_200_2 = 356,
00396 uid_1_2_840_10008_5_1_4_1_1_200_3 = 357,
00397 uid_1_2_840_10008_5_1_4_1_1_200_4 = 358,
00398 uid_1_2_840_10008_5_1_4_1_1_200_5 = 359,
00399 uid_1_2_840_10008_5_1_4_1_1_200_6 = 360,
00400 uid_1_2_840_10008_5_1_4_1_1_481_10 = 361,
00401 uid_1_2_840_10008_5_1_4_1_1_481_11 = 362,
00402 uid_1_2_840_10008_5_1_4_1_1_501_1 = 363,
00403 uid_1_2_840_10008_5_1_4_1_1_501_2_1 = 364,
00404 uid_1_2_840_10008_5_1_4_1_1_501_2_2 = 365,
00405 uid_1_2_840_10008_5_1_4_1_1_501_3 = 366,
00406 uid_1_2_840_10008_5_1_4_1_1_501_4 = 367,
00407 uid_1_2_840_10008_5_1_4_1_1_501_5 = 368,
00408 uid_1_2_840_10008_5_1_4_1_1_501_6 = 369,
00409 uid_1_2_840_10008_5_1_4_1_1_601_1 = 370,
00410 uid_1_2_840_10008_5_1_4_1_1_601_2 = 371,
00411 uid_1_2_840_10008_5_1_4_1_2_4_2 = 372,
00412 uid_1_2_840_10008_5_1_4_1_2_4_3 = 373,
00413 uid_1_2_840_10008_5_1_4_1_2_5_3 = 374,
00414 uid_1_2_840_10008_5_1_4_20_1 = 375,
00415 uid_1_2_840_10008_5_1_4_20_2 = 376,
00416 uid_1_2_840_10008_5_1_4_20_3 = 377,
00417 uid_1_2_840_10008_5_1_4_34_5_1 = 378,
00418 uid_1_2_840_10008_5_1_4_34_6 = 379,
00419 uid_1_2_840_10008_5_1_4_34_6_1 = 380,
00420 uid_1_2_840_10008_5_1_4_34_6_2 = 381,
00421 uid_1_2_840_10008_5_1_4_34_6_3 = 382,
00422 uid_1_2_840_10008_5_1_4_34_6_4 = 383,
00423 uid_1_2_840_10008_5_1_4_34_7 = 384,
00424 uid_1_2_840_10008_5_1_4_34_8 = 385,
00425 uid_1_2_840_10008_5_1_4_34_9 = 386,
00426 uid_1_2_840_10008_5_1_4_34_10 = 387,
00427 uid_1_2_840_10008_5_1_4_38_4 = 388,
00428 uid_1_2_840_10008_5_1_4_39_1 = 389,
00429 uid_1_2_840_10008_5_1_4_39_2 = 390,
00430 uid_1_2_840_10008_5_1_4_39_3 = 391,
00431 uid_1_2_840_10008_5_1_4_39_4 = 392,
00432 uid_1_2_840_10008_5_1_4_43_1 = 393,
00433 uid_1_2_840_10008_5_1_4_43_2 = 394,
00434 uid_1_2_840_10008_5_1_4_43_3 = 395,
00435 uid_1_2_840_10008_5_1_4_43_4 = 396,
00436 uid_1_2_840_10008_5_1_4_44_1 = 397,
00437 uid_1_2_840_10008_5_1_4_44_2 = 398,
00438 uid_1_2_840_10008_5_1_4_44_3 = 399,
00439 uid_1_2_840_10008_5_1_4_44_4 = 400,
00440 uid_1_2_840_10008_5_1_4_45_1 = 401,
00441 uid_1_2_840_10008_5_1_4_45_2 = 402,
00442 uid_1_2_840_10008_5_1_4_45_3 = 403,
00443 uid_1_2_840_10008_5_1_4_45_4 = 404,
00444 uid_1_2_840_10008_7_1_1 = 405,
00445 uid_1_2_840_10008_7_1_2 = 406,
00446 uid_1_2_840_10008_8_1_1 = 407,
00447 uid_1_2_840_10008_5_1_4_1_1_4_3 = 408,
00448 uid_1_2_840_10008_15_1_1 = 409
00449 //
00450 //
00452 //
00454 //
00455 // Optionally private UIDs
00456 //
00457 #if 0
00458 uid_1_2_840_113619_4_2,
00459 uid_1_2_840_113619_4_3,
00460 uid_1_3_12_2_1107_5_9_1,
00461 uid_1_2_840_113619_4_26,
00462 uid_1_2_840_113619_4_30,
00463 uid_2_16_840_1_113709_1_5_1,
00464 uid_2_16_840_1_113709_1_2_2,
00465 uid_1_2_840_113543_6_6_1_3_10002,
00466 uid_1_2_392_200036_9116_7_8_1_1_1,
00467 uid_1_2_392_200036_9125_1_1_2,
```

```

00468 uid_1_2_840_113619_4_27,
00469 uid_1_3_46_670589_11_0_0_12_1,
00470 uid_1_3_46_670589_11_0_0_12_2,
00471 uid_1_3_46_670589_11_0_0_12_4,
00472 uid_1_3_46_670589_2_3_1_1,
00473 uid_1_3_46_670589_2_4_1_1,
00474 uid_1_3_46_670589_2_5_1_1,
00475 uid_1_3_46_670589_5_0_1,
00476 uid_1_3_46_670589_5_0_1_1,
00477 uid_1_3_46_670589_5_0_10,
00478 uid_1_3_46_670589_5_0_11,
00479 uid_1_3_46_670589_5_0_11_1,
00480 uid_1_3_46_670589_5_0_12,
00481 uid_1_3_46_670589_5_0_13,
00482 uid_1_3_46_670589_5_0_14,
00483 uid_1_3_46_670589_5_0_2,
00484 uid_1_3_46_670589_5_0_2_1,
00485 uid_1_3_46_670589_5_0_3,
00486 uid_1_3_46_670589_5_0_3_1,
00487 uid_1_3_46_670589_5_0_4,
00488 uid_1_3_46_670589_5_0_7,
00489 uid_1_3_46_670589_5_0_8,
00490 uid_1_3_46_670589_5_0_9,
00491 uid_1_2_752_24_3_7_6,
00492 uid_1_2_752_24_3_7_7,
00493 uid_1_2_840_113619_5_2,
00494 uid_1_3_46_670589_33_1_4_1
00495 #endif
00496 //
00497 //
00499
00500 } TSType;
00501 typedef enum {
00502 VerificationSOPClass = 1, // Verification SOP Class
00503 ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM = 2, // Implicit VR Little Endian: Default Transfer
    Syntax for DICOM
00504 ExplicitVRLittleEndian = 3, // Explicit VR Little Endian
00505 DeflatedExplicitVRLittleEndian = 4, // Deflated Explicit VR Little Endian
00506 ExplicitVRBigEndian = 5, // Explicit VR Big Endian
00507 JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImageCompression = 6, // JPEG Baseline (Process
    1): Default Transfer Syntax for Lossy JPEG 8 Bit Image Compression
00508 JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG12BitImageCompressionProcess4only = 7, // JPEG
    Extended (Process 2 & 4): Default Transfer Syntax for Lossy JPEG 12 Bit Image Compression (Process 4 only)
00509 JPEGExtendedProcess35Retired = 8, // JPEG Extended (Process 3 & 5)
00510 JPEGSpectralSelectionNonHierarchicalProcess68Retired = 9, // JPEG Spectral Selection, Non-Hierarchical
    (Process 6 & 8)
00511 JPEGSpectralSelectionNonHierarchicalProcess79Retired = 10, // JPEG Spectral Selection, Non-Hierarchical
    (Process 7 & 9)
00512 JPEGFullProgressionNonHierarchicalProcess1012Retired = 11, // JPEG Full Progression, Non-Hierarchical
    (Process 10 & 12)
00513 JPEGFullProgressionNonHierarchicalProcess1113Retired = 12, // JPEG Full Progression, Non-Hierarchical
    (Process 11 & 13)
00514 JPEGLosslessNonHierarchicalProcess14 = 13, // JPEG Lossless, Non-Hierarchical (Process 14)
00515 JPEGLosslessNonHierarchicalProcess15Retired = 14, // JPEG Lossless, Non-Hierarchical (Process 15)
00516 JPEGExtendedHierarchicalProcess1618Retired = 15, // JPEG Extended, Hierarchical (Process 16 & 18)
00517 JPEGExtendedHierarchicalProcess1719Retired = 16, // JPEG Extended, Hierarchical (Process 17 & 19)
00518 JPEGSpectralSelectionHierarchicalProcess2022Retired = 17, // JPEG Spectral Selection, Hierarchical
    (Process 20 & 22)
00519 JPEGSpectralSelectionHierarchicalProcess2123Retired = 18, // JPEG Spectral Selection, Hierarchical
    (Process 21 & 23)
00520 JPEGFullProgressionHierarchicalProcess2426Retired = 19, // JPEG Full Progression, Hierarchical (Process 24
    & 26)
00521 JPEGFullProgressionHierarchicalProcess2527Retired = 20, // JPEG Full Progression, Hierarchical (Process 25
    & 27)
00522 JPEGLosslessHierarchicalProcess28Retired = 21, // JPEG Lossless, Hierarchical (Process 28)
00523 JPEGLosslessHierarchicalProcess29Retired = 22, // JPEG Lossless, Hierarchical (Process 29)
00524
    JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLosslessJPEGImageCompression
    = 23, // JPEG Lossless, Non-Hierarchical, First-Order Prediction (Process 14 [Selection Value 1]): Default
    Transfer Syntax for Lossless JPEG Image Compression
00525 JPEGLSLosslessImageCompression = 24, // JPEG-LS Lossless Image Compression
00526 JPEGLSLossyNearLosslessImageCompression = 25, // JPEG-LS Lossy (Near-Lossless) Image Compression
00527 JPEG2000ImageCompressionLosslessOnly = 26, // JPEG 2000 Image Compression (Lossless Only)
00528 JPEG2000ImageCompression = 27, // JPEG 2000 Image Compression
00529 JPEG2000Part2MulticomponentImageCompressionLosslessOnly = 28, // JPEG 2000 Part 2 Multi-component Image
    Compression (Lossless Only)
00530 JPEG2000Part2MulticomponentImageCompression = 29, // JPEG 2000 Part 2 Multi-component Image Compression
00531 JPIPReferenced = 30, // JPIP Referenced
00532 JPIPReferencedDeflate = 31, // JPIP Referenced Deflate
00533 MPEG2MainProfileMainLevel = 32, // MPEG2 Main Profile @ Main Level
00534 RLELossless = 33, // RLE Lossless

```

```
00535 RFC2557MIMEencapsulation = 34, // RFC 2557 MIME encapsulation
00536 XMLEncoding = 35, // XML Encoding
00537 MediaStorageDirectoryStorage = 36, // Media Storage Directory Storage
00538 TalairachBrainAtlasFrameofReference = 37, // Talairach Brain Atlas Frame of Reference
00539 SPM2T1FrameofReference = 38, // SPM2 T1 Frame of Reference
00540 SPM2T2FrameofReference = 39, // SPM2 T2 Frame of Reference
00541 SPM2PDFFrameofReference = 40, // SPM2 PD Frame of Reference
00542 SPM2EPIFrameofReference = 41, // SPM2 EPI Frame of Reference
00543 SPM2FILTI1FrameofReference = 42, // SPM2 FIL T1 Frame of Reference
00544 SPM2PETFrameofReference = 43, // SPM2 PET Frame of Reference
00545 SPM2TRANSMFrameofReference = 44, // SPM2 TRANSM Frame of Reference
00546 SPM2SPECTFrameofReference = 45, // SPM2 SPECT Frame of Reference
00547 SPM2GRAYFrameofReference = 46, // SPM2 GRAY Frame of Reference
00548 SPM2WHITEFrameofReference = 47, // SPM2 WHITE Frame of Reference
00549 SPM2CSFFrameofReference = 48, // SPM2 CSF Frame of Reference
00550 SPM2BRAINMASKFrameofReference = 49, // SPM2 BRAINMASK Frame of Reference
00551 SPM2AVG305T1FrameofReference = 50, // SPM2 AVG305T1 Frame of Reference
00552 SPM2AVG152T1FrameofReference = 51, // SPM2 AVG152T1 Frame of Reference
00553 SPM2AVG152T2FrameofReference = 52, // SPM2 AVG152T2 Frame of Reference
00554 SPM2AVG152PDFFrameofReference = 53, // SPM2 AVG152PD Frame of Reference
00555 SPM2SINGLESUBJT1FrameofReference = 54, // SPM2 SINGLESUBJT1 Frame of Reference
00556 ICBM452T1FrameofReference = 55, // ICBM 452 T1 Frame of Reference
00557 ICBMSingleSubjectMRIFrameofReference = 56, // ICBM Single Subject MRI Frame of Reference
00558 BasicStudyContentNotificationSOPClassRetired = 57, // Basic Study Content Notification SOP Class
00559 StorageCommitmentPushModelSOPClass = 58, // Storage Commitment Push Model SOP Class
00560 StorageCommitmentPushModelSOPInstance = 59, // Storage Commitment Push Model SOP Instance
00561 StorageCommitmentPullModelSOPClassRetired = 60, // Storage Commitment Pull Model SOP Class
00562 StorageCommitmentPullModelSOPInstanceRetired = 61, // Storage Commitment Pull Model SOP Instance
00563 ProceduralEventLoggingSOPClass = 62, // Procedural Event Logging SOP Class
00564 ProceduralEventLoggingSOPInstance = 63, // Procedural Event Logging SOP Instance
00565 SubstanceAdministrationLoggingSOPClass = 64, // Substance Administration Logging SOP Class
00566 SubstanceAdministrationLoggingSOPInstance = 65, // Substance Administration Logging SOP Instance
00567 DICOMUIDRegistry = 66, // DICOM UID Registry
00568 DICOMControlledTerminology = 67, // DICOM Controlled Terminology
00569 DICOMApplicationContextName = 68, // DICOM Application Context Name
00570 DetachedPatientManagementSOPClassRetired = 69, // Detached Patient Management SOP Class
00571 DetachedPatientManagementMetaSOPClassRetired = 70, // Detached Patient Management Meta SOP Class
00572 DetachedVisitManagementSOPClassRetired = 71, // Detached Visit Management SOP Class
00573 DetachedStudyManagementSOPClassRetired = 72, // Detached Study Management SOP Class
00574 StudyComponentManagementSOPClassRetired = 73, // Study Component Management SOP Class
00575 ModalityPerformedProcedureStepSOPClass = 74, // Modality Performed Procedure Step SOP Class
00576 ModalityPerformedProcedureStepRetrieveSOPClass = 75, // Modality Performed Procedure Step Retrieve SOP
    Class
00577 ModalityPerformedProcedureStepNotificationSOPClass = 76, // Modality Performed Procedure Step Notification
    SOP Class
00578 DetachedResultsManagementSOPClassRetired = 77, // Detached Results Management SOP Class
00579 DetachedResultsManagementMetaSOPClassRetired = 78, // Detached Results Management Meta SOP Class
00580 DetachedStudyManagementMetaSOPClassRetired = 79, // Detached Study Management Meta SOP Class
00581 DetachedInterpretationManagementSOPClassRetired = 80, // Detached Interpretation Management SOP Class
00582 StorageServiceClass = 81, // Storage Service Class
00583 BasicFilmSessionSOPClass = 82, // Basic Film Session SOP Class
00584 BasicFilmBoxSOPClass = 83, // Basic Film Box SOP Class
00585 BasicGrayscaleImageBoxSOPClass = 84, // Basic Grayscale Image Box SOP Class
00586 BasicColorImageBoxSOPClass = 85, // Basic Color Image Box SOP Class
00587 ReferencedImageBoxSOPClassRetired = 86, // Referenced Image Box SOP Class
00588 BasicGrayscalePrintManagementMetaSOPClass = 87, // Basic Grayscale Print Management Meta SOP Class
00589 ReferencedGrayscalePrintManagementMetaSOPClassRetired = 88, // Referenced Grayscale Print Management Meta
    SOP Class
00590 PrintJobSOPClass = 89, // Print Job SOP Class
00591 BasicAnnotationBoxSOPClass = 90, // Basic Annotation Box SOP Class
00592 PrinterSOPClass = 91, // Printer SOP Class
00593 PrinterConfigurationRetrievalSOPClass = 92, // Printer Configuration Retrieval SOP Class
00594 PrinterSOPInstance = 93, // Printer SOP Instance
00595 PrinterConfigurationRetrievalSOPInstance = 94, // Printer Configuration Retrieval SOP Instance
00596 BasicColorPrintManagementMetaSOPClass = 95, // Basic Color Print Management Meta SOP Class
00597 ReferencedColorPrintManagementMetaSOPClassRetired = 96, // Referenced Color Print Management Meta SOP
    Class
00598 VOILUTBoxSOPClass = 97, // VOI LUT Box SOP Class
00599 PresentationLUTSOPClass = 98, // Presentation LUT SOP Class
00600 ImageOverlayBoxSOPClassRetired = 99, // Image Overlay Box SOP Class
00601 BasicPrintImageOverlayBoxSOPClassRetired = 100, // Basic Print Image Overlay Box SOP Class
00602 PrintQueueSOPInstanceRetired = 101, // Print Queue SOP Instance
00603 PrintQueueManagementSOPClassRetired = 102, // Print Queue Management SOP Class
00604 StoredPrintStorageSOPClassRetired = 103, // Stored Print Storage SOP Class
00605 HardcopyGrayscaleImageStorageSOPClassRetired = 104, // Hardcopy Grayscale Image Storage SOP Class
00606 HardcopyColorImageStorageSOPClassRetired = 105, // Hardcopy Color Image Storage SOP Class
00607 PullPrintRequestSOPClassRetired = 106, // Pull Print Request SOP Class
00608 PullStoredPrintManagementMetaSOPClassRetired = 107, // Pull Stored Print Management Meta SOP Class
00609 MediaCreationManagementSOPClassUID = 108, // Media Creation Management SOP Class UID
00610 ComputedRadiographyImageStorage = 109, // Computed Radiography Image Storage
00611 DigitalXRayImageStorageForPresentation = 110, // Digital X-Ray Image Storage - For Presentation
```

```
00612 DigitalXRayImageStorageForProcessing = 111, // Digital X-Ray Image Storage - For Processing
00613 DigitalMammographyXRayImageStorageForPresentation = 112, // Digital Mammography X-Ray Image Storage - For
    Presentation
00614 DigitalMammographyXRayImageStorageForProcessing = 113, // Digital Mammography X-Ray Image Storage - For
    Processing
00615 DigitalIntraoralXRayImageStorageForPresentation = 114, // Digital Intra-oral X-Ray Image Storage - For
    Presentation
00616 DigitalIntraoralXRayImageStorageForProcessing = 115, // Digital Intra-oral X-Ray Image Storage - For
    Processing
00617 CTImageStorage = 116, // CT Image Storage
00618 EnhancedCTImageStorage = 117, // Enhanced CT Image Storage
00619 UltrasoundMultiframeImageStorageRetired = 118, // Ultrasound Multi-frame Image Storage
00620 UltrasoundMultiframeImageStorage = 119, // Ultrasound Multi-frame Image Storage
00621 MRImageStorage = 120, // MR Image Storage
00622 EnhancedMRImageStorage = 121, // Enhanced MR Image Storage
00623 MRSpectroscopyStorage = 122, // MR Spectroscopy Storage
00624 NuclearMedicineImageStorageRetired = 123, // Nuclear Medicine Image Storage
00625 UltrasoundImageStorageRetired = 124, // Ultrasound Image Storage
00626 UltrasoundImageStorage = 125, // Ultrasound Image Storage
00627 SecondaryCaptureImageStorage = 126, // Secondary Capture Image Storage
00628 MultiframeSingleBitSecondaryCaptureImageStorage = 127, // Multi-frame Single Bit Secondary Capture Image
    Storage
00629 MultiframeGrayscaleByteSecondaryCaptureImageStorage = 128, // Multi-frame Grayscale Byte Secondary Capture
    Image Storage
00630 MultiframeGrayscaleWordSecondaryCaptureImageStorage = 129, // Multi-frame Grayscale Word Secondary Capture
    Image Storage
00631 MultiframeTrueColorSecondaryCaptureImageStorage = 130, // Multi-frame True Color Secondary Capture Image
    Storage
00632 StandaloneOverlayStorageRetired = 131, // Standalone Overlay Storage
00633 StandaloneCurveStorageRetired = 132, // Standalone Curve Storage
00634 WaveformStorageTrialRetired = 133, // Waveform Storage - Trial
00635 ECG12leadWaveformStorage = 134, // 12-lead ECG Waveform Storage
00636 GeneralECGWaveformStorage = 135, // General ECG Waveform Storage
00637 AmbulatoryECGWaveformStorage = 136, // Ambulatory ECG Waveform Storage
00638 HemodynamicWaveformStorage = 137, // Hemodynamic Waveform Storage
00639 CardiacElectrophysiologyWaveformStorage = 138, // Cardiac Electrophysiology Waveform Storage
00640 BasicVoiceAudioWaveformStorage = 139, // Basic Voice Audio Waveform Storage
00641 StandaloneModalityLUTStorageRetired = 140, // Standalone Modality LUT Storage
00642 StandaloneVOILUTStorageRetired = 141, // Standalone VOI LUT Storage
00643 GrayscaleSoftcopyPresentationStateStorageSOPClass = 142, // Grayscale Softcopy Presentation State Storage
    SOP Class
00644 ColorSoftcopyPresentationStateStorageSOPClass = 143, // Color Softcopy Presentation State Storage SOP
    Class
00645 PseudoColorSoftcopyPresentationStateStorageSOPClass = 144, // Pseudo-Color Softcopy Presentation State
    Storage SOP Class
00646 BlendingSoftcopyPresentationStateStorageSOPClass = 145, // Blending Softcopy Presentation State Storage
    SOP Class
00647 XRayAngiographicImageStorage = 146, // X-Ray Angiographic Image Storage
00648 EnhancedXAImageStorage = 147, // Enhanced XA Image Storage
00649 XRayRadiofluoroscopicImageStorage = 148, // X-Ray Radiofluoroscopic Image Storage
00650 EnhancedXRFImageStorage = 149, // Enhanced XRF Image Storage
00651 XRay3DAngiographicImageStorage = 150, // X-Ray 3D Angiographic Image Storage
00652 XRay3DCraniofacialImageStorage = 151, // X-Ray 3D Craniofacial Image Storage
00653 XRayAngiographicBiPlaneImageStorageRetired = 152, // X-Ray Angiographic Bi-Plane Image Storage
00654 NuclearMedicineImageStorage = 153, // Nuclear Medicine Image Storage
00655 RawDataStorage = 154, // Raw Data Storage
00656 SpatialRegistrationStorage = 155, // Spatial Registration Storage
00657 SpatialFiducialsStorage = 156, // Spatial Fiducials Storage
00658 DeformableSpatialRegistrationStorage = 157, // Deformable Spatial Registration Storage
00659 SegmentationStorage = 158, // Segmentation Storage
00660 RealWorldValueMappingStorage = 159, // Real World Value Mapping Storage
00661 VLImageStorageTrialRetired = 160, // VL Image Storage - Trial
00662 VLMultiframeImageStorageTrialRetired = 161, // VL Multi-frame Image Storage - Trial
00663 VLEndoscopicImageStorage = 162, // VL Endoscopic Image Storage
00664 VideoEndoscopicImageStorage = 163, // Video Endoscopic Image Storage
00665 VLMicroscopicImageStorage = 164, // VL Microscopic Image Storage
00666 VideoMicroscopicImageStorage = 165, // Video Microscopic Image Storage
00667 VLSlideCoordinatesMicroscopicImageStorage = 166, // VL Slide-Coordinates Microscopic Image Storage
00668 VLPhotographicImageStorage = 167, // VL Photographic Image Storage
00669 VideoPhotographicImageStorage = 168, // Video Photographic Image Storage
00670 OphthalmicPhotography8BitImageStorage = 169, // Ophthalmic Photography 8 Bit Image Storage
00671 OphthalmicPhotography16BitImageStorage = 170, // Ophthalmic Photography 16 Bit Image Storage
00672 StereometricRelationshipStorage = 171, // Stereometric Relationship Storage
00673 OphthalmicTomographyImageStorage = 172, // Ophthalmic Tomography Image Storage
00674 TextSRStorageTrialRetired = 173, // Text SR Storage - Trial
00675 AudioSRStorageTrialRetired = 174, // Audio SR Storage - Trial
00676 DetailSRStorageTrialRetired = 175, // Detail SR Storage - Trial
00677 ComprehensiveSRStorageTrialRetired = 176, // Comprehensive SR Storage - Trial
00678 BasicTextSRStorage = 177, // Basic Text SR Storage
00679 EnhancedSRStorage = 178, // Enhanced SR Storage
00680 ComprehensiveSRStorage = 179, // Comprehensive SR Storage
```

```
00681 ProcedureLogStorage = 180, // Procedure Log Storage
00682 MammographyCADSRStorage = 181, // Mammography CAD SR Storage
00683 KeyObjectSelectionDocumentStorage = 182, // Key Object Selection Document Storage
00684 ChestCADSRStorage = 183, // Chest CAD SR Storage
00685 XRayRadiationDoseSRStorage = 184, // X-Ray Radiation Dose SR Storage
00686 EncapsulatedPDFStorage = 185, // Encapsulated PDF Storage
00687 EncapsulatedCDAStorage = 186, // Encapsulated CDA Storage
00688 PositronEmissionTomographyImageStorage = 187, // Positron Emission Tomography Image Storage
00689 StandalonePETCurveStorageRetired = 188, // Standalone PET Curve Storage
00690 RTImageStorage = 189, // RT Image Storage
00691 RTDoseStorage = 190, // RT Dose Storage
00692 RTStructureSetStorage = 191, // RT Structure Set Storage
00693 RTBeamsTreatmentRecordStorage = 192, // RT Beams Treatment Record Storage
00694 RTPlanStorage = 193, // RT Plan Storage
00695 RTBrachyTreatmentRecordStorage = 194, // RT Brachy Treatment Record Storage
00696 RTTreatmentSummaryRecordStorage = 195, // RT Treatment Summary Record Storage
00697 RTIonPlanStorage = 196, // RT Ion Plan Storage
00698 RTIonBeamsTreatmentRecordStorage = 197, // RT Ion Beams Treatment Record Storage
00699 PatientRootQueryRetrieveInformationModelFIND = 198, // Patient Root Query/Retrieve Information Model -
    FIND
00700 PatientRootQueryRetrieveInformationModelMOVE = 199, // Patient Root Query/Retrieve Information Model -
    MOVE
00701 PatientRootQueryRetrieveInformationModelGET = 200, // Patient Root Query/Retrieve Information Model - GET
00702 StudyRootQueryRetrieveInformationModelFIND = 201, // Study Root Query/Retrieve Information Model - FIND
00703 StudyRootQueryRetrieveInformationModelMOVE = 202, // Study Root Query/Retrieve Information Model - MOVE
00704 StudyRootQueryRetrieveInformationModelGET = 203, // Study Root Query/Retrieve Information Model - GET
00705 PatientStudyOnlyQueryRetrieveInformationModelFINDRetired = 204, // Patient/Study Only Query/Retrieve
    Information Model - FIND
00706 PatientStudyOnlyQueryRetrieveInformationModelMOVERetired = 205, // Patient/Study Only Query/Retrieve
    Information Model - MOVE
00707 PatientStudyOnlyQueryRetrieveInformationModelGETRetired = 206, // Patient/Study Only Query/Retrieve
    Information Model - GET
00708 ModalityWorklistInformationModelFIND = 207, // Modality Worklist Information Model - FIND
00709 GeneralPurposeWorklistInformationModelFIND = 208, // General Purpose Worklist Information Model - FIND
00710 GeneralPurposeScheduledProcedureStepSOPClass = 209, // General Purpose Scheduled Procedure Step SOP Class
00711 GeneralPurposePerformedProcedureStepSOPClass = 210, // General Purpose Performed Procedure Step SOP Class
00712 GeneralPurposeWorklistManagementMetaSOPClass = 211, // General Purpose Worklist Management Meta SOP Class
00713 InstanceAvailabilityNotificationSOPClass = 212, // Instance Availability Notification SOP Class
00714 RTBeamsDeliveryInstructionStorageSupplement74FrozenDraft = 213, // RT Beams Delivery Instruction Storage
    (Supplement 74 Frozen Draft)
00715 RTConventionalMachineVerificationSupplement74FrozenDraft = 214, // RT Conventional Machine Verification
    (Supplement 74 Frozen Draft)
00716 RTIonMachineVerificationSupplement74FrozenDraft = 215, // RT Ion Machine Verification (Supplement 74
    Frozen Draft)
00717 UnifiedWorklistandProcedureStepServiceClass = 216, // Unified Worklist and Procedure Step Service Class
00718 UnifiedProcedureStepPushSOPClass = 217, // Unified Procedure Step - Push SOP Class
00719 UnifiedProcedureStepWatchSOPClass = 218, // Unified Procedure Step - Watch SOP Class
00720 UnifiedProcedureStepPullSOPClass = 219, // Unified Procedure Step - Pull SOP Class
00721 UnifiedProcedureStepEventSOPClass = 220, // Unified Procedure Step - Event SOP Class
00722 UnifiedWorklistandProcedureStepSOPInstance = 221, // Unified Worklist and Procedure Step SOP Instance
00723 GeneralRelevantPatientInformationQuery = 222, // General Relevant Patient Information Query
00724 BreastImagingRelevantPatientInformationQuery = 223, // Breast Imaging Relevant Patient Information Query
00725 CardiacRelevantPatientInformationQuery = 224, // Cardiac Relevant Patient Information Query
00726 HangingProtocolStorage = 225, // Hanging Protocol Storage
00727 HangingProtocolInformationModelFIND = 226, // Hanging Protocol Information Model - FIND
00728 HangingProtocolInformationModelMOVE = 227, // Hanging Protocol Information Model - MOVE
00729 ProductCharacteristicsQuerySOPClass = 228, // Product Characteristics Query SOP Class
00730 SubstanceApprovalQuerySOPClass = 229, // Substance Approval Query SOP Class
00731 dicomDeviceName = 230, // dicomDeviceName
00732 dicomDescription = 231, // dicomDescription
00733 dicomManufacturer = 232, // dicomManufacturer
00734 dicomManufacturerModelName = 233, // dicomManufacturerModelName
00735 dicomSoftwareVersion = 234, // dicomSoftwareVersion
00736 dicomVendorData = 235, // dicomVendorData
00737 dicomAETitle = 236, // dicomAETitle
00738 dicomNetworkConnectionReference = 237, // dicomNetworkConnectionReference
00739 dicomApplicationCluster = 238, // dicomApplicationCluster
00740 dicomAssociationInitiator = 239, // dicomAssociationInitiator
00741 dicomAssociationAcceptor = 240, // dicomAssociationAcceptor
00742 dicomHostname = 241, // dicomHostname
00743 dicomPort = 242, // dicomPort
00744 dicomSOPClass = 243, // dicomSOPClass
00745 dicomTransferRole = 244, // dicomTransferRole
00746 dicomTransferSyntax = 245, // dicomTransferSyntax
00747 dicomPrimaryDeviceType = 246, // dicomPrimaryDeviceType
00748 dicomRelatedDeviceReference = 247, // dicomRelatedDeviceReference
00749 dicomPreferredCalledAETitle = 248, // dicomPreferredCalledAETitle
00750 dicomTLSCyphersuite = 249, // dicomTLSCyphersuite
00751 dicomAuthorizedNodeCertificateReference = 250, // dicomAuthorizedNodeCertificateReference
00752 dicomThisNodeCertificateReference = 251, // dicomThisNodeCertificateReference
00753 dicomInstalled = 252, // dicomInstalled
```



```

00754 dicomStationName = 253, // dicomStationName
00755 dicomDeviceSerialNumber = 254, // dicomDeviceSerialNumber
00756 dicomInstitutionName = 255, // dicomInstitutionName
00757 dicomInstitutionAddress = 256, // dicomInstitutionAddress
00758 dicomInstitutionDepartmentName = 257, // dicomInstitutionDepartmentName
00759 dicomIssuerOfPatientID = 258, // dicomIssuerOfPatientID
00760 dicomPreferredCallingAETitle = 259, // dicomPreferredCallingAETitle
00761 dicomSupportedCharacterSet = 260, // dicomSupportedCharacterSet
00762 dicomConfigurationRoot = 261, // dicomConfigurationRoot
00763 dicomDevicesRoot = 262, // dicomDevicesRoot
00764 dicomUniqueAETitlesRegistryRoot = 263, // dicomUniqueAETitlesRegistryRoot
00765 dicomDevice = 264, // dicomDevice
00766 dicomNetworkAE = 265, // dicomNetworkAE
00767 dicomNetworkConnection = 266, // dicomNetworkConnection
00768 dicomUniqueAETitle = 267, // dicomUniqueAETitle
00769 dicomTransferCapability = 268, // dicomTransferCapability
00770 //
00771 VLWholeSlideMicroscopyImageStorage = 269,
00772 EnhancedUSVolumeStorage = 270,
00773 SurfaceSegmentationStorage = 271,
00774 BreastTomosynthesisImageStorage = 272,
00775 LegacyConvertedEnhancedCTImageStorage = 273,
00776 LegacyConvertedEnhancedMRImageStorage = 274,
00777 LegacyConvertedEnhancedPETImageStorage = 275,
00778 MPEG2MainProfileHighLevel = 276,
00779 MPEG4AVCH_264HighProfileLevel4_1 = 277,
00780 MPEG4AVCH_264BDcompatibleHighProfileLevel4_1 = 278,
00781
00783 //
00784 // 2019b
00785 //
00786 PETColorPaletteSOPInstance = 279,
00787 HotMetalBlueColorPaletteSOPInstance = 280,
00788 PET20StepColorPaletteSOPInstance = 281,
00789 SpringColorPaletteSOPInstance = 282,
00790 SummerColorPaletteSOPInstance = 283,
00791 FallColorPaletteSOPInstance = 284,
00792 WinterColorPaletteSOPInstance = 285,
00793 Papyrus3ImplicitVRLittleEndian = 286,
00794 AdultMouseAnatomyOntology = 287,
00795 UberonOntology = 288,
00796 IntegratedTaxonomicInformationSystemITISTaxonomicSerialNumberTSN = 289,
00797 MouseGenomeInitiativeMGI = 290,
00798 PubChemCompoundCID = 291,
00799 ICD11 = 292,
00800 NewYorkUniversityMelanomaClinicalCooperativeGroup = 293,
00801 MayoClinicNonradiologicalImagesSBSAnatomicalSurfaceRegionGuide = 294,
00802 ImageBiomarkerStandardisationInitiative = 295,
00803 RadiomicsOntology = 296,
00804 DisplaySystemSOPClass = 297,
00805 DisplaySystemSOPInstance = 298,
00806 GeneralAudioWaveformStorage = 299,
00807 ArterialPulseWaveformStorage = 300,
00808 RespiratoryWaveformStorage = 301,
00809 XAXRFGrayscaleSoftcopyPresentationStateStorage = 302,
00810 GrayscalePlanarMPRVolumetricPresentationStateStorage = 303,
00811 MPEG4AVCH_264HighProfileLevel4_2For2DVideo = 304,
00812 MPEG4AVCH_264HighProfileLevel4_2For3DVideo = 305,
00813 MPEG4AVCH_264StereoHighProfileLevel4_2 = 306,
00814 HEVCH_265MainProfileLevel5_1 = 307,
00815 HEVCH_265Main10ProfileLevel5_1 = 308,
00816 HotIronColorPaletteSOPInstance = 309,
00817 CompositingPlanarMPRVolumetricPresentationStateStorage = 310,
00818 AdvancedBlendingPresentationStateStorage = 311,
00819 VolumeRenderingVolumetricPresentationStateStorage = 312,
00820 SegmentedVolumeRenderingVolumetricPresentationStateStorage = 313,
00821 MultipleVolumeRenderingVolumetricPresentationStateStorage = 314,
00822 Null0 = 315,
00823 BreastProjectionXRayImageStorageForPresentation = 316,
00824 BreastProjectionXRayImageStorageForProcessing = 317,
00825 IntravascularOpticalCoherenceTomographyImageStorageForPresentation = 318,
00826 IntravascularOpticalCoherenceTomographyImageStorageForProcessing = 319,
00827 ParametricMapStorage = 320,
00828 Null1 = 321,
00829 TractographyResultsStorage = 322,
00830 SurfaceScanMeshStorage = 323,
00831 SurfaceScanPointCloudStorage = 324,
00832 WideFieldOphthalmicPhotographyStereographicProjectionImageStorage = 325,
00833 WideFieldOphthalmicPhotography3DCoordinatesImageStorage = 326,
00834 OphthalmicOpticalCoherenceTomographyEnFaceImageStorage = 327,
00835 OphthalmicOpticalCoherenceTomographyBscanVolumeAnalysisStorage = 328,

```

00836	LensometryMeasurementsStorage	= 329,
00837	AutorefractometryMeasurementsStorage	= 330,
00838	KeratometryMeasurementsStorage	= 331,
00839	SubjectiveRefractionMeasurementsStorage	= 332,
00840	VisualAcuityMeasurementsStorage	= 333,
00841	SpectaclePrescriptionReportStorage	= 334,
00842	OphthalmicAxialMeasurementsStorage	= 335,
00843	IntraocularLensCalculationsStorage	= 336,
00844	MacularGridThicknessandVolumeReportStorage	= 337,
00845	OphthalmicVisualFieldStaticPerimetryMeasurementsStorage	= 338,
00846	OphthalmicThicknessMapStorage	= 339,
00847	CornealTopographyMapStorage	= 340,
00848	Comprehensive3DSRStorage	= 341,
00849	ExtensibleSRStorage	= 342,
00850	RadiopharmaceuticalRadiationDoseSRStorage	= 343,
00851	ColonCADSRStorage	= 344,
00852	ImplantationPlanSRStorage	= 345,
00853	AcquisitionContextSRStorage	= 346,
00854	SimplifiedAdultEchoSRStorage	= 347,
00855	PatientRadiationDoseSRStorage	= 348,
00856	PlannedImagingAgentAdministrationSRStorage	= 349,
00857	PerformedImagingAgentAdministrationSRStorage	= 350,
00858	ContentAssessmentResultsStorage	= 351,
00859	EncapsulatedSTLStorage	= 352,
00860	EnhancedPETImageStorage	= 353,
00861	BasicStructuredDisplayStorage	= 354,
00862	CTDefinedProcedureProtocolStorage	= 355,
00863	CTPerformedProcedureProtocolStorage	= 356,
00864	ProtocolApprovalStorage	= 357,
00865	ProtocolApprovalInformationModelFIND	= 358,
00866	ProtocolApprovalInformationModelMOVE	= 359,
00867	ProtocolApprovalInformationModelGET	= 360,
00868	RTPhysicianIntentStorage	= 361,
00869	RTSegmentAnnotationStorage	= 362,
00870	DICOSCTImageStorage	= 363,
00871	DICOSDigitalXRayImageStorageForPresentation	= 364,
00872	DICOSDigitalXRayImageStorageForProcessing	= 365,
00873	DICOSThreatDetectionReportStorage	= 366,
00874	DICOS2DAITStorage	= 367,
00875	DICOS3DAITStorage	= 368,
00876	DICOSQuadrupoleResonanceQRStorage	= 369,
00877	EddyCurrentImageStorage	= 370,
00878	EddyCurrentMultiframeImageStorage	= 371,
00879	CompositeInstanceRootRetrieveMOVE	= 372,
00880	CompositeInstanceRootRetrieveGET	= 373,
00881	CompositeInstanceRetrieveWithoutBulkDataGET	= 374,
00882	DefinedProcedureProtocolInformationModelFIND	= 375,
00883	DefinedProcedureProtocolInformationModelMOVE	= 376,
00884	DefinedProcedureProtocolInformationModelGET	= 377,
00885	UPSFilteredGlobalSubscriptionSOPInstance	= 378,
00886	UnifiedWorklistandProcedureStepServiceClass1	= 379,
00887	UnifiedProcedureStepPushSOPClass1	= 380,
00888	UnifiedProcedureStepWatchSOPClass1	= 381,
00889	UnifiedProcedureStepPullSOPClass1	= 382,
00890	UnifiedProcedureStepEventSOPClass1	= 383,
00891	RTBeamsDeliveryInstructionStorage	= 384,
00892	RTConventionalMachineVerification	= 385,
00893	RTIonMachineVerification	= 386,
00894	RTBrachyApplicationSetupDeliveryInstructionStorage	= 387,
00895	HangingProtocolInformationModelGET	= 388,
00896	ColorPaletteStorage	= 389,
00897	ColorPaletteQueryRetrieveInformationModelFIND	= 390,
00898	ColorPaletteQueryRetrieveInformationModelMOVE	= 391,
00899	ColorPaletteQueryRetrieveInformationModelGET	= 392,
00900	GenericImplantTemplateStorage	= 393,
00901	GenericImplantTemplateInformationModelFIND	= 394,
00902	GenericImplantTemplateInformationModelMOVE	= 395,
00903	GenericImplantTemplateInformationModelGET	= 396,
00904	ImplantAssemblyTemplateStorage	= 397,
00905	ImplantAssemblyTemplateInformationModelFIND	= 398,
00906	ImplantAssemblyTemplateInformationModelMOVE	= 399,
00907	ImplantAssemblyTemplateInformationModelGET	= 400,
00908	ImplantTemplateGroupStorage	= 401,
00909	ImplantTemplateGroupInformationModelFIND	= 402,
00910	ImplantTemplateGroupInformationModelMOVE	= 403,
00911	ImplantTemplateGroupInformationModelGET	= 404,
00912	NativeDICOMModel	= 405,
00913	AbstractMultiDimensionalImageModel	= 406,
00914	DICOMContentMappingResource	= 407,
00915	EnhancedMRColorImageStorage	= 408,
00916	UniversalCoordinatedTime	= 409

```

00917 //
00918 //
00920
00922 //
00923 // Optionally private UIDs
00924 //
00925 #if 0
00926 Private_1_2_840_113619_4_2,
00927 Private_1_2_840_113619_4_3,
00928 Private_1_3_12_2_1107_5_9_1,
00929 Private_1_2_840_113619_4_26,
00930 Private_1_2_840_113619_4_30,
00931 Private_2_16_840_1_113709_1_5_1,
00932 Private_2_16_840_1_113709_1_2_2,
00933 Private_1_2_840_113543_6_6_1_3_10002,
00934 Private_1_2_392_200036_9116_7_8_1_1_1,
00935 Private_1_2_392_200036_9125_1_1_2,
00936 Private_1_2_840_113619_4_27,
00937 Private_1_3_46_670589_11_0_0_12_1,
00938 Private_1_3_46_670589_11_0_0_12_2,
00939 Private_1_3_46_670589_11_0_0_12_4,
00940 Private_1_3_46_670589_2_3_1_1,
00941 Private_1_3_46_670589_2_4_1_1,
00942 Private_1_3_46_670589_2_5_1_1,
00943 Private_1_3_46_670589_5_0_1,
00944 Private_1_3_46_670589_5_0_1_1,
00945 Private_1_3_46_670589_5_0_10,
00946 Private_1_3_46_670589_5_0_11,
00947 Private_1_3_46_670589_5_0_11_1,
00948 Private_1_3_46_670589_5_0_12,
00949 Private_1_3_46_670589_5_0_13,
00950 Private_1_3_46_670589_5_0_14,
00951 Private_1_3_46_670589_5_0_2,
00952 Private_1_3_46_670589_5_0_2_1,
00953 Private_1_3_46_670589_5_0_3,
00954 Private_1_3_46_670589_5_0_3_1,
00955 Private_1_3_46_670589_5_0_4,
00956 Private_1_3_46_670589_5_0_7,
00957 Private_1_3_46_670589_5_0_8,
00958 Private_1_3_46_670589_5_0_9,
00959 Private_1_2_752_24_3_7_6,
00960 Private_1_2_752_24_3_7_7,
00961 Private_1_2_840_113619_5_2,
00962 Private_1_3_46_670589_33_1_4_1
00963 #endif
00964 //
00965 //
00967
00968 } TSName;
00969
00970
00971 typedef const char* const (*TransferSyntaxStringsType)[2];
00972 static TransferSyntaxStringsType GetTransferSyntaxStrings();
00973 static const char * const *GetTransferSyntaxString(unsigned int ts);
00974 static unsigned int GetNumberOfTransferSyntaxStrings();
00975
00976
00977 // TODO: Because I would like a dual signature for TSType and TSName, C++ won't let me do it...
00978 static const char* GetUIDString(/*TSType*/ unsigned int ts);
00979 static const char* GetUIDName(/*TSType*/ unsigned int ts);
00980
00981 bool SetFromUID(const char *str);
00982
00983 const char *GetName() const;
00984
00985 const char *GetString() const;
00986
00987 operator TSType () const { return TSField; }
00988
00989 private:
00990 TSField TSField;
00991 };
00992 //-----
00993 inline std::ostream &operator<<(std::ostream &_os, const UIDs &uid)
00994 {
00995     _os << uid.GetString() << " -> " << uid.GetName();
00996     return _os;
00997 }
00998 }
00999 } // end namespace gdcmm

```



```

01007
01008 #endif //GDCMUIDS_H

```

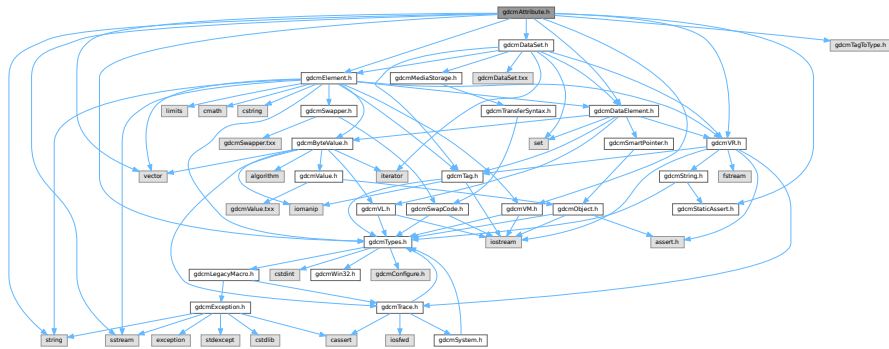
## 11.109 gdcmAttribute.h File Reference

```

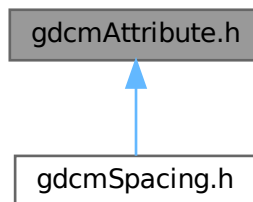
#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmTagToType.h"
#include "gdcmVM.h"
#include "gdcmElement.h"
#include "gdcmDataElement.h"
#include "gdcmDataSet.h"
#include "gdcmStaticAssert.h"
#include <string>
#include <vector>
#include <sstream>

```

Include dependency graph for gdcmAttribute.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::Attribute< Group, Element, TVR, TVM >`  
*Attribute class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary.*
- class `gdcm::Attribute< Group, Element, TVR, VM::VM1 >`
- class `gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >`
- class `gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >`
- class `gdcm::Attribute< Group, Element, TVR, VM::VM1_n >`
- class `gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >`
- class `gdcm::Attribute< Group, Element, TVR, VM::VM2_n >`
- class `gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >`
- class `gdcm::Attribute< Group, Element, TVR, VM::VM3_n >`
- class `gdcm::VRVLSize< 0 >`
- class `gdcm::VRVLSize< 1 >`

## Namespaces

- namespace `gdcm`

## 11.110 gdcmAttribute.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMATATTRIBUTE_H
00015 #define GDCMATATTRIBUTE_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmVR.h"
00019 #include "gdcmTagToType.h"
00020 #include "gdcmVM.h"
00021 #include "gdcmElement.h"
00022 #include "gdcmDataElement.h"
00023 #include "gdcmDataSet.h"
00024 #include "gdcmStaticAssert.h"
00025
00026 #include <string>
00027 #include <vector>
00028 #include <sstream>
00029
00030 namespace gdcm_ns
00031 {
00032
00033   struct void_;
00034
00035   // Declaration, also serve as forward declaration
00036   template<int T> class VRVLSize;
00037
00038   // Implementation when VL is coded on 16 bits:
00039   template<> class VRVLSize<0> {
00040   public:

```

```

00041 static inline uint16_t Read(std::istream &_is) {
00042     uint16_t l;
00043     _is.read((char*)&l, 2);
00044     return l;
00045 }
00046
00047 static inline void Write(std::ostream &os) { (void)os;
00048 }
00049 };
00050 // Implementation when VL is coded on 32 bits:
00051 template<> class VRVLSize<1> {
00052 public:
00053     static inline uint32_t Read(std::istream &_is) {
00054         char dummy[2];
00055         _is.read(dummy, 2);
00056
00057         uint32_t l;
00058         _is.read((char*)&l, 4);
00059         return l;
00060     }
00061
00062     static inline void Write(std::ostream &os) { (void)os;
00063     }
00064 };
00065
00066 template<uint16_t Group, uint16_t Element,
00067         long long TVR = TagToType<Group, Element>::VRType, // can the user override this value ?
00068         int TVM = TagToType<Group, Element>::VMType // can the user override this value ?
00069         /*typename SQAttribute = void_*/ > // if only I had variadic template...
00070 class Attribute
00071 {
00072 public:
00073     typedef typename VRToType<TVR>::Type ArrayType;
00074     enum { VMType = VMToLength<TVM>::Length };
00075     ArrayType Internal[VMToLength<TVM>::Length];
00076
00077     // Make sure that user specified VR/VM are compatible with the public dictionary:
00078     GDCM_STATIC_ASSERT( ((VR::VRType)TVR & (VR::VRType)(TagToType<Group, Element>::VRType)) );
00079     GDCM_STATIC_ASSERT( ((VM::VMType)TVM & (VM::VMType)(TagToType<Group, Element>::VMType)) );
00080     GDCM_STATIC_ASSERT( (((VR::VRType)TVR & VR::VR_VM1) && ((VM::VMType)TVM == VM::VM1) )
00081         || !((VR::VRType)TVR & VR::VR_VM1) );
00082
00083     static Tag GetTag() { return Tag(Group,Element); }
00084     static VR GetVR() { return (VR::VRType)TVR; }
00085     static VM GetVM() { return (VM::VMType)TVM; }
00086
00087     // The following two methods do make sense only in case of public element,
00088     // when the template is instantiated with private element the VR/VM are simply
00089     // defaulted to allow everything (see gdcmTagToType.h default template for TagToType)
00090     static VR GetDictVR() { return (VR::VRType)(TagToType<Group, Element>::VRType); }
00091     static VM GetDictVM() { return (VM::VMType)(TagToType<Group, Element>::VMType); }
00092
00093     // Some extra dummy checks:
00094     // Data Elements with a VR of SQ, OF, OW, OB or UN shall always have a Value Multiplicity of one.
00095
00096     unsigned int GetNumberOfValues() const {
00097         return VMToLength<TVM>::Length;
00098     }
00099
00100     // Implementation of Print is common to all Mode (ASCII/Binary)
00101     // TODO: Can we print a \ when in ASCII...well I don't think so
00102     // it would mean we used a bad VM then, right ?
00103     void Print(std::ostream &os) const {
00104         os << GetTag() << " ";
00105         os << TagToType<Group,Element>::GetVRString() << " ";
00106         os << TagToType<Group,Element>::GetVMString() << " ";
00107         os << Internal[0]; // VM is at least guarantee to be one
00108         for(unsigned int i=1; i<GetNumberOfValues(); ++i)
00109             os << "," << Internal[i];
00110     }
00111
00112     // copy:
00113     //ArrayType GetValue(unsigned int idx = 0) {
00114     //    assert( idx < GetNumberOfValues() );
00115     //    return Internal[idx];
00116     //}
00117     //ArrayType operator[] (unsigned int idx) {
00118     //    return GetValue(idx);
00119     //}
00120     // FIXME: is this always a good idea ?
00121     // I do not think so, I prefer operator
00122     //operator ArrayType () const { return Internal[0]; }

```

```

00137
00138 bool operator==(const Attribute &att) const
00139 {
00140     return std::equal(Internal, Internal+GetNumberOfValues(),
00141         att.GetValues());
00142 }
00143 bool operator!=(const Attribute &att) const
00144 {
00145     return !std::equal(Internal, Internal+GetNumberOfValues(),
00146         att.GetValues());
00147 }
00148 bool operator<(const Attribute &att) const
00149 {
00150     return std::lexicographical_compare(Internal, Internal+GetNumberOfValues(),
00151         att.GetValues(), att.GetValues() + att.GetNumberOfValues() );
00152 }
00153
00154 ArrayType &GetValue(unsigned int idx = 0) {
00155     assert( idx < GetNumberOfValues() );
00156     return Internal[idx];
00157 }
00158 ArrayType & operator[] (unsigned int idx) {
00159     return GetValue(idx);
00160 }
00161 // const reference
00162 ArrayType const &GetValue(unsigned int idx = 0) const {
00163     assert( idx < GetNumberOfValues() );
00164     return Internal[idx];
00165 }
00166 ArrayType const & operator[] (unsigned int idx) const {
00167     return GetValue(idx);
00168 }
00169 void SetValue(ArrayType v, unsigned int idx = 0) {
00170     assert( idx < GetNumberOfValues() );
00171     Internal[idx] = v;
00172 }
00173 void SetValues(const ArrayType* array, unsigned int numel = VMType ) {
00174     assert( array && numel && numel == GetNumberOfValues() );
00175     // std::copy is smarter than a memcpy, and will call memcpy when POD type
00176     std::copy(array, array+numel, Internal);
00177 }
00178 const ArrayType* GetValues() const {
00179     return Internal;
00180 }
00181
00182 // API to talk to the run-time layer: gdcm::DataElement
00183 DataElement GetAsDataElement() const {
00184     DataElement ret( GetTag() );
00185     std::ostringstream os;
00186     // os.imbue(std::locale::classic()); // This is not required AFAIK
00187     EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
00188         GetNumberOfValues(), os);
00189     ret.SetVR( GetVR() );
00190     assert( ret.GetVR() != VR::SQ );
00191     if( (VR::VRType)VRToEncoding<TVR>::Mode == VR::VRASCII )
00192     {
00193         if( GetVR() != VR::UI )
00194         {
00195             if( os.str().size() % 2 )
00196             {
00197                 os << " ";
00198             }
00199         }
00200     }
00201     VL::Type osStrSize = (VL::Type)os.str().size();
00202     ret.SetByteValue( os.str().c_str(), osStrSize );
00203     return ret;
00204 }
00205
00206 void SetFromDataElement(DataElement const &de) {
00207     // This is kind of hackish but since I do not generate other element than the first one: 0x6000 I
    should be ok:
00208     assert( Tag(Group,Element) == de.GetTag() || Group == 0x6000 || Group == 0x5000 );
00209     assert( GetVR() != VR::INVALID );
00210     assert( GetVR().Compatible( de.GetVR() ) || de.GetVR() == VR::INVALID ); // In case of VR::INVALID
    cannot use the & operator
00211     if( de.IsEmpty() ) return;
00212     const ByteValue *bv = de.GetByteValue();
00213 #ifdef GDCM_WORDS_BIGENDIAN
00214     if( de.GetVR() == VR::UN /*|| de.GetVR() == VR::INVALID*/ )
00215 #else

```

```

00216     if( de.GetVR() == VR::UN || de.GetVR() == VR::INVALID )
00217 #endif
00218     {
00219         SetByteValue(bv);
00220     }
00221     else
00222     {
00223         SetByteValueNoSwap(bv);
00224     }
00225 }
00226 void Set(DataSet const &ds) {
00227     SetFromDataElement( Tag(Group,Element) );
00228 }
00229 void SetFromDataSet(DataSet const &ds) {
00230     if( ds.FindDataElement( Tag(Group,Element) ) &&
00231         !ds.GetDataElement( Tag(Group,Element) ).IsEmpty() )
00232     {
00233         SetFromDataElement( ds.GetDataElement( Tag(Group,Element) ) );
00234     }
00235 }
00236 protected:
00237 void SetByteValueNoSwap(const ByteValue *bv) {
00238     if( !bv ) return; // That would be bad...
00239     assert( bv->GetPointer() && bv->GetLength() ); // [123]C element can be empty
00240     //if( VRToEncoding<TVR>::Mode == VR::VRBINARY )
00241     // {
00242     // // always do a copy !
00243     // SetValues(bv->GetPointer(), bv->GetLength());
00244     // }
00245     //else
00246     {
00247         std::stringstream ss;
00248         std::string s = std::string( bv->GetPointer(), bv->GetLength() );
00249         ss.str( s );
00250         EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadNoSwap(Internal,
00251             GetNumberOfValues(),ss);
00252     }
00253 }
00254 void SetByteValue(const ByteValue *bv) {
00255     if( !bv ) return; // That would be bad...
00256     assert( bv->GetPointer() && bv->GetLength() ); // [123]C element can be empty
00257     //if( VRToEncoding<TVR>::Mode == VR::VRBINARY )
00258     // {
00259     // // always do a copy !
00260     // SetValues(bv->GetPointer(), bv->GetLength());
00261     // }
00262     //else
00263     {
00264         std::stringstream ss;
00265         std::string s = std::string( bv->GetPointer(), bv->GetLength() );
00266         ss.str( s );
00267         EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
00268             GetNumberOfValues(),ss);
00269     }
00270 }
00271 #if 0 // TODO FIXME the implicit way:
00272 // explicit:
00273 void Read(std::istream &_is) {
00274     const uint16_t cref[] = { Group, Element };
00275     uint16_t c[2];
00276     _is.read((char*)&c, sizeof(c));
00277     assert( c[0] == cref[0] && c[1] == cref[1] );
00278     char vr[2];
00279     _is.read(vr, 2); // Check consistency ?
00280     const uint32_t lref = GetLength() * sizeof( typename VRToType<TVR>::Type );
00281     uint32_t l = VRVLSize< (TVR & VR::VL32) >::Read(_is);
00282     l /= sizeof( typename VRToType<TVR>::Type );
00283     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
00284         l,_is);
00285 }
00286 void Write(std::ostream &_os) const {
00287     uint16_t c[] = { Group, Element };
00288     _os.write((char*)&c, 4);
00289     uint32_t l = GetLength() * sizeof( typename VRToType<TVR>::Type );
00290     _os.write((char*)&l, 4);
00291     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
00292         GetLength(),_os);
00293 }
00294 void Read(std::istream &_is) {
00295     uint16_t cref[] = { Group, Element };
00296     uint16_t c[2];

```

```

00297     _is.read((char*)&c, 4);
00298     const uint32_t lref = GetLength() * sizeof( typename VRTToType<TVR>::Type );
00299     uint32_t l;
00300     _is.read((char*)&l, 4);
00301     l /= sizeof( typename VRTToType<TVR>::Type );
00302     return EncodingImplementation<VRTToEncoding<TVR>::Mode>::Read(Internal,
00303         l,_is);
00304 }
00305 void Write(std::ostream &_os) const {
00306     uint16_t c[] = { Group, Element };
00307     _os.write((char*)&c, 4);
00308     uint32_t l = GetLength() * sizeof( typename VRTToType<TVR>::Type );
00309     _os.write((char*)&l, 4);
00310     return EncodingImplementation<VRTToEncoding<TVR>::Mode>::Write(Internal,
00311         GetLength(),_os);
00312 }
00313 #endif
00314
00315 };
00316
00317 template<uint16_t Group, uint16_t Element, long long TVR >
00318 class Attribute<Group,Element,TVR,VM::VM1>
00319 {
00320 public:
00321     typedef typename VRTToType<TVR>::Type ArrayType;
00322     enum { VMType = VMToLength<VM::VM1>::Length };
00323     //ArrayType Internal[VMToLength<TVM>::Length];
00324     ArrayType Internal;
00325     GDCM_STATIC_ASSERT( VMToLength<VM::VM1>::Length == 1 );
00326
00327     // Make sure that user specified VR/VM are compatible with the public dictionary:
00328     GDCM_STATIC_ASSERT( ((VR::VRTType)TVR & (VR::VRTType)(TagToType<Group, Element>::VRTType)) );
00329     GDCM_STATIC_ASSERT( ((VM::VMType)VM::VM1 & (VM::VMType)(TagToType<Group, Element>::VMType)) );
00330     GDCM_STATIC_ASSERT( (((VR::VRTType)TVR & VR::VR_VM1) && ((VM::VMType)VM::VM1 == VM::VM1) )
00331         || !((VR::VRTType)TVR & VR::VR_VM1) );
00332
00333     static Tag GetTag() { return Tag(Group,Element); }
00334     static VR GetVR() { return (VR::VRTType)TVR; }
00335     static VM GetVM() { return (VM::VMType)VM::VM1; }
00336
00337     // The following two methods do make sense only in case of public element,
00338     // when the template is intanciated with private element the VR/VM are simply
00339     // defaulted to allow everything (see gdcmTagToType.h default template for TagToType)
00340     static VR GetDictVR() { return (VR::VRTType)(TagToType<Group, Element>::VRTType); }
00341     static VM GetDictVM() { return (VM::VMType)(TagToType<Group, Element>::VMType); }
00342
00343     // Some extra dummy checks:
00344     // Data Elements with a VR of SQ, OF, OW, OB or UN shall always have a Value Multiplicity of one.
00345
00346     unsigned int GetNumberOfValues() const {
00347         return VMToLength<VM::VM1>::Length;
00348     }
00349     // Implementation of Print is common to all Mode (ASCII/Binary)
00350     // TODO: Can we print a \ when in ASCII...well I don't think so
00351     // it would mean we used a bad VM then, right ?
00352     void Print(std::ostream &os) const {
00353         os << GetTag() << " ";
00354         os << TagToType<Group,Element>::GetVRString() << " ";
00355         os << TagToType<Group,Element>::GetVMString() << " ";
00356         os << Internal; // VM is at least guarantee to be one
00357     }
00358     // copy:
00359     //ArrayType GetValue(unsigned int idx = 0) {
00360     //    assert( idx < GetNumberOfValues() );
00361     //    return Internal[idx];
00362     //}
00363     //ArrayType operator[] (unsigned int idx) {
00364     //    return GetValue(idx);
00365     //}
00366     // FIXME: is this always a good idea ?
00367     // I do not think so, I prefer operator
00368     //operator ArrayType () const { return Internal[0]; }
00369
00370     bool operator==(const Attribute &att) const
00371     {
00372         return std::equal(&Internal, &Internal+GetNumberOfValues(),
00373             att.GetValues());
00374     }
00375     bool operator!=(const Attribute &att) const
00376     {
00377         return !std::equal(&Internal, &Internal+GetNumberOfValues(),

```

```

00378     att.GetValues();
00379 }
00380 bool operator<(const Attribute &att) const
00381 {
00382     return std::lexicographical_compare(&Internal, &Internal+GetNumberOfValues(),
00383     att.GetValues(), att.GetValues() + att.GetNumberOfValues() );
00384 }
00385
00386 ArrayType &GetValue() {
00387 //     assert( idx < GetNumberOfValues() );
00388     return Internal;
00389 }
00390 // ArrayType & operator[] (unsigned int idx) {
00391 //     return GetValue(idx);
00392 // }
00393 // const reference
00394 ArrayType const &GetValue() const {
00395 //assert( idx < GetNumberOfValues() );
00396     return Internal;
00397 }
00398 //ArrayType const & operator[] () const {
00399 //     return GetValue();
00400 //}
00401 void SetValue(ArrayType v) {
00402 //     assert( idx < GetNumberOfValues() );
00403     Internal = v;
00404 }
00405 /* void SetValues(const ArrayType* array, unsigned int numel = VMType ) {
00406     assert( array && numel && numel == GetNumberOfValues() );
00407     // std::copy is smarter than a memcpy, and will call memcpy when POD type
00408     std::copy(array, array+numel, Internal);
00409 }
00410 */
00411
00412 // FIXME Should we remove this function ?
00413 const ArrayType* GetValues() const {
00414     return &Internal;
00415 }
00416
00417 // API to talk to the run-time layer: gdcm::DataElement
00418 DataElement GetAsDataElement() const {
00419     DataElement ret( Tag(Group,Element) );
00420     std::ostringstream os;
00421     // os.imbue(std::locale::classic()); // This is not required AFAIK
00422     EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(&Internal,
00423     GetNumberOfValues(),os);
00424     ret.SetVR( GetVR() );
00425     assert( ret.GetVR() != VR::SQ );
00426     if( (VR::VRType)VRToEncoding<TVR>::Mode == VR::VRASCII )
00427     {
00428         if( GetVR() != VR::UI )
00429         {
00430             if( os.str().size() % 2 )
00431             {
00432                 os << " ";
00433             }
00434         }
00435     }
00436     VL::Type osStrSize = (VL::Type)os.str().size();
00437     ret.SetByteValue( os.str().c_str(), osStrSize );
00438     return ret;
00439 }
00440
00441 void SetFromDataElement(DataElement const &de) {
00442 // This is kind of hackish but since I do not generate other element than the first one: 0x6000 I
00443 // should be ok:
00444     assert( Tag(Group,Element) == de.GetTag() || Group == 0x6000 || Group == 0x5000 );
00445     assert( GetVR() != VR::INVALID );
00446     assert( GetVR().Compatible( de.GetVR() ) || de.GetVR() == VR::INVALID ); // In case of VR::INVALID
00447     // cannot use the & operator
00448     if( de.IsEmpty() ) return;
00449     const ByteValue *bv = de.GetByteValue();
00450 #ifdef GDCM_WORDS_BIGENDIAN
00451     if( de.GetVR() == VR::UN /*|| de.GetVR() == VR::INVALID*/ )
00452     #endif
00453     {
00454         SetByteValue(bv);
00455     }
00456     else

```

```

00457     {
00458         SetByteValueNoSwap(bv);
00459     }
00460 }
00461 void Set(DataSet const &ds) {
00462     SetFromDataElement( ds.GetDataElement( Tag(Group,Element) ) );
00463 }
00464 void SetFromDataSet(DataSet const &ds) {
00465     if( ds.FindDataElement( Tag(Group,Element) ) &&
00466         !ds.GetDataElement( Tag(Group,Element) ).IsEmpty() )
00467     {
00468         SetFromDataElement( ds.GetDataElement( Tag(Group,Element) ) );
00469     }
00470 }
00471 protected:
00472 void SetByteValueNoSwap(const ByteValue *bv) {
00473     if( !bv ) return; // That would be bad...
00474     assert( bv->GetPointer() && bv->GetLength() ); // [123]C element can be empty
00475     //if( VRToEncoding<TVR>::Mode == VR::VRBINARY )
00476     // {
00477     //     // always do a copy !
00478     //     SetValues(bv->GetPointer(), bv->GetLength());
00479     // }
00480     //else
00481     {
00482         std::stringstream ss;
00483         std::string s = std::string( bv->GetPointer(), bv->GetLength() );
00484         ss.str( s );
00485         EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadNoSwap(&Internal,
00486             GetNumberOfValues(),ss);
00487     }
00488 }
00489 void SetByteValue(const ByteValue *bv) {
00490     if( !bv ) return; // That would be bad...
00491     assert( bv->GetPointer() && bv->GetLength() ); // [123]C element can be empty
00492     //if( VRToEncoding<TVR>::Mode == VR::VRBINARY )
00493     // {
00494     //     // always do a copy !
00495     //     SetValues(bv->GetPointer(), bv->GetLength());
00496     // }
00497     //else
00498     {
00499         std::stringstream ss;
00500         std::string s = std::string( bv->GetPointer(), bv->GetLength() );
00501         ss.str( s );
00502         EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(&Internal,
00503             GetNumberOfValues(),ss);
00504     }
00505 }
00506 #if 0 // TODO FIXME the implicit way:
00507 // explicit:
00508 void Read(std::istream &_is) {
00509     const uint16_t cref[] = { Group, Element };
00510     uint16_t c[2];
00511     _is.read((char*)&c, sizeof(c));
00512     assert( c[0] == cref[0] && c[1] == cref[1] );
00513     char vr[2];
00514     _is.read(vr, 2); // Check consistency ?
00515     const uint32_t lref = GetLength() * sizeof( typename VRToType<TVR>::Type );
00516     uint32_t l = VRVLSize<(TVR & VR::VL32)>::Read(_is);
00517     l /= sizeof( typename VRToType<TVR>::Type );
00518     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
00519         l,_is);
00520 }
00521 void Write(std::ostream &_os) const {
00522     uint16_t c[] = { Group, Element };
00523     _os.write((char*)&c, 4);
00524     uint32_t l = GetLength() * sizeof( typename VRToType<TVR>::Type );
00525     _os.write((char*)&l, 4);
00526     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
00527         GetLength(),_os);
00528 }
00529 void Read(std::istream &_is) {
00530     uint16_t cref[] = { Group, Element };
00531     uint16_t c[2];
00532     _is.read((char*)&c, 4);
00533     const uint32_t lref = GetLength() * sizeof( typename VRToType<TVR>::Type );
00534     uint32_t l;
00535     _is.read((char*)&l, 4);
00536     l /= sizeof( typename VRToType<TVR>::Type );
00537     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,

```



```

00538     l,_is);
00539 }
00540 void Write(std::ostream &_os) const {
00541     uint16_t c[] = { Group, Element };
00542     _os.write((char*)&c, 4);
00543     uint32_t l = GetLength() * sizeof( typename VRToType<TVR>::Type );
00544     _os.write((char*)&l, 4);
00545     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
00546         GetLength(),_os);
00547 }
00548 #endif
00549
00550 };
00551
00552 // No need to repeat default template arg, since primary template
00553 // will be used to generate the default arguments
00554 template<uint16_t Group, uint16_t Element, long long TVR >
00555 class Attribute<Group,Element,TVR,VM::VM1_n>
00556 {
00557 public:
00558     typedef typename VRToType<TVR>::Type ArrayType;
00559
00560     // Make sure that user specified VR/VM are compatible with the public dictionary:
00561     GDCM_STATIC_ASSERT( ((VR::VRType)TVR & (VR::VRType)(TagToType<Group, Element>::VRType)) );
00562     GDCM_STATIC_ASSERT( (VM::VM1_n & (VM::VMType)(TagToType<Group, Element>::VMType)) );
00563     GDCM_STATIC_ASSERT( (((VR::VRType)TVR & VR::VR_VM1) && ((VM::VMType)TagToType<Group,Element>::VMType ==
VM::VM1) )
00564         || !((VR::VRType)TVR & VR::VR_VM1) ) );
00565
00566     static Tag GetTag() { return Tag(Group,Element); }
00567     static VR GetVR() { return (VR::VRType)TVR; }
00568     static VM GetVM() { return VM::VM1_n; }
00569
00570     static VR GetDictVR() { return (VR::VRType)(TagToType<Group, Element>::VRType); }
00571     static VM GetDictVM() { return GetVM(); }
00572
00573     // This the way to prevent default initialization
00574     explicit Attribute() { Internal=nullptr; Length=0; Own = true; }
00575     ~Attribute() {
00576         if( Own ) {
00577             delete[] Internal;
00578         }
00579         Internal = nullptr; // paranoid
00580     }
00581
00582     unsigned int GetNumberOfValues() const { return Length; }
00583
00584     void SetNumberOfValues(unsigned int numel)
00585     {
00586         SetValues(nullptr, numel, true);
00587     }
00588
00589     const ArrayType* GetValues() const {
00590         return Internal;
00591     }
00592     void Print(std::ostream &os) const {
00593         os << GetTag() << " ";
00594         os << GetVR() << " ";
00595         os << GetVM() << " ";
00596         os << Internal[0]; // VM is at least guarantee to be one
00597         for(unsigned int i=1; i<GetNumberOfValues(); ++i)
00598             os << "," << Internal[i];
00599     }
00600     ArrayType &GetValue(unsigned int idx = 0) {
00601         assert( idx < GetNumberOfValues() );
00602         return Internal[idx];
00603     }
00604     ArrayType &operator[] (unsigned int idx) {
00605         return GetValue(idx);
00606     }
00607     // const reference
00608     ArrayType const &GetValue(unsigned int idx = 0) const {
00609         assert( idx < GetNumberOfValues() );
00610         return Internal[idx];
00611     }
00612     ArrayType const &operator[] (unsigned int idx) const {
00613         return GetValue(idx);
00614     }
00615     void SetValue(unsigned int idx, ArrayType v) {
00616         assert( idx < GetNumberOfValues() );
00617         Internal[idx] = v;

```

```

00618     }
00619     void SetValue(ArrayType v) { SetValue(0, v); }
00620
00621     void SetValues(const ArrayType *array, unsigned int numel, bool own = false)
00622     {
00623         if( Internal ) // were we used before ?
00624         {
00625             // yes !
00626             if( Own ) delete[] Internal;
00627             Internal = nullptr;
00628         }
00629         Own = own;
00630         Length = numel;
00631         assert( Internal == nullptr );
00632         if( own ) // make a copy:
00633         {
00634             Internal = new ArrayType[numel];
00635             if( array && numel )
00636                 std::copy(array, array+numel, Internal);
00637         }
00638         else // pass pointer
00639         {
00640             Internal = const_cast<ArrayType*>(array);
00641         }
00642         // postcondition
00643         assert( numel == GetNumberOfValues() );
00644     }
00645
00646     DataElement GetAsDataElement() const {
00647         DataElement ret( GetTag() );
00648         std::ostringstream os;
00649         if( Internal )
00650         {
00651             EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
00652                 GetNumberOfValues(),os);
00653             if( (VR::VRType)VRToEncoding<TVR>::Mode == VR::VRASCII )
00654             {
00655                 if( GetVR() != VR::UI )
00656                 {
00657                     if( os.str().size() % 2 )
00658                     {
00659                         os << " ";
00660                     }
00661                 }
00662             }
00663         }
00664         ret.SetVR( GetVR() );
00665         assert( ret.GetVR() != VR::SQ );
00666         VL::Type osStrSize = (VL::Type) os.str().size();
00667         ret.SetByteValue( os.str().c_str(), osStrSize);
00668         return ret;
00669     }
00670
00671     void SetFromDataElement(DataElement const &de) {
00672         // This is kind of hackish but since I do not generate other element than the first one: 0x6000 I
00673         // should be ok:
00674         assert( GetTag() == de.GetTag() || GetTag().GetGroup() == 0x6000
00675             || GetTag().GetGroup() == 0x5000 );
00676         assert( GetVR().Compatible( de.GetVR() ) ); // In case of VR::INVALID cannot use the & operator
00677         assert( !de.IsEmpty() );
00678         const ByteValue *bv = de.GetByteValue();
00679         SetByteValue(bv);
00680     }
00681
00682     void Set(DataSet const &ds) {
00683         SetFromDataElement( ds.GetDataElement( GetTag() ) );
00684     }
00685
00686     void SetFromDataSet(DataSet const &ds) {
00687         if( ds.FindDataElement( GetTag() ) &&
00688             !ds.GetDataElement( GetTag() ).IsEmpty() )
00689         {
00690             SetFromDataElement( ds.GetDataElement( GetTag() ) );
00691         }
00692     }
00693
00694 protected:
00695     void SetByteValue(const ByteValue *bv) {
00696         assert( bv ); // FIXME
00697         std::stringstream ss;
00698         std::string s = std::string( bv->GetPointer(), bv->GetLength() );
00699         Length = bv->GetLength(); // HACK FIXME
00700         ss.str( s );
00701         ArrayType *internal;
00702         ArrayType buffer[256];

```

```

00698     if( bv->GetLength() < 256 )
00699     {
00700         internal = buffer;
00701     }
00702     else
00703     {
00704         internal = new ArrayType[ (VL::Type)bv->GetLength() ]; // over allocation
00705     }
00706     EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadComputeLength(internal, Length, ss);
00707     SetValues( internal, Length, true );
00708     if( !(bv->GetLength() < 256) )
00709     {
00710         delete[] internal;
00711     }
00712     //EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
00713     //  GetNumberOfValues(),ss);
00714 }
00715
00716 private:
00717     ArrayType *Internal;
00718     unsigned int Length;
00719     bool Own : 1;
00720 };
00721
00722 template<uint16_t Group, uint16_t Element, long long TVR>
00723 class Attribute<Group,Element,TVR,VM::VM1_3> : public Attribute<Group,Element,TVR,VM::VM1_n>
00724 {
00725 public:
00726     VM GetVM() const { return VM::VM1_3; }
00727 };
00728
00729 template<uint16_t Group, uint16_t Element, long long TVR>
00730 class Attribute<Group,Element,TVR,VM::VM1_8> : public Attribute<Group,Element,TVR,VM::VM1_n>
00731 {
00732 public:
00733     VM GetVM() const { return VM::VM1_8; }
00734 };
00735
00736 template<uint16_t Group, uint16_t Element, long long TVR>
00737 class Attribute<Group,Element,TVR,VM::VM2_n> : public Attribute<Group,Element,TVR,VM::VM1_n>
00738 {
00739 public:
00740     VM GetVM() const { return VM::VM2_n; }
00741 };
00742
00743 template<uint16_t Group, uint16_t Element, long long TVR>
00744 class Attribute<Group,Element,TVR,VM::VM2_2n> : public Attribute<Group,Element,TVR,VM::VM2_n>
00745 {
00746 public:
00747     static VM GetVM() { return VM::VM2_2n; }
00748 };
00749
00750 template<uint16_t Group, uint16_t Element, long long TVR>
00751 class Attribute<Group,Element,TVR,VM::VM3_n> : public Attribute<Group,Element,TVR,VM::VM1_n>
00752 {
00753 public:
00754     static VM GetVM() { return VM::VM3_n; }
00755 };
00756
00757 template<uint16_t Group, uint16_t Element, long long TVR>
00758 class Attribute<Group,Element,TVR,VM::VM3_3n> : public Attribute<Group,Element,TVR,VM::VM3_n>
00759 {
00760 public:
00761     static VM GetVM() { return VM::VM3_3n; }
00762 };
00763
00764
00765 // For particular case for ASCII string
00766 // WARNING: This template explicitly instantiates a particular
00767 // EncodingImplementation THEREFORE it is required to be declared after the
00768 // EncodingImplementation is needs (doh!)
00769 #if 0
00770 template<int TVM>
00771 class Attribute<TVM>
00772 {
00773 public:
00774     Attribute(const char array[])
00775     {
00776         unsigned int i = 0;
00777         const char sep = '\\';
00778         std::string sarray = array;

```

```

00779     std::string::size_type pos1 = 0;
00780     std::string::size_type pos2 = sarray.find(sep, pos1+1);
00781     while(pos2 != std::string::npos)
00782     {
00783         Internal[i++] = sarray.substr(pos1, pos2-pos1);
00784         pos1 = pos2+1;
00785         pos2 = sarray.find(sep, pos1+1);
00786     }
00787     Internal[i] = sarray.substr(pos1, pos2-pos1);
00788     // Shouldn't we do the contrary, since we know how many separators
00789     // (and default behavior is to discard anything after the VM declared
00790     assert( GetLength()-1 == i );
00791 }
00792
00793 unsigned long GetLength() const {
00794     return VMTToLength<TVM>::Length;
00795 }
00796 // Implementation of Print is common to all Mode (ASCII/Binary)
00797 void Print(std::ostream &_os) const {
00798     _os << Internal[0]; // VM is at least guarantee to be one
00799     for(int i=1; i<VMTToLength<TVM>::Length; ++i)
00800         _os << ", " << Internal[i];
00801 }
00802
00803 void Read(std::istream &_is) {
00804     EncodingImplementation<VR::VRASCII>::Read(Internal, GetLength(), _is);
00805 }
00806 void Write(std::ostream &_os) const {
00807     EncodingImplementation<VR::VRASCII>::Write(Internal, GetLength(), _os);
00808 }
00809 private:
00810     typename String Internal[VMTToLength<TVM>::Length];
00811 };
00812
00813 template< int TVM>
00814 class Attribute<VR::PN, TVM> : public StringAttribute<TVM>
00815 {
00816 };
00817 #endif
00818
00819 #if 0
00820
00821 // Implementation for the undefined length (dynamically allocated array)
00822 template<int TVR>
00823 class Attribute<TVR, VM::VM1_n>
00824 {
00825 public:
00826     // This the way to prevent default initialization
00827     explicit Attribute() { Internal=0; Length=0; }
00828     ~Attribute() {
00829         delete[] Internal;
00830         Internal = 0;
00831     }
00832
00833     // Length manipulation
00834     // SetLength should really be protected anyway...all operation
00835     // should go through SetArray
00836     unsigned long GetLength() const { return Length; }
00837     typedef typename VRToType<TVR>::Type ArrayType;
00838     void SetLength(unsigned long len) {
00839         const unsigned int size = sizeof(ArrayType);
00840         if( len ) {
00841             if( len > Length ) {
00842                 // perform realloc
00843                 assert( (len / size) * size == len );
00844                 ArrayType *internal = new ArrayType[len / size];
00845                 memcpy(internal, Internal, Length * size);
00846                 delete[] Internal;
00847                 Internal = internal;
00848             }
00849         }
00850         Length = len / size;
00851     }
00852
00853     // If save is set to zero user should not delete the pointer
00854     //void SetArray(const typename VRToType<TVR>::Type *array, int len, bool save = false)
00855     void SetArray(const ArrayType *array, unsigned long len,
00856                 bool save = false) {
00857         if( save ) {
00858             SetLength(len); // realloc
00859             memcpy(Internal, array, len/sizeof(ArrayType)*);

```

```

00860     }
00861     else {
00862         // TODO rewrite this stupid code:
00863         Length = len;
00864         //Internal = array;
00865         assert(0);
00866     }
00867 }
00868 // Implementation of Print is common to all Mode (ASCII/Binary)
00869 void Print(std::ostream &_os) const {
00870     assert( Length );
00871     assert( Internal );
00872     _os << Internal[0]; // VM is at least guarantee to be one
00873     const unsigned long length = GetLength() < 25 ? GetLength() : 25;
00874     for(unsigned long i=1; i<length; ++i)
00875         _os << ", " << Internal[i];
00876 }
00877 void Read(std::istream &_is) {
00878     EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
00879         GetLength(),_is);
00880 }
00881 void Write(std::ostream &_os) const {
00882     EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
00883         GetLength(),_os);
00884 }
00885
00886 Attribute(const Attribute&_val) {
00887     if( this != &_amp;_val) {
00888         *this = _val;
00889     }
00890 }
00891
00892 Attribute &operator=(const Attribute &_val) {
00893     Length = 0; // SYITF
00894     Internal = 0;
00895     SetArray(_val.Internal, _val.Length, true);
00896     return *this;
00897 }
00898
00899 private:
00900     typename VRToType<TVR>::Type *Internal;
00901     unsigned long Length; // unsigned int ??
00902 };
00903
00904 //template <int TVM = VM::VM1_n>
00905 //class Attribute<VR::OB, TVM > : public Attribute<VR::OB, VM::VM1_n> {};
00906
00907 // Partial specialization for derivatives of 1-n : 2-n, 3-n ...
00908 template<int TVR>
00909 class Attribute<TVR, VM::VM2_n> : public Attribute<TVR, VM::VM1_n>
00910 {
00911 public:
00912     typedef Attribute<TVR, VM::VM1_n> Parent;
00913     void SetLength(int len) {
00914         if( len <= 1 ) return;
00915         Parent::SetLength(len);
00916     }
00917 };
00918 template<int TVR>
00919 class Attribute<TVR, VM::VM2_2n> : public Attribute<TVR, VM::VM2_n>
00920 {
00921 public:
00922     typedef Attribute<TVR, VM::VM2_n> Parent;
00923     void SetLength(int len) {
00924         if( len % 2 ) return;
00925         Parent::SetLength(len);
00926     }
00927 };
00928 template<int TVR>
00929 class Attribute<TVR, VM::VM3_n> : public Attribute<TVR, VM::VM1_n>
00930 {
00931 public:
00932     typedef Attribute<TVR, VM::VM1_n> Parent;
00933     void SetLength(int len) {
00934         if( len <= 2 ) return;
00935         Parent::SetLength(len);
00936     }
00937 };
00938 template<int TVR>
00939 class Attribute<TVR, VM::VM3_3n> : public Attribute<TVR, VM::VM3_n>
00940 {

```

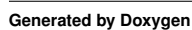
```

00941 public:
00942     typedef Attribute<TVR, VM::VM3_n> Parent;
00943     void SetLength(int len) {
00944         if( len % 3 ) return;
00945         Parent::SetLength(len);
00946     }
00947 };
00948
00949
00950 //template<int T> struct VRToLength;
00951 //template<> struct VRToLength<VR::AS>
00952 //{ enum { Length = VM::VM1 }; }
00953 //template<>
00954 //class Attribute<VR::AS> : public Attribute<VR::AS, VRToLength<VR::AS>::Length >
00955
00956 // only 0010 1010 AS 1 Patient's Age
00957 template<>
00958 class Attribute<VR::AS, VM::VM5>
00959 {
00960 public:
00961     char Internal[VMToLength<VM::VM5>::Length];
00962     void Print(std::ostream &_os) const {
00963         _os << Internal;
00964     }
00965 };
00966
00967 template<>
00968 class Attribute<VR::OB, VM::VM1> : public Attribute<VR::OB, VM::VM1_n> {};
00969 // Make it impossible to compile any other cases:
00970 template<int TVM> class Attribute<VR::OB, TVM>;
00971
00972 // Same for OW:
00973 template<>
00974 class Attribute<VR::OW, VM::VM1> : public Attribute<VR::OW, VM::VM1_n> {};
00975 // Make it impossible to compile any other cases:
00976 template<int TVM> class Attribute<VR::OW, TVM>;
00977 #endif
00978
00979 #if 0
00980 template<>
00981 class Attribute<0x7fe0,0x0010, VR::OW, VM::VM1>
00982 {
00983 public:
00984     char *Internal;
00985     unsigned long Length; // unsigned int ??
00986
00987     void Print(std::ostream &_os) const {
00988         _os << Internal[0];
00989     }
00990     void SetBytes(char *bytes, unsigned long length) {
00991         Internal = bytes;
00992         Length = length;
00993     }
00994     void Read(std::istream &_is) {
00995         uint16_t c[2];
00996         _is.read((char*)&c, 4);
00997         uint32_t l;
00998         _is.read((char*)&l, 4);
00999         Length = l;
01000         _is.read( Internal, Length );
01001     }
01002     void Write(std::ostream &_os) const {
01003         uint16_t c[] = {0x7fe0, 0x0010};
01004         _os.write((char*)&c, 4);
01005         _os.write((char*)&Length, 4);
01006         _os.write( Internal, Length );
01007     }
01008 };
01009 #endif
01010
01011 /*
01012 // Removing Attribute for SQ for now...
01013 template<uint16_t Group, uint16_t Element, typename SQA>
01014 class Attribute<Group,Element, VR::SQ, VM::VM1, SQA>
01015 {
01016 public:
01017     SQA sqa;
01018     void Print(std::ostream &_os) const {
01019         _os << Tag(Group,Element);
01020         sqa.Print(_os << std::endl << '\t');
01021     }

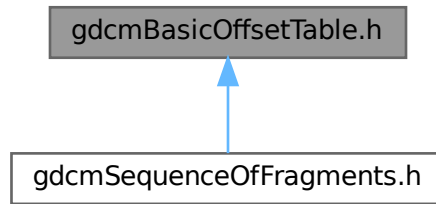
```

## 11.111 gdcmBasicOffsetTable.h File Reference

Include dependency graph for `gdcmBasicOffsetTable.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcmbasicOffsetTable](#)  
Class to represent a [BasicOffsetTable](#).

## Namespaces

- namespace [gdcmbasic](#)

## Functions

- `std::ostream & gdcmbasic::operator<< (std::ostream &os, const BasicOffsetTable &val)`

## 11.112 gdcmbasicOffsetTable.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcmbasic.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012  =====*/
00013
00014
00015  #ifndef GDCMBASICOFFSETTABLE_H
00016  #define GDCMBASICOFFSETTABLE_H
00017
00018  #include "gdcmbasicFragment.h"
00019
00020  namespace gdcmbasic_ns
00021  {
00022  class GDCM_EXPORT BasicOffsetTable : public Fragment
  
```



```

00027 {
00028 //protected:
00029 // void SetTag(const Tag &t);
00030 public:
00031 BasicOffsetTable() : Fragment() {}
00032 friend std::ostream &operator<<(std::ostream &os, const BasicOffsetTable &val);
00033
00034 /*
00035 VL GetLength() const {
00036     assert( !ValueLengthField.IsUndefined() );
00037     assert( !ValueField || ValueField->GetLength() == ValueLengthField );
00038     return TagField.GetLength() + ValueLengthField.GetLength()
00039         + ValueLengthField;
00040 }
00041 */
00042
00043 template <typename TSwap>
00044 std::istream &Read(std::istream &is) {
00045     // Superclass
00046     const Tag itemStart(0xffff, 0xe000);
00047     if( !TagField.Read<TSwap>(is) )
00048     {
00049         assert(0 && "Should not happen");
00050         return is;
00051     }
00052     //assert( TagField == itemStart );
00053     if( TagField != itemStart )
00054     {
00055         // Bug_Siemens_PrivateIconNoItem.dcm
00056         //gdcmbDebugMacro( "Could be Bug_Siemens_PrivateIconNoItem.dcm" );
00057         ParseException pe;
00058         pe.SetLastElement(*this);
00059         //throw "SIEMENS Icon thingy";
00060         throw pe;
00061     }
00062     if( !ValueLengthField.Read<TSwap>(is) )
00063     {
00064         assert(0 && "Should not happen");
00065         return is;
00066     }
00067     // Self
00068     SmartPointer<ByteValue> bv = new ByteValue;
00069     bv->SetLength(ValueLengthField);
00070     if( !bv->Read<TSwap>(is) )
00071     {
00072         gdcmbAssertAlwaysMacro(0 && "Should not happen");
00073         return is;
00074     }
00075     ValueField = bv;
00076     return is;
00077 }
00078
00079 /*
00080 template <typename TSwap>
00081 std::ostream &Write(std::ostream &os) const {
00082     const Tag itemStart(0xffff, 0xe000);
00083     const Tag seqDelItem(0xffff, 0xe0dd);
00084     if( !TagField.Write<TSwap>(os) )
00085     {
00086         assert(0 && "Should not happen");
00087         return os;
00088     }
00089     assert( TagField == itemStart );
00090     if( !ValueLengthField.Write<TSwap>(os) )
00091     {
00092         assert(0 && "Should not happen");
00093         return os;
00094     }
00095     if( ValueLengthField )
00096     {
00097         // Self
00098         const ByteValue *bv = GetByteValue();
00099         assert( bv );
00100         assert( bv->GetLength() == ValueLengthField );
00101         if( !bv->Write<TSwap>(os) )
00102         {
00103             assert(0 && "Should not happen");
00104             return os;
00105         }
00106     }
00107     return os;

```

```

00108     }
00109     */
00110 };
00111 //-----
00112 inline std::ostream &operator<<(std::ostream &os, const BasicOffsetTable &val)
00113 {
00114     os << " BasicOffsetTable Length=" << val.ValueLengthField << std::endl;
00115     if( val.ValueField )
00116     {
00117         const ByteValue *bv = val.GetByteValue();
00118         assert( bv );
00119         os << *bv;
00120     }
00121 }
00122 return os;
00123 }
00124
00125
00126 } // end namespace gdcms
00127
00128 #endif //GDCMBASICOFFSETTABLE_H

```

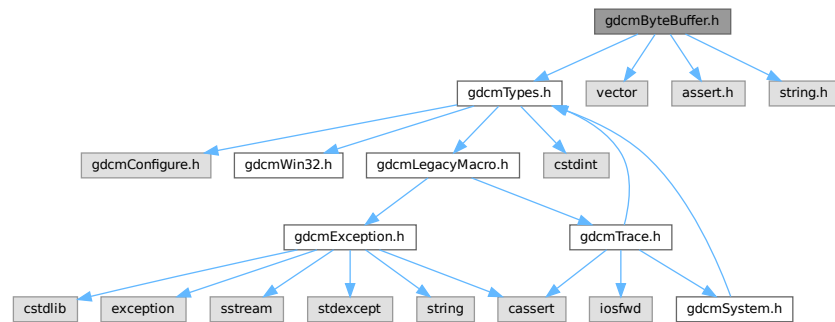
## 11.113 gdcmsByteBuffer.h File Reference

```

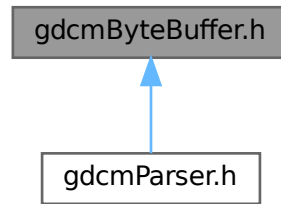
#include "gdcmsTypes.h"
#include <vector>
#include <assert.h>
#include <string.h>

```

Include dependency graph for gdcmsByteBuffer.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::ByteBuffer](#)  
*ByteBuffer.*

## Namespaces

- namespace [gdcm](#)

## 11.114 gdcmByteBuffer.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMBYTEBUFFER_H
00015  #define GDCMBYTEBUFFER_H
00016
00017  #include "gdcmTypes.h"
00018  #include <vector>
00019  #include <assert.h>
00020  #include <string.h> // memmove
00021
00022  #error should not be used
00023
00024  namespace gdcm
00025  {
00034  class ByteBuffer
00035  {
00036  public:
00037      static const int InitBufferSize = 1024;
00038      ByteBuffer() : Start(0), End(0), Limit(0) {}
  
```

```

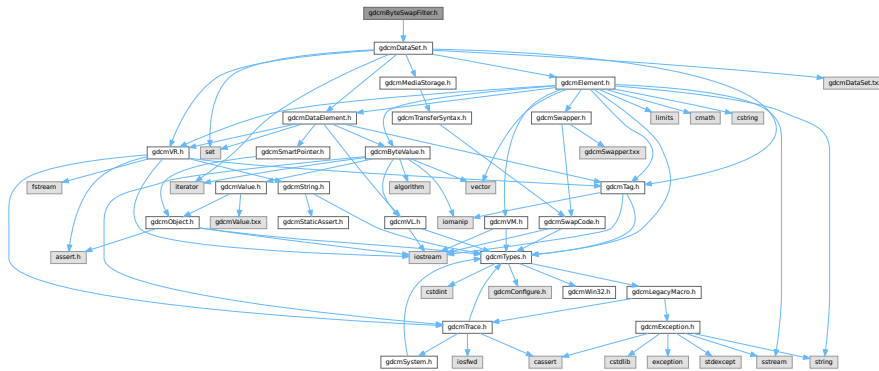
00039 char *Get(int len)
00040 {
00041     char *buffer = &Internal[0];
00042     if (len > Limit - End)
00043     {
00044         // FIXME avoid integer overflow
00045         int neededSize = len + (End - Start);
00046         if (neededSize <= Limit - buffer)
00047         {
00048             memmove(buffer, Start, End - Start);
00049             End = buffer + (End - Start);
00050             Start = buffer;
00051         }
00052     else
00053     {
00054         char *newBuf;
00055         int bufferSize = Limit - Start;
00056         if ( bufferSize == 0 )
00057         {
00058             bufferSize = InitBufferSize;
00059         }
00060         do
00061         {
00062             bufferSize *= 2;
00063         } while (bufferSize < neededSize);
00064         //newBuf = malloc(bufferSize);
00065         try
00066         {
00067             Internal.reserve(bufferSize);
00068             newBuf = &Internal[0];
00069         }
00070         catch(...)
00071         {
00072             //errorCode = NoMemoryError;
00073             return 0;
00074         }
00075         Limit = newBuf + bufferSize;
00076     }
00077     if (Start)
00078     {
00079         memcpy(newBuf, Start, End - Start);
00080     }
00081     End = newBuf + (End - Start);
00082     Start = /*buffer =*/ newBuf;
00083 }
00084 }
00085 assert( (int)Internal.capacity() >= len );
00086 return End;
00087 }
00088
00089 void UpdatePosition() {}
00090 void ShiftEnd(int len) {
00091     End += len;
00092 }
00093 const char *GetStart() const {
00094     return Start;
00095 }
00096
00097 private:
00098     typedef std::vector<char> CharVector;
00099     const char *Start;
00100     char *End;
00101     const char *Limit;
00102     CharVector Internal;
00103 };
00104
00105 } // end namespace gdcmm
00106
00107 #endif //GDCMBYTEBUFFER_H

```

## 11.115 gdcmByteSwapFilter.h File Reference

```
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmByteSwapFilter.h:



### Classes

- class [gdcm::ByteSwapFilter](#)  
*ByteSwapFilter.*

### Namespaces

- namespace [gdcm](#)

## 11.116 gdcmByteSwapFilter.h

[Go to the documentation of this file.](#)

```
00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMBYTESWAPFILTER_H
00015  #define GDCMBYTESWAPFILTER_H
00016
00017  #include "gdcmDataSet.h"
00018
00019  namespace gdcm
00020  {
00021
00027  class GDCM_EXPORT ByteSwapFilter
00028  {
00029  public:
```

```

00030     ByteSwapFilter(DataSet& ds):DS(ds),ByteSwapTag(false) {}
00031     ~ByteSwapFilter() = default;
00032     ByteSwapFilter(const ByteSwapFilter &) = delete;
00033     ByteSwapFilter& operator=(const ByteSwapFilter &) = delete;
00034
00035     bool ByteSwap();
00036     void SetByteSwapTag(bool b) { ByteSwapTag = b; }
00037
00038 private:
00039     DataSet &DS;
00040     bool ByteSwapTag;
00041
00042 };
00043
00044 } // end namespace gdcm
00045
00046 #endif //GDCMBYTESWAPFILTER_H

```

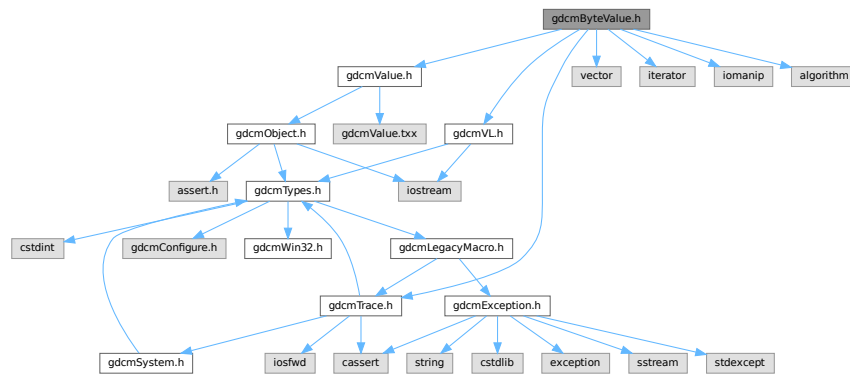
## 11.117 gdcmByteValue.h File Reference

```

#include "gdcmValue.h"
#include "gdcmTrace.h"
#include "gdcmVL.h"
#include <vector>
#include <iterator>
#include <iomanip>
#include <algorithm>

```

Include dependency graph for gdcmByteValue.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::ByteValue](#)  
Class to represent binary value (array of bytes)

## Namespaces

- namespace [gdcm](#)

## 11.118 gdcmByteValue.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMBYTEVALUE_H
00015 #define GDCMBYTEVALUE_H
00016
00017 #include "gdcmValue.h"
00018 #include "gdcmTrace.h"
00019 #include "gdcmVL.h"
00020
00021 #include <vector>
00022 #include <iterator>
00023 #include <iomanip>
00024 #include <algorithm>
00025
00026 namespace gdcm_ns
00027 {
00028   #if !defined(SWIGPYTHON) && !defined(SWIGSHARP) && !defined(SWIGJAVA) && !defined(SWIGPHP)
00029     using namespace gdcm;
00030   #endif
00031   class GDCM_EXPORT ByteValue : public Value
00032   {
00033   public:
00034     ByteValue(const char* array = nullptr, VL const &vl = 0):
00035       Internal(array, array+vl), Length(vl) {
00036       if( vl.IsOdd() )
00037       {
00038         gdcmDebugMacro( "Odd length" );
00039         Internal.resize(vl+1);
00040         ++Length;
00041       }
00042     }
00043     ByteValue(std::vector<char> &v):Internal(v),Length((uint32_t)v.size()) {}
00044     //ByteValue(std::ostringstream const &os) {
00045     //  (void)os;
00046     //  assert(0); // TODO
00047     //}
00048     ~ByteValue() override {
00049       Internal.clear();
00050     }
00051
00052     // When 'dumping' dicom file we still have some information from
00053     // Either the VR: eg LO (private tag)
00054     void PrintASCII(std::ostream &os, VL maxlength) const;
00055
00056     void PrintHex(std::ostream &os, VL maxlength) const;
00057
00058     // Either from Element Number (== 0x0000)
00059     void PrintGroupLength(std::ostream &os) {
00060       assert( Length == 2 );
00061       (void)os;
00062     }
00063
00064     bool IsEmpty() const {
00065     #if 0
00066       if( Internal.empty() ) assert( Length == 0 );
00067     #endif
00068     }
00069   };
00070
00071   #if 0
00072     if( Internal.empty() ) assert( Length == 0 );
00073   #endif

```

```

00072     return Internal.empty();
00073 #else
00074     return Length == 0;
00075 #endif
00076 }
00077 VL GetLength() const override { return Length; }
00078
00079 VL ComputeLength() const { return Length + Length % 2; }
00080 // Does a reallocation
00081 void SetLength(VL vl) override;
00082
00083 operator const std::vector<char>& () const { return Internal; }
00084
00085 ByteValue &operator=(const ByteValue &val) {
00086     Internal = val.Internal;
00087     Length = val.Length;
00088     return *this;
00089 }
00090
00091 bool operator==(const ByteValue &val) const {
00092     if( Length != val.Length )
00093         return false;
00094     if( Internal == val.Internal )
00095         return true;
00096     return false;
00097 }
00098 bool operator==(const Value &val) const override
00099 {
00100     const ByteValue &bv = dynamic_cast<const ByteValue>(val);
00101     return Length == bv.Length && Internal == bv.Internal;
00102 }
00103
00104 void Append(ByteValue const &bv);
00105
00106 void Clear() override {
00107     Internal.clear();
00108 }
00109 // Use that only if you understand what you are doing
00110 const char *GetPointer() const {
00111     if(!Internal.empty()) return &Internal[0];
00112     return nullptr;
00113 }
00114 // Use that only if you really understand what you are doing
00115 const void *GetVoidPointer() const {
00116     if(!Internal.empty()) return &Internal[0];
00117     return nullptr;
00118 }
00119 void *GetVoidPointer() {
00120     if(!Internal.empty()) return &Internal[0];
00121     return nullptr;
00122 }
00123 void Fill(char c) {
00124     //if( Internal.empty() ) return;
00125     std::vector<char>::iterator it = Internal.begin();
00126     for(; it != Internal.end(); ++it) *it = c;
00127 }
00128 bool GetBuffer(char *buffer, unsigned long length) const;
00129 bool WriteBuffer(std::ostream &os) const {
00130     if( Length ) {
00131         //assert( Internal.size() <= Length );
00132         assert( !(Internal.size() % 2) );
00133         os.write(&Internal[0], Internal.size() );
00134     }
00135     return true;
00136 }
00137
00138 template <typename TSwap, typename TType>
00139 std::istream &Read(std::istream &is, bool readvalues = true) {
00140     // If Length is odd we have detected that in SetLength
00141     // and calling std::vector::resize make sure to allocate *AND*
00142     // initialize values to 0 so we are sure to have a \0 at the end
00143     // even in this case
00144     if(Length)
00145     {
00146         if( readvalues )
00147         {
00148             is.read(&Internal[0], Length);
00149             assert( Internal.size() == Length || Internal.size() == Length + 1 );
00150             TSwap::SwapArray((TType*)GetVoidPointer(), Internal.size() / sizeof(TType) );
00151         }
00152         else

```



```

00153     {
00154         is.seekg(Length, std::ios::cur);
00155     }
00156 }
00157 return is;
00158 }
00159
00160 template <typename TSwap>
00161 std::istream &Read(std::istream &is) {
00162     return Read<TSwap,uint8_t>(is);
00163 }
00164
00165 template <typename TSwap, typename TType>
00166 std::ostream const &Write(std::ostream &os) const {
00167     assert( !(Internal.size() % 2) );
00168     if( !Internal.empty() ) {
00169         //os.write(&Internal[0], Internal.size());
00170         std::vector<char> copy = Internal;
00171         TSwap::SwapArray((TType*)(void*)&copy[0], Internal.size() / sizeof(TType) );
00172         os.write(&copy[0], copy.size());
00173     }
00174     return os;
00175 }
00176
00177 template <typename TSwap>
00178 std::ostream const &Write(std::ostream &os) const {
00179     return Write<TSwap,uint8_t>(os);
00180 }
00181
00182 bool IsPrintable(VL length) const {
00183     assert( length <= Length );
00184     for(unsigned int i=0; i<length; i++)
00185     {
00186         if ( i == (length-1) && Internal[i] == '\0' ) continue;
00187         if ( !( isprint((unsigned char)Internal[i]) || isspace((unsigned char)Internal[i]) ) )
00188         {
00189             //gdcMWarningMacro( "Cannot print : " << i );
00190             return false;
00191         }
00192     }
00193     return true;
00194 }
00195
00196 void PrintPNXML(std::ostream &os) const;
00197 void PrintASCIIXML(std::ostream &os) const;
00198 void PrintHexXML(std::ostream &os) const;
00199 protected:
00200 void Print(std::ostream &os) const override {
00201     // This is perfectly valid to have a Length = 0 , so we cannot check
00202     // the length for printing
00203     if( !Internal.empty() )
00204     {
00205         if( IsPrintable(Length) )
00206         {
00207             // WARNING: Internal.end() != Internal.begin()+Length
00208             std::vector<char>::size_type length = Length;
00209             if( Internal.back() == 0 ) --length;
00210             std::copy(Internal.begin(), Internal.begin()+length,
00211                 std::ostream_iterator<char>(os));
00212         }
00213         else
00214             os << "Loaded:" << Internal.size();
00215     }
00216     else
00217     {
00218         //os << "Not Loaded";
00219         os << "(no value available)";
00220     }
00221 }
00222
00223 /*
00224 //Introduce check for invalid XML characters
00225 friend std::ostream& operator<<(std::ostream &os,const char c);
00226 */
00227
00228 void SetLengthOnly(VL vl) override {
00229     Length = vl;
00230 }
00231
00232 private:
00233     std::vector<char> Internal;

```

```

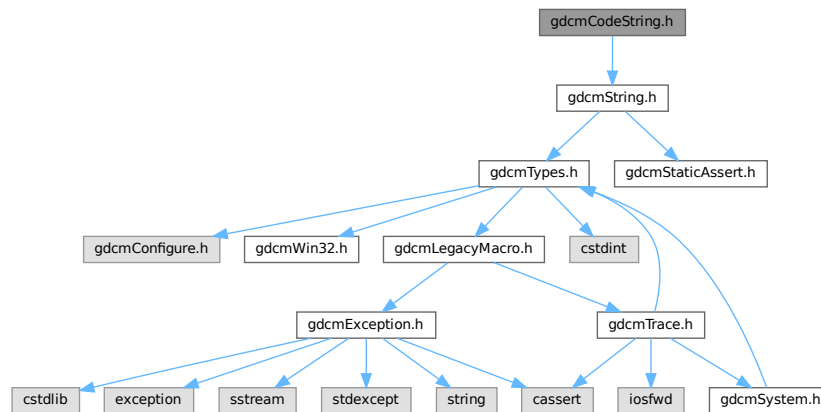
00241
00242 // WARNING Length IS NOT Internal.size() some *featured* DICOM
00243 // implementation define odd length, we always load them as even number
00244 // of byte, so we need to keep the right Length
00245 VL Length;
00246 };
00247
00248 } // end namespace gdcn_ns
00249
00250 #endif //GDCMBYTEVALUE_H

```

## 11.119 gdcnCodeString.h File Reference

```
#include "gdcnString.h"
```

Include dependency graph for gdcnCodeString.h:



### Classes

- class [gdcn::CodeString](#)  
*CodeString.*

### Namespaces

- namespace [gdcn](#)

### Functions

- bool [gdcn::operator!=](#) (const [CodeString](#) &ref, const [CodeString](#) &cs)
- std::ostream & [gdcn::operator<<](#) (std::ostream &os, const [CodeString](#) &str)
- bool [gdcn::operator==](#) (const [CodeString](#) &ref, const [CodeString](#) &cs)

## 11.120 gdcmCodeString.h

[Go to the documentation of this file.](#)

```

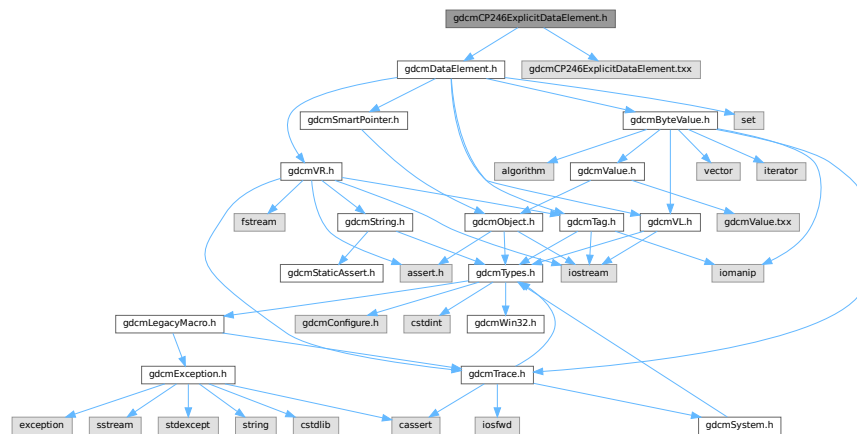
00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMCODESTRING_H
00015  #define GDCMCODESTRING_H
00016
00017  #include "gdcmString.h"
00018
00019  namespace gdcm
00020  {
00021
00022  // Note to myself: because not all wrapped language support exception
00023  // we could not support throwing an exception during object construction.
00024  class GDCM_EXPORT CodeString
00025  {
00026  {
00027      friend std::ostream& operator<( std::ostream& os, const CodeString& str);
00028      friend bool operator==(const CodeString &ref, const CodeString& cs);
00029      friend bool operator!=(const CodeString &ref, const CodeString& cs);
00030      typedef String<'\\',16> InternalClass;
00031  public:
00032      typedef InternalClass::value_type          value_type;
00033      typedef InternalClass::pointer             pointer;
00034      typedef InternalClass::reference           reference;
00035      typedef InternalClass::const_reference     const_reference;
00036      typedef InternalClass::size_type           size_type;
00037      typedef InternalClass::difference_type     difference_type;
00038      typedef InternalClass::iterator            iterator;
00039      typedef InternalClass::const_iterator      const_iterator;
00040      typedef InternalClass::reverse_iterator    reverse_iterator;
00041      typedef InternalClass::const_reverse_iterator const_reverse_iterator;
00042
00043      CodeString(): Internal() {}
00044      CodeString(const value_type* s): Internal(s) { Internal = Internal.Trim(); }
00045      CodeString(const value_type* s, size_type n): Internal(s, n) {
00046          Internal = Internal.Trim(); }
00047      CodeString(const InternalClass& s, size_type pos=0, size_type n=InternalClass::npos):
00048          Internal(s, pos, n) { Internal = Internal.Trim(); }
00049
00050      bool IsValid() const;
00051
00052      std::string GetAsString() const {
00053          return Internal;
00054      }
00055
00056      size_type Size() const { return Internal.size(); }
00057
00058  protected:
00059      std::string TrimInternal() const {
00060          return Internal.Trim();
00061      }
00062
00063  private:
00064      String<'\\',16> Internal;
00065      };
00066
00067  inline std::ostream& operator<( std::ostream& os, const CodeString& str)
00068  {
00069      os << str.Internal;
00070      return os;
00071  }
00072
00073  inline bool operator==(const CodeString &ref, const CodeString& cs)
00074  {
00075      return ref.Internal == cs.Internal;
00076  }
00077
00078  inline bool operator!=(const CodeString &ref, const CodeString& cs)

```

```
00098 {
00099     return ref.Internal != cs.Internal;
00100 }
00101
00102
00103 } // end namespace gdcms
00104
00105 #endif //GDCMCODESTRING_H
```

## 11.121 gdcmCP246ExplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmCP246ExplicitDataElement.hxx"
Include dependency graph for gdcmCP246ExplicitDataElement.h:
```



## Classes

- class `gdcm::CP246ExplicitDataElement`  
*Class to read/write a [DataElement](#) as CP246Explicit Data [Element](#).*

## Namespaces

- namespace **gdcm**

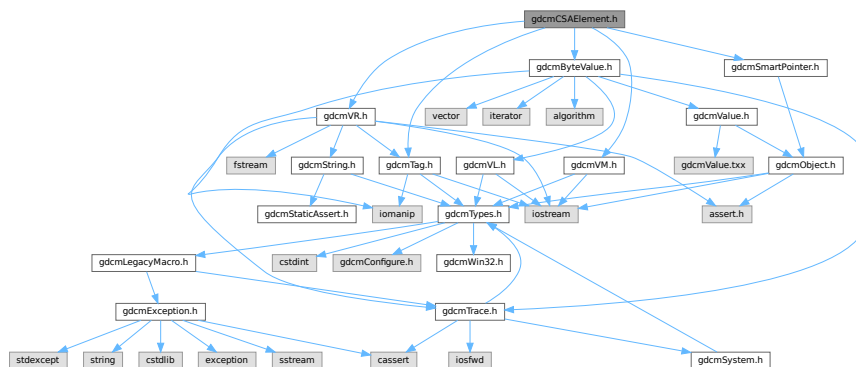
11.122 **gdcmCP246ExplicitDataElement.h**

[Go to the documentation of this file.](#)

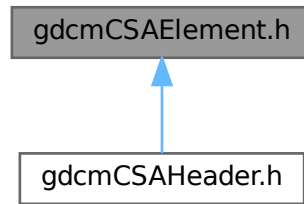
```
00001 /*****
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
```

## 11.123 gdcmsCSAElement.h File Reference

Include dependency graph for `gdcmsCAElement.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::CSAElement](#)  
*Class to represent a CSA [Element](#).*

## Namespaces

- namespace [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const CSAElement &val)`

## 11.124 gdcmCSAElement.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMCSAELEMENT_H
00015  #define GDCMCSAELEMENT_H
00016
00017  #include "gdcmTag.h"
00018  #include "gdcmVM.h"
00019  #include "gdcmVR.h"
00020  #include "gdcmByteValue.h"
00021  #include "gdcmSmartPointer.h"
00022

```

```

00023 namespace gdcm
00024 {
00029 class GDCM_EXPORT CSAElement
00030 {
00031 public:
00032     CSAElement(unsigned int kf = 0):KeyField(kf) {}
00033
00034     friend std::ostream& operator<(std::ostream &os, const CSAElement &val);
00035
00037     unsigned int GetKey() const { return KeyField; }
00038     void SetKey(unsigned int key) { KeyField = key; }
00039
00041     const char *GetName() const { return NameField.c_str(); }
00042     void SetName(const char *name) { NameField = name; }
00043
00045     const VM& GetVM() const { return ValueMultiplicityField; }
00046     void SetVM(const VM &vm) { ValueMultiplicityField = vm; }
00047
00049     VR const &GetVR() const { return VRField; }
00050     void SetVR(VR const &vr) { VRField = vr; }
00051
00053     unsigned int GetSyngoDT() const { return SyngoDTField; }
00054     void SetSyngoDT(unsigned int syngodt) { SyngoDTField = syngodt; }
00055
00057     unsigned int GetNoOfItems() const { return NoOfItemsField; }
00058     void SetNoOfItems(unsigned int items) { NoOfItemsField = items; }
00059
00061     Value const &GetValue() const { return *DataField; }
00062     Value &GetValue() { return *DataField; }
00063     void SetValue(Value const &vl) {
00064         //assert( DataField == 0 );
00065         DataField = vl;
00066     }
00068     bool IsEmpty() const { return DataField == nullptr; }
00069
00071     void SetByteValue(const char *array, VL length)
00072     {
00073         ByteValue *bv = new ByteValue(array,length);
00074         SetValue( *bv );
00075     }
00078     const ByteValue* GetByteValue() const {
00079         // Get the raw pointer from the gdcm::SmartPointer
00080         const ByteValue *bv = dynamic_cast<const ByteValue*>(DataField.GetPointer());
00081         return bv; // Will return NULL if not ByteValue
00082     }
00083
00084     CSAElement(const CSAElement &_val)
00085     {
00086         if( this != &_val)
00087         {
00088             *this = _val;
00089         }
00090     }
00091
00092     bool operator<(const CSAElement &de) const
00093     {
00094         return GetKey() < de.GetKey();
00095     }
00096     CSAElement &operator=(const CSAElement &de)
00097     = default;
00098
00099     bool operator==(const CSAElement &de) const
00100     {
00101         return KeyField == de.KeyField
00102             && NameField == de.NameField
00103             && ValueMultiplicityField == de.ValueMultiplicityField
00104             && VRField == de.VRField
00105             && SyngoDTField == de.SyngoDTField
00106             //&& ValueField == de.ValueField;
00107         ;
00108     }
00109
00110 protected:
00111     unsigned int KeyField;
00112     std::string NameField;
00113     VM ValueMultiplicityField;
00114     VR VRField;
00115     unsigned int SyngoDTField;
00116     unsigned int NoOfItemsField;
00117     typedef SmartPointer<Value> DataPtr;
00118     DataPtr DataField;

```

```

00119 };
00120 //-----
00121 inline std::ostream& operator<<(std::ostream &os, const CSAElement &val)
00122 {
00123     os << val.KeyField;
00124     os << " - '" << val.NameField;
00125     os << "' VM " << val.ValueMultiplicityField;
00126     os << ", VR " << val.VRField;
00127     os << ", SyngoDT " << val.SyngoDTField;
00128     os << ", NoOfItems " << val.NoOfItemsField;
00129     os << ", Data ";
00130     if( val.DataField )
00131     {
00132         //val.DataField->Print( os << "' " );
00133         const ByteValue * bv = dynamic_cast<ByteValue*>(&*val.DataField);
00134         assert( bv );
00135         const char * p = bv->GetPointer();
00136         std::string str(p, p + bv->GetLength() );
00137         if( val.ValueMultiplicityField == VM::VM1 )
00138         {
00139             os << "' " << str.c_str() << "' ";
00140         }
00141         else
00142         {
00143             std::istringstream is( str );
00144             std::string s;
00145             bool sep = false;
00146             while( std::getline(is, s, '\\') )
00147             {
00148                 if( sep )
00149                 {
00150                     os << '\\';
00151                 }
00152                 sep = true;
00153                 os << "' " << s.c_str() << "' ";
00154             }
00155             //bv->Print( os << "' " );
00156             //os << "' ";
00157         }
00158     }
00159     return os;
00160 }
00161
00162 } // end namespace gdcm
00163
00164 #endif //GDCMCSAELEMENT_H

```

## 11.125 gdcmCSAHeader.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmDataSet.h"
#include "gdcmCSAElement.h"
#include "gdcmMrProtocol.h"

```





```

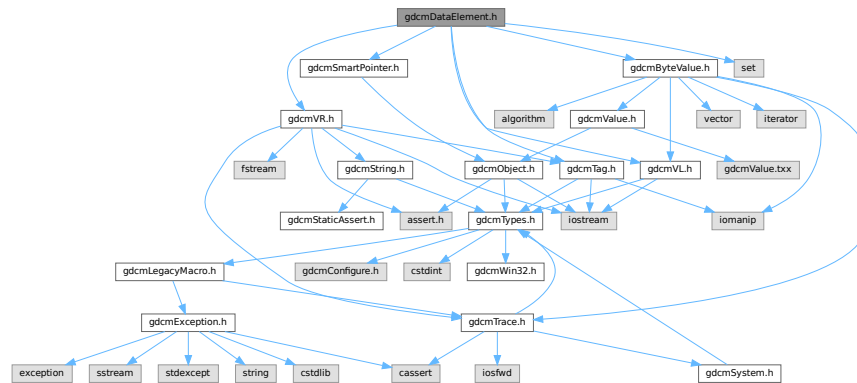
00022 namespace gdcmm
00023 {
00024 /*
00025  * Everything done in this code is for the sole purpose of writing interoperable
00026  * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
00027  * If you believe anything in this code violates any law or any of your rights,
00028  * please contact us (gdcmm-developers@lists.sourceforge.net) so that we can
00029  * find a solution.
00030  */
00031 //-----
00032
00033 class DataElement;
00034 class PrivateTag;
00063 class GDCM_EXPORT CSAHeader
00064 {
00065     friend std::ostream& operator<<(std::ostream &_os, const CSAHeader &d);
00066 public :
00067     CSAHeader():InternalDataSet(),InternalType(UNKNOWN),InterfileData(nullptr) {}
00068     ~CSAHeader() = default;
00069
00071     typedef enum {
00072         UNKNOWN = 0,
00073         SV10,
00074         NOMAGIC,
00075         DATASET_FORMAT,
00076         INTERFILE,
00077         ZEROED_OUT
00078     } CSAHeaderType;
00079
00081     bool LoadFromDataElement(DataElement const &de);
00082
00084     void Print(std::ostream &os) const;
00085
00087     const DataSet& GetDataSet() const { return InternalDataSet; }
00088
00090     const char * GetInterfile() const { return InterfileData; }
00091
00094     CSAHeaderType GetFormat() const;
00095
00098     static const PrivateTag & GetCSAImageHeaderInfoTag();
00099
00102     static const PrivateTag & GetCSASeriesHeaderInfoTag();
00103
00106     static const PrivateTag & GetCSADataInfo();
00107
00110     const CSAElement &GetCSAElementByName(const char *name);
00111
00114     bool FindCSAElementByName(const char *name);
00115
00117     bool GetMrProtocol(const DataSet &ds, MrProtocol &mrProtocol);
00118
00119 protected:
00120     const CSAElement& GetCSAEEnd() const;
00121
00122 private:
00123     std::set<CSAElement> InternalCSADataSet;
00124     DataSet InternalDataSet;
00125     CSAHeaderType InternalType;
00126     Tag DataElementTag;
00127     static CSAElement CSAEEnd;
00128     const char *InterfileData;
00129 };
00130 //-----
00131 inline std::ostream& operator<<(std::ostream &os, const CSAHeader &d)
00132 {
00133     d.Print(os);
00134     return os;
00135 }
00136
00137 } // end namespace gdcmm
00138 //-----
00139 #endif //GDCMCSAHEADER_H

```

## 11.127 gdcmDataElement.h File Reference

```
#include "gdcmTag.h"
#include "gdcmVL.h"
#include "gdcmVR.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"
#include <set>
```

Include dependency graph for gdcmDataElement.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [gdcm::DataElement](#)  
*Class to represent a Data [Element](#) either Implicit or Explicit.*

### Namespaces

- namespace [gdcm](#)

### Functions

- bool [gdcm::operator!=](#) (const [DataElement](#) &lhs, const [DataElement](#) &rhs)
- std::ostream & [gdcm::operator<<](#) (std::ostream &os, const [DataElement](#) &val)

## 11.128 gdcmDataElement.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMDATAELEMENT_H
00015  #define GDCMDATAELEMENT_H
00016
00017  #include "gdcmTag.h"
00018  #include "gdcmVL.h"
00019  #include "gdcmVR.h"
00020  #include "gdcmByteValue.h"
00021  #include "gdcmSmartPointer.h"
00022
00023  #include <set>
00024
00025  namespace gdcm_ns
00026  {
00027    // Data Element
00028    // Contains multiple fields:
00029    // -> Tag
00030    // -> Optional VR (Explicit Transfer Syntax)
00031    // -> ValueLength
00032    // -> Value
00033    // TODO: This class SHOULD be pure virtual. I don't want a user
00034    // to shoot himself in the foot.
00035
00036    class SequenceOfItems;
00037    class SequenceOfFragments;
00038    class GDCM_EXPORT DataElement
00039    {
00040    public:
00041      DataElement(const Tag& t = Tag(0), const VL& vl = 0, const VR &vr =
VR::INVALID):TagField(t),ValueLengthField(vl),VRField(vr),ValueField(nullptr) {}
00042      //DataElement( Attribute const &att );
00043
00044      friend std::ostream& operator<<(std::ostream &_os, const DataElement &_val);
00045
00046      const Tag& GetTag() const { return TagField; }
00047      Tag& GetTag() { return TagField; }
00048      void SetTag(const Tag &t) { TagField = t; }
00049
00050      const VL& GetVL() const { return ValueLengthField; }
00051      VL& GetVL() { return ValueLengthField; }
00052      void SetVL(const VL &vl) { ValueLengthField = vl; }
00053      void SetVLToUndefined();
00054
00055      VR const &GetVR() const { return VRField; }
00056      void SetVR(VR const &vr) {
00057        if( vr.IsVRFile() )
00058          VRField = vr;
00059      }
00060
00061      Value const &GetValue() const { gdcmAssertAlwaysMacro(ValueField); return *ValueField; }
00062      Value &GetValue() {
00063        gdcmAssertAlwaysMacro(ValueField);
00064        return *ValueField;
00065      }
00066      void SetValue(Value const & vl) {
00067        //assert( ValueField == 0 );
00068        ValueField = vl;
00069        ValueLengthField = vl.GetLength();
00070      }
00071      bool IsEmpty() const { return ValueField == nullptr || (GetByteValue() && GetByteValue()->IsEmpty()); }
00072
00073      void Empty() { ValueField = nullptr; ValueLengthField = 0; }
00074
00075      void Clear()

```

```

00113     {
00114         TagField = 0;
00115         VRField = VR::INVALID;
00116         ValueField = nullptr;
00117         ValueLengthField = 0;
00118     }
00119
00120     // Helper:
00121 void SetByteValue(const char *array, VL length)
00122     {
00123         ByteValue *bv = new ByteValue(array, length);
00124         SetValue( *bv );
00125     }
00126
00127 const ByteValue* GetByteValue() const {
00128     // Get the raw pointer from the gdcm::SmartPointer
00129     const ByteValue *bv = dynamic_cast<const ByteValue*>(ValueField.GetPointer());
00130     return bv; // Will return NULL if not ByteValue
00131 }
00132
00133 SmartPointer<SequenceOfItems> GetValueAsSQ() const;
00134
00135 const SequenceOfFragments* GetSequenceOfFragments() const;
00136 SequenceOfFragments* GetSequenceOfFragments();
00137
00138 bool IsUndefinedLength() const {
00139     return ValueLengthField.IsUndefined();
00140 }
00141
00142 DataElement(const DataElement &_val)
00143     {
00144         if( this != &_amp;_val)
00145         {
00146             *this = _val;
00147         }
00148     }
00149
00150 bool operator<(const DataElement &de) const
00151     {
00152         return GetTag() < de.GetTag();
00153     }
00154
00155 DataElement &operator=(const DataElement &)
00156     = default;
00157
00158 bool operator==(const DataElement &de) const
00159     {
00160         bool b = TagField == de.TagField
00161             && ValueLengthField == de.ValueLengthField
00162             && VRField == de.VRField;
00163         if( !ValueField && !de.ValueField )
00164         {
00165             return b;
00166         }
00167         if( ValueField && de.ValueField )
00168         {
00169             return b && (*ValueField == *de.ValueField);
00170         }
00171         // ValueField != de.ValueField
00172         return false;
00173     }
00174
00175 // The following functionalities are dependent on:
00176 // # The Transfer Syntax: Explicit or Implicit
00177 // # The Byte encoding: Little Endian / Big Endian
00178
00179 /*
00180 * The following was inspired by a C++ idiom: Curiously Recurring Template Pattern
00181 * Ref: http://en.wikipedia.org/wiki/Curiously\_Recurring\_Template\_Pattern
00182 * The typename TDE is typically a derived class *without* any data
00183 * while TSwap is a simple template parameter to achieve byteswapping (and allow factorization of
00184 * highly identical code)
00185 */
00186 template <typename TDE>
00187 VL GetLength() const {
00188     return static_cast<const TDE*>(this)->GetLength();
00189 }
00190
00191 template <typename TDE, typename TSwap>
00192 std::istream &Read(std::istream &is) {
00193     return static_cast<TDE*>(this)->template Read<TSwap>(is);
00194 }
00195
00196
00197
00198
00199

```

```

00210     template <typename TDE, typename TSwap>
00211     std::istream &ReadOrSkip(std::istream &is, std::set<Tag> const &skiptags) {
00212         (void)skiptags;
00213         return static_cast<TDE*>(this)->template Read<TSwap>(is);
00214     }
00215
00216     template <typename TDE, typename TSwap>
00217     std::istream &ReadPreValue(std::istream &is, std::set<Tag> const &skiptags) {
00218         (void)skiptags;
00219         return static_cast<TDE*>(this)->template ReadPreValue<TSwap>(is);
00220     }
00221     template <typename TDE, typename TSwap>
00222     std::istream &ReadValue(std::istream &is, std::set<Tag> const &skiptags) {
00223         (void)skiptags;
00224         return static_cast<TDE*>(this)->template ReadValue<TSwap>(is);
00225     }
00226     template <typename TDE, typename TSwap>
00227     std::istream &ReadValueWithLength(std::istream &is, VL &length, std::set<Tag> const &skiptags) {
00228         (void)skiptags;
00229         return static_cast<TDE*>(this)->template ReadValueWithLength<TSwap>(is, length);
00230     }
00231
00232     template <typename TDE, typename TSwap>
00233     std::istream &ReadWithLength(std::istream &is, VL &length) {
00234         return static_cast<TDE*>(this)->template ReadWithLength<TSwap>(is, length);
00235     }
00236
00237     template <typename TDE, typename TSwap>
00238     const std::ostream &Write(std::ostream &os) const {
00239         return static_cast<const TDE*>(this)->template Write<TSwap>(os);
00240     }
00241
00242 protected:
00243     Tag TagField;
00244     // This is the value read from the file, might be different from the length of Value Field
00245     VL ValueLengthField; // Can be 0xFFFFFFFF
00246
00247     // Value Representation
00248     VR VRField;
00249     typedef SmartPointer<Value> ValuePtr;
00250     ValuePtr ValueField;
00251
00252     void SetValueFieldLength( VL vl, bool readvalues );
00253 };
00254 //-----
00255 inline std::ostream& operator<<(std::ostream &os, const DataElement &val)
00256 {
00257     os << val.TagField;
00258     os << "\t" << val.VRField;
00259     os << "\t" << val.ValueLengthField;
00260     if( val.ValueField )
00261     {
00262         val.ValueField->Print( os << "\t" );
00263     }
00264     return os;
00265 }
00266
00267 inline bool operator!=(const DataElement& lhs, const DataElement& rhs)
00268 {
00269     return ! ( lhs == rhs );
00270 }
00271
00272 } // end namespace gdcms_ns
00273
00274 #endif //GDCMDATAELEMENT_H

```

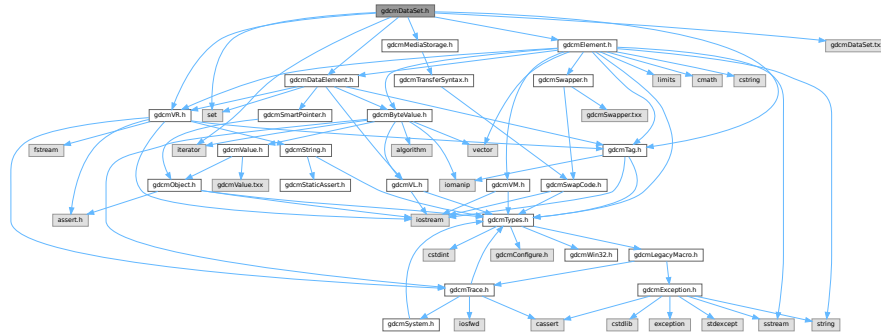
## 11.129 gdcmsDataSet.h File Reference

```

#include "gdcmsDataElement.h"
#include "gdcmsTag.h"
#include "gdcmsVR.h"
#include "gdcmsElement.h"
#include "gdcmsMediaStorage.h"

```

```
#include <set>
#include <iterator>
#include "gdcmDataSet.txx"
Include dependency graph for gdcmDataSet.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::DataElementException](#)
- class [gdcm::DataSet](#)

*Class to represent a Data Set (which contains Data Elements)*

## Namespaces

- namespace [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const DataSet &val)`

## 11.130 gdcmDataSet.h

[Go to the documentation of this file.](#)

```
00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
```

```

00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 #ifndef GDCMDATASET_H
00015 #define GDCMDATASET_H
00016
00017 #include "gdcmDataElement.h"
00018 #include "gdcmTag.h"
00019 #include "gdcmVR.h"
00020 #include "gdcmElement.h"
00021 #include "gdcmMediaStorage.h"
00022
00023 #include <set>
00024 #include <iterator>
00025
00026 namespace gdcm_ns
00027 {
00028     class GDCM_EXPORT DataElementException : public std::exception {};
00029
00030     class PrivateTag;
00031     class GDCM_EXPORT DataSet
00032     {
00033     public:
00034         typedef std::set<DataElement> DataElementSet;
00035         typedef DataElementSet::const_iterator ConstIterator;
00036         typedef DataElementSet::iterator Iterator;
00037         typedef DataElementSet::size_type SizeType;
00038         //typedef typename DataElementSet::iterator iterator;
00039         ConstIterator Begin() const { return DES.begin(); }
00040         Iterator Begin() { return DES.begin(); }
00041         ConstIterator End() const { return DES.end(); }
00042         Iterator End() { return DES.end(); }
00043         const DataElementSet &GetDES() const { return DES; }
00044         DataElementSet &GetDES() { return DES; }
00045         void Clear() {
00046             DES.clear();
00047             assert( DES.empty() );
00048         }
00049         SizeType Size() const {
00050             return DES.size();
00051         }
00052
00053         void Print(std::ostream &os, std::string const &indent = "") const {
00054             // CT_Phillips_JPEG2K_Decompr_Problem.dcm has a SQ of length == 0
00055             //int s = DES.size();
00056             //assert( s );
00057             //std::copy(DES.begin(), DES.end(),
00058             //    std::ostream_iterator<DataElement>(os, "\n"));
00059             ConstIterator it = DES.begin();
00060             for( ; it != DES.end(); ++it)
00061             {
00062                 os << indent << *it << "\n";
00063             }
00064         }
00065
00066     template <typename TDE>
00067     unsigned int ComputeGroupLength(Tag const &tag) const
00068     {
00069         assert( tag.GetElement() != 0x0 );
00070         const DataElement r(tag);
00071         ConstIterator it = DES.find(r);
00072         unsigned int res = 0;
00073         for( ++it; it != DES.end()
00074             && it->GetTag().GetGroup() == tag.GetGroup(); ++it)
00075         {
00076             assert( it->GetTag().GetElement() != 0x0 );
00077             assert( it->GetTag().GetGroup() == tag.GetGroup() );
00078             res += it->GetLength<TDE>();
00079         }
00080         return res;
00081     }
00082
00083     template <typename TDE>
00084     VL GetLength() const {
00085         if( DES.empty() ) return 0;
00086         assert( !DES.empty() );

```



```

00113     VL l1 = 0;
00114     assert( l1 == 0 );
00115     ConstIterator it = DES.begin();
00116     for( ; it != DES.end(); ++it)
00117     {
00118         assert( !(it->GetLength<TDE>().IsUndefined()) );
00119         if ( it->GetTag() != Tag(0xfffe,0xe00d) )
00120         {
00121             l1 += it->GetLength<TDE>();
00122         }
00123     }
00124     return l1;
00125 }
00126 void Insert(const DataElement& de) {
00127     // FIXME: there is a special case where a dataset can have value < 0x8, see:
00128     // $ gdcmDump --csa gdcmData/SIEMENS-JPEG-CorruptFrag.dcm
00129     if( de.GetTag().GetGroup() >= 0x0008 || de.GetTag().GetGroup() == 0x4 )
00130     {
00131         // prevent user error:
00132         if( de.GetTag() == Tag(0xfffe,0xe00d)
00133             || de.GetTag() == Tag(0xfffe,0xe0dd)
00134             || de.GetTag() == Tag(0xfffe,0xe000) )
00135         {
00136             // do nothing
00137         }
00138         else
00139         {
00140             InsertDataElement( de );
00141         }
00142     }
00143     else
00144     {
00145         gdcmErrorMacro( "Cannot add element with group < 0x0008 and != 0x4 in the dataset: " « de.GetTag()
00146     );
00147     }
00148 }
00149 void Replace(const DataElement& de) {
00150     ConstIterator it = DES.find(de);
00151     if( it != DES.end() )
00152     {
00153         // detect loop:
00154         gdcmAssertAlwaysMacro( &*it != &de );
00155         DES.erase(it);
00156     }
00157     DES.insert(de);
00158 }
00159 void ReplaceEmpty(const DataElement& de) {
00160     ConstIterator it = DES.find(de);
00161     if( it != DES.end() && it->IsEmpty() )
00162     {
00163         // detect loop:
00164         gdcmAssertAlwaysMacro( &*it != &de );
00165         DES.erase(it);
00166     }
00167     DES.insert(de);
00168 }
00169 void Remove(const Tag& tag) {
00170     DataElementSet::size_type count = DES.erase(tag);
00171     assert( count == 0 || count == 1 );
00172     return count;
00173 }
00174 //DataElement& GetDataElement(const Tag &t) {
00175 //    DataElement r(t);
00176 //    Iterator it = DES.find(r);
00177 //    if( it != DES.end() )
00178 //        return *it;
00179 //    return GetDEEnd();
00180 // }
00181 const DataElement& GetDataElement(const Tag &t) const {
00182     const DataElement r(t);
00183     ConstIterator it = DES.find(r);
00184     if( it != DES.end() )
00185         return *it;
00186     return GetDEEnd();
00187 }
00188 const DataElement& operator[] (const Tag &t) const { return GetDataElement(t); }
00189 const DataElement& operator() (uint16_t group, uint16_t element) const { return GetDataElement(
00190     Tag(group,element) ); }
00191 std::string GetPrivateCreator(const Tag &t) const;
00192

```

```

00203 PrivateTag GetPrivateTag(const Tag &t) const;
00204
00206 bool FindDataElement(const PrivateTag &t) const;
00208 const DataElement& GetDataElement(const PrivateTag &t) const;
00209
00210 // DUMB: this only search within the level of the current DataSet
00211 bool FindDataElement(const Tag &t) const {
00212     const auto it = GetDataElement(t);
00213     // Return if tag is found
00214     return it != GetDEEnd();
00215 }
00216
00217 // WARNING:
00218 // This only search at the same level as the DataSet is !
00219 const DataElement& FindNextDataElement(const Tag &t) const {
00220     const DataElement r(t);
00221     ConstIterator it = DES.lower_bound(r);
00222     if( it != DES.end() )
00223         return *it;
00224     return GetDEEnd();
00225 }
00226
00228 bool IsEmpty() const { return DES.empty(); }
00229
00230 DataSet& operator=(DataSet const &)
00231 = default;
00232
00233 template <typename TDE, typename TSwap>
00234 std::istream &ReadNested(std::istream &is);
00235
00236 template <typename TDE, typename TSwap>
00237 std::istream &Read(std::istream &is);
00238
00239 template <typename TDE, typename TSwap>
00240 std::istream &ReadUpToTag(std::istream &is, const Tag &t, std::set<Tag> const &skiptags);
00241
00242 template <typename TDE, typename TSwap>
00243 std::istream &ReadUpToTagWithLength(std::istream &is, const Tag &t, std::set<Tag> const &skiptags, VL &
length);
00244
00245 template <typename TDE, typename TSwap>
00246 std::istream &ReadSelectedTags(std::istream &is, const std::set<Tag> &tags, bool readvalues = true);
00247 template <typename TDE, typename TSwap>
00248 std::istream &ReadSelectedTagsWithLength(std::istream &is, const std::set<Tag> &tags, VL &length, bool
readvalues = true);
00249
00250 template <typename TDE, typename TSwap>
00251 std::istream &ReadSelectedPrivateTags(std::istream &is, const std::set<PrivateTag> &tags, bool
readvalues = true);
00252 template <typename TDE, typename TSwap>
00253 std::istream &ReadSelectedPrivateTagsWithLength(std::istream &is, const std::set<PrivateTag> &tags, VL
&length, bool readvalues = true);
00254
00255 template <typename TDE, typename TSwap>
00256 std::ostream const &Write(std::ostream &os) const;
00257
00258 template <typename TDE, typename TSwap>
00259 std::istream &ReadWithLength(std::istream &is, VL &length);
00260
00261 MediaStorage GetMediaStorage() const;
00262
00263 protected:
00264     /* GetDEEnd is a Win32 only issue, one cannot use a dllexport
00265     * static member data in an inline function, otherwise symbol
00266     * will get reported as missing in any dll using the inlined function
00267     */
00268     const DataElement& GetDEEnd() const;
00269
00270     // This function is not safe, it does not check for the value of the tag
00271     // so depending whether we are getting called from a dataset or file meta header
00272     // the condition is different
00273     void InsertDataElement(const DataElement& de) {
00274         //if( de.GetTag() == Tag(0xffff,0xe00d) ) return;
00275         //if( de.GetTag() == Tag(0xffff,0xe0dd) ) return;
00276 #ifndef NDEBUG
00277         std::pair<Iterator,bool> pr = DES.insert(de);
00278         if( pr.second == false )
00279         {
00280             gdcWarningMacro( "DataElement: " << de << " was already found, skipping duplicate entry.\n"
00281             "Original entry kept is: " << *pr.first );
00282         }

```

```

00283 #else
00284     DES.insert(de);
00285 #endif
00286     assert( de.IsEmpty() || de.GetVL() == de.GetValue().GetLength() );
00287 }
00288
00289 protected:
00290     // Internal function, that will compute the actual Tag (if found) of
00291     // a requested Private Tag (XXXX,YY,"PRIVATE")
00292     Tag ComputeDataElement(const PrivateTag & t) const;
00293
00294 private:
00295     DataElementSet DES;
00296     static DataElement DEEnd;
00297     friend std::ostream& operator<<(std::ostream &_os, const DataSet &);
00298 };
00299 //-----
00300 inline std::ostream& operator<<(std::ostream &os, const DataSet &val)
00301 {
00302     val.Print(os);
00303     return os;
00304 }
00305
00306 #if defined(SWIGPYTHON) || defined(SWIGCSharp) || defined(SWIGJAVA) || defined(SWIGPHP)
00307 /*
00308  * HACK: I need this temp class to be able to manipulate a std::set from python,
00309  * swig does not support wrapping of simple class like std::set...
00310  */
00311 class SWIGDataSet
00312 {
00313 public:
00314     SWIGDataSet(DataSet &des):Internal(des),it(des.Begin()) {}
00315     const DataElement& GetCurrent() const { return *it; }
00316     void Start() { it = Internal.Begin(); }
00317     bool IsAtEnd() const { return it == Internal.End(); }
00318     void Next() { ++it; }
00319 private:
00320     DataSet & Internal;
00321     DataSet::ConstIterator it;
00322 };
00323 #endif /* SWIG */
00324
00330 } // end namespace gdcm_ns
00331
00332 #include "gdcmDataSet.txx"
00333
00334 #endif //GDCMDATASET_H

```

## 11.131 gdcmDataSetEvent.h File Reference

```

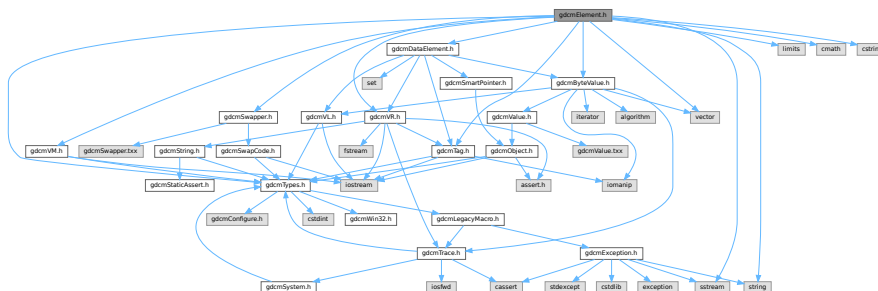
#include "gdcmEvent.h"
#include "gdcmDataSet.h"

```



### 11.133 gdcmElement.h File Reference

Include dependency graph for `gdcmElement.h`:



## Classes

- class [gdcm::Element< TVR, TVM >](#)  
*Element class.*
- class [gdcm::Element< TVR, VM::VM1\\_2 >](#)
- class [gdcm::Element< TVR, VM::VM1\\_n >](#)
- class [gdcm::Element< TVR, VM::VM2\\_2n >](#)
- class [gdcm::Element< TVR, VM::VM2\\_n >](#)
- class [gdcm::Element< TVR, VM::VM3\\_3n >](#)
- class [gdcm::Element< TVR, VM::VM3\\_4 >](#)
- class [gdcm::Element< TVR, VM::VM3\\_n >](#)
- class [gdcm::Element< VR::AS, VM::VM5 >](#)
- class [gdcm::Element< VR::OB, VM::VM1 >](#)
- class [gdcm::Element< VR::OW, VM::VM1 >](#)
- class [gdcm::ElementDisableCombinations< TVR, TVM >](#)  
*A class which is used to produce compile errors for an invalid combination of template parameters.*
- class [gdcm::ElementDisableCombinations< VR::OB, VM::VM1\\_n >](#)
- class [gdcm::ElementDisableCombinations< VR::OW, VM::VM1\\_n >](#)
- class [gdcm::EncodingImplementation< VR::VRASCII >](#)
- class [gdcm::EncodingImplementation< VR::VRBINARY >](#)
- struct [gdcm::ignore\\_char](#)

## Namespaces

- namespace [gdcm](#)

## Functions

- static int [gdcm::add1](#) (char \*buf, int n)
- [ignore\\_char](#) const [gdcm::backslash](#) ("\\")
- static void [gdcm::clean](#) (char \*mant)
- static int [gdcm::doround](#) (char \*buf, unsigned int n)
- std::istream & [gdcm::operator>>](#) (std::istream &in, [ignore\\_char](#) const &ic)
- static int [gdcm::roundat](#) (char \*buf, size\_t bufLen, unsigned int i, int iexp)
- template<typename Float >  
static void [gdcm::x16printf](#) (char \*buf, int size, Float f)

## 11.134 gdcmElement.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
```

```

00012
00013 =====*/
00014 #ifndef GDCMELEMENT_H
00015 #define GDCMELEMENT_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmVR.h"
00019 #include "gdcmTag.h"
00020 #include "gdcmVM.h"
00021 #include "gdcmByteValue.h"
00022 #include "gdcmDataElement.h"
00023 #include "gdcmSwapper.h"
00024
00025 #include <string>
00026 #include <vector>
00027 #include <sstream>
00028 #include <limits>
00029 #include <cmath>
00030 #include <cstring>
00031
00032 namespace gdcm_ns
00033 {
00034
00035 // Forward declaration
00041 template<long long T> class EncodingImplementation;
00042
00043
00051 template <long long TVR, int TVM>
00052 class ElementDisableCombinations {};
00053 template <>
00054 class ElementDisableCombinations<VR::OB, VM::VM1_n> {};
00055 template <>
00056 class ElementDisableCombinations<VR::OW, VM::VM1_n> {};
00057 // Make it impossible to compile these other cases
00058 template <int TVM>
00059 class ElementDisableCombinations<VR::OB, TVM>;
00060 template <int TVM>
00061 class ElementDisableCombinations<VR::OW, TVM>;
00062
00068 template<long long TVR, int TVM>
00069 class Element
00070 {
00071     enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, TVM> ) };
00072 public:
00073     typename VRToType<TVR>::Type Internal[VMToLength<TVM>::Length];
00074     typedef typename VRToType<TVR>::Type Type;
00075
00076     static VR GetVR() { return (VR::VRType)TVR; }
00077     static VM GetVM() { return (VM::VMType)TVM; }
00078
00079     unsigned long GetLength() const {
00080         return VMToLength<TVM>::Length;
00081     }
00082     // Implementation of Print is common to all Mode (ASCII/Binary)
00083     // TODO: Can we print a \ when in ASCII...well I don't think so
00084     // it would mean we used a bad VM then, right?
00085     void Print(std::ostream &_os) const {
00086         _os << Internal[0]; // VM is at least guarantee to be one
00087         for(int i=1; i<VMToLength<TVM>::Length; ++i)
00088             _os << ", " << Internal[i];
00089     }
00090
00091     const typename VRToType<TVR>::Type *GetValues() const {
00092         return Internal;
00093     }
00094     const typename VRToType<TVR>::Type &GetValue(unsigned int idx = 0) const {
00095         assert( idx < VMToLength<TVM>::Length );
00096         return Internal[idx];
00097     }
00098     typename VRToType<TVR>::Type &GetValue(unsigned int idx = 0) {
00099         assert( idx < VMToLength<TVM>::Length );
00100         return Internal[idx];
00101     }
00102     typename VRToType<TVR>::Type operator[] (unsigned int idx) const {
00103         return GetValue(idx);
00104     }
00105     void SetValue(typename VRToType<TVR>::Type v, unsigned int idx = 0) {
00106         assert( idx < VMToLength<TVM>::Length );
00107         Internal[idx] = v;
00108     }
00109 }

```

```

00110 void SetFromDataElement(DataElement const &de) {
00111     const ByteValue *bv = de.GetByteValue();
00112     if( !bv ) return;
00113 #ifdef GDCM_WORDS_BIGENDIAN
00114     if( de.GetVR() == VR::UN /*|| de.GetVR() == VR::INVALID*/ )
00115 #else
00116     if( de.GetVR() == VR::UN || de.GetVR() == VR::INVALID )
00117 #endif
00118     {
00119         Set(de.GetValue());
00120     }
00121     else
00122     {
00123         SetNoSwap(de.GetValue());
00124     }
00125 }
00126
00127 DataElement GetAsDataElement() const {
00128     DataElement ret;
00129     std::ostringstream os;
00130     EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
00131         GetLength(),os);
00132     ret.SetVR( (VR::VRType)TVR );
00133     assert( ret.GetVR() != VR::SQ );
00134     if( (VR::VRType)VRToEncoding<TVR>::Mode == VR::VRASCII )
00135     {
00136         if( GetVR() != VR::UI )
00137         {
00138             if( os.str().size() % 2 )
00139             {
00140                 os << " ";
00141             }
00142         }
00143     }
00144     VL::Type osStrSize = (VL::Type)os.str().size();
00145     ret.SetByteValue( os.str().c_str(), osStrSize );
00146     return ret;
00147 }
00148
00149 void Read(std::istream &_is) {
00150     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
00151         GetLength(),_is);
00152 }
00153
00154 void Write(std::ostream &_os) const {
00155     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
00156         GetLength(),_os);
00157 }
00158
00159 // FIXME: remove this function
00160 // this is only used in gdcm::SplitMosaicFilter / to pass value of a CSAElement
00161 void Set(Value const &v) {
00162     const ByteValue *bv = dynamic_cast<const ByteValue*>(&v);
00163     if( bv ) {
00164         //memcpy(Internal, bv->GetPointer(), bv->GetLength());
00165         std::stringstream ss;
00166         std::string s = std::string( bv->GetPointer(), bv->GetLength() );
00167         ss.str( s );
00168         EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
00169             GetLength(),ss);
00170     }
00171 }
00172 protected:
00173 void SetNoSwap(Value const &v) {
00174     const ByteValue *bv = dynamic_cast<const ByteValue*>(&v);
00175     assert( bv ); // That would be bad...
00176     //memcpy(Internal, bv->GetPointer(), bv->GetLength());
00177     std::stringstream ss;
00178     std::string s = std::string( bv->GetPointer(), bv->GetLength() );
00179     ss.str( s );
00180     EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadNoSwap(Internal,
00181         GetLength(),ss);
00182 }
00183 };
00184
00185 struct ignore_char {
00186     ignore_char(char c): m_char(c) {}
00187     char m_char;
00188 };
00189 ignore_char const backslash('\\');
00190

```



```

00191 inline std::istream& operator> (std::istream& in, ignore_char const& ic) {
00192     if (!in.eof())
00193         in.clear(in.rdstate() & ~std::ios_base::failbit);
00194     if (in.get() != ic.m_char)
00195         in.setstate(std::ios_base::failbit);
00196     return in;
00197 }
00198
00199
00200 // Implementation to perform formatted read and write
00201 template<> class EncodingImplementation<VR::VRASCII> {
00202 public:
00203     template<typename T> // FIXME this should be VRToType<TVR>::Type
00204     static inline void ReadComputeLength(T* data, unsigned int &length,
00205                                         std::istream &_is) {
00206         assert( data );
00207         //assert( length ); // != 0
00208         length = 0;
00209         assert( _is );
00210         #if 0
00211             char sep;
00212             while( _is >> data[length++] )
00213             {
00214                 // Get the separator in between the values
00215                 assert( _is );
00216                 _is.get(sep);
00217                 assert( sep == '\\\\' || sep == ' ' ); // FIXME: Bad use of assert
00218                 if( sep == ' ' ) length--; // FIXME
00219             }
00220         #else
00221             while( _is >> std::ws >> data[length++] >> std::ws >> backslash )
00222             {
00223             }
00224         #endif
00225     }
00226
00227     template<typename T> // FIXME this should be VRToType<TVR>::Type
00228     static inline void Read(T* data, unsigned long length,
00229                             std::istream &_is) {
00230         assert( data );
00231         assert( length ); // != 0
00232         assert( _is );
00233         // FIXME BUG: what if >> operation fails ?
00234         // gdcmData/MR00010001.dcm / SpacingBetweenSlices
00235         _is >> std::ws >> data[0];
00236         char sep;
00237         //std::cout << "GetLength: " << af->GetLength() << std::endl;
00238         for(unsigned long i=1; i<length; ++i) {
00239             //assert( _is );
00240             // Get the separator in between the values
00241             _is >> std::ws >> sep; // _is.get(sep);
00242             //assert( sep == '\\\\' ); // FIXME: Bad use of assert
00243             _is >> std::ws >> data[i];
00244         }
00245     }
00246
00247     template<typename T>
00248     static inline void ReadNoSwap(T* data, unsigned long length,
00249                                   std::istream &_is) {
00250         Read(data, length, _is);
00251     }
00252
00253     template<typename T>
00254     static inline void Write(const T* data, unsigned long length,
00255                              std::ostream &_os) {
00256         assert( data );
00257         assert( length );
00258         assert( _os );
00259         _os << data[0];
00260         for(unsigned long i=1; i<length; ++i) {
00261             assert( _os );
00262             _os << "\\\" << data[i];
00263         }
00264     };
00265
00266     // #define VRDS16ILLEGAL
00267     #ifdef VRDS16ILLEGAL
00268     template < typename Float >
00269     std::string to_string ( Float data ) {
00270         std::stringstream in;

```

```

00272 // in.imbue(std::locale::classic()); // This is not required AFAIK
00273 int const digits =
00274     static_cast< int >{
00275         - std::log( std::numeric_limits<Float>::epsilon() )
00276         / static_cast< Float >( std::log( 10.0 ) ) );
00277 if ( in « std::dec « std::setprecision(*2*/digits) « data ) {
00278     return ( in.str() );
00279 } else {
00280     throw "Impossible Conversion"; // should not happen ...
00281 }
00282 }
00283 #else
00284 //
00285     http://stackoverflow.com/questions/32631178/writing-ieee-754-1985-double-as-ascii-on-a-limited-16-bytes-string
00286 static inline void clean(char *mant) {
00287     char *ix = mant + strlen(mant) - 1;
00288     while ((*ix == '0' || *ix == ' ') && (ix > mant)) {
00289         *ix-- = '\0';
00290     }
00291     if (*ix == '.') {
00292         *ix = '\0';
00293     }
00294 }
00295 static int add1(char *buf, int n) {
00296     if (n < 0) return 1;
00297     if (buf[n] == '9') {
00298         buf[n] = '0';
00299         return add1(buf, n-1);
00300     }
00301     else {
00302         buf[n] = (char)(buf[n] + 1);
00303     }
00304     return 0;
00305 }
00306 }
00307 static int doround(char *buf, unsigned int n) {
00308     char c;
00309     if (n >= strlen(buf)) return 0;
00310     c = buf[n];
00311     buf[n] = 0;
00312     if ((c >= '5') && (c <= '9')) return add1(buf, n-1);
00313     return 0;
00314 }
00315 }
00316 #if defined(_MSC_VER) && (_MSC_VER < 1900)
00317 #define snprintf _snprintf
00318 #endif
00319 #endif
00320 static int roundat(char *buf, size_t bufLen, unsigned int i, int iexp) {
00321     if (doround(buf, i) != 0) {
00322         iexp += 1;
00323         switch(iexp) {
00324             case -2:
00325                 strcpy(buf, ".01");
00326                 break;
00327             case -1:
00328                 strcpy(buf, ".1");
00329                 break;
00330             case 0:
00331                 strcpy(buf, "1.");
00332                 break;
00333             case 1:
00334                 strcpy(buf, "10");
00335                 break;
00336             case 2:
00337                 strcpy(buf, "100");
00338                 break;
00339             default:
00340                 snprintf(buf, bufLen, "1e%d", iexp);
00341         }
00342         return 1;
00343     }
00344     return 0;
00345 }
00346 }
00347 template < typename Float >
00348 static void x16printf(char *buf, int size, Float f) {
00349     char line[40];
00350     char *mant = line + 1;

```

```

00352     int iexp, lexp, i;
00353     char exp[6];
00354
00355     if (f < 0) {
00356         f = -f;
00357         size -= 1;
00358         *buf++ = '-';
00359     }
00360     snprintf(line, sizeof(line), "%1.16e", f);
00361     if (line[0] == '-') {
00362         f = -f;
00363         size -= 1;
00364         *buf++ = '-';
00365         snprintf(line, sizeof(line), "%1.16e", f);
00366     }
00367     *mant = line[0];
00368     i = (int)strcspn(mant, "eE");
00369     mant[i] = '\0';
00370     iexp = (int)strtol(mant + i + 1, nullptr, 10);
00371     lexp = snprintf(exp, sizeof(exp), "e%d", iexp);
00372     if ((iexp >= size) || (iexp < -3)) {
00373         i = roundat(mant, sizeof(line) - 1, size - 1 - lexp, iexp);
00374         if (i == 1) {
00375             strcpy(buf, mant);
00376             return;
00377         }
00378         buf[0] = mant[0];
00379         buf[1] = '.';
00380         strncpy(buf + i + 2, mant + 1, size - 2 - lexp);
00381         buf[size - lexp] = 0;
00382         clean(buf);
00383         strcat(buf, exp);
00384     }
00385     else if (iexp >= size - 2) {
00386         roundat(mant, sizeof(line) - 1, iexp + 1, iexp);
00387         strcpy(buf, mant);
00388     }
00389     else if (iexp >= 0) {
00390         i = roundat(mant, sizeof(line) - 1, size - 1, iexp);
00391         if (i == 1) {
00392             strcpy(buf, mant);
00393             return;
00394         }
00395         strncpy(buf, mant, iexp + 1);
00396         buf[iexp + 1] = '.';
00397         strncpy(buf + iexp + 2, mant + iexp + 1, size - iexp - 1);
00398         buf[size] = 0;
00399         clean(buf);
00400     }
00401     else {
00402         int j;
00403         i = roundat(mant, sizeof(line) - 1, size + 1 + iexp, iexp);
00404         if (i == 1) {
00405             strcpy(buf, mant);
00406             return;
00407         }
00408         buf[0] = '.';
00409         for (j=0; j< -1 - iexp; j++) {
00410             buf[j+1] = '0';
00411         }
00412         strncpy(buf - iexp, mant, size + 1 + iexp);
00413         buf[size] = 0;
00414         clean(buf);
00415     }
00416 }
00417 #if defined(_MSC_VER) && (_MSC_VER < 1900)
00418 #undef snprintf
00419 #endif
00420
00421 #endif
00422
00423 template<> inline void EncodingImplementation<VR:VRASCII>::Write(const double* data, unsigned long
length, std::ostream &_os) {
00424     assert( data );
00425     assert( length );
00426     assert( _os );
00427 #ifdef VRDS16ILLEGAL
00428     _os << to_string(data[0]);
00429 #else
00430     char buf[16+1];
00431     x16printf(buf, 16, data[0]);

```

```

00432     _os << buf;
00433 #endif
00434     for(unsigned long i=1; i<length; ++i) {
00435         assert( _os );
00436 #ifdef VRDS16ILLEGAL
00437         _os << "\\\" << to_string(data[i]);
00438 #else
00439         x16printf(buf, 16, data[i]);
00440         _os << "\\\" << buf;
00441 #endif
00442     }
00443 }
00444
00445
00446 // Implementation to perform binary read and write
00447 // TODO rewrite operation so that either:
00448 // #1. dummy implementation use a pointer to Internal and do ++p (faster)
00449 // #2. Actually do some meta programming to unroll the loop
00450 // (no notion of order in VM ...)
00451 template<> class EncodingImplementation<VR::VRBINARY> {
00452 public:
00453     template<typename T> // FIXME this should be VRToType<TVR>::Type
00454         static inline void ReadComputeLength(T* data, unsigned int &length,
00455             std::istream &_is) {
00456             const unsigned int type_size = sizeof(T);
00457             assert( data ); // Can we read from pointer ?
00458             //assert( length );
00459             length /= type_size;
00460             assert( _is ); // Is stream valid ?
00461             _is.read( reinterpret_cast<char*>(data+0), type_size);
00462             for(unsigned long i=1; i<length; ++i) {
00463                 assert( _is );
00464                 _is.read( reinterpret_cast<char*>(data+i), type_size );
00465             }
00466         }
00467     template<typename T>
00468     static inline void ReadNoSwap(T* data, unsigned long length,
00469         std::istream &_is) {
00470         const unsigned int type_size = sizeof(T);
00471         assert( data ); // Can we read from pointer ?
00472         assert( length );
00473         assert( _is ); // Is stream valid ?
00474         _is.read( reinterpret_cast<char*>(data+0), type_size);
00475         for(unsigned long i=1; i<length; ++i) {
00476             if( _is )
00477                 _is.read( reinterpret_cast<char*>(data+i), type_size );
00478         }
00479         //ByteSwap<T>::SwapRangeFromSwapCodeIntoSystem(data,
00480             // _is.GetSwapCode(), length);
00481         //SwapperNoOp::SwapArray(data,length);
00482     }
00483     template<typename T>
00484     static inline void Read(T* data, unsigned long length,
00485         std::istream &_is) {
00486         const unsigned int type_size = sizeof(T);
00487         assert( data ); // Can we read from pointer ?
00488         assert( length );
00489         assert( _is ); // Is stream valid ?
00490         _is.read( reinterpret_cast<char*>(data+0), type_size);
00491         for(unsigned long i=1; i<length; ++i) {
00492             if( _is )
00493                 _is.read( reinterpret_cast<char*>(data+i), type_size );
00494         }
00495         //ByteSwap<T>::SwapRangeFromSwapCodeIntoSystem(data,
00496             // _is.GetSwapCode(), length);
00497         SwapperNoOp::SwapArray(data,length);
00498     }
00499     template<typename T>
00500     static inline void Write(const T* data, unsigned long length,
00501         std::ostream &_os) {
00502         const unsigned int type_size = sizeof(T);
00503         assert( data ); // Can we write into pointer ?
00504         assert( length );
00505         assert( _os ); // Is stream valid ?
00506         //ByteSwap<T>::SwapRangeFromSwapCodeIntoSystem((T*)data,
00507             // _os.GetSwapCode(), length);
00508         T swappedData = SwapperNoOp::Swap(data[0]);
00509         _os.write( reinterpret_cast<const char*>(&swappedData), type_size);
00510         for(unsigned long i=1; i<length;++i) {
00511             assert( _os );
00512             swappedData = SwapperNoOp::Swap(data[i]);

```

```

00513     _os.write( reinterpret_cast<const char*>(&swappedData), type_size );
00514 }
00515 //ByteSwap<T>::SwapRangeFromSwapCodeIntoSystem((T*)data,
00516 // _os.GetSwapCode(), length);
00517 }
00518 };
00519
00520 // For particular case for ASCII string
00521 // WARNING: This template explicitly instantiates a particular
00522 // EncodingImplementation THEREFORE it is required to be declared after the
00523 // EncodingImplementation is needs (doh!)
00524 #if 0
00525 template<int TVM>
00526 class Element<TVM>
00527 {
00528 public:
00529     Element(const char array[])
00530     {
00531         unsigned int i = 0;
00532         const char sep = '\\';
00533         std::string sarray = array;
00534         std::string::size_type pos1 = 0;
00535         std::string::size_type pos2 = sarray.find(sep, pos1+1);
00536         while(pos2 != std::string::npos)
00537         {
00538             Internal[i++] = sarray.substr(pos1, pos2-pos1);
00539             pos1 = pos2+1;
00540             pos2 = sarray.find(sep, pos1+1);
00541         }
00542         Internal[i] = sarray.substr(pos1, pos2-pos1);
00543         // Shouldn't we do the contrary, since we know how many separators
00544         // (and default behavior is to discard anything after the VM declared
00545         assert( GetLength()-1 == i );
00546     }
00547
00548     unsigned long GetLength() const {
00549         return VMToLength<TVM>::Length;
00550     }
00551     // Implementation of Print is common to all Mode (ASCII/Binary)
00552     void Print(std::ostream &_os) const {
00553         _os << Internal[0]; // VM is at least guarantee to be one
00554         for(int i=1; i<VMToLength<TVM>::Length; ++i)
00555             _os << ", " << Internal[i];
00556     }
00557
00558     void Read(std::istream &_is) {
00559         EncodingImplementation<VR::VRASCII>::Read(Internal, GetLength(), _is);
00560     }
00561     void Write(std::ostream &_os) const {
00562         EncodingImplementation<VR::VRASCII>::Write(Internal, GetLength(), _os);
00563     }
00564 private:
00565     typename String Internal[VMToLength<TVM>::Length];
00566 };
00567
00568 template< int TVM>
00569 class Element<VR::PN, TVM> : public StringElement<TVM>
00570 {
00571     enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<VR::PN, TVM> ) };
00572 };
00573 #endif
00574
00575 // Implementation for the undefined length (dynamically allocated array)
00576 template<long long TVR>
00577 class Element<TVR, VM::VM1_n>
00578 {
00579     enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, VM::VM1_n> ) };
00580 public:
00581     // This the way to prevent default initialization
00582     explicit Element() { Internal=nullptr; Length=0; Save = false; }
00583     ~Element() {
00584         if( Save ) {
00585             delete[] Internal;
00586         }
00587         Internal = nullptr;
00588     }
00589
00590     static VR GetVR() { return (VR::VRType)TVR; }
00591     static VM GetVM() { return VM::VM1_n; }
00592
00593     // Length manipulation

```

```

00594 // SetLength should really be protected anyway...all operation
00595 // should go through SetArray
00596 unsigned long GetLength() const { return Length; }
00597 typedef typename VRToType<TVR>::Type Type;
00598
00599 void SetLength(unsigned long len) {
00600     const unsigned int size = sizeof(Type);
00601     if( len ) {
00602         if( len > Length ) {
00603             // perform realloc
00604             assert( (len / size) * size == len );
00605             Type *internal = new Type[len / size];
00606             assert( Save == false );
00607             Save = true; // ???
00608             if( Internal )
00609             {
00610                 memcpy(internal, Internal, len);
00611                 delete[] Internal;
00612             }
00613             Internal = internal;
00614         }
00615     }
00616     Length = len / size;
00617 }
00618
00619 // If save is set to zero user should not delete the pointer
00620 //void SetArray(const typename VRToType<TVR>::Type *array, int len, bool save = false)
00621 void SetArray(const Type *array, unsigned long len,
00622     bool save = false) {
00623     if( save ) {
00624         SetLength(len); // realloc
00625         memcpy(Internal, array, len/*sizeof(Type)*/);
00626         assert( Save == false );
00627     }
00628     else {
00629         // TODO rewrite this stupid code:
00630         assert( Length == 0 );
00631         assert( Internal == nullptr );
00632         assert( Save == false );
00633         Length = len / sizeof(Type);
00634         //assert( (len / sizeof(Type)) * sizeof(Type) == len );
00635         // MR00010001.dcm is a tough kid: 0019,105a is supposed to be VR::FL, VM::VM3 but
00636         // length is 14 bytes instead of 12 bytes. Simply consider value is total garbage.
00637         if( (len / sizeof(Type)) * sizeof(Type) != len ) { Internal = nullptr; Length = 0; }
00638         else Internal = const_cast<Type*>(array);
00639     }
00640     Save = save;
00641 }
00642 void SetValue(typename VRToType<TVR>::Type v, unsigned int idx = 0) {
00643     assert( idx < Length );
00644     Internal[idx] = v;
00645 }
00646 const typename VRToType<TVR>::Type &GetValue(unsigned int idx = 0) const {
00647     assert( idx < Length );
00648     return Internal[idx];
00649 }
00650 typename VRToType<TVR>::Type &GetValue(unsigned int idx = 0) {
00651     //assert( idx < Length );
00652     return Internal[idx];
00653 }
00654 typename VRToType<TVR>::Type operator[] (unsigned int idx) const {
00655     return GetValue(idx);
00656 }
00657 void Set(Value const &v) {
00658     const ByteValue *bv = dynamic_cast<const ByteValue*>(&v);
00659     assert( bv ); // That would be bad...
00660     if( (VR::VRType) (VRToEncoding<TVR>::Mode) == VR::VRBINARY )
00661     {
00662         const Type* array = (const Type*)bv->GetVoidPointer();
00663         if( array ) {
00664             assert( array ); // That would be bad...
00665             assert( Internal == nullptr );
00666             SetArray(array, bv->GetLength() ); }
00667     }
00668     else
00669     {
00670         std::stringstream ss;
00671         std::string s = std::string( bv->GetPointer(), bv->GetLength() );
00672         ss.str( s );
00673         EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
00674             GetLength(), ss);

```

```

00675     }
00676 }
00677 void SetFromDataElement(DataElement const &de) {
00678     const ByteValue *bv = de.GetByteValue();
00679     if( !bv ) return;
00680 #ifdef GDCM_WORDS_BIGENDIAN
00681     if( de.GetVR() == VR::UN /*|| de.GetVR() == VR::INVALID*/ )
00682 #else
00683     if( de.GetVR() == VR::UN || de.GetVR() == VR::INVALID )
00684 #endif
00685     {
00686         Set(de.GetValue());
00687     }
00688     else
00689     {
00690         SetNoSwap(de.GetValue());
00691     }
00692 }
00693
00694
00695 // Need to be placed after definition of EncodingImplementation<VR::VRASCII>
00696 void WriteASCII(std::ostream &os) const {
00697     return EncodingImplementation<VR::VRASCII>::Write(Internal, GetLength(), os);
00698 }
00699
00700 // Implementation of Print is common to all Mode (ASCII/Binary)
00701 void Print(std::ostream &_os) const {
00702     assert( Length );
00703     assert( Internal );
00704     _os << Internal[0]; // VM is at least guarantee to be one
00705     const unsigned long length = GetLength() < 25 ? GetLength() : 25;
00706     for(unsigned long i=1; i<length; ++i)
00707         _os << ", " << Internal[i];
00708 }
00709 void Read(std::istream &_is) {
00710     if( !Internal ) return;
00711     EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
00712         GetLength(), _is);
00713 }
00714 //void ReadComputeLength(std::istream &_is) {
00715 //    if( !Internal ) return;
00716 //    EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadComputeLength(Internal,
00717 //        Length, _is);
00718 // }
00719 void Write(std::ostream &_os) const {
00720     EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
00721         GetLength(), _os);
00722 }
00723
00724 DataElement GetAsDataElement() const {
00725     DataElement ret;
00726     ret.SetVR( (VR::VRType)TVR );
00727     assert( ret.GetVR() != VR::SQ );
00728     if( Internal )
00729     {
00730         std::ostringstream os;
00731         EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
00732             GetLength(), os);
00733         if( (VR::VRType)VRToEncoding<TVR>::Mode == VR::VRASCII )
00734         {
00735             if( GetVR() != VR::UI )
00736             {
00737                 if( os.str().size() % 2 )
00738                 {
00739                     os << " ";
00740                 }
00741             }
00742         }
00743         VL::Type osStrSize = (VL::Type)os.str().size();
00744         ret.SetByteValue( os.str().c_str(), osStrSize );
00745     }
00746     return ret;
00747 }
00748
00749 Element(const Element&_val) {
00750     if( this != &_amp;val ) {
00751         *this = _amp;val;
00752     }
00753 }
00754
00755 Element &operator=(const Element &_val) {

```

```

00756     Length = 0; // SYITF
00757     Internal = 0;
00758     SetArray(_val.Internal, _val.Length, true);
00759     return *this;
00760 }
00761 protected:
00762 void SetNoSwap(Value const &v) {
00763     const ByteValue *bv = dynamic_cast<const ByteValue*>(&v);
00764     assert( bv ); // That would be bad...
00765     if( (VR::VRType) (VRToEncoding<TVR>::Mode) == VR::VRBINARY )
00766     {
00767         const Type* array = (const Type*)bv->GetPointer();
00768         if( array ) {
00769             assert( array ); // That would be bad...
00770             assert( Internal == nullptr );
00771             SetArray(array, bv->GetLength() ); }
00772     }
00773     else
00774     {
00775         std::stringstream ss;
00776         std::string s = std::string( bv->GetPointer(), bv->GetLength() );
00777         ss.str( s );
00778         EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadNoSwap(Internal,
00779             GetLength(),ss);
00780     }
00781 }
00782
00783 private:
00784     typename VRToType<TVR>::Type *Internal;
00785     unsigned long Length; // unsigned int ??
00786     bool Save;
00787 };
00788
00789 //template <int TVM = VM::VM1_n>
00790 //class Element<VR::OB, TVM > : public Element<VR::OB, VM::VM1_n> {};
00791
00792 // Partial specialization for derivatives of 1-n : 2-n, 3-n ...
00793 template<long long TVR>
00794 class Element<TVR, VM::VM1_2> : public Element<TVR, VM::VM1_n>
00795 {
00796 public:
00797     typedef Element<TVR, VM::VM1_n> Parent;
00798     void SetLength(int len) {
00799         if( len != 1 && len != 2 ) return;
00800         Parent::SetLength(len);
00801     }
00802 };
00803 template<long long TVR>
00804 class Element<TVR, VM::VM2_n> : public Element<TVR, VM::VM1_n>
00805 {
00806     enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, VM::VM2_n> ) };
00807 public:
00808     typedef Element<TVR, VM::VM1_n> Parent;
00809     void SetLength(int len) {
00810         if( len <= 1 ) return;
00811         Parent::SetLength(len);
00812     }
00813 };
00814 template<long long TVR>
00815 class Element<TVR, VM::VM2_2n> : public Element<TVR, VM::VM2_n>
00816 {
00817     enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, VM::VM2_2n> ) };
00818 public:
00819     typedef Element<TVR, VM::VM2_n> Parent;
00820     void SetLength(int len) {
00821         if( len % 2 ) return;
00822         Parent::SetLength(len);
00823     }
00824 };
00825 template<long long TVR>
00826 class Element<TVR, VM::VM3_n> : public Element<TVR, VM::VM1_n>
00827 {
00828     enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, VM::VM3_n> ) };
00829 public:
00830     typedef Element<TVR, VM::VM1_n> Parent;
00831     void SetLength(int len) {
00832         if( len <= 2 ) return;
00833         Parent::SetLength(len);
00834     }
00835 };
00836 template<long long TVR>

```



```

00837 class Element<TVR, VM::VM3_3n> : public Element<TVR, VM::VM3_n>
00838 {
00839     enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, VM::VM3_3n> ) };
00840 public:
00841     typedef Element<TVR, VM::VM3_n> Parent;
00842     void SetLength(int len) {
00843         if( len % 3 ) return;
00844         Parent::SetLength(len);
00845     }
00846 };
00847 template<long long TVR>
00848 class Element<TVR, VM::VM3_4> : public Element<TVR, VM::VM1_n>
00849 {
00850 public:
00851     typedef Element<TVR, VM::VM1_n> Parent;
00852     void SetLength(int len) {
00853         if( len != 3 && len != 4 ) return;
00854         Parent::SetLength(len);
00855     }
00856 };
00857
00858
00859 //template<int T> struct VRToLength;
00860 //template<> struct VRToLength<VR::AS>
00861 //{ enum { Length = VM::VM1 }; }
00862 //template<>
00863 //class Element<VR::AS> : public Element<VR::AS, VRToLength<VR::AS>::Length >
00864
00865 // only 0010 1010 AS 1 Patient's Age
00866 template<>
00867 class Element<VR::AS, VM::VM5>
00868 {
00869     enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<VR::AS, VM::VM5> ) };
00870 public:
00871     char Internal[VMToLength<VM::VM5>::Length * sizeof( VRToType<VR::AS>::Type )];
00872     void Print(std::ostream &_os) const {
00873         _os << Internal;
00874     }
00875     unsigned long GetLength() const {
00876         return VMToLength<VM::VM5>::Length;
00877     }
00878 };
00879
00880
00881 template<>
00882 class Element<VR::OB, VM::VM1> : public Element<VR::OB, VM::VM1_n> {};
00883
00884 // Same for OW:
00885 template<>
00886 class Element<VR::OW, VM::VM1> : public Element<VR::OW, VM::VM1_n> {};
00887
00888
00889 } // namespace gdcm_ns
00890
00891 #endif //GDCMELEMENT_H

```

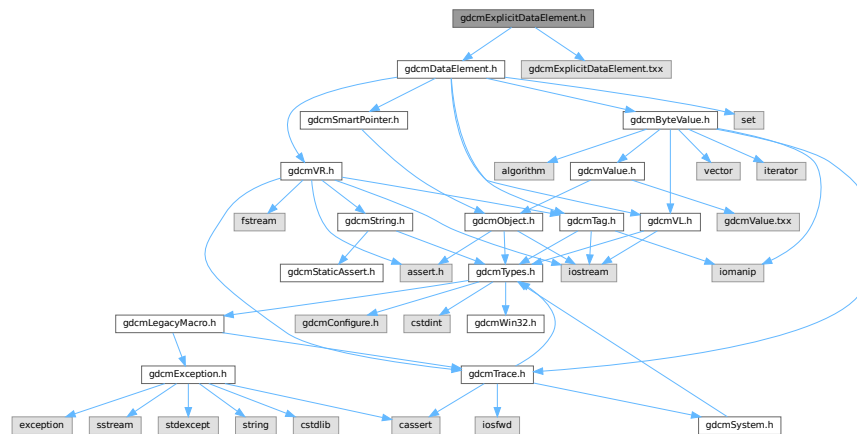
## 11.135 gdcmExplicitDataElement.h File Reference

```

#include "gdcmDataElement.h"
#include "gdcmExplicitDataElement.txx"

```

Include dependency graph for `gdcmExplicitDataElement.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::ExplicitDataElement`  
Class to read/write a *DataElement* as *Explicit Data Element*.

## Namespaces

- namespace `gdcm`

## 11.136 gdcmExplicitDataElement.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012

```

```

00013 =====*/
00014 #ifndef GDCMEXPLICITDATAELEMENT_H
00015 #define GDCMEXPLICITDATAELEMENT_H
00016
00017 #include "gdcmDataElement.h"
00018
00019 namespace gdcm_ns
00020 {
00021     class GDCM_EXPORT ExplicitDataElement : public DataElement
00022     {
00023     public:
00024         VL GetLength() const;
00025
00026         template <typename TSwap>
00027         std::istream &Read(std::istream &is);
00028
00029         template <typename TSwap>
00030         std::istream &ReadPreValue(std::istream &is);
00031
00032         template <typename TSwap>
00033         std::istream &ReadValue(std::istream &is, bool readvalues = true);
00034
00035         template <typename TSwap>
00036         std::istream &ReadWithLength(std::istream &is, VL & length);
00037
00038         template <typename TSwap>
00039         const std::ostream &Write(std::ostream &os) const;
00040     };
00041 } // end namespace gdcm_ns
00042
00043 #include "gdcmExplicitDataElement.txx"
00044
00045 #endif //GDCMEXPLICITDATAELEMENT_H

```

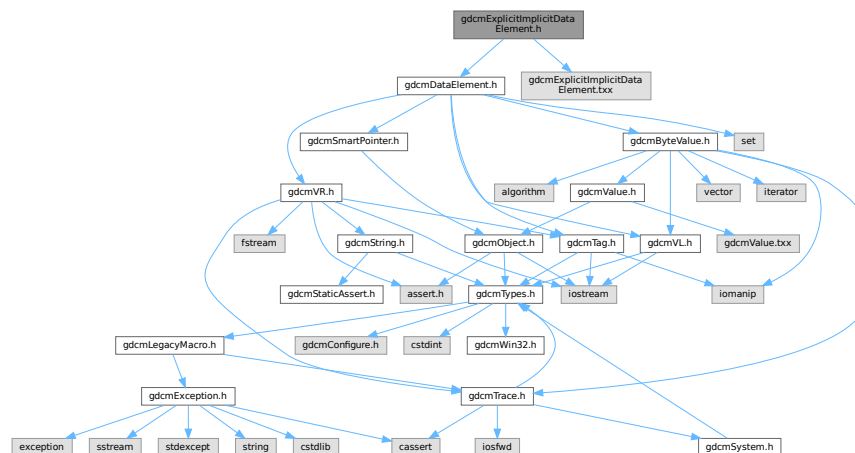
## 11.137 gdcmExplicitImplicitDataElement.h File Reference

```

#include "gdcmDataElement.h"
#include "gdcmExplicitImplicitDataElement.txx"

```

Include dependency graph for gdcmExplicitImplicitDataElement.h:



### Classes

- class [gdcm::ExplicitImplicitDataElement](#)  
Class to read/write a [DataElement](#) as [ExplicitImplicit Data Element](#).

## Namespaces

- namespace [gdcm](#)

## 11.138 gdcmExplicitImplicitDataElement.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMEXPLICITIMPLICITDATAELEMENT_H
00015 #define GDCMEXPLICITIMPLICITDATAELEMENT_H
00016
00017 #include "gdcmDataElement.h"
00018
00019 namespace gdcm
00020 {
00021   // Data Element (ExplicitImplicit)
00022   class GDCM_EXPORT ExplicitImplicitDataElement : public DataElement
00023   {
00024   public:
00025     VL GetLength() const;
00026
00027     template <typename TSwap>
00028     std::istream &Read(std::istream &is);
00029
00030     template <typename TSwap>
00031     std::istream &ReadPreValue(std::istream &is);
00032
00033     template <typename TSwap>
00034     std::istream &ReadValue(std::istream &is, bool readvalues = true);
00035
00036     template <typename TSwap>
00037     std::istream &ReadWithLength(std::istream &is, VL & length)
00038     {
00039       (void)length;
00040       return Read<TSwap>(is);
00041     }
00042
00043     // PURPOSELY do not provide an implementation for writing !
00044     template <typename TSwap>
00045     //const std::ostream &Write(std::ostream &os) const;
00046   };
00047
00048 } // end namespace gdcm
00049
00050 #include "gdcmExplicitImplicitDataElement.txx"
00051 #endif //GDCMEXPLICITIMPLICITDATAELEMENT_H

```

## 11.139 gdcmFile.h File Reference

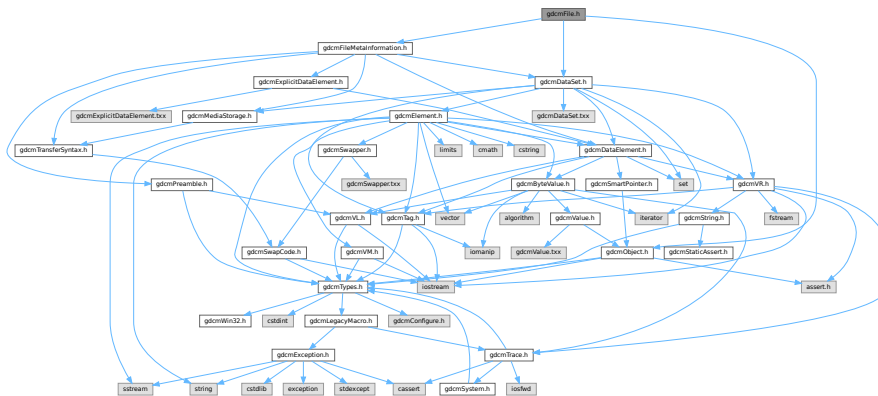
```

#include "gdcmObject.h"
#include "gdcmDataSet.h"

```

```
#include "gdcmFileMetaInformation.h"
```

Include dependency graph for gdcmFile.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::File`  
a DICOM File

## Namespaces

- namespace `gdcm`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const File &val)`

## 11.140 gdcmFile.h

[Go to the documentation of this file.](#)

```
00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
```

```

00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 #ifndef GDCMFILE_H
00015 #define GDCMFILE_H
00016
00017 #include "gdcmObject.h"
00018 #include "gdcmDataSet.h"
00019 #include "gdcmFileMetaInformation.h"
00020
00021 namespace gdcm_ns
00022 {
00023
00024     class GDCM_EXPORT File : public Object
00025     {
00026     public:
00027         File();
00028         ~File() override;
00029
00030         friend std::ostream &operator<<(std::ostream &os, const File &val);
00031
00032         std::istream &Read(std::istream &is);
00033
00034         std::ostream const &Write(std::ostream &os) const;
00035
00036         const FileMetaInformation &GetHeader() const { return Header; }
00037
00038         FileMetaInformation &GetHeader() { return Header; }
00039
00040         void SetHeader( const FileMetaInformation &fmi ) { Header = fmi; }
00041
00042         const DataSet &GetDataSet() const { return DS; }
00043
00044         DataSet &GetDataSet() { return DS; }
00045
00046         void SetDataSet( const DataSet &ds ) { DS = ds; }
00047
00048     private:
00049         FileMetaInformation Header;
00050         DataSet DS;
00051     };
00052
00053     //-----
00054     inline std::ostream& operator<<(std::ostream &os, const File &val)
00055     {
00056         os << val.GetHeader() << std::endl;
00057         //os << val.GetDataSet() << std::endl; // FIXME
00058         assert(0);
00059         return os;
00060     }
00061
00062 } // end namespace gdcm_ns
00063
00064 #endif //GDCMFILE_H

```

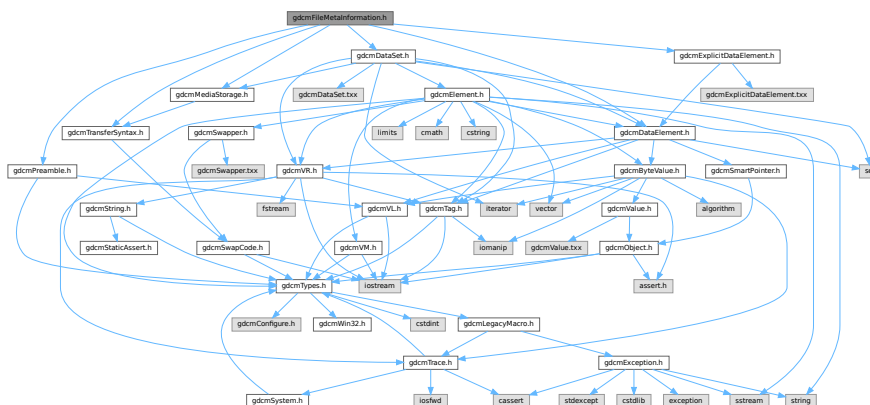
## 11.141 gdcmFileMetaInformation.h File Reference

```

#include "gdcmPreamble.h"
#include "gdcmDataSet.h"
#include "gdcmDataElement.h"
#include "gdcmMediaStorage.h"
#include "gdcmTransferSyntax.h"
#include "gdcmExplicitDataElement.h"

```

Include dependency graph for gdcmFileMetaInformation.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::FileMetaInformation](#)  
Class to represent a [File](#) Meta Information.

## Namespaces

- namespace [gdcm](#)

## Functions

- std::ostream & [gdcm::operator<<](#) (std::ostream &os, const [FileMetaInformation](#) &val)

## 11.142 gdcmFileMetaInformation.h

[Go to the documentation of this file.](#)

```
00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
```

```

00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 #ifndef GDCMFILEMETAINFORMATION_H
00015 #define GDCMFILEMETAINFORMATION_H
00016
00017 #include "gdcmPreamble.h"
00018 #include "gdcmDataSet.h"
00019 #include "gdcmDataElement.h"
00020 #include "gdcmMediaStorage.h"
00021 #include "gdcmTransferSyntax.h"
00022 #include "gdcmExplicitDataElement.h"
00023
00024 namespace gdcm_ns
00025 {
00040 class GDCM_EXPORT FileMetaInformation : public DataSet
00041 {
00042 public:
00043     // FIXME: TransferSyntax::TS_END -> TransferSyntax::ImplicitDataElement
00044     FileMetaInformation();
00045     ~FileMetaInformation();
00046
00047     friend std::ostream &operator<(std::ostream &_os, const FileMetaInformation &_val);
00048
00049     bool IsValid() const { return true; }
00050
00051     TransferSyntax::NegociatedType GetMetaInformationTS() const { return MetaInformationTS; }
00052     void SetDataSetTransferSyntax(const TransferSyntax &ts);
00053     const TransferSyntax &GetDataSetTransferSyntax() const { return DataSetTS; }
00054     MediaStorage GetMediaStorage() const;
00055     std::string GetMediaStorageAsString() const;
00056
00057     // FIXME: no virtual function means: duplicate code...
00058     void Insert(const DataElement& de) {
00059         if ( de.GetTag().GetGroup() == 0x0002 )
00060         {
00061             InsertDataElement( de );
00062         }
00063         else
00064         {
00065             gdcmErrorMacro( "Cannot add element with group != 0x0002 in the file meta header: " << de );
00066         }
00067     }
00068     void Replace(const DataElement& de) {
00069         Remove(de.GetTag());
00070         Insert(de);
00071     }
00072
00074     std::istream &Read(std::istream &is);
00075     std::istream &ReadCompat(std::istream &is);
00076
00078     std::ostream &Write(std::ostream &os) const;
00079
00081     void FillFromDataSet(DataSet const &ds);
00082
00084     const Preamble &GetPreamble() const { return P; }
00085     Preamble &GetPreamble() { return P; }
00086     void SetPreamble(const Preamble &p) { P = p; }
00087
00089     static void SetImplementationClassUID(const char * imp);
00090     static void AppendImplementationClassUID(const char * imp);
00091     static const char *GetImplementationClassUID();
00092     static void SetImplementationVersionName(const char * version);
00093     static const char *GetImplementationVersionName();
00094     static void SetSourceApplicationEntityTitle(const char * title);
00095     static const char *GetSourceApplicationEntityTitle();
00096
00097     FileMetaInformation(FileMetaInformation const& fmi) = default;
00098     FileMetaInformation& operator=(const FileMetaInformation& fmi) = default;
00099
00100     VL GetFullLength() const {
00101         return P.GetLength() + DataSet::GetLength<ExplicitDataElement>();
00102     }
00103
00104 protected:
00105     void ComputeDataSetTransferSyntax(); // FIXME
00106
00107     template <typename TSwap>

```



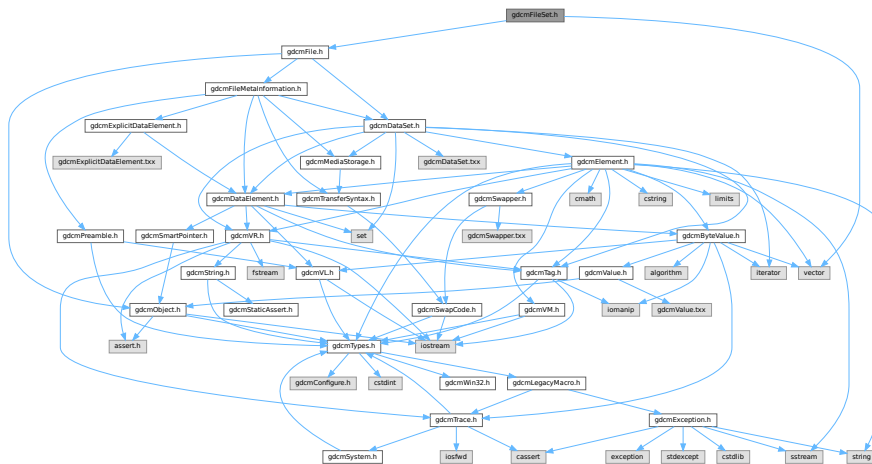
```

00108 std::istream &ReadCompatInternal(std::istream &is);
00109
00110 void Default();
00111 void ComputeDataSetMediaStorageSOPClass();
00112
00113 TransferSyntax DataSetTS;
00114 TransferSyntax::NegociatedType MetaInformationTS;
00115 MediaStorage::MSType DataSetMS;
00116
00117 protected:
00118     static const char * GetFileMetaInformationVersion();
00119     static const char * GetGDCMImplementationClassUID();
00120     static const char * GetGDCMImplementationVersionName();
00121     static const char * GetGDCMSourceApplicationEntityTitle();
00122
00123 private:
00124     Preamble P;
00125
00126 //static stuff:
00127     static const char GDCM_FILE_META_INFORMATION_VERSION[];
00128     static const char GDCM_IMPLEMENTATION_CLASS_UID[];
00129     static const char GDCM_IMPLEMENTATION_VERSION_NAME[];
00130     static const char GDCM_SOURCE_APPLICATION_ENTITY_TITLE[];
00131     static std::string ImplementationClassUID;
00132     static std::string ImplementationVersionName;
00133     static std::string SourceApplicationEntityTitle;
00134 };
00135 //-----
00136 inline std::ostream& operator<<(std::ostream &os, const FileMetaInformation &val)
00137 {
00138     os << val.GetPreamble() << std::endl;
00139     val.Print( os );
00140     return os;
00141 }
00142
00143 } // end namespace gdcm_ns
00144
00145 #endif //GDCMFILEMETAINFORMATION H

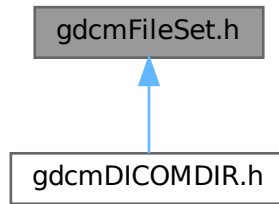
```

### 11.143 gdcMFileSet.h File Reference

```
#include "gdcmFile.h"
#include <vector>
Include dependency graph for gdcmFileSet.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcmdicom::FileSet](#)

## Namespaces

- namespace [gdcmdicom](#)

## Functions

- `std::ostream & gdcmdicom::operator<< (std::ostream &os, const FileSet &f)`

## 11.144 gdcmdicomFileSet.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcmdicom.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012  =====*/
00013  #ifndef GDCMFILESET_H
00014  #define GDCMFILESET_H
00015
00016  #include "gdcmdicomFile.h"
00017  #include <vector>
00018
00019  namespace gdcmdicom
00020  {
00021  {
00026  class GDCM_EXPORT FileSet
00027  {
00028  friend std::ostream& operator<<(std::ostream &_os, const FileSet &d);
  
```

```

00029 public:
00030     FileSet():Files() {}
00031     typedef std::string FileType;
00032     typedef std::vector<FileType> FileTypes;
00033
00034     void AddFile(File const & ) {}
00035
00036     bool AddFile(const char *filename);
00037
00038     void SetFiles(FileTypes const &files);
00039     FileTypes const &GetFiles() const {
00040         return Files;
00041     }
00042
00043 private:
00044     FileTypes Files;
00045 };
00046 //-----
00047 inline std::ostream& operator<<(std::ostream &os, const FileSet &F)
00048 {
00049     (void)f; // FIXME
00050     return os;
00051 }
00052
00053 } // end namespace gdcm
00054 #endif //GDCMFILESET_H

```

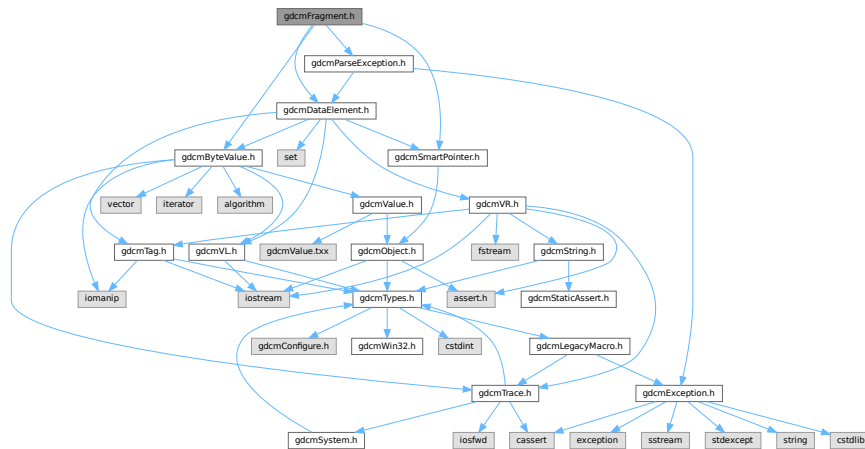
## 11.145 gdcmFragment.h File Reference

```

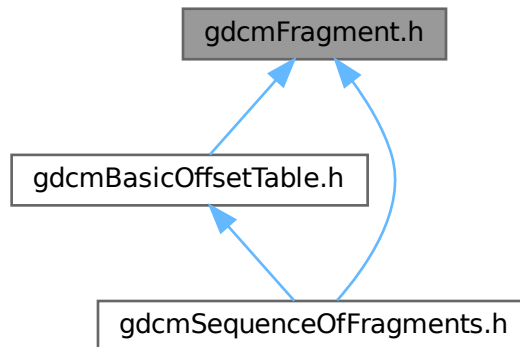
#include "gdcmDataElement.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"
#include "gdcmParseException.h"

```

Include dependency graph for gdcmFragment.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcmm::Fragment](#)  
Class to represent a *Fragment*.

## Namespaces

- namespace [gdcmm](#)

## Functions

- `std::ostream & gdcmm::operator<< (std::ostream &os, const Fragment &val)`

## 11.146 gdcmmFragment.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMMFRAGMENT_H
00015  #define GDCMMFRAGMENT_H
00016

```

```

00017 #include "gdcmDataElement.h"
00018 #include "gdcmByteValue.h"
00019 #include "gdcmSmartPointer.h"
00020 #include "gdcmParseException.h"
00021
00022 namespace gdcm_ns
00023 {
00024
00025     // Implementation detail:
00026     // I think Fragment should be a protected subclass of DataElement:
00027     // looking somewhat like this:
00028     /*
00029     class GDCM_EXPORT Fragment : protected DataElement
00030     {
00031     public:
00032         using DataElement::GetTag;
00033         using DataElement::GetVL;
00034         using DataElement::SetByteValue;
00035         using DataElement::GetByteValue;
00036         using DataElement::GetValue;
00037     */
00038     // Instead I am only hiding the SetTag member...
00039
00040     class GDCM_EXPORT Fragment : public DataElement
00041     {
00042     //protected:
00043     // void SetTag(const Tag &t);
00044     public:
00045         Fragment() : DataElement(Tag(0xffff, 0x000), 0) {}
00046         friend std::ostream &operator<<(std::ostream &os, const Fragment &val);
00047
00048         VL GetLength() const;
00049
00050         VL ComputeLength() const;
00051
00052         template <typename TSwap>
00053         std::istream &Read(std::istream &is)
00054         {
00055             ReadPreValue<TSwap>(is);
00056             return ReadValue<TSwap>(is);
00057         }
00058
00059         template <typename TSwap>
00060         std::istream &ReadPreValue(std::istream &is)
00061         {
00062             TagField.Read<TSwap>(is);
00063             if( !is )
00064             {
00065                 // BogusItemStartItemEnd.dcm
00066                 throw Exception( "Problem #1" );
00067             }
00068             if( !ValueLengthField.Read<TSwap>(is) )
00069             {
00070                 // GENESIS_SIGNA-JPEG-CorruptFrag.dcm
00071                 // JPEG fragment is declared to have 61902, but in fact really is only 61901
00072                 // so we end up reading 0xddff,0x00e0, and VL = 0x0 (1 byte)
00073                 throw Exception( "Problem #2" );
00074             }
00075
00076 #ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
00077             const Tag itemStart(0xffff, 0x000);
00078             const Tag seqDelItem(0xffff,0x00dd);
00079             if( TagField != itemStart && TagField != seqDelItem )
00080             {
00081                 throw Exception( "Problem #3" );
00082             }
00083 #endif
00084             return is;
00085         }
00086
00087         template <typename TSwap>
00088         std::istream &ReadValue(std::istream &is)
00089         {
00090             // Self
00091             SmartPointer<ByteValue> bv = new ByteValue;
00092             bv->SetLength(ValueLengthField);
00093             if( !bv->Read<TSwap>(is) )
00094             {
00095                 // Fragment is incomplete, but is a itemStart, let's try to push it anyway...
00096                 gdcmWarningMacro( "Fragment could not be read" );
00097                 //bv->SetLength(is.gcount());
00098                 ValueField = bv;
00099             }
00100         }
00101     };

```

```

00101         ParseException pe;
00102         pe.SetLastElement( *this );
00103         throw pe;
00104     }
00105     ValueField = bv;
00106     return is;
00107 }
00108
00109 template <typename TSwap>
00110 std::istream &ReadBacktrack(std::istream &is)
00111 {
00112     const Tag itemStart(0xfffe, 0xe000);
00113     const Tag seqDelItem(0xfffe, 0xe0dd);
00114
00115     bool cont = true;
00116     const std::streampos start = is.tellg();
00117     const int max = 10;
00118     int offset = 0;
00119     while( cont )
00120     {
00121         TagField.Read<TSwap>(is);
00122         assert( is );
00123         if( TagField != itemStart && TagField != seqDelItem )
00124         {
00125             ++offset;
00126             is.seekg( (std::streampos)((size_t)start - offset) );
00127             gdcmWarningMacro( "Fuzzy Search, backtrack: " < (start - is.tellg()) < " Offset: " < is.tellg() );
00128             if( offset > max )
00129             {
00130                 gdcmErrorMacro( "Giving up" );
00131                 throw "Impossible to backtrack";
00132             }
00133         }
00134         else
00135         {
00136             cont = false;
00137         }
00138     }
00139     assert( TagField == itemStart || TagField == seqDelItem );
00140     if( !ValueLengthField.Read<TSwap>(is) )
00141     {
00142         return is;
00143     }
00144
00145     // Self
00146     SmartPointer<ByteValue> bv = new ByteValue;
00147     bv->SetLength(ValueLengthField);
00148     if( !bv->Read<TSwap>(is) )
00149     {
00150         // Fragment is incomplete, but is a itemStart, let's try to push it anyway...
00151         gdcmWarningMacro( "Fragment could not be read" );
00152         //bv->SetLength(is.gcount());
00153         ValueField = bv;
00154         ParseException pe;
00155         pe.SetLastElement( *this );
00156         throw pe;
00157     }
00158     ValueField = bv;
00159     return is;
00160 }
00161
00162 template <typename TSwap>
00163 std::ostream &Write(std::ostream &os) const {
00164     const Tag itemStart(0xfffe, 0xe000);
00165     const Tag seqDelItem(0xfffe, 0xe0dd);
00166     if( !TagField.Write<TSwap>(os) )
00167     {
00168         assert(0 && "Should not happen");
00169         return os;
00170     }
00171     assert( TagField == itemStart
00172         || TagField == seqDelItem );
00173     const ByteValue *bv = GetByteValue();
00174     // VL
00175     // The following piece of code is hard to read in order to support such broken file as:
00176     // CompressedLossy.dcm
00177     if( IsEmpty() )
00178     {
00179         //assert( bv );
00180         VL zero = 0;

```

```

00182         if( !zero.Write<TSwap>(os) )
00183         {
00184             assert(0 && "Should not happen");
00185             return os;
00186         }
00187     }
00188     else
00189     {
00190         assert( ValueLengthField );
00191         assert( !ValueLengthField.IsUndefined() );
00192         const VL actualLen = bv->ComputeLength();
00193         assert( actualLen == ValueLengthField || actualLen == ValueLengthField + 1 );
00194         if( !actualLen.Write<TSwap>(os) )
00195         {
00196             assert(0 && "Should not happen");
00197             return os;
00198         }
00199     }
00200     // Value
00201     if( ValueLengthField && bv )
00202     {
00203         // Self
00204         assert( bv );
00205         assert( bv->GetLength() == ValueLengthField );
00206         if( !bv->Write<TSwap>(os) )
00207         {
00208             assert(0 && "Should not happen");
00209             return os;
00210         }
00211     }
00212     return os;
00213 }
00214 };
00215 //-----
00216 inline std::ostream &operator<<(std::ostream &os, const Fragment &val)
00217 {
00218     os << "Tag: " << val.TagField;
00219     os << "\tVL: " << val.ValueLengthField;
00220     if( val.ValueField )
00221     {
00222         os << "\t" << *(val.ValueField);
00223     }
00224     return os;
00225 }
00226 }
00227
00228 } // end namespace gdcm_ns
00229
00230 #endif //GDCMFRAGMENT_H

```

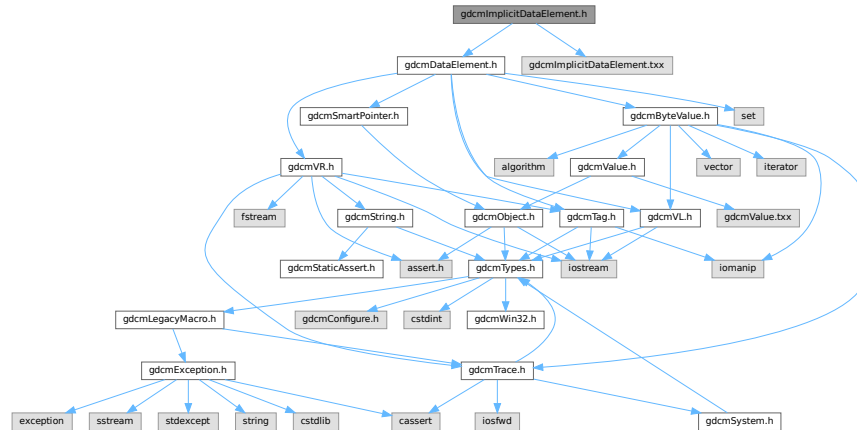
## 11.147 gdcmImplicitDataElement.h File Reference

```

#include "gdcmDataElement.h"
#include "gdcmImplicitDataElement.txx"

```

Include dependency graph for `gdcmImplicitDataElement.h`:



## Classes

- class `gdcm::ImplicitDataElement`  
Class to represent an Implicit VR Data Element.

## Namespaces

- namespace `gdcm`

## 11.148 gdcmImplicitDataElement.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 #ifndef GDCMIMPLICITDATAELEMENT_H
00015 #define GDCMIMPLICITDATAELEMENT_H
00016
00017 #include "gdcmDataElement.h"
00018
00019 namespace gdcm_ns
00020 {
00021
00022
00023
00024
00025
00026 class GDCM_EXPORT ImplicitDataElement : public DataElement
00027 {
00028 public:
00029     VL GetLength() const;
00030

```

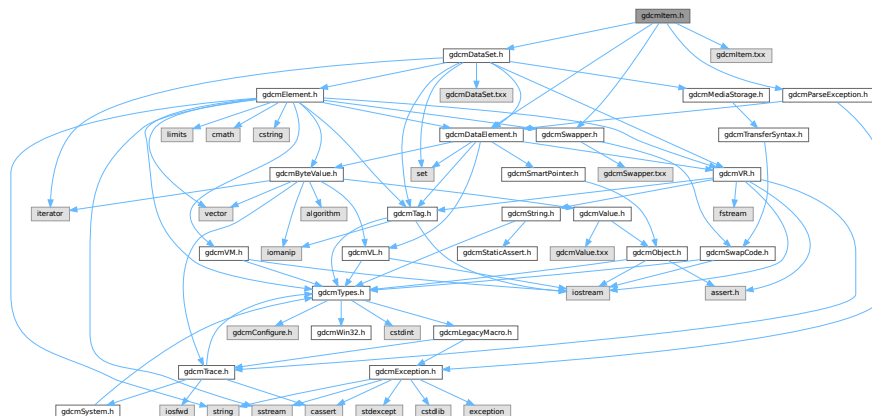


```
00031     template <typename TSwap>
00032         std::istream &Read(std::istream& is);
00033
00034     template <typename TSwap>
00035         std::istream &ReadPreValue(std::istream& is);
00036
00037     template <typename TSwap>
00038         std::istream &ReadValue(std::istream& is, bool readvalues = true);
00039
00040     template <typename TSwap>
00041         std::istream &ReadWithLength(std::istream& is, VL & length, bool readvalues = true);
00042
00043     template <typename TSwap>
00044         std::istream &ReadValueWithLength(std::istream& is, VL & length, bool readvalues = true);
00045
00046     template <typename TSwap>
00047         const std::ostream &Write(std::ostream& os) const;
00048 };
00049
00050 } // end namespace gdcms
00051
00052 #include "gdcmsImplicitDataElement.txx"
00053
00054 #endif //GDCMIMPLICITDATAELEMENT_H
```

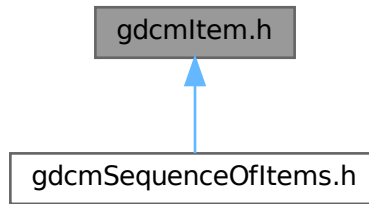
## 11.149 gdcmltem.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmDataSet.h"
#include "gdcmParseException.h"
#include "gdcmSwapper.h"
#include "gdcmItem.hxx"
```

Include dependency graph for gdcmlItem.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcmltem::Item](#)  
*Class to represent an [Item](#).*

## Namespaces

- namespace [gdcmltem](#)

## Functions

- `std::ostream & gdcmltem::operator<< (std::ostream &os, const Item &val)`

## 11.150 gdcmltem.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcmltem.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012  =====*/
00013
00014
00015  #ifndef GDCMITEM_H
00016  #define GDCMITEM_H
00017
00018  #include "gdcmltemDataElement.h"
00019  #include "gdcmltemDataSet.h"
00020  #include "gdcmltemParseException.h"
00021  #include "gdcmltemSwapper.h"
00022

```

```

00023 #ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
00024 #include "gdcmByteSwapFilter.h"
00025 #endif
00026
00027 namespace gdcm_ns
00028 {
00029
00030 class DataSet;
00045 class GDCM_EXPORT Item : public DataElement
00046 {
00047 public:
00048     Item() : DataElement(Tag(0xfffe, 0xe000), 0xffffffff, NestedDataSet()) {}
00049     friend std::ostream& operator<< (std::ostream &os, const Item &val);
00050
00051     void Clear() {
00052         this->DataElement::Clear();
00053         NestedDataSet.Clear();
00054     }
00055
00056     template <typename TDE>
00057     VL GetLength() const;
00058
00059     void InsertDataElement(const DataElement & de) {
00060         NestedDataSet.Insert(de);
00061         // Update the length
00062         if( !IsUndefinedLength() )
00063         {
00064             assert( 0 && "InsertDataElement" );
00065             //ValueLengthField += de.GetLength();
00066         }
00067     }
00068     const DataElement& GetDataElement(const Tag& t) const
00069     {
00070         return NestedDataSet.GetDataElement(t);
00071     }
00072
00073     // Completely defines it with the nested dataset
00074     // destroy anything present
00075     void SetNestedDataSet(const DataSet& nested)
00076     {
00077         NestedDataSet = nested;
00078     }
00079     // Return a const ref to the Nested Data Set
00080     const DataSet &GetNestedDataSet() const
00081     {
00082         return NestedDataSet;
00083     }
00084     DataSet &GetNestedDataSet()
00085     {
00086         return NestedDataSet;
00087     }
00088
00089     //Value const & GetValue() const { return *NestedDataSet; }
00090
00091     Item(Item const &val):DataElement(val)
00092     {
00093         NestedDataSet = val.NestedDataSet;
00094     }
00095
00096     template <typename TDE, typename TSwap>
00097     std::istream &Read(std::istream &is) {
00098         // Superclass
00099         {
00100             DataSet &nested = NestedDataSet;
00101             nested.Clear();
00102             assert( nested.IsEmpty() );
00103         }
00104         if( !TagField.Read<TSwap>(is) )
00105         {
00106             throw Exception("Should not happen (item)");
00107             return is;
00108         }
00109     #ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
00110         // MR_Philips_Intera_SwitchIndianess_noLgtSQItem_in_trueLgtSeq.dcm
00111         if( TagField == Tag(0xfeff, 0x00e0)
00112             || TagField == Tag(0xfeff, 0xdde0) )
00113         {
00114             gdcmWarningMacro( "ByteSwaping Private SQ: " << TagField );
00115             // Invert previously read TagField since wrong endianness:
00116             TagField = Tag( SwapperDoOp::Swap( TagField.GetGroup() ), SwapperDoOp::Swap( TagField.GetElement() )
);

```

```

00117     assert ( TagField == Tag(0xfffe, 0xe000)
00118             || TagField == Tag(0xfffe, 0xe0dd) );
00119
00120     if( !ValueLengthField.Read<SwapperDoOp>(is) )
00121     {
00122         assert(0 && "Should not happen");
00123         return is;
00124     }
00125     // Self
00126     // Some file written by GDCM 1.0 we write 0xFFFFFFFF instead of 0x0
00127     if( TagField == Tag(0xfffe,0xe0dd) )
00128     {
00129         if( ValueLengthField )
00130         {
00131             gdcmErrorMacro( "ValueLengthField is not 0" );
00132         }
00133     }
00134     //else if( ValueLengthField == 0 )
00135     // {
00136     //     //assert( TagField == Tag( 0xfffe, 0xe0dd) );
00137     //     if( TagField != Tag( 0xfffe, 0xe0dd) )
00138     //     {
00139     //         gdcmErrorMacro( "SQ: " << TagField << " has a length of 0" );
00140     //     }
00141     // }
00142     else if( ValueLengthField.IsUndefined() )
00143     {
00144         DataSet &nested = NestedDataSet;
00145         nested.Clear();
00146         assert( nested.IsEmpty() );
00147         std::streampos start = is.tellg();
00148         try
00149         {
00150             nested.template ReadNested<TDE,SwapperDoOp>(is);
00151             ByteSwapFilter bsf(nested);
00152             bsf.ByteSwap();
00153         }
00154         catch(ParseException &pe)
00155         {
00156             (void)pe;
00157             //
00158             MR_Philips_Intera_PrivateSequenceExplicitVR_in_SQ_2001_e05f_item_wrong_lgt_use_NOSHADOWSEQ.dcm
00159             // You have to byteswap the length but not the tag...sigh
00160             gdcmWarningMacro( "Attempt to read nested Item without byteswapping the Value Length." );
00161             start -= is.tellg();
00162             assert( start < 0 );
00163             is.seekg( start, std::ios::cur );
00164             nested.Clear();
00165             nested.template ReadNested<TDE,SwapperNoOp>(is);
00166             ByteSwapFilter bsf(nested);
00167             // Tag are read in big endian, need to byteswap them back...
00168             bsf.SetByteSwapTag(true);
00169             bsf.ByteSwap();
00170         }
00171         catch(Exception &e)
00172         {
00173             // MR_Philips_Intera_No_PrivateSequenceImplicitVR.dcm
00174             throw e;
00175         }
00176         catch(...)
00177         {
00178             assert(0);
00179         }
00180     }
00181     else /* if( ValueLengthField.IsUndefined() ) */
00182     {
00183         DataSet &nested = NestedDataSet;
00184         nested.Clear();
00185         assert( nested.IsEmpty() );
00186         nested.template ReadWithLength<TDE,SwapperDoOp>(is, ValueLengthField);
00187         ByteSwapFilter bsf(nested);
00188         bsf.ByteSwap();
00189     }
00190     return is;
00191 }
00192 // http://groups.google.com/group/comp.protocols.dicom/msg/c07efcf5e759fc83
00193 // Bug_Philips_ItemTag_3F3F.dcm
00194 if( TagField == Tag(0x3f3f, 0x3f00) )
00195 {
00196     //TagField = Tag(0xfffe, 0xe000);
00197 }

```

```

00197 #endif
00198     if( TagField != Tag(0xffff, 0xe000) && TagField != Tag(0xffff, 0xe0dd) )
00199     {
00200         gdcmDebugMacro( "Invalid Item, found tag: " « TagField);
00201         throw Exception( "Not a valid Item" );
00202     }
00203     assert( TagField == Tag(0xffff, 0xe000) || TagField == Tag(0xffff, 0xe0dd) );
00204
00205     if( !ValueLengthField.Read<TSwap>(is) )
00206     {
00207         assert(0 && "Should not happen");
00208         return is;
00209     }
00210     // Self
00211     if( TagField == Tag(0xffff,0xe0dd) )
00212     {
00213         // Some file written by GDCM 1.0 were written with 0xFFFFFFFF instead of 0x0
00214         if( ValueLengthField )
00215         {
00216             gdcmDebugMacro( "ValueLengthField is not 0 but " « ValueLengthField );
00217         }
00218     }
00219     else if( ValueLengthField.IsUndefined() )
00220     {
00221         DataSet &nested = NestedDataSet;
00222         nested.Clear();
00223         assert( nested.IsEmpty() );
00224         nested.template ReadNested<TDE,TSwap>(is);
00225     }
00226     else /* if( ValueLengthField.IsUndefined() ) */
00227     {
00228         assert( !ValueLengthField.IsUndefined() );
00229         DataSet &nested = NestedDataSet;
00230         nested.Clear();
00231         assert( nested.IsEmpty() );
00232         nested.template ReadWithLength<TDE,TSwap>(is, ValueLengthField);
00233     }
00234
00235     return is;
00236 }
00237
00238 template <typename TDE, typename TSwap>
00239 const std::ostream &Write(std::ostream &os) const {
00240 #ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
00241     if( TagField == Tag(0x3f3f,0x3f00) && false )
00242     {
00243         Tag t(0xffff, 0xe000);
00244         t.Write<TSwap>(os);
00245     }
00246     else
00247 #endif
00248     {
00249         assert ( TagField == Tag(0xffff, 0xe000)
00250             || TagField == Tag(0xffff, 0xe0dd) );
00251         // Not sure how this happen
00252         if( TagField == Tag(0xffff, 0xe0dd) )
00253         {
00254             gdcmWarningMacro( "SeqDelItem found in defined length Sequence" );
00255             assert( ValueLengthField == 0 );
00256             assert( NestedDataSet.Size() == 0 );
00257         }
00258         if( !TagField.Write<TSwap>(os) )
00259         {
00260             assert(0 && "Should not happen");
00261             return os;
00262         }
00263     }
00264     if( ValueLengthField.IsUndefined() )
00265     {
00266         if( !ValueLengthField.Write<TSwap>(os) )
00267         {
00268             assert(0 && "Should not happen");
00269             return os;
00270         }
00271     }
00272     else
00273     {
00274         const VL dummy = NestedDataSet.GetLength<TDE>();
00275         assert( dummy % 2 == 0 );
00276         //assert( ValueLengthField == dummy );
00277         if( !dummy.Write<TSwap>(os) )

```

```

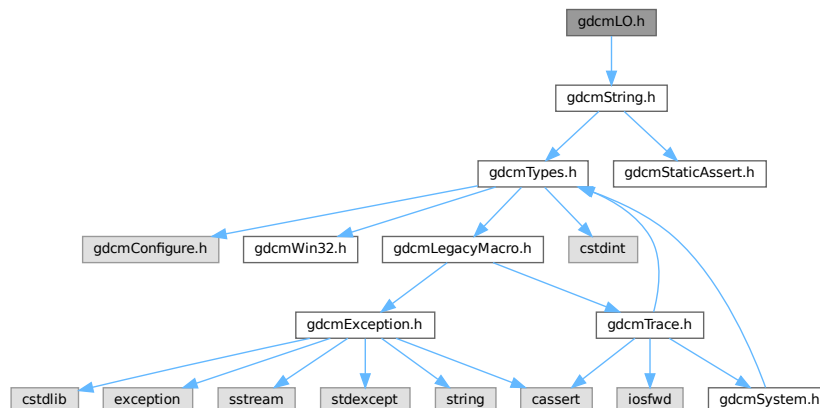
00278     {
00279         assert(0 && "Should not happen");
00280         return os;
00281     }
00282 }
00283 // Self
00284 NestedDataSet.Write<TDE,TSwap>(os);
00285 if( ValueLengthField.IsUndefined() )
00286 {
00287     const Tag itemDelItem(0xfffe,0xe00d);
00288     itemDelItem.Write<TSwap>(os);
00289     VL zero = 0;
00290     zero.Write<TSwap>(os);
00291 }
00292
00293 return os;
00294 }
00295
00296 /*
00297 There are three special SQ related Data Elements that are not ruled by the VR encoding rules conveyed
00298 by the Transfer Syntax. They shall be encoded as Implicit VR. These special Data Elements are Item
00299 (FFFE,E000), Item Delimitation Item (FFFE,E00D), and Sequence Delimitation Item (FFFE,E0DD).
00300 However, the Data Set within the Value Field of the Data Element Item (FFFE,E000) shall be encoded
00301 according to the rules conveyed by the Transfer Syntax.
00302 */
00303 bool FindDataElement(const Tag &t) const {
00304     return NestedDataSet.FindDataElement( t );
00305 }
00306
00307 private:
00308     /* NESTED DATA SET a Data Set contained within a Data Element of an other Data Set.
00309      * May be nested recursively.
00310      * Only Data Elements with VR = SQ may, themselves, contain Data Sets
00311      */
00312     DataSet NestedDataSet;
00313 };
00314 //-----
00315 inline std::ostream& operator<<(std::ostream& os, const Item &val)
00316 {
00317     os << val.TagField;
00318     os << "\t" << val.ValueLengthField << "\n";
00319     val.NestedDataSet.Print( os, "\t" );
00320
00321     return os;
00322 }
00323
00324
00325 } // end namespace gdcm_ns
00326
00327 #include "gdcmItem.txx"
00328
00329 #endif //GDCMITEM_H

```

## 11.151 gdcmLO.h File Reference

```
#include "gdcmString.h"
```

Include dependency graph for gdcmLO.h:



### Classes

- class [gdcm::LO](#)  
[LO](#).

### Namespaces

- namespace [gdcm](#)

## 11.152 gdcmLO.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012  =====*/
00013  #ifndef GDCMLO_H
00014  #define GDCMLO_H
00015
00016  #include "gdcmString.h"
00017
00018  namespace gdcm

```

```

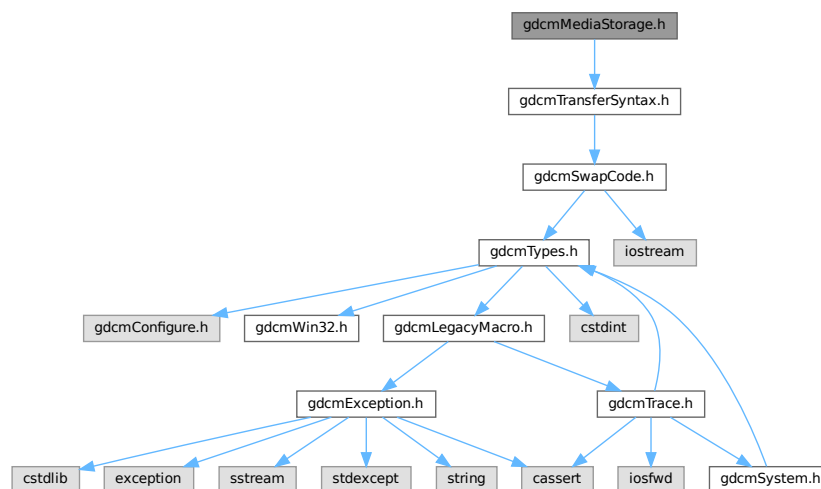
00020 {
00021
00027 class /*GDCM_EXPORT*/ LO : public String<'\\',64> /* PLEASE do not export me */
00028 {
00029 public:
00030     // typedef are not inherited:
00031     typedef String<'\\',64> Superclass;
00032     typedef Superclass::value_type      value_type;
00033     typedef Superclass::pointer         pointer;
00034     typedef Superclass::reference        reference;
00035     typedef Superclass::const_reference  const_reference;
00036     typedef Superclass::size_type        size_type;
00037     typedef Superclass::difference_type  difference_type;
00038     typedef Superclass::iterator         iterator;
00039     typedef Superclass::const_iterator   const_iterator;
00040     typedef Superclass::reverse_iterator reverse_iterator;
00041     typedef Superclass::const_reverse_iterator const_reverse_iterator;
00042
00043     // LO constructors.
00044     LO(): Superclass() {}
00045     LO(const value_type* s): Superclass(s) {}
00046     LO(const value_type* s, size_type n): Superclass(s, n) {}
00047     LO(const Superclass& s, size_type pos=0, size_type n=npos):
00048         Superclass(s, pos, n) {}
00049
00050     bool IsValid() const {
00051         if( !Superclass::IsValid() ) return false;
00052         // Implementation specific:
00053         return true;
00054     }
00055 };
00056
00057 } // end namespace gdcmm
00058
00059 #endif //GDCMLO_H

```

## 11.153 gdcmmMediaStorage.h File Reference

#include "gdcmmTransferSyntax.h"

Include dependency graph for gdcmmMediaStorage.h:





This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::MediaStorage](#)  
*MediaStorage.*

## Namespaces

- namespace [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const MediaStorage &ms)`

## 11.154 gdcmMediaStorage.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 #ifndef GDCMMEDIASTORAGE_H
00015 #define GDCMMEDIASTORAGE_H
00016
00017 #include "gdcmTransferSyntax.h"
00018
00019 namespace gdcm { class Tag; }
00020 namespace gdcm_ns
00021 {
00022     #if !defined(SWIGPYTHON) && !defined(SWIGSHARP) && !defined(SWIGJAVA) && !defined(SWIGPHP)
00023     using namespace gdcm;
00024     #endif
00025     class DataSet;
00026     class FileMetaInformation;
00027     class File;
00028
00029     // WARNING: This class will be deprecated in the future. There is no reason to extend this class.
00030     // Please check the gdcm::UIDs class if adding new well known UID.
00031
00043     class GDCM_EXPORT MediaStorage
00044     {
00045     public:
00046         typedef enum {
00047             MediaStorageDirectoryStorage = 0,
00048             ComputedRadiographyImageStorage,
00049             DigitalXRayImageStorageForPresentation,

```

```

00050     DigitalXRayImageStorageForProcessing,
00051     DigitalMammographyImageStorageForPresentation,
00052     DigitalMammographyImageStorageForProcessing,
00053     DigitalIntraoralXrayImageStorageForPresentation,
00054     DigitalIntraoralXrayImageStorageForProcessing,
00055     CTImageStorage,
00056     EnhancedCTImageStorage,
00057     UltrasoundImageStorageRetired,
00058     UltrasoundImageStorage,
00059     UltrasoundMultiFrameImageStorageRetired,
00060     UltrasoundMultiFrameImageStorage,
00061     MRImageStorage,
00062     EnhancedMRImageStorage,
00063     MRSpectroscopyStorage,
00064     NuclearMedicineImageStorageRetired,
00065     SecondaryCaptureImageStorage,
00066     MultiframeSingleBitSecondaryCaptureImageStorage,
00067     MultiframeGrayscaleByteSecondaryCaptureImageStorage,
00068     MultiframeGrayscaleWordSecondaryCaptureImageStorage,
00069     MultiframeTrueColorSecondaryCaptureImageStorage,
00070     StandaloneOverlayStorage,
00071     StandaloneCurveStorage,
00072     LeadECGWaveformStorage, // 12-
00073     GeneralECGWaveformStorage,
00074     AmbulatoryECGWaveformStorage,
00075     HemodynamicWaveformStorage,
00076     CardiacElectrophysiologyWaveformStorage,
00077     BasicVoiceAudioWaveformStorage,
00078     StandaloneModalityLUTStorage,
00079     StandaloneVOILUTStorage,
00080     GrayscaleSoftcopyPresentationStateStorageSOPClass,
00081     XRayAngiographicImageStorage,
00082     XRayRadiofluoroscopicImageStorage,
00083     XRayAngiographicBiPlaneImageStorageRetired,
00084     NuclearMedicineImageStorage,
00085     RawDataStorage,
00086     SpacialRegistrationStorage, // Spatial
00087     SpacialFiducialsStorage, // Spatial..
00088     PETImageStorage,
00089     RTImageStorage,
00090     RTDoseStorage,
00091     RTStructureSetStorage,
00092     RTPlanStorage,
00093     CSANonImageStorage,
00094     Philips3D,
00095     EnhancedSR,
00096     BasicTextSR,
00097     HardcopyGrayscaleImageStorage,
00098     ComprehensiveSR,
00099     DetachedStudyManagementSOPClass,
00100     EncapsulatedPDFStorage,
00101     EncapsulatedCDASTorage,
00102     StudyComponentManagementSOPClass,
00103     DetachedVisitManagementSOPClass,
00104     DetachedPatientManagementSOPClass,
00105     VideoEndoscopicImageStorage,
00106     GeneralElectricMagneticResonanceImageStorage,
00107     GEPrivate3DModelStorage,
00108     ToshibaPrivateDataStorage,
00109     MammographyCADSR,
00110     KeyObjectSelectionDocument,
00111     HangingProtocolStorage,
00112     ModalityPerformedProcedureStepSOPClass,
00113     PhilipsPrivateMRSyntheticImageStorage,
00114     VLPhotographicImageStorage,
00115     SegmentationStorage, // "1.2.840.10008.5.1.4.1.1.66.4"
00116     RTIonPlanStorage, // 1.2.840.10008.5.1.4.1.1.481.8
00117     XRay3DAngiographicImageStorage, // 1.2.840.10008.5.1.4.1.1.13.1.1
00118     EnhancedXAImageStorage,
00119     RTIonBeamsTreatmentRecordStorage, // 1.2.840.10008.5.1.4.1.1.481.9
00120     SurfaceSegmentationStorage, // "1.2.840.10008.5.1.4.1.1.66.5"
00121     VLWholeSlideMicroscopyImageStorage, // 1.2.840.10008.5.1.4.1.1.77.1.6
00122     RTTreatmentSummaryRecordStorage, // 1.2.840.10008.5.1.4.1.1.481.7
00123     EnhancedUSVolumeStorage, // 1.2.840.10008.5.1.4.1.1.6.2
00124     XRayRadiationDoseSR, // 1.2.840.10008.5.1.4.1.1.88.67
00125     VLEndoscopicImageStorage, // 1.2.840.10008.5.1.4.1.1.77.1.1
00126     BreastTomosynthesisImageStorage, // 1.2.840.10008.5.1.4.1.1.13.1.3
00127     FujiPrivateCRImageStorage, // 1.2.392.200036.9125.1.1.2
00128     OphthalmicPhotography8BitImageStorage, // 1.2.840.10008.5.1.4.1.1.77.1.5.1
00129     OphthalmicTomographyImageStorage, // 1.2.840.10008.5.1.4.1.1.77.1.5.4
00130     VLMicroscopicImageStorage,

```

```

00131     EnhancedPETImageStorage,
00132     VideoPhotographicImageStorage,
00133     XRay3DCraniofacialImageStorage,
00134     IVOCForPresentation,
00135     IVOCForProcessing,
00136     LegacyConvertedEnhancedCTImageStorage,
00137     LegacyConvertedEnhancedMRImageStorage,
00138     LegacyConvertedEnhancedPETImageStorage,
00139     BreastProjectionXRayImageStorageForPresentation,
00140     BreastProjectionXRayImageStorageForProcessing,
00141     HardcopyColorImageStorage,
00142     EnhancedMRColorImageStorage,
00143     FujiPrivateMammoCRImageStorage,
00144     OphthalmicPhotographyl6BitImageStorage,
00145     VideoMicroscopicImageStorage,
00146     MS_END
00147 } MStype; // Media Storage Type
00148
00149 typedef enum {
00150     NoObject = 0, // DICOMDIR
00151     Video, // Most common, include image, video and volume
00152     Waveform, // Isn't it simply a 1D video ?
00153     Audio, // ???
00154     PDF,
00155     URI, // URL...
00156     Segmentation, // TODO
00157     ObjectEnd
00158 } ObjectType;
00159
00161 static const char* GetMSString(MStype ts);
00162
00164 const char* GetString() const;
00165 static MStype GetMStype(const char *str);
00166
00167 MediaStorage(MStype type = MS_END):MSField(type) {}
00168
00171 static bool IsImage(MStype ts);
00172
00173 operator MStype () const { return MSField; }
00174
00175 const char *GetModality() const;
00176 unsigned int GetModalityDimension() const;
00177
00178 static unsigned int GetNumberOfMStype();
00179 static unsigned int GetNumberOfMSString();
00180 static unsigned int GetNumberOfModality();
00181
00182
00187 bool SetFromFile(File const &file);
00188
00191 bool SetFromDataSet(DataSet const &ds); // Will get the SOP Class UID
00192 bool SetFromHeader(FileMetaInformation const &fmi); // Will get the Media Storage SOP Class UID
00193 bool SetFromModality(DataSet const &ds);
00194 void GuessFromModality(const char *modality, unsigned int dimension = 2);
00195
00196 friend std::ostream &operator<<(std::ostream &os, const MediaStorage &ms);
00197
00198 bool IsUndefined() const { return MSField == MS_END; }
00199
00200 protected:
00201 void SetFromSourceImageSequence(DataSet const &ds);
00202
00203 private:
00204 bool SetFromDataSetOrHeader(DataSet const &ds, const Tag &tag);
00205
00206 std::string GetFromDataSetOrHeader(DataSet const &ds, const Tag &tag);
00207 std::string GetFromHeader(FileMetaInformation const &fmi);
00208 std::string GetFromDataSet(DataSet const &ds);
00209
00210 private:
00211 MStype MSField;
00212 };
00213 //-----
00214 inline std::ostream &operator<<(std::ostream &_os, const MediaStorage &ms)
00215 {
00216     const char *msstring = MediaStorage::GetMSString(ms);
00217     _os << (msstring ? msstring : "INVALID MEDIA STORAGE");
00218     return _os;
00219 }
00220 }
00221

```



## Namespaces

- namespace [gdcm](#)

## Functions

- [std::ostream & gdcm::operator<<](#) (std::ostream &os, const [MrProtocol](#) &d)

## 11.156 gdcmMrProtocol.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMMRPROTOCOL_H
00015 #define GDCMMRPROTOCOL_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmDataSet.h"
00019
00020 namespace gdcm
00021 {
00022     class ByteValue;
00023     /*
00024      * Everything done in this code is for the sole purpose of writing interoperable
00025      * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
00026      * If you believe anything in this code violates any law or any of your rights,
00027      * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
00028      * find a solution.
00029      */
00030     //-----
00031
00032     class DataElement;
00033     class GDCM_EXPORT MrProtocol
00034     {
00035     public:
00036         friend std::ostream& operator<<(std::ostream &os, const MrProtocol &d);
00037         MrProtocol();
00038         ~MrProtocol();
00039
00040         bool Load( const ByteValue * bv, const char * str, int version );
00041         void Print(std::ostream &os) const;
00042
00043         int GetVersion() const;
00044
00045         const char * GetMrProtocolByName(const char *name) const;
00046
00047         bool FindMrProtocolByName(const char *name) const;
00048
00049         struct Vector3
00050         {
00051             double dSag;
00052             double dCor;
00053             double dTra;
00054         };
00055         struct Slice
00056         {
00057             Vector3 Normal;
00058             Vector3 Position;
00059         };
00060     };

```

```

00063     struct SliceArray
00064     {
00065         std::vector< Slice > Slices;
00066     };
00067     bool GetSliceArray( MrProtocol::SliceArray & sa ) const;
00068
00069 private:
00070     struct Element;
00071     struct Internals;
00072     Internals *Pimpl;
00073 };
00074 //-----
00075 inline std::ostream& operator<<(std::ostream &os, const MrProtocol &d)
00076 {
00077     d.Print( os );
00078     return os;
00079 }
00080
00081 } // end namespace gdcmm
00082 //-----
00083 #endif //GDCMMRPROTOCOL_H

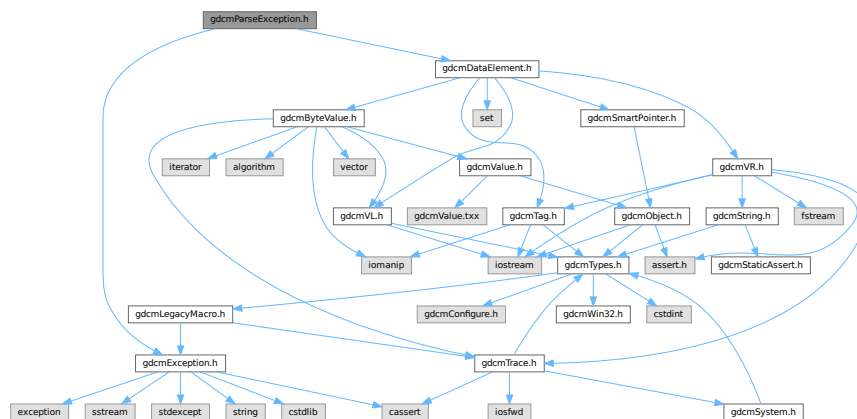
```

## 11.157 gdcmmParseException.h File Reference

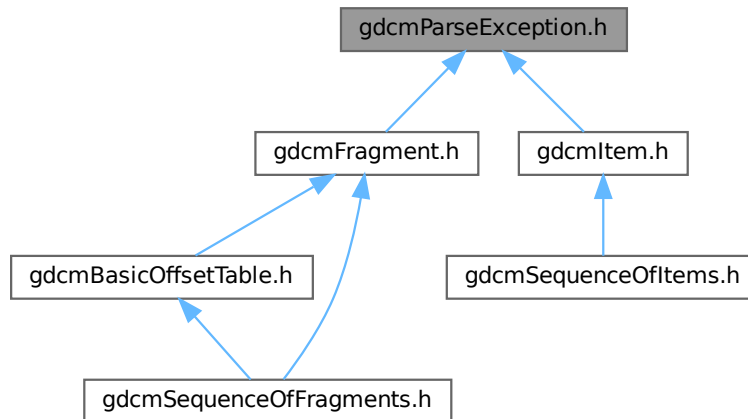
```
#include "gdcmmException.h"
```

```
#include "gdcmmDataElement.h"
```

Include dependency graph for gdcmmParseException.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::ParseException](#)  
*ParseException* Standard exception handling object.

## Namespaces

- namespace [gdcm](#)

## 11.158 gdcmParseException.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMPARSEEXCEPTION_H
00015 #define GDCMPARSEEXCEPTION_H
00016
00017 #include "gdcmException.h"
00018 #include "gdcmDataElement.h"
00019
00020 // Disable clang warning "dynamic exception specifications are deprecated".
00021 // We need to be C++03 and C++11 compatible, and if we remove the 'throw()'

```

```

00022 // specifier we'll get an error in C++03 by not matching the superclass.
00023 #if defined(__clang__) && defined(__has_warning)
00024 # if __has_warning("-Wdeprecated")
00025 #   pragma clang diagnostic push
00026 #   pragma clang diagnostic ignored "-Wdeprecated"
00027 # endif
00028 #endif
00029
00030 namespace gdcms
00031 {
00032     class ParseException : public Exception
00033     {
00034     public:
00035         ParseException() = default;
00036         ~ParseException() throw() override {}
00037
00038         ParseException &operator= ( const ParseException &orig )
00039         {
00040             LastElement = orig.LastElement;
00041             return *this;
00042         }
00043         ParseException(const ParseException& orig):Exception(orig)
00044         {
00045             LastElement = orig.LastElement;
00046         }
00047
00048         /* virtual bool operator==( const ParseException &orig )
00049         {
00050             return true;
00051         }*/
00052
00053         /*
00054         // Multiple calls to what ??
00055         const char* what() const throw()
00056         {
00057             static std::string strwhat;
00058             std::ostringstream oswhat;
00059             oswhat << File << ":" << Line << ":\n";
00060             oswhat << Description;
00061             strwhat = oswhat.str();
00062             return strwhat.c_str();
00063         }
00064         */
00065         void SetLastElement(DataElement& de)
00066         {
00067             LastElement = de;
00068         }
00069         const DataElement& GetLastElement() const { return LastElement; }
00070
00071     private:
00072         // Store last parsed element before error:
00073         DataElement LastElement;
00074     };
00075 } // end namespace gdcms
00076
00077 // Undo warning suppression.
00078 #if defined(__clang__) && defined(__has_warning)
00079 # if __has_warning("-Wdeprecated")
00080 #   pragma clang diagnostic pop
00081 # endif
00082 #endif
00083
00084 #endif

```

## 11.159 gdcmsParser.h File Reference

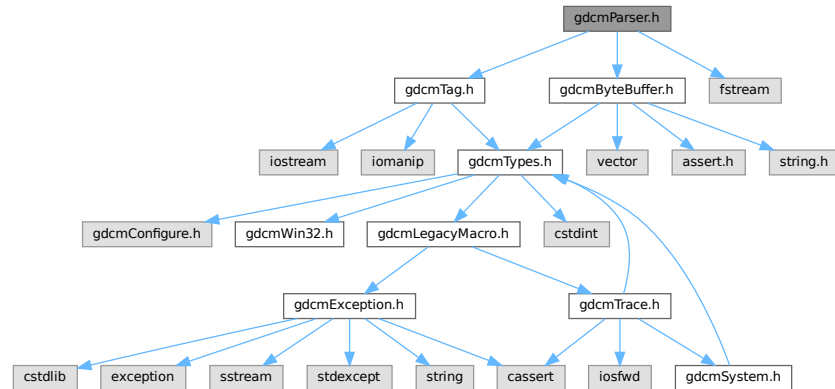
```

#include "gdcmsTag.h"
#include "gdcmsByteBuffer.h"
#include <fstream>

```



Include dependency graph for gdcmParser.h:



## Classes

- class [gdcm::Parser](#)  
*Parser* ala *XML\_Parser* from *expat* (SAX)

## Namespaces

- namespace [gdcm](#)

## 11.160 gdcmParser.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012   =====*/
00013
00014
00015 #ifndef GDCMPARSER_H
00016 #define GDCMPARSER_H
00017
00018 #include "gdcmTag.h"
00019 #error do not use
00020 #include "gdcmByteBuffer.h"
00021
00022 #include <fstream> // std::ifstream
00023
00024 namespace gdcm
00025 {
00032 class GDCM_EXPORT Parser /*: private IStream*/

```

```

00033 {
00034 public:
00035     typedef enum {
00036         NoError,
00037         NoMemoryError,
00038         SyntaxError,
00039         NoElementsError,
00040         TagMismatchError,
00041         DuplicateAttributeError,
00042         JunkAfterDocElementError,
00043         UndefinedEntityError,
00044         UnexpectedStateError
00045     } ErrorType;
00046
00047     Parser() : UserData(0), Buffer(), ErrorCode(NoError) {}
00048     ~Parser() {}
00049
00050     // Parse some more of the document. The string s is a buffer containing
00051     // part (or perhaps all) of the document. The number of bytes of s that
00052     // are part of the document is indicated by len. This means that s
00053     // doesn't have to be null terminated. It also means that if len is
00054     // larger than the number of bytes in the block of memory that s points
00055     // at, then a memory fault is likely. The isFinal parameter informs the
00056     // parser that this is the last piece of the document. Frequently, the
00057     // last piece is empty (i.e. len is zero.) If a parse error occurred,
00058     // it returns 0. Otherwise it returns a non-zero value.
00059     bool Parse(const char* s, int len, bool isFinal);
00060
00061     // Set handlers for start and end tags. Attributes are passed to the
00062     // start handler as a pointer to a vector of char pointers. Each
00063     // attribute seen in a start (or empty) tag occupies 2 consecutive places
00064     // in this vector: the attribute name followed by the attribute value.
00065     // These pairs are terminated by a null pointer.
00066     typedef void (*StartElementHandler) (void *userData,
00067                                         const Tag &tag,
00068                                         const char *atts[]);
00069     typedef void (*EndElementHandler) (void *userData, const Tag &name);
00070     void SetElementHandler(StartElementHandler start, EndElementHandler end);
00071
00072     // Return what type of error has occurred.
00073     ErrorType GetErrorCode() const;
00074
00075     // Return a string describing the error corresponding to code.
00076     // The code should be one of the enums that can be returned from
00077     // GetErrorCode.
00078     static const char *GetErrorString(ErrorType const &err);
00079
00080     // Return the byte offset of the position.
00081     unsigned long GetCurrentByteIndex() const;
00082
00083     // Miscellaneous functions
00084
00085     // The functions in this section either obtain state information from
00086     // the parser or can be used to dynamically set parser options.
00087
00088     // This sets the user data pointer that gets passed to handlers.
00089     void SetUserData(void *userData);
00090
00091     // This returns the user data pointer that gets passed to handlers.
00092     void * GetUserData() const;
00093
00094 protected:
00095
00096     // This is just like Parse, except in this case expat provides the buffer.
00097     // By obtaining the buffer from expat with the GetBuffer function,
00098     // the application can avoid double copying of the input.
00099     bool ParseBuffer(int len, bool isFinal);
00100
00101     // Obtain a buffer of size len to read a piece of the document into.
00102     // A NULL value is returned if expat can't allocate enough memory for
00103     // this buffer. This has to be called prior to every call to ParseBuffer.
00104     char *GetBuffer(int len);
00105
00106     ErrorType Process();
00107
00108 private:
00109     std::ifstream Stream;
00110     void* UserData;
00111     ByteBuffer Buffer;
00112     ErrorType ErrorCode;
00113

```

## 11.161 gdcnPDBElement.h File Reference

```
graph BT
    gdc[PDBElement.h] --> gdh[gdcPDBHeader.h]
```

- class `gdcm::PDBElement`  
*Class to represent a PDB Element.*

## Namespaces

- namespace `gdcm`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const PDBelement &val)`

## 11.162 gdcmPDBelement.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMPDBeLEMENT_H
00015 #define GDCMPDBeLEMENT_H
00016
00017 #include "gdcmTag.h"
00018 #include "gdcmVM.h"
00019 #include "gdcmVR.h"
00020 #include "gdcmByteValue.h"
00021 #include "gdcmSmartPointer.h"
00022
00023 namespace gdcm
00024 {
00025     class GDCM_EXPORT PDBelement
00026     {
00027     public:
00028         PDBelement() = default;
00029
00030         friend std::ostream& operator<<(std::ostream &os, const PDBelement &val);
00031
00032         const char *GetName() const { return NameField.c_str(); }
00033         void SetName(const char *name) { NameField = name; }
00034
00035         const char *GetValue() const { return ValueField.c_str(); }
00036         void SetValue(const char *value) { ValueField = value; }
00037
00038         bool operator==(const PDBelement &de) const
00039         {
00040             return ValueField == de.ValueField
00041                 && NameField == de.NameField;
00042         }
00043
00044     protected:
00045         std::string NameField;
00046         std::string ValueField;
00047     };
00048
00049 //-----
00050 inline std::ostream& operator<<(std::ostream &os, const PDBelement &val)
00051 {
00052     os << val.NameField;
00053     os << " \n";
00054     os << val.ValueField;
00055     os << " \n";
00056
00057     return os;
00058 }
00059
00060 } // end namespace gdcm
00061
00062 #endif //GDCMPDBeLEMENT_H

```



```

00014 #ifndef GDCMPDBHEADER_H
00015 #define GDCMPDBHEADER_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmDataSet.h"
00019 #include "gdcmPDBElement.h"
00020
00021 namespace gdcm
00022 {
00023
00024 /*
00025  * Everything done in this code is for the sole purpose of writing interoperable
00026  * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
00027  * If you believe anything in this code violates any law or any of your rights,
00028  * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
00029  * find a solution.
00030  */
00031 //-----
00032
00033 class DataElement;
00034 class PrivateTag;
00053 class GDCM_EXPORT PDBHeader
00054 {
00055     friend std::ostream& operator<<(std::ostream &_os, const PDBHeader &d);
00056 public:
00057     PDBHeader() = default;
00058     ~PDBHeader() = default;
00059
00061     bool LoadFromDataElement(DataElement const &de);
00062
00064     void Print(std::ostream &os) const;
00065
00067     static const PrivateTag & GetPDBInfoTag();
00068
00071     const PDBElement &GetPDBElementByName(const char *name);
00072
00074     bool FindPDBElementByName(const char *name);
00075
00076 protected:
00077     const PDBElement& GetPDBEEnd() const;
00078
00079 private:
00080     int readprotocoldatablock(const char *input, size_t inputlen, bool verbose);
00081     std::vector<PDBElement> InternalPDBDataSet;
00082     static PDBElement PDBEEnd;
00083     bool IsXML;
00084     std::string xmltxt;
00085 };
00086 //-----
00087 inline std::ostream& operator<<(std::ostream &os, const PDBHeader &d)
00088 {
00089     d.Print( os );
00090     return os;
00091 }
00092
00093 } // end namespace gdcm
00094 //-----
00095 #endif //GDCMPDBHEADER_H

```

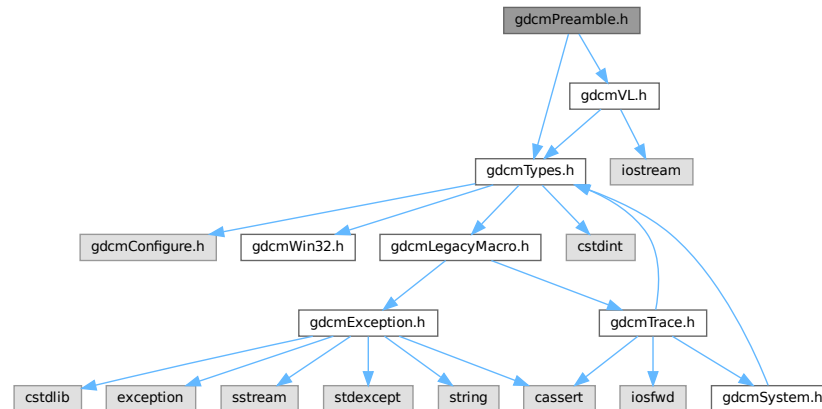
## 11.165 gdcmPreamble.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmVL.h"

```

Include dependency graph for gdcmPreamble.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::Preamble](#)  
*DICOM Preamble (Part 10)*

## Namespaces

- namespace [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Preamble &val)`

## 11.166 gdcmPreamble.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMPREAMBLE_H
00015  #define GDCMPREAMBLE_H
00016
00017  #include "gdcmTypes.h"
00018  #include "gdcmVL.h"
00019
00020  namespace gdcm
00021  {
00022
00026  class GDCM_EXPORT Preamble
00027  {
00028  public:
00029      Preamble();
00030      ~Preamble();
00031
00032      friend std::ostream &operator<<(std::ostream &_os, const Preamble &_val);
00033
00035      void Clear();
00036
00038      void Valid();
00039      void Create();
00040      void Remove();
00041
00043      std::istream &Read(std::istream &is);
00044
00046      std::ostream const &Write(std::ostream &os) const;
00047
00049      void Print(std::ostream &os) const;
00050
00052      const char *GetInternal() const { return Internal; }
00053
00055      bool IsEmpty() const { return !Internal; }
00056
00058      VL GetLength() const { return 128 + 4; }
00059
00060      Preamble(Preamble const &):Internal(nullptr)
00061      {
00062          Create();
00063      }
00064      Preamble& operator=(Preamble const &)
00065      {
00066          Create();
00067          return *this;
00068      }
00069  protected:
00070      //
00071      bool IsValid() const {
00072          // is (IsValid == true) => Internal was read
00073          return true;
00074      }
00075
00076  private:
00077      char *Internal;
00078  };
00079  //-----
00080  inline std::ostream& operator<<(std::ostream &os, const Preamble &val)
00081  {
00082      os << val.Internal;
00083      return os;
00084  }
00085
00086  } // end namespace gdcm
00087

```

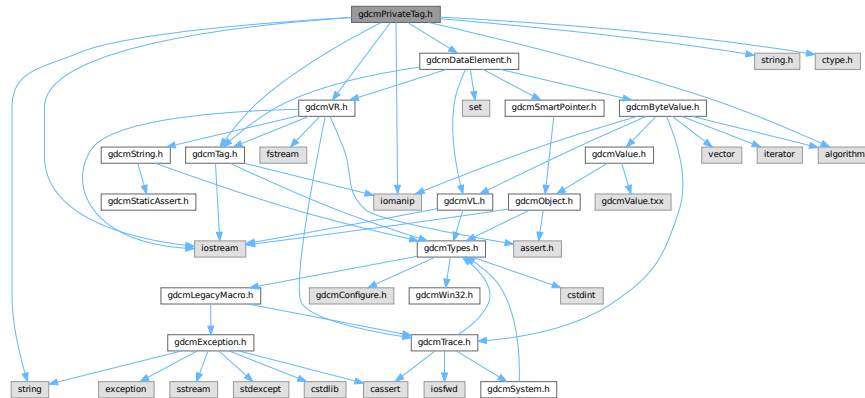


```
00088 #endif //GDCMPREAMBLE_H
```

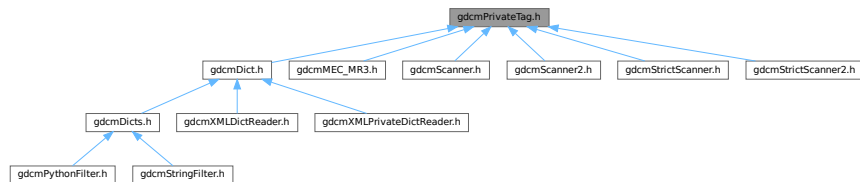
## 11.167 gdcmPrivateTag.h File Reference

```
#include "gdcmTag.h"
#include "gdcmVR.h"
#include "gdcmDataElement.h"
#include <iostream>
#include <iomanip>
#include <string>
#include <algorithm>
#include <string.h>
#include <ctype.h>
```

Include dependency graph for gdcmPrivateTag.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [gdcm::PrivateTag](#)

*Class to represent a Private DICOM Data [Element](#) ([Attribute](#)) [Tag](#) ([Group](#), [Element](#), [Owner](#))*

## Namespaces

- namespace [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const PrivateTag &val)`

## 11.168 gdcmPrivateTag.h

[Go to the documentation of this file.](#)

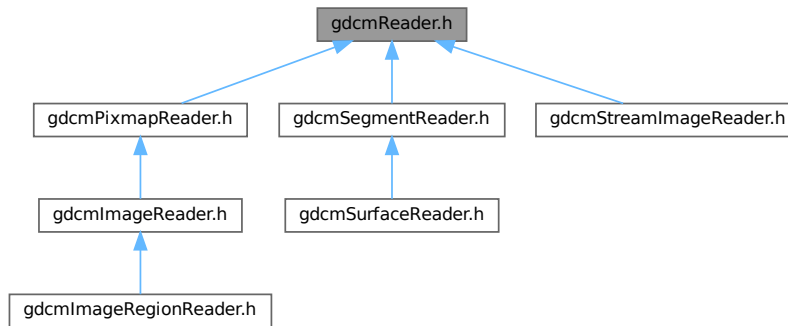
```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 #ifndef GDCMPRIVATETAG_H
00015 #define GDCMPRIVATETAG_H
00016
00017 #include "gdcmTag.h"
00018 #include "gdcmVR.h"
00019 #include "gdcmDataElement.h"
00020
00021 #include <iostream>
00022 #include <iomanip>
00023 #include <string>
00024 #include <algorithm>
00025
00026 #include <string.h> // strlen
00027 #include <ctype.h> // tolower
00028
00029 namespace gdcm_ns
00030 {
00031
00032     // TODO: We could save some space since we only store 8bits for element
00033     class GDCM_EXPORT PrivateTag : public Tag
00034     {
00035     public:
00036         PrivateTag(uint16_t group = 0, uint16_t element = 0, const char *owner =
00037             ""):Tag(group,element),Owner(owner ? LOComp::Trim(owner) : "") {
00038             // truncate the high bits
00039             SetElement( (uint8_t)element );
00040         }
00041         PrivateTag( Tag const & t, const char *owner = ""):Tag(t),Owner(owner ? LOComp::Trim(owner) : "") {
00042             // truncate the high bits
00043             SetElement( (uint8_t)t.GetElement());
00044         }
00045
00046         const char *GetOwner() const { return Owner.c_str(); }
00047         void SetOwner(const char *owner) { if(owner) Owner = LOComp::Trim(owner); }
00048
00049         PrivateTag &operator=(const PrivateTag &_val)
00050         {
00051             SetElementTag( _val.GetElementTag() );
00052             Owner = _val.Owner;
00053             return *this;
00054         }
00055
00056         bool operator==(const Tag &_val) const
00057         {
00058             return GetElementTag() == _val.GetElementTag();
00059         }
00060
00061     };
00062
00063 
```

## 11.169 gdcmReader.h File Reference

[illegible]

This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::Reader](#)  
*Reader* ala DOM (Document *Object* Model)

## Namespaces

- namespace [gdcm](#)

## 11.170 gdcmReader.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMREADER_H
00015 #define GDCMREADER_H
00016
00017 #include "gdcmFile.h"
00018
00019 #include <fstream>
00020
00021 namespace gdcm_ns
00022 {
00023     class StreamImageReader;
00024     class GDCM_EXPORT Reader
00025     {
00026     public:
00027         Reader();
00028         virtual ~Reader();
00029     };
00030 }

```

```

00058
00060 virtual bool Read(); // Execute()
00061
00064 void SetFileName(const char *filename_native);
00065
00067 void SetStream(std::istream &input_stream) {
00068     Stream = &input_stream;
00069 }
00070
00072 const File &GetFile() const { return *F; }
00073
00075 File &GetFile() { return *F; }
00076
00078 void SetFile(File& file) { F = &file; }
00079
00082 bool ReadUpToTag(const Tag &tag, std::set<Tag> const &skiptags = std::set<Tag>() );
00083
00085 bool ReadSelectedTags(std::set<Tag> const &tags, bool readvalues = true);
00086
00088 bool ReadSelectedPrivateTags(std::set<PrivateTag> const &ptags, bool readvalues = true);
00089
00092 bool CanRead() const;
00093
00096 size_t GetStreamCurrentPosition() const;
00097
00098 protected:
00099     bool ReadPreamble();
00100     bool ReadMetaInformation();
00101     bool ReadDataSet();
00102
00103     SmartPointer<File> F;
00104
00105     friend class StreamImageReader; //need to be friended to be able to grab the GetStreamPtr
00106
00107     //this function is added for the StreamImageReader, which needs to read
00108     //up to the pixel data and then stops right before reading the pixel data.
00109     //it's used to get that position, so that reading can continue
00110     //apace once the read function is called.
00111     //so, this function gets the stream directly, and then allows for position information
00112     //from the tellg function, and allows for stream/pointer manip in order
00113     //to read the pixel data. Note, of course, that reading pixel elements
00114     //will still have to be subject to endianness swaps, if necessary.
00115     std::istream* GetStreamPtr() const { return Stream; }
00116
00117 private:
00118     template <typename T_Caller>
00119     bool InternalReadCommon(const T_Caller &caller);
00120     TransferSyntax GuessTransferSyntax();
00121     std::istream *Stream;
00122     std::ifstream *Ifstream;
00123
00124     // prevent copy/move to avoid 2 ifstream leak
00125     Reader(const Reader &) = delete;
00126     Reader &operator=(const Reader &) = delete;
00127     Reader(const Reader &&) = delete;
00128     Reader &operator=(const Reader &&) = delete;
00129 };
00130
00137 } // end namespace gdcm_ns
00138
00139
00140 #endif //GDCMREADER_H

```

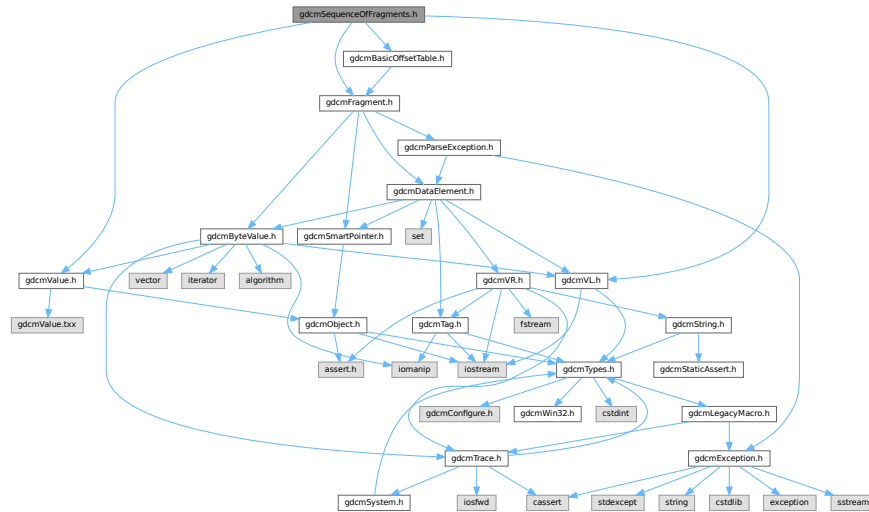
## 11.171 gdcmSequenceOfFragments.h File Reference

```

#include "gdcmValue.h"
#include "gdcmVL.h"
#include "gdcmFragment.h"
#include "gdcmBasicOffsetTable.h"

```

Include dependency graph for `gdcSequenceOfFragments.h`:



## Classes

- class [gdc::SequenceOfFragments](#)  
Class to represent a Sequence Of Fragments.

## Namespaces

- namespace [gdc](#)

## 11.172 gdcSequenceOfFragments.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdc.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 #ifndef GDCMSEQUENCEOFFRAGMENTS_H
00015 #define GDCMSEQUENCEOFFRAGMENTS_H
00016
00017 #include "gdcValue.h"
00018 #include "gdcVL.h"
00019 #include "gdcFragment.h"
00020 #include "gdcBasicOffsetTable.h"
00021
00022 namespace gdc_ns
00023 {

```

```

00024
00025 // FIXME gdcmSequenceOfItems and gdcmSequenceOfFragments
00026 // should be rethink (duplicate code)
00031 class GDCM_EXPORT SequenceOfFragments : public Value
00032 {
00033 public:
00034 // Typdefs:
00035 typedef std::vector<Fragment> FragmentVector;
00036 typedef FragmentVector::size_type SizeType;
00037 typedef FragmentVector::iterator Iterator;
00038 typedef FragmentVector::const_iterator ConstIterator;
00039 Iterator Begin() { return Fragments.begin(); }
00040 Iterator End() { return Fragments.end(); }
00041 ConstIterator Begin() const { return Fragments.begin(); }
00042 ConstIterator End() const { return Fragments.end(); }
00043
00044 SequenceOfFragments():Table(),SequenceLengthField(0xFFFFFFFF) { }
00046
00048 VL GetLength() const override {
00049     return SequenceLengthField;
00050 }
00051
00053 void SetLength(VL length) override {
00054     SequenceLengthField = length;
00055 }
00056
00058 void Clear() override;
00059
00061 void AddFragment(Fragment const &item);
00062
00063 // Compute the length of all fragments (and fragments only!).
00064 // Basically the size of the PixelData as stored (in bytes).
00065 unsigned long ComputeByteLength() const;
00066
00067 // Compute the length of fragments (in bytes)+ length of tag...
00068 // to be used for computation of Group Length
00069 VL ComputeLength() const;
00070
00071 // Get the buffer
00072 bool GetBuffer(char *buffer, unsigned long length) const;
00073 bool GetFragBuffer(unsigned int fragNb, char *buffer, unsigned long &length) const;
00074
00075 SizeType GetNumberOfFragments() const;
00076 const Fragment& GetFragment(SizeType num) const;
00077
00078 // Write the buffer of each fragment (call WriteBuffer on all Fragments, which are
00079 // ByteValue). No Table information is written.
00080 bool WriteBuffer(std::ostream &os) const;
00081
00082 const BasicOffsetTable &GetTable() const { return Table; }
00083 BasicOffsetTable &GetTable() { return Table; }
00084
00085 template <typename TSwap>
00086 std::istream& Read(std::istream &is, bool readvalues = true)
00087 {
00088     assert( SequenceLengthField.IsUndefined() );
00089     ReadPreValue<TSwap>(is);
00090     return ReadValue<TSwap>(is, readvalues);
00091 }
00092
00093 template <typename TSwap>
00094 std::istream& ReadPreValue(std::istream &is)
00095 {
00096     // First item is the basic offset table:
00097     #if 0
00098     try
00099     {
00100         Table.Read<TSwap>(is);
00101         gdcmDebugMacro( "Table: " << Table );
00102     }
00103     catch(...)
00104     {
00105         // throw "SIEMENS Icon thingy";
00106         // Bug_Siemens_PrivateIconNoItem.dcm
00107         // First thing first let's rewind
00108         is.seekg(-4, std::ios::cur);
00109         // FF D8 <=> Start of Image (SOI) marker
00110         // FF E0 <=> APP0 Reserved for Application Use
00111         if ( Table.GetTag() == Tag(0xd8ff,0xe0ff) )
00112         {
00113             Table = BasicOffsetTable(); // clear up stuff

```

```

00114         //Table.SetByteValue( "", 0 );
00115         Fragment frag;
00116         if( FillFragmentWithJPEG( frag, is ) )
00117         {
00118             Fragments.push_back( frag );
00119         }
00120         return is;
00121     }
00122     else
00123     {
00124         throw "Catch me if you can";
00125         //assert(0);
00126     }
00127 }
00128 #else
00129     Table.Read<TSwap>(is);
00130     gdcMDebugMacro( "Table: " « Table );
00131 #endif
00132     return is;
00133 }
00134
00135 template <typename TSwap>
00136 std::istream& ReadValue(std::istream &is, bool /*readvalues*/)
00137 {
00138     const Tag seqDelItem(0xfffe,0xe0dd);
00139     // not used for now...
00140     Fragment frag;
00141     try
00142     {
00143         while( frag.Read<TSwap>(is) && frag.GetTag() != seqDelItem )
00144         {
00145             //gdcMDebugMacro( "Frag: " « frag );
00146             Fragments.push_back( frag );
00147         }
00148         assert( frag.GetTag() == seqDelItem && frag.GetVL() == 0 );
00149     }
00150     catch(Exception &ex)
00151     {
00152         (void)ex;
00153 #ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
00154         // that's ok ! In all cases the whole file was read, because
00155         // Fragment::Read only fail on eof() reached 1.
00156         // SIEMENS-JPEG-CorruptFrag.dcm is more difficult to deal with, we have a
00157         // partial fragment, read we decide to add it anyway to the stack of
00158         // fragments (eof was reached so we need to clear error bit)
00159         if( frag.GetTag() == Tag(0xfffe,0xe000) )
00160         {
00161             gdcMWarningMacro( "Pixel Data Fragment could be corrupted. Use file at own risk" );
00162             Fragments.push_back( frag );
00163             is.clear(); // clear the error bit
00164         }
00165         // 2. GENESIS_SIGNA-JPEG-CorruptFrag.dcm
00166         else if ( frag.GetTag() == Tag(0xddff,0x00e0) )
00167         {
00168             assert( Fragments.size() == 1 );
00169             const ByteValue *bv = Fragments[0].GetByteValue();
00170             assert( (unsigned char)bv->GetPointer()[ bv->GetLength() - 1 ] == 0xfe );
00171             // Yes this is an extra copy, this is a bug anyway, go fix YOUR code
00172             Fragments[0].SetByteValue( bv->GetPointer(), bv->GetLength() - 1 );
00173             gdcMWarningMacro( "JPEG Fragment length was declared with an extra byte"
00174                 " at the end: stripped !" );
00175             is.clear(); // clear the error bit
00176         }
00177         // 3. LEICA/WSI
00178         else if ( (frag.GetTag().GetGroup() == 0x00ff)
00179             && ((frag.GetTag().GetElement() & 0x00ff) == 0xe0) )
00180         {
00181             // Looks like there is a mess with offset and odd byte array
00182             // We are going first to backtrack one byte back, and then use a
00183             // ReadBacktrack function which in turn may backtrack up to 10 bytes
00184             // backward. This appears to be working on a set of DICOM/WSI files from
00185             // LEICA
00186             gdcMWarningMacro( "Trying to fix the even-but-odd value length bug #1" );
00187             assert( Fragments.size() );
00188             const size_t lastf = Fragments.size() - 1;
00189             const ByteValue *bv = Fragments[ lastf ].GetByteValue();
00190             const char *a = bv->GetPointer();
00191             gdcMAssertAlwaysMacro( (unsigned char)a[ bv->GetLength() - 1 ] == 0xfe );
00192             Fragments[ lastf ].SetByteValue( bv->GetPointer(), bv->GetLength() - 1 );
00193             is.seekg( -9, std::ios::cur );
00194             assert( is.good() );

```



```

00195     while( frag.ReadBacktrack<TSwap>(is) && frag.GetTag() != seqDelItem )
00196     {
00197         gdcmDebugMacro( "Frag: " << frag );
00198         Fragments.push_back( frag );
00199     }
00200     assert( frag.GetTag() == seqDelItem && frag.GetVL() == 0 );
00201 }
00202 // 4. LEICA/WSI (bis)
00203 else if ( frag.GetTag().GetGroup() == 0xe000 )
00204 {
00205     // Looks like there is a mess with offset and odd byte array
00206     // We are going first to backtrack one byte back, and then use a
00207     // ReadBacktrack function which in turn may backtrack up to 10 bytes
00208     // backward. This appears to be working on a set of DICOM/WSI files from
00209     // LEICA
00210     gdcmWarningMacro( "Trying to fix the even-but-odd value length bug #2" );
00211     assert( Fragments.size() );
00212     const size_t lastf = Fragments.size() - 1;
00213     const ByteValue *bv = Fragments[ lastf ].GetByteValue();
00214     const char *a = bv->GetPointer();
00215     gdcmAssertAlwaysMacro( (unsigned char)a[ bv->GetLength() - 2 ] == 0xfe );
00216     Fragments[ lastf ].SetByteValue( bv->GetPointer(), bv->GetLength() - 2 );
00217     is.seekg( -10, std::ios::cur );
00218     assert( is.good() );
00219     while( frag.ReadBacktrack<TSwap>(is) && frag.GetTag() != seqDelItem )
00220     {
00221         gdcmDebugMacro( "Frag: " << frag );
00222         Fragments.push_back( frag );
00223     }
00224     assert( frag.GetTag() == seqDelItem && frag.GetVL() == 0 );
00225 }
00226 // 5. LEICA/WSI (ter)
00227 else if ( (frag.GetTag().GetGroup() & 0x00ff) == 0x00e0
00228 && (frag.GetTag().GetElement() & 0xff00) == 0x0000 )
00229 {
00230     // Looks like there is a mess with offset and odd byte array
00231     // We are going first to backtrack one byte back, and then use a
00232     // ReadBacktrack function which in turn may backtrack up to 10 bytes
00233     // backward. This appears to be working on a set of DICOM/WSI files from
00234     // LEICA
00235     gdcmWarningMacro( "Trying to fix the even-but-odd value length bug #3" );
00236     assert( Fragments.size() );
00237     const size_t lastf = Fragments.size() - 1;
00238     const ByteValue *bv = Fragments[ lastf ].GetByteValue();
00239     const char *a = bv->GetPointer();
00240     gdcmAssertAlwaysMacro( (unsigned char)a[ bv->GetLength() - 3 ] == 0xfe );
00241     Fragments[ lastf ].SetByteValue( bv->GetPointer(), bv->GetLength() - 3 );
00242     is.seekg( -11, std::ios::cur );
00243     assert( is.good() );
00244     while( frag.ReadBacktrack<TSwap>(is) && frag.GetTag() != seqDelItem )
00245     {
00246         gdcmDebugMacro( "Frag: " << frag );
00247         Fragments.push_back( frag );
00248     }
00249     assert( frag.GetTag() == seqDelItem && frag.GetVL() == 0 );
00250 }
00251 else
00252 {
00253     // 3. gdcm-JPEG-LossLess3a.dcm: easy case, an extra tag was found
00254     // instead of terminator (eof is the next char)
00255     gdcmWarningMacro( "Reading failed at Tag:" << frag.GetTag() << " Index #"
00256 << Fragments.size() << " Offset " << is.tellg() << ". Use file at own risk."
00257 << ex.what() );
00258 }
00259 #endif /* GDCM_SUPPORT_BROKEN_IMPLEMENTATION */
00260 }
00261
00262 return is;
00263 }
00264
00265 template <typename TSwap>
00266 std::ostream const &Write(std::ostream &os) const
00267 {
00268     if( !Table.Write<TSwap>(os) )
00269     {
00270         assert(0 && "Should not happen");
00271         return os;
00272     }
00273     for(ConstIterator it = Begin(); it != End(); ++it)
00274     {
00275         it->Write<TSwap>(os);

```

```

00276     }
00277     // seq del item is not stored, write it !
00278     const Tag seqDelItem(0xfffe,0xe0dd);
00279     seqDelItem.Write<TSwap>(os);
00280     VL zero = 0;
00281     zero.Write<TSwap>(os);
00282
00283     return os;
00284 }
00285
00286 // #if defined(SWIGPYTHON) || defined(SWIGSHARP) || defined(SWIGJAVA)
00287 // For now leave it there, this does not make sense in the C++ layer
00288 // Create a new object
00289 static SmartPointer<SequenceOfFragments> New()
00290 {
00291     return new SequenceOfFragments();
00292 }
00293 // #endif
00294
00295 protected:
00296 public:
00297     void Print(std::ostream &os) const override {
00298         os << "SQ L= " << SequenceLengthField << "\n";
00299         os << "Table:" << Table << "\n";
00300         for(ConstIterator it = Begin(); it != End(); ++it)
00301         {
00302             os << " " << *it << "\n";
00303         }
00304         assert( SequenceLengthField.IsUndefined() );
00305         {
00306             const Tag seqDelItem(0xfffe,0xe0dd);
00307             VL zero = 0;
00308             os << seqDelItem;
00309             os << "\t" << zero;
00310         }
00311     }
00312     bool operator==(const Value &val) const override
00313     {
00314         const SequenceOfFragments &sqf = dynamic_cast<const SequenceOfFragments*>(val);
00315         return Table == sqf.Table &&
00316             SequenceLengthField == sqf.SequenceLengthField &&
00317             Fragments == sqf.Fragments;
00318     }
00319
00320 private:
00321     BasicOffsetTable Table;
00322     VL SequenceLengthField;
00323     FragmentVector Fragments;
00324
00325 private:
00326     bool FillFragmentWithJPEG( Fragment & frag, std::istream & is );
00327 };
00328
00329 } // end namespace gdcm_ns
00330
00331 #endif //GDCMSEQUENCEOFFRAGMENTS_H

```

## 11.173 gdcmSequenceOfItems.h File Reference

```

#include "gdcmValue.h"
#include "gdcmItem.h"
#include <vector>
#include <cstring>
#include "gdcmSequenceOfItems.txx"

```

[illegible]

- class `gdcm::SequenceOfItems`  
*Class to represent a Sequence Of Items.*

- namespace **gdcm**

[Go to the documentation of this file.](#)

Generated by Doxygen

```

00043 typedef std::vector< Item > ItemVector;
00044 typedef ItemVector::size_type SizeType;
00045 typedef ItemVector::iterator Iterator;
00046 typedef ItemVector::const_iterator ConstIterator;
00047 Iterator Begin() { return Items.begin(); }
00048 Iterator End() { return Items.end(); }
00049 ConstIterator Begin() const { return Items.begin(); }
00050 ConstIterator End() const { return Items.end(); }
00051
00053 SequenceOfItems():SequenceLengthField(0xFFFFFFFF) { }
00054 //SequenceOfItems(VL const &vl = 0xFFFFFFFF):SequenceLengthField(vl),NType(type) { }
00055
00057 VL GetLength() const override { return SequenceLengthField; }
00059 void SetLength(VL length) override {
00060     SequenceLengthField = length;
00061 }
00063 void SetLengthToUndefined();
00065 bool IsUndefinedLength() const {
00066     return SequenceLengthField.IsUndefined();
00067 }
00068
00069 template <typename TDE>
00070 VL ComputeLength() const;
00071
00073 void Clear() override;
00074
00076 void AddItem(Item const &item);
00077
00079 Item & AddNewUndefinedLengthItem();
00080
00083 bool RemoveItemByIndex( const SizeType index );
00084
00085 bool IsEmpty() const { return Items.empty(); }
00086 SizeType GetNumberOfItems() const { return Items.size(); }
00087 void SetNumberOfItems(SizeType n) { Items.resize(n); }
00088
00089 /* WARNING: first item is #1 (see DICOM standard)
00090 * Each Item shall be implicitly assigned an ordinal position starting with the value 1 for the
00091 * first Item in the Sequence, and incremented by 1 with each subsequent Item. The last Item in the
00092 * Sequence shall have an ordinal position equal to the number of Items in the Sequence.
00093 */
00094 const Item &GetItem(SizeType position) const;
00095 Item &GetItem(SizeType position);
00096
00097 SequenceOfItems &operator=(const SequenceOfItems &val) {
00098     SequenceLengthField = val.SequenceLengthField;
00099     Items = val.Items;
00100     return *this;
00101 }
00102
00103 template <typename TDE, typename TSwap>
00104 std::istream &Read(std::istream &is, bool readvalues = true)
00105 {
00106     (void)readvalues;
00107     const Tag seqDelItem(0xfffe,0xe0dd);
00108     if( SequenceLengthField.IsUndefined() )
00109     {
00110         Item item;
00111         while( item.Read<TDE,TSwap>(is) && item.GetTag() != seqDelItem )
00112         {
00113             //gdcmDebugMacro( "Item: " « item );
00114             assert( item.GetTag() != seqDelItem );
00115             Items.push_back( item );
00116             item.Clear();
00117         }
00118         //assert( item.GetTag() == seqDelItem && item.GetVL() == 0 );
00119     }
00120     else
00121     {
00122         Item item;
00123         VL l = 0;
00124         //is.seekg( SequenceLengthField, std::ios::cur ); return is;
00125         while( l != SequenceLengthField )
00126         {
00127             try
00128             {
00129                 item.Read<TDE,TSwap>(is);
00130             }
00131             catch( Exception &ex )
00132             {
00133                 if( strcmp( ex.GetDescription(), "Changed Length" ) == 0 )

```

```

00134         {
00135             VL newlength = 1 + item.template GetLength<TDE>();
00136             if( newlength > SequenceLengthField )
00137             {
00138                 // BogugsItemAndSequenceLength.dcm
00139                 gdcmWarningMacro( "SQ length is wrong" );
00140                 SequenceLengthField = newlength;
00141             }
00142         }
00143         else
00144         {
00145             throw ex;
00146         }
00147     }
00148 #ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
00149     if( item.GetTag() == seqDelItem )
00150     {
00151         gdcmWarningMacro( "SeqDelItem found in defined length Sequence. Skipping" );
00152         assert( item.GetVL() == 0 );
00153         assert( item.GetNestedDataSet().Size() == 0 );
00154         // we need to pay attention that the length of the Sequence of Items will be wrong
00155         // this way. Indeed by not adding this item we are changing the size of this sqi
00156     }
00157     else // Not a seq del item marker
00158 #endif
00159     {
00160         // By design we never load them. If we were to load those attribute
00161         // as normal item it would become very complex to convert a sequence
00162         // from defined length to undefined length with the risk to write two
00163         // seq del marker
00164         Items.push_back( item );
00165     }
00166     l += item.template GetLength<TDE>();
00167     if( l > SequenceLengthField )
00168     {
00169         gdcmDebugMacro( "Found: Length of Item larger than expected" );
00170         throw "Length of Item larger than expected";
00171     }
00172     assert( l <= SequenceLengthField );
00173     //std::cerr << "sqi debug len: " << is.tellg() << " " << l << " " << SequenceLengthField << std::endl;
00174 #ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
00175     // MR_Philips_Intera_No_PrivateSequenceImplicitVR.dcm
00176     // (0x2005, 0x1080): for some reason computation of length fails...
00177     if( SequenceLengthField == 778 && l == 774 )
00178     {
00179         gdcmWarningMacro( "PMS: Super bad hack" );
00180         SequenceLengthField = l;
00181         throw Exception( "Wrong Length" );
00182         //l = SequenceLengthField;
00183     }
00184     // Bug_Philips_ItemTag_3F3F
00185     // (0x2005, 0x1080): Because we do not handle fully the bug at the item
00186     // level we need to check here too
00187     else if ( SequenceLengthField == 444 && l == 3*71 )
00188     {
00189         // This one is a double bug. Item length is wrong and impact SQ length
00190         gdcmWarningMacro( "PMS: Super bad hack" );
00191         l = SequenceLengthField;
00192     }
00193 #endif
00194     }
00195     assert( l == SequenceLengthField );
00196 }
00197 return is;
00198 }
00199
00200 template <typename TDE,typename TSwap>
00201 std::ostream const &Write(std::ostream &os) const
00202 {
00203     typename ItemVector::const_iterator it = Items.begin();
00204     for(;it != Items.end(); ++it)
00205     {
00206         it->Write<TDE,TSwap>(os);
00207     }
00208     if( SequenceLengthField.IsUndefined() )
00209     {
00210         // seq del item is not stored, write it !
00211         const Tag seqDelItem(0xfffe,0xe0dd);
00212         seqDelItem.Write<TSwap>(os);
00213         VL zero = 0;
00214         zero.Write<TSwap>(os);

```

```

00215     }
00216
00217     return os;
00218 }
00219
00220 //protected:
00221 void Print(std::ostream &os) const override {
00222     os << "\t(" << SequenceLengthField << ")\n";
00223     ItemVector::const_iterator it =
00224         Items.begin();
00225     for(; it != Items.end(); ++it)
00226     {
00227         os << "  " << *it;
00228     }
00229     if( SequenceLengthField.IsUndefined() )
00230     {
00231         const Tag seqDelItem(0xffff,0xe0dd);
00232         VL zero = 0;
00233         os << seqDelItem;
00234         os << "\t" << zero;
00235     }
00236 }
00237
00238 static SmartPointer<SequenceOfItems> New()
00239 {
00240     return new SequenceOfItems;
00241 }
00242 bool FindDataElement(const Tag &t) const;
00243
00244 bool operator==(const Value &val) const override
00245 {
00246     const SequenceOfItems &sqi = dynamic_cast<const SequenceOfItems&>(val);
00247     return SequenceLengthField == sqi.SequenceLengthField &&
00248         Items == sqi.Items;
00249 }
00250
00251 private:
00252 public:
00253     VL SequenceLengthField;
00254     ItemVector Items;
00255 };
00256
00257 } // end namespace gdcms_ns
00258
00259 } // end namespace gdcms_ns
00260
00261 #include "gdcmsSequenceOfItems.txx"
00262
00263 #endif //GDCMSEQUENCEOFITEMS_H

```

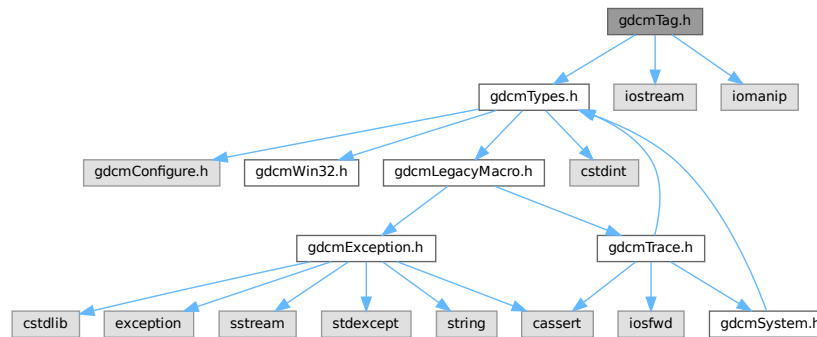
## 11.175 gdcmsTag.h File Reference

```

#include "gdcmsTypes.h"
#include <iostream>
#include <iomanip>

```

Include dependency graph for gdcmTag.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::Tag](#)  
Class to represent a DICOM Data *Element* (*Attribute*) *Tag* (Group, *Element*).

## Namespaces

- namespace [gdcm](#)

## Functions

- std::ostream & [gdcm::operator<<](#) (std::ostream &\_os, const [Tag](#) &\_val)
- std::istream & [gdcm::operator>>](#) (std::istream &\_is, [Tag](#) &\_val)

## 11.176 gdcmTag.h

[Go to the documentation of this file.](#)

```

00001 /x=====
00002
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008

```

```

00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE. See the above copyright notice for more information.
00012
00013      =====*/
00014 #ifndef GDCMTAG_H
00015 #define GDCMTAG_H
00016
00017 #include "gdcmTypes.h"
00018
00019 #include <iostream>
00020 #include <iomanip>
00021
00022 namespace gdcm
00023 {
00024
00025     class GDCM_EXPORT Tag
00026     {
00027     public:
00028         Tag(uint16_t group, uint16_t element) {
00029             ElementTag.tags[0] = group; ElementTag.tags[1] = element;
00030         }
00031         Tag(uint32_t tag = 0) {
00032             SetElementTag(tag);
00033         }
00034
00035         friend std::ostream& operator<<(std::ostream &_os, const Tag &_val);
00036         friend std::istream& operator>>(std::istream &_is, Tag &_val);
00037
00038         uint16_t GetGroup() const { return ElementTag.tags[0]; }
00039         uint16_t GetElement() const { return ElementTag.tags[1]; }
00040         void SetGroup(uint16_t group) { ElementTag.tags[0] = group; }
00041         void SetElement(uint16_t element) { ElementTag.tags[1] = element; }
00042         void SetElementTag(uint16_t group, uint16_t element) {
00043             ElementTag.tags[0] = group; ElementTag.tags[1] = element;
00044         }
00045
00046         uint32_t GetElementTag() const {
00047 #ifndef GDCM_WORDS_BIGENDIAN
00048             return (ElementTag.tag<16) | (ElementTag.tag>16);
00049 #else
00050             return ElementTag.tag;
00051 #endif
00052         }
00053
00054         void SetElementTag(uint32_t tag) {
00055 #ifndef GDCM_WORDS_BIGENDIAN
00056             tag = ( (tag<16) | (tag>16) );
00057 #endif
00058             ElementTag.tag = tag;
00059         }
00060
00061         const uint16_t &operator[](const unsigned int &_id) const
00062         {
00063             assert(_id<2);
00064             return ElementTag.tags[_id];
00065         }
00066         uint16_t &operator[](const unsigned int &_id)
00067         {
00068             assert(_id<2);
00069             return ElementTag.tags[_id];
00070         }
00071
00072         Tag &operator=(const Tag &_val)
00073         {
00074             ElementTag.tag = _val.ElementTag.tag;
00075             return *this;
00076         }
00077
00078         bool operator==(const Tag &_val) const
00079         {
00080             return ElementTag.tag == _val.ElementTag.tag;
00081         }
00082         bool operator!=(const Tag &_val) const
00083         {
00084             return ElementTag.tag != _val.ElementTag.tag;
00085         }
00086
00087         // FIXME FIXME FIXME TODO
00088         // the following is pretty dumb. Since we have control over who is group
00089         // and who is element, we should reverse them in little endian and big endian case

```



```

00117 // since what we really want is fast comparison and not guarantee that group is in #0
00118 // ...
00119 bool operator<(const Tag &_val) const
00120 {
00121 #ifndef GDCM_WORDS_BIGENDIAN
00122     if( ElementTag.tags[0] < _val.ElementTag.tags[0] )
00123         return true;
00124     if( ElementTag.tags[0] == _val.ElementTag.tags[0]
00125         && ElementTag.tags[1] < _val.ElementTag.tags[1] )
00126         return true;
00127     return false;
00128 #else
00129     // Plain comparison is enough!
00130     return ( ElementTag.tag < _val.ElementTag.tag );
00131 #endif
00132 }
00133 bool operator<=(const Tag &t2) const
00134 {
00135     const Tag &t1 = *this;
00136     return t1 == t2 || t1 < t2;
00137 }
00138 Tag(const Tag &_val)
00139 {
00140     ElementTag.tag = _val.ElementTag.tag;
00141 }
00142 uint32_t GetLength() const { return 4; }
00143 bool IsPublic() const { return !(ElementTag.tags[0] % 2); }
00144 bool IsPrivate() const { return !IsPublic(); }
00145
00146 //-----
00147 template <typename TSwap>
00148 std::istream &Read(std::istream &is)
00149 {
00150     if( is.read(ElementTag.bytes, 4) )
00151         TSwap::SwapArray(ElementTag.tags, 2);
00152     return is;
00153 }
00154
00155 template <typename TSwap>
00156 const std::ostream &Write(std::ostream &os) const
00157 {
00158     uint16_t copy[2];
00159     copy[0] = ElementTag.tags[0];
00160     copy[1] = ElementTag.tags[1];
00161     TSwap::SwapArray(copy, 2);
00162     return os.write((char*)(copy), 4);
00163 }
00164
00165 Tag GetPrivateCreator() const
00166 {
00167     // See PS 3.5 - 7.8.1 PRIVATE DATA ELEMENT TAGS
00168     // eg: 0x0123,0x1425 -> 0x0123,0x0014
00169     if( IsPrivate() && !IsPrivateCreator() )
00170     {
00171         Tag r = *this;
00172         r.SetElement( (uint16_t)(GetElement() >> 8) );
00173         return r;
00174     }
00175     if( IsPrivateCreator() ) return *this;
00176     return Tag(0x0,0x0);
00177 }
00178
00179 void SetPrivateCreator(Tag const &t)
00180 {
00181     // See PS 3.5 - 7.8.1 PRIVATE DATA ELEMENT TAGS
00182     // eg: 0x0123,0x0045 -> 0x0123,0x4567
00183     assert( t.IsPrivate() /*&& t.IsPrivateCreator()*/ );
00184     const uint16_t element = (uint16_t)(t.GetElement() << 8);
00185     const uint16_t base = (uint16_t)(GetElement() << 8);
00186     SetElement( (uint16_t)((base >> 8) + element) );
00187     SetGroup( t.GetGroup() );
00188 }
00189
00190 bool IsPrivateCreator() const
00191 {
00192     return IsPrivate() && (GetElement() <= 0xFF && GetElement() >= 0x10);
00193 }
00194
00195 bool IsIllegal() const

```

```

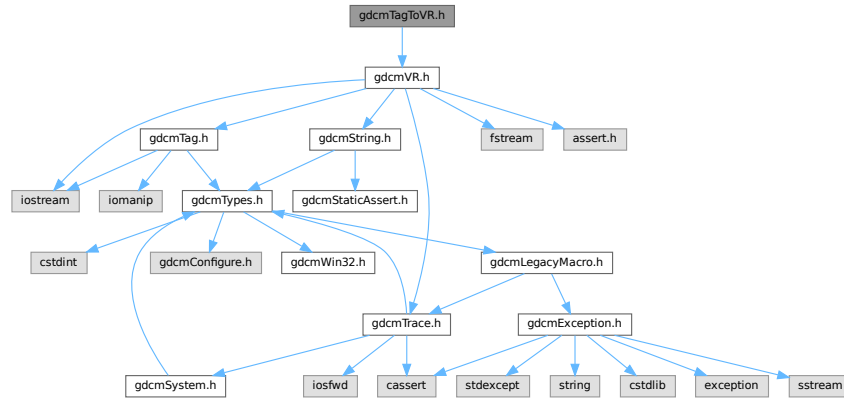
00213     {
00214         // DICOM reserved those groups:
00215         return GetGroup() == 0x0001 || GetGroup() == 0x0003 || GetGroup() == 0x0005 || GetGroup() == 0x0007
00216         // This is a very special case, in private group, one cannot use element [0x01,0x09] ...
00217         // || (IsPrivate() && !IsPrivateCreator() && !IsGroupLength());
00218         || (IsPrivate() && GetElement() > 0x0 && GetElement() < 0x10 );
00219     }
00220
00222 bool IsGroupLength() const
00223 {
00224     return GetElement() == 0x0;
00225 }
00226
00228 bool IsGroupXX(const Tag &t) const
00229 {
00230     if( t.GetElement() == GetElement() )
00231     {
00232         if( t.IsPrivate() ) return false;
00233         uint16_t group = (uint16_t)((GetGroup() » 8 ) « 8);
00234         return group == t.GetGroup();
00235     }
00236     return false;
00237 }
00238
00244 bool ReadFromCommaSeparatedString(const char *str);
00245
00248 bool ReadFromContinuousString(const char *str);
00249
00252 std::string PrintAsContinuousString() const;
00253
00255 std::string PrintAsContinuousUpperCaseString() const;
00256
00259 bool ReadFromPipeSeparatedString(const char *str);
00260
00263 std::string PrintAsPipeSeparatedString() const;
00264
00265 private:
00266     union { uint32_t tag; uint16_t tags[2]; char bytes[4]; } ElementTag;
00267 };
00268 //-----
00269 inline std::istream& operator>(std::istream &_is, Tag &_val)
00270 {
00271     char c;
00272     _is > c;
00273     uint16_t a, b;
00274     _is > std::hex > a;
00275     //_is > std::hex > _val[0];
00276     //_is > std::hex > _val.ElementTag.tags[0];
00277     _is > c;
00278     //_is > _val[1];
00279     //_is > std::hex > _val.ElementTag.tags[1];
00280     _is > std::hex > b;
00281     _is > c;
00282     _val.SetGroup( a );
00283     _val.SetElement( b );
00284     return _is;
00285 }
00286
00287 inline std::ostream& operator<(std::ostream &_os, const Tag &_val)
00288 {
00289     _os.setf( std::ios::right);
00290     _os < std::hex < ' (' < std::setw( 4 ) < std::setfill( '0' )
00291     < _val[0] < ', ' < std::setw( 4 ) < std::setfill( '0' )
00292     < _val[1] < ')' < std::setfill( ' ' ) < std::dec;
00293     return _os;
00294 }
00295
00296 } // end namespace gdcm
00297
00298 #endif //GDCMTAG_H

```

## 11.177 gdcmTagToVR.h File Reference

```
#include "gdcmVR.h"
```

Include dependency graph for gdcmTagToVR.h:



### Namespaces

- namespace [gdcm](#)

### Functions

- [VR::VRType gdcm::GetVRFromTag](#) ([Tag](#) const &tag)

## 11.178 gdcmTagToVR.h

[Go to the documentation of this file.](#)

```

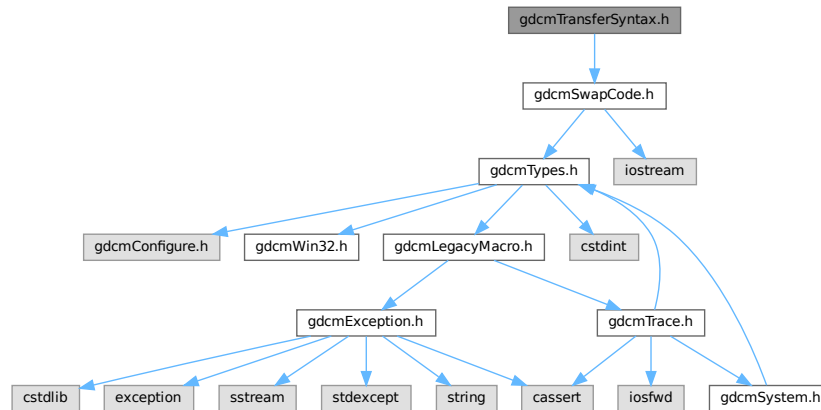
00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMTAGTOVR_H
00015 #define GDCMTAGTOVR_H
00016
00017 #include "gdcmVR.h"
00018
00019 namespace gdcm
00020 {
00021   class Tag;
00022   VR::VRType GetVRFromTag( Tag const & tag );
00023 }
00024
00025 #endif // GDCMTAGTOVR_H

```

## 11.179 gdcmTransferSyntax.h File Reference

```
#include "gdcmSwapCode.h"
```

Include dependency graph for gdcmTransferSyntax.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [gdcm::TransferSyntax](#)  
*Class to manipulate Transfer Syntax.*

### Namespaces

- namespace [gdcm](#)

### Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const TransferSyntax &ts)`

## 11.180 gdcmTransferSyntax.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMTRANSFERSYNTAX_H
00015  #define GDCMTRANSFERSYNTAX_H
00016
00017  #include "gdcmSwapCode.h"
00018
00019  namespace gdcm
00020  {
00021
00022  class GDCM_EXPORT TransferSyntax
00023  {
00024  public:
00025      typedef enum {
00026          Unknown = 0,
00027          Explicit,
00028          Implicit
00029      } NegotiatedType;
00030
00031  #if 0
00032      //NOT FLEXIBLE, since forces user to update lib every time new module
00033      //comes out...
00034      // TODO
00035      typedef enum {
00036          NoSpacing = 0,
00037          PixelSpacing,
00038          ImagerPixelSpacing,
00039          PixelAspectRatio
00040      } ImageSpacingType;
00041      ImageSpacingType GetImageSpacing();
00042  #endif
00043
00044      typedef enum {
00045          ImplicitVRLittleEndian = 0,
00046          ImplicitVRBigEndianPrivateGE,
00047          ExplicitVRLittleEndian,
00048          DeflatedExplicitVRLittleEndian,
00049          ExplicitVRBigEndian,
00050          JPEGBaselineProcess1,
00051          JPEGExtendedProcess2_4,
00052          JPEGExtendedProcess3_5,
00053          JPEGsSpectralSelectionProcess6_8,
00054          JPEGFullProgressionProcess10_12,
00055          JPEGLosslessProcess14,
00056          JPEGLosslessProcess14_1,
00057          JPEGLSLossless,
00058          JPEGLSNearLossless,
00059          JPEG2000Lossless,
00060          JPEG2000,
00061          JPEG2000Part2Lossless,
00062          JPEG2000Part2,
00063          RLELossless,
00064          MPEG2MainProfile,
00065          ImplicitVRBigEndianACRNEEMA,
00066          WeirdPapryus,
00067          CT_private_ELE,
00068          JPIPReferenced,
00069          MPEG2MainProfileHighLevel,
00070          MPEG4AVCH264HighProfileLevel4_1,
00071          MPEG4AVCH264BDcompatibleHighProfileLevel4_1,
00072          TS_END
00073      } TSType;
00074
00075      // Return the string as written in the official DICOM dict from
00076      // a custom enum type

```

```

00094 static const char* GetTSString(TSType ts);
00095 static TSType GetTSType(const char *str);
00096
00097 NegotiatedType GetNegociatedType() const;
00098
00102 SwapCode GetSwapCode() const;
00103
00104 bool IsValid() const { return TSField != TS_END; }
00105
00106 operator TSType () const { return TSField; }
00107
00108 // FIXME: ImplicitVRLittleEndian used to be the default, but nowadays
00109 // this is rather the ExplicitVRLittleEndian instead...should be change the default ?
00110 TransferSyntax(TSType type = ImplicitVRLittleEndian):TSField(type) {}
00111
00112 // return if dataset is encoded or not (Deflate Explicit VR)
00113 bool IsEncoded() const;
00114
00115 bool IsImplicit() const;
00116 bool IsExplicit() const;
00117
00118 bool IsEncapsulated() const;
00119
00121 bool IsLossy() const;
00123 bool IsLossless() const;
00125 bool CanStoreLossy() const;
00126
00127 const char *GetString() const { return TransferSyntax::GetTSString(TSField); }
00128
00129 friend std::ostream &operator<<(std::ostream &os, const TransferSyntax &ts);
00130 private:
00131 // DO NOT EXPOSE the following. Internal details of TransferSyntax
00132 bool IsImplicit(TSType ts) const;
00133 bool IsExplicit(TSType ts) const;
00134 bool IsLittleEndian(TSType ts) const;
00135 bool IsBigEndian(TSType ts) const;
00136
00137 TSType TSField;
00138 };
00139 //-----
00140 inline std::ostream &operator<<(std::ostream &_os, const TransferSyntax &ts)
00141 {
00142     _os << TransferSyntax::GetTSString(ts);
00143     return _os;
00144 }
00145 }
00146
00147 } // end namespace gdcm
00148
00149 #endif //GDCMTRANSFERSYNTAX_H

```

## 11.181 gdcmUNExplicitDataElement.h File Reference

```

#include "gdcmDataElement.h"
#include "gdcmUNExplicitDataElement.txx"

```

- class `gdcm::UNExplicitDataElement`  
*Class to read/write a `DataElement` as UNExplicit Data `Element`.*

- namespace **gdcm**

[Go to the documentation of this file.](#)

```
00001  /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMUNEXPLICITDATAELEMENT_H
00015  #define GDCMUNEXPLICITDATAELEMENT_H
00016
00017  #include "gdcmDataElement.h"
00018
00019  namespace gdcms
00020  {
00021  // Data Element (UNExplicit)
00022
00023  class GDCM_EXPORT UNExplicitDataElement : public DataElement
00024  {
00025  public:
00026      VL GetLength() const;
00027
00028  }
```

```

00031  template <typename TSwap>
00032  std::istream &Read(std::istream &is);
00033
00034  template <typename TSwap>
00035  std::istream &ReadPreValue(std::istream &is);
00036
00037  template <typename TSwap>
00038  std::istream &ReadValue(std::istream &is, bool readvalues = true);
00039
00040  template <typename TSwap>
00041  std::istream &ReadWithLength(std::istream &is, VL & length);
00042
00043  // PURPOSELY do not provide an implementation for writing !
00044  //template <typename TSwap>
00045  //const std::ostream &Write(std::ostream &os) const;
00046 };
00047
00048 } // end namespace gdcm
00049
00050 #include "gdcmUNExplicitDataElement.txx"
00051
00052 #endif //GDCMUNEXPLICITDATAELEMENT_H

```

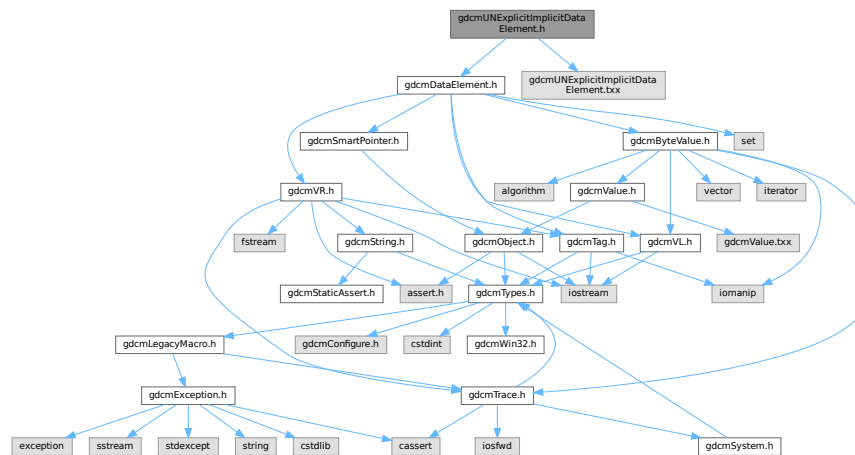
## 11.183 gdcmUNExplicitImplicitDataElement.h File Reference

```

#include "gdcmDataElement.h"
#include "gdcmUNExplicitImplicitDataElement.txx"

```

Include dependency graph for gdcmUNExplicitImplicitDataElement.h:



### Classes

- class [gdcm::UNExplicitImplicitDataElement](#)  
Class to read/write a *DataElement* as *ExplicitImplicit Data Element*.

### Namespaces

- namespace [gdcm](#)



## 11.184 gdcmUNExplicitImplicitDataElement.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMUNEXPLICITIMPLICITDATAELEMENT_H
00015  #define GDCMUNEXPLICITIMPLICITDATAELEMENT_H
00016
00017  #include "gdcmDataElement.h"
00018
00019  namespace gdcm
00020  {
00021  // Data Element (ExplicitImplicit)
00022  class GDCM_EXPORT UNExplicitImplicitDataElement : public DataElement
00023  {
00024  public:
00025      VL GetLength() const;
00026
00027      template <typename TSwap>
00028      std::istream &Read(std::istream &is);
00029
00030      template <typename TSwap>
00031      std::istream &ReadPreValue(std::istream &is);
00032
00033      template <typename TSwap>
00034      std::istream &ReadValue(std::istream &is);
00035
00036      // PURPOSELY do not provide an implementation for writing !
00037      //template <typename TSwap>
00038      //const std::ostream &Write(std::ostream &os) const;
00039  };
00040
00041  } // end namespace gdcm
00042
00043  #include "gdcmUNExplicitImplicitDataElement.txx"
00044
00045  #endif //GDCMUNEXPLICITIMPLICITDATAELEMENT_H

```

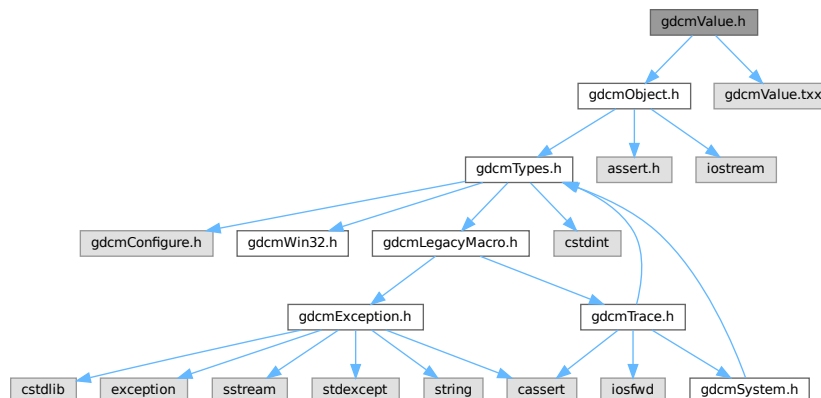
## 11.185 gdcmValue.h File Reference

```

#include "gdcmObject.h"
#include "gdcmValue.txx"

```

Include dependency graph for `gdcmValue.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::Value`  
Class to represent the value of a Data *Element*.

## Namespaces

- namespace `gdcm`

## 11.186 `gdcmValue.h`

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/

```

```

00014 #ifndef GDCMVALUE_H
00015 #define GDCMVALUE_H
00016
00017 #include "gdcmObject.h"
00018
00019 namespace gdcm { class VL; }
00020 namespace gdcm_ns
00021 {
00022     #if !defined(SWIGPYTHON) && !defined(SWIGSHARP) && !defined(SWIGJAVA) && !defined(SWIGPHP)
00023     using namespace gdcm;
00024     #endif
00031     class GDCM_EXPORT Value : public Object
00032     {
00033     public:
00034         Value() = default;
00035         ~Value() override = default;
00036
00037         virtual VL GetLength() const = 0;
00038         virtual void SetLength(VL l) = 0;
00039
00040         virtual void Clear() = 0;
00041
00042         virtual bool operator==(const Value &val) const = 0;
00043
00044     protected:
00045         friend class DataElement;
00046         virtual void SetLengthOnly(VL l);
00047     };
00048
00049 } // end namespace gdcm_ns
00050
00051 #include "gdcmValue.txx"
00052
00053 #endif //GDCMVALUE_H

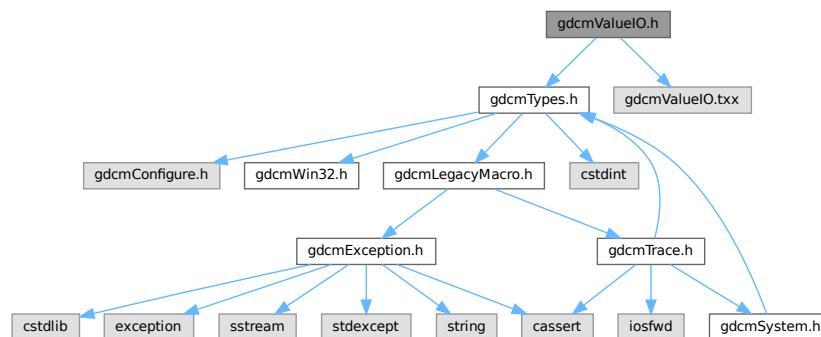
```

## 11.187 gdcmValueIO.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmValueIO.txx"
```

Include dependency graph for gdcmValueIO.h:



### Classes

- class [gdcm::ValueIO< TDE, TSwap, TType >](#)

*Class to dispatch template calls.*

## Namespaces

- namespace `gdcm`

## 11.188 gdcmValueIO.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMVALUEIO_H
00015 #define GDCMVALUEIO_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm_ns
00020 {
00021     template <typename TDE, typename TSwap, typename TType=uint8_t>
00022     class /*GDCM_EXPORT*/ ValueIO
00023     {
00024     public:
00025         static std::istream &Read(std::istream &is, Value& v, bool readvalues);
00026         static const std::ostream &Write(std::ostream &os, const Value& v);
00027     };
00028 } // end namespace gdcm_ns
00029
00030 #include "gdcmValueIO.txx"
00031
00032 #endif //GDCMVALUEIO_H

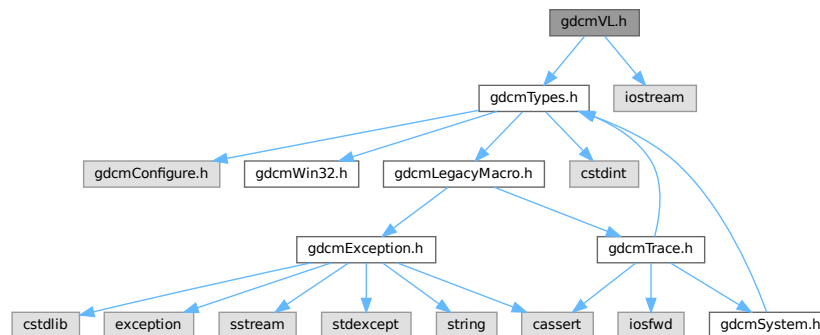
```

## 11.189 gdcmVL.h File Reference

```
#include "gdcmTypes.h"
```

```
#include <iostream>
```

Include dependency graph for gdcmVL.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::VL`  
*Value Length.*

## Namespaces

- namespace `gdcm`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const VL &val)`

## 11.190 gdcmVL.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012   =====*/
00013
00014 #ifndef GDCMVL_H
00015 #define GDCMVL_H
00016
00017 #include "gdcmTypes.h"
00018
00019 #include <iostream>
00020
00021 namespace gdcm
00022 {
00023
00024   class GDCM_EXPORT VL
00025   {
00026   public:
00027     typedef uint32_t Type;
00028     VL(uint32_t vl = 0) : ValueLength(vl) { }
00029
00030     // FIXME: ugly
00031     static uint32_t GetVL32Max() { return 0xFFFFFFFF; }
00032     static uint16_t GetVL16Max() { return 0xFFFF; }
00033
00034     bool IsUndefined() const {
00035       return ValueLength == 0xFFFFFFFF;
00036     }
00037   }
00038
00039 }

```

```

00042 void SetToUndefined() {
00043     ValueLength = 0xFFFFFFFF;
00044 }
00045
00047 bool IsOdd() const {
00048     return !IsUndefined() && ValueLength % 2;
00049 }
00050
00052 VL& operator+=(VL const &vl) {
00053     ValueLength += vl.ValueLength;
00054     return *this;
00055 }
00056 VL& operator++() {
00057     ++ValueLength;
00058     return *this;
00059 }
00060 VL operator++(int) {
00061     uint32_t tmp(ValueLength);
00062     ++ValueLength;
00063     return tmp;
00064 }
00065
00066 operator uint32_t () const { return ValueLength; }
00067
00068 VL GetLength() const {
00069     // VL cannot know it's length...well in implicit yes...
00070     // TODO: need to check we cannot call this function from an Explicit element
00071     return 4;
00072 }
00073
00074 friend std::ostream& operator<<(std::ostream& os, const VL& vl);
00075
00076 // PURPOSELY not implemented (could not differentiate 16bits vs 32bits VL)
00077 //friend std::istream& operator>>(std::istream& is, VL& n);
00078
00079 template <typename TSwap>
00080 std::istream &Read(std::istream &is)
00081 {
00082     is.read((char*)(&ValueLength), sizeof(uint32_t));
00083     TSwap::SwapArray(&ValueLength,1);
00084     return is;
00085 }
00086
00087 template <typename TSwap>
00088 std::istream &Read16(std::istream &is)
00089 {
00090     uint16_t copy;
00091     is.read((char*)(&copy), sizeof(uint16_t));
00092     TSwap::SwapArray(&copy,1);
00093     ValueLength = copy;
00094     assert( ValueLength <= 65535 /*UINT16_MAX*/ ); // ?? doh !
00095     return is;
00096 }
00097
00098 template <typename TSwap>
00099 const std::ostream &Write(std::ostream &os) const
00100 {
00101     uint32_t copy = ValueLength;
00102     if( IsOdd() )
00103     {
00104         ++copy;
00105     }
00106     TSwap::SwapArray(&copy,1);
00107     return os.write((char*)(&copy), sizeof(uint32_t));
00108 }
00109
00110 template <typename TSwap>
00111 const std::ostream &Write16(std::ostream &os) const
00112 {
00113     assert( ValueLength <= 65535 /*UINT16_MAX*/ );
00114     uint16_t copy = (uint16_t)ValueLength;
00115     if( IsOdd() )
00116     {
00117         ++copy;
00118     }
00119     TSwap::SwapArray(&copy,1);
00120     return os.write((char*)(&copy), sizeof(uint16_t));
00121 }
00122
00123 private:
00124     uint32_t ValueLength;

```

```

00125 };
00126 //-----
00127 inline std::ostream& operator<<(std::ostream& os, const VL& val)
00128 {
00129     os << /*std::hex <<*/ val.ValueLength;
00130     return os;
00131 }
00132
00133 } // end namespace gdcm
00134
00135 #endif //GDCMVL_H

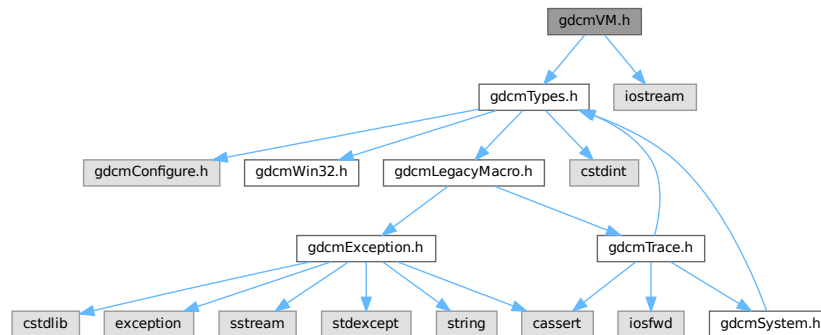
```

## 11.191 gdcmVM.h File Reference

```

#include "gdcmTypes.h"
#include <iostream>
Include dependency graph for gdcmVM.h:

```



This graph shows which files directly or indirectly include this file:



### Classes

- class [gdcm::VM](#)

*Value Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n.*

### Namespaces

- namespace [gdcm](#)

## Macros

- `#define TYPETOLENGTH`(type, length)

## Functions

- `std::ostream & gdcmm::operator<<` (std::ostream &\_os, const VM &\_val)

## 11.191.1 Macro Definition Documentation

### 11.191.1.1 TYPETOLENGTH

```
#define TYPETOLENGTH(
    type,
    length )
```

#### Value:

```
template< > struct VMToLength<VM::type> \
{ enum { Length = length }; };
```

## 11.192 gdcmmVM.h

[Go to the documentation of this file.](#)

```
00001 /*=====
00002
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMMVM_H
00015 #define GDCMMVM_H
00016
00017 #include "gdcmmTypes.h"
00018 #include <iostream>
00019
00020 namespace gdcmm
00021 {
00022
00023 class GDCMM_EXPORT VM
00024 {
00025 public:
00026     typedef enum {
00027         VM0 = 0, // aka the invalid VM
00028         VM1 = 1,
00029         VM2 = 2,
00030         VM3 = 4,
00031         VM4 = 8,
00032         VM5 = 16,
00033         VM6 = 32,
00034         VM8 = 64,
00035         VM9 = 128,
00036         VM10 = 256,
00037         VM12 = 512, //1024,
00038         VM16 = 1024, //2048,
00039         VM18 = 2048, //4096,
```



```

00084     VM24 = 4096, //8192,
00085     VM28 = 8192, //16384,
00086     VM32 = 16384, //32768,
00087     VM35 = 32768, //65536,
00088     VM99 = 65536, //131072,
00089     VM256 = 131072, //262144,
00090     VM1_2 = VM1 | VM2,
00091     VM1_3 = VM1 | VM2 | VM3,
00092     VM1_4 = VM1 | VM2 | VM3 | VM4,
00093     VM1_5 = VM1 | VM2 | VM3 | VM4 | VM5,
00094     VM1_8 = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8,
00095 // The following need some work:
00096     VM1_32 = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32,
00097     VM1_99 = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99,
00098     VM1_n = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256,
00099     VM2_2n = VM2 | VM4 | VM6 | VM8 | VM16 | VM24 | VM32 | VM99 | VM256,
00100     VM2_n = VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256,
00101     VM3_4 = VM3 | VM4,
00102     VM3_3n = VM3 | VM6 | VM9 | VM24 | VM99 | VM256,
00103     VM3_n = VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256,
00104     VM4_4n = VM4 | VM16 | VM24 | VM32 | VM99 | VM256,
00105     VM6_6n = VM6 | VM12 | VM18 | VM24 | VM99 | VM256,
00106     VM6_n = VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256,
00107     VM7_7n,
00108     VM30_30n,
00109     VM47_47n,
00110     VM_END = VM1_n + 1 // Custom tag to count number of entry
00111 } VMType;
00112
00113 static const char* GetVMString(VMType vm);
00114 static VMType GetVMType(const char *vm);
00115
00116 static bool IsValid(int vm1, VMType vm2);
00117 //bool IsValid() { return VMField != VM0 && VMField < VM_END; }
00118
00119 bool Compatible(VM const &vm) const;
00120
00121 static VMType GetVMTypeFromLength(size_t length, unsigned int size);
00122 static size_t GetNumberOfElementsFromArray(const char *array, size_t length);
00123
00124 VM(VMType type = VM0):VMField(type) {}
00125 operator VMType () const { return VMField; }
00126 unsigned int GetLength() const;
00127
00128 friend std::ostream &operator<<(std::ostream &os, const VM &vm);
00129 protected:
00130     static unsigned int GetIndex(VMType vm);
00131
00132 private:
00133     VMType VMField;
00134 };
00135 //-----
00136 inline std::ostream& operator<<(std::ostream& _os, const VM &_val)
00137 {
00138     assert( VM::GetVMString(_val) );
00139     _os << VM::GetVMString(_val);
00140     return _os;
00141 }
00142
00143 //template <int TVM> struct LengthToVM;
00144 //template <> struct LengthToVM<1>
00145 //{ enum { TVM = VM::VM1 }; };
00146
00147 template<int T> struct VMToLength;
00148 #define TYPETOLENGTH(type,length) \
00149     template<> struct VMToLength<VM::type> \
00150     { enum { Length = length }; };
00151 // TODO: Could be generated from XML file
00152 //TYPETOLENGTH(VM0,1)
00153 TYPETOLENGTH(VM1,1)
00154 TYPETOLENGTH(VM2,2)
00155 TYPETOLENGTH(VM3,3)
00156 TYPETOLENGTH(VM4,4)
00157 TYPETOLENGTH(VM5,5)
00158 TYPETOLENGTH(VM6,6)
00159 TYPETOLENGTH(VM8,8)
00160 TYPETOLENGTH(VM9,9)
00161 TYPETOLENGTH(VM10,10)
00162 TYPETOLENGTH(VM12,12)
00163 TYPETOLENGTH(VM16,16)
00164 TYPETOLENGTH(VM18,18)

```

```

00173 TYPETOLENGTH (VM24, 24)
00174 TYPETOLENGTH (VM28, 28)
00175 TYPETOLENGTH (VM32, 32)
00176 TYPETOLENGTH (VM35, 35)
00177 TYPETOLENGTH (VM99, 99)
00178 TYPETOLENGTH (VM256, 256)
00179 //TYPETOLENGTH (VM1_2, 2)
00180 //TYPETOLENGTH (VM1_3, 3)
00181 //TYPETOLENGTH (VM1_8, 8)
00182 //TYPETOLENGTH (VM1_32, 32)
00183 //TYPETOLENGTH (VM1_99, 99)
00184 //TYPETOLENGTH (VM1_n,
00185 //TYPETOLENGTH (VM2_2n,
00186 //TYPETOLENGTH (VM2_n,
00187 //TYPETOLENGTH (VM3_3n,
00188 //TYPETOLENGTH (VM3_n,
00189
00190 } // end namespace gdcm
00191
00192 #endif //GDCMVM_H

```

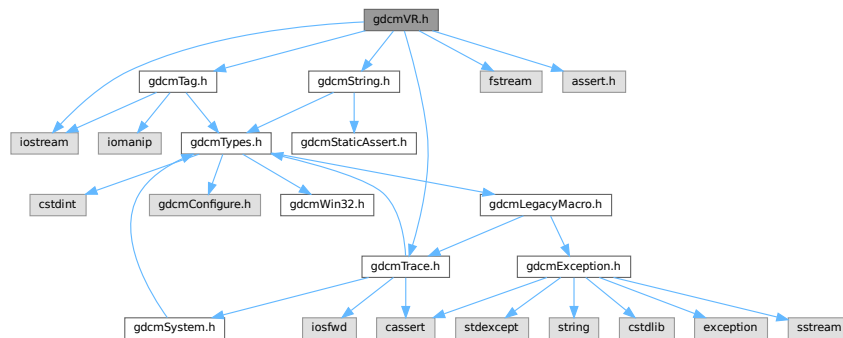
## 11.193 gdcmVR.h File Reference

```

#include "gdcmTag.h"
#include "gdcmTrace.h"
#include "gdcmString.h"
#include <iostream>
#include <fstream>
#include <assert.h>

```

Include dependency graph for gdcmVR.h:



This graph shows which files directly or indirectly include this file:



### Classes

- struct [gdcm::UI](#)
- class [gdcm::VR](#)  
*VR class.*

## Namespaces

- namespace [gdcm](#)

## Macros

- `#define TYPETOENCODING(type, rep, rtype)`
- `#define VRTypeTemplateCase(type)`

## Typedefs

- `typedef String<'\', 16 > gdcm::AECComp`
- `typedef String<'\', 64 > gdcm::ASComp`
- `typedef String<'\', 16 > gdcm::CSComp`
- `typedef String<'\', 64 > gdcm::DAComp`
- `typedef String<'\', 64 > gdcm::DTComp`
- `typedef String<'\', 64 > gdcm::LOComp`
- `typedef String<'\', 64 > gdcm::LTComp`
- `typedef String<'\', 64 > gdcm::PNComp`
- `typedef String<'\', 64 > gdcm::SHComp`
- `typedef String<'\', 64 > gdcm::STComp`
- `typedef String<'\', 16 > gdcm::TMComp`
- `typedef String<'\', 4294967294 > gdcm::UCComp`
- `typedef String<'\', 64, 0 > gdcm::UIComp`
- `typedef String<'\', 4294967294 > gdcm::URComp`
- `typedef String<'\', 64 > gdcm::UTComp`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const UI &_val)`
- `std::ostream & gdcm::operator<< (std::ostream &_os, const VR &val)`

## 11.193.1 Macro Definition Documentation

### 11.193.1.1 TYPETOENCODING

```
#define TYPETOENCODING(
    type,
    rep,
    rtype )
```

#### Value:

```
template<> struct VRToEncoding<VR::type> \
{ enum:long long { Mode = VR::rep }; }; \
template<> struct VRToType<VR::type> \
{ typedef rtype Type; };
```

### 11.193.1.2 VRTypeTemplateCase

```
#define VRTypeTemplateCase(
    type )
```

#### Value:

```
case VR::type: \
    return sizeof ( VRToType<VR::type>::Type );
```

## 11.194 gdcmmVR.h

[Go to the documentation of this file.](#)

```
00001 /*=====
00002
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMMVR_H
00015 #define GDCMMVR_H
00016
00017 #include "gdcmmTag.h"
00018 #include "gdcmmTrace.h"
00019 #include "gdcmmString.h"
00020
00021 #include <iostream>
00022 #include <fstream>
00023 #include <assert.h>
00024
00025 //these defines are here to ensure compilation on sunos gcc
00026 #if defined (CS)
00027 # undef CS
00028 #endif
00029 #if defined (DS)
00030 # undef DS
00031 #endif
00032 #if defined (SS)
00033 # undef SS
00034 #endif
00035
00036 namespace gdcmm
00037 {
00038
00039 class GDCMM_EXPORT VR
00040 {
00041 public:
00042     enum VRType : long long {
00043         // Warning: Do not write if ( vr & VR::INVALID ) but if ( vr == VR::INVALID )
00044         INVALID = 0, // For Item/(Seq) Item Delimitation Item
00045         AE = 1,
00046         AS = 2,
00047         AT = 4,
00048         CS = 8,
00049         DA = 16,
00050         DS = 32,
00051         DT = 64,
00052         FD = 128,
00053         FL = 256,
00054         IS = 512,
00055         LO = 1024,
00056         LT = 2048,
00057         OB = 4096,
00058         OD = 134217728, // 2^27
00059         OF = 8192,
```

```

00075     OL = 268435456, // 2^28
00076     OV = 2147483648, // 2^31
00077     OW = 16384,
00078     PN = 32768,
00079     SH = 65536,
00080     SL = 131072,
00081     SQ = 262144,
00082     SS = 524288,
00083     ST = 1048576,
00084     SV = 4294967296, // 2^32
00085     TM = 2097152,
00086     UC = 536870912, // 2^29
00087     UI = 4194304,
00088     UL = 8388608,
00089     UN = 16777216,
00090     UR = 1073741824, // 2^30
00091     US = 33554432,
00092     UT = 67108864,
00093     UV = 8589934592, // 2^33
00094     OB_OW = OB | OW,
00095     US_SS = US | SS,
00096     US_SS_OW = US | SS | OW,
00097     US_OW = US | OW,
00098     // The following do not have a VRString equivalent (ie cannot be found in PS 3.6)
00099     VL16 = AE | AS | AT | CS | DA | DS | DT | FD | FL | IS | LO | LT | PN | SH | SL | SS | ST | TM | UI |
UL | US, // if( VR & VL16 ) => VR has its VL coded over 16bits
00100     VL32 = OB | OW | OD | OF | OL | OV | SQ | SV | UC | UN | UR | UT | UV, // if( VR & VL32 ) => VR has
its VL coded over 32bits
00101     VRASCII = AE | AS | CS | DA | DS | DT | IS | LO | LT | PN | SH | ST | TM | UC | UI | UR | UT,
00102     VRBINARY = AT | FL | FD | OB | OD | OF | OL | OV | OW | SL | SQ | SS | SV | UL | UN | US | UV, //
FIXME: UN ?
00103     // PS 3.5:
00104     // Data Elements with a VR of SQ, OD, OF, OL, OW, OB or UN shall always have a Value Multiplicity of
one.
00105     // GDCM is adding a couple more: AS, LT, ST, UT
00106     VR_VM1 = AS | LT | ST | UT | SQ | OF | OL | OV | OD | OW | OB | UN, // All those VR have a VM1
00107     VRALL = VRASCII | VRBINARY,
00108     VR_END = UV+1 // Invalid VR, need to be max(VRType)+1
00109 };
00110
00111 static const char *GetVRString(VRType vr);
00112
00113 // This function will only look at the very first two chars nothing else
00114 static VRType GetVRTypeFromFile(const char *vr);
00115
00116 // You need to make sure end of string is \0
00117 static VRType GetVRType(const char *vr);
00118 static const char *GetVRStringFromFile(VRType vr);
00119
00120 static bool IsValid(const char *vr);
00121 // Check if vr1 is valid against vr2,
00122 // Typically vr1 is read from the file and vr2 is taken from the dict
00123 static bool IsValid(const char *vr1, VRType vr2);
00124 //static bool IsValid(const VRType &vr1, const VRType &vr2);
00125 // Find out if the string read is byte swapped
00126 static bool IsSwap(const char *vr);
00127
00128 // Size read on disk
00129 // FIXME: int ?
00130 int GetLength() const {
00131     return VR::GetLength(VRField);
00132 }
00133 unsigned int GetSizeof() const;
00134 static uint32_t GetLength(VRType vr) {
00135     //if( vr == VR::INVALID ) return 4;
00136     if( vr & VL32 )
00137     {
00138         return 4;
00139     }
00140     else
00141         return 2;
00142 }
00143
00144 // Some use of template metaprograming with ugly macro
00145 static bool IsBinary(VRType vr);
00146 static bool IsASCII(VRType vr);
00147 // TODO: REMOVE ME
00148 static bool CanDisplay(VRType vr);
00149 // TODO: REMOVE ME
00150 static bool IsBinary2(VRType vr);
00151 // TODO: REMOVE ME

```

```

00152     static bool IsASCII2(VRType vr);
00153
00154     VR(VRType vr = INVALID):VRField(vr) { }
00155     //VR(VR const &vr):VRField(vr.VRField) { }
00156     std::istream &Read(std::istream &is)
00157     {
00158         char vr[2];
00159         is.read(vr, 2);
00160         VRField = GetVRTypeFromFile(vr);
00161         assert( VRField != VR::VR_END );
00162         if( VRField == VR::INVALID )
00163         {
00164             // \0\2 Data/TheralysGDCM120Bug.dcm
00165             // \0\0
00166             Data/MR_Philips_Intera_PrivateSequenceExplicitVR_in_SQ_2001_e05f_item_wrong_lgt_use_NOSHADOWSEQ.dcm
00167             // \0\4 Data/BugGDCM2_UndefItemWrongVL.dcm
00168             // \44\0 Data/gdcm-MR-PHILIPS-16-Multi-Seq.dcm
00169             // \0\20 Data/ExplicitVRforPublicElementsImplicitVRforShadowElements.dcm
00170             // \0\3 Data/DMCPACS_ExplicitImplicit_BogusIOP.dcm
00171             // \0\4 Data/THERALYS-12-MONO2-Uncompressed-Even_Length_Tag.dcm
00172             // \0\4 Data/PrivateGEImplicitVRBigEndianTransferSyntax16Bits.dcm
00173             // \0\4 Data/GE_DLX-8-MONO2-PrivateSyntax.dcm
00174             throw Exception( "INVALID VR" );
00175         }
00176         if( VRField & VL32 )
00177         {
00178             #if 0
00179                 // For some reason this seems slower on my linux box...
00180                 is.seekg(2, std::ios::cur );
00181             #else
00182                 char dumb[2];
00183                 is.read(dumb, 2);
00184                 if( !(dumb[0] == 0 && dumb[1] == 0 ) )
00185                 {
00186                     // JDDICOM_Sample4.dcm
00187                     gdcmDebugMacro( "32bits VR contains non zero bytes. Skipped" );
00188                 }
00189             #endif
00190             return is;
00191         }
00192
00193     const std::ostream &Write(std::ostream &os) const
00194     {
00195         VRType vrfield = VRField;
00196         gdcmAssertAlwaysMacro( !IsDual() );
00197         if( vrfield == VR::INVALID )
00198         {
00199             //vrfield = VR::UN;
00200         }
00201         const char *vr = GetVRString(vrfield);
00202         //assert( strlen( vr ) == 2 );
00203         assert( vr[0] && vr[1] && vr[2] == 0 );
00204         os.write(vr, 2);
00205         // See PS 3.5, Data Element Structure With Explicit VR
00206         if( vrfield & VL32 )
00207         {
00208             const char dumb[2] = {0, 0};
00209             os.write(dumb,2);
00210         }
00211         return os;
00212     }
00213     friend std::ostream &operator<<(std::ostream &os, const VR &vr);
00214
00215     operator VRType () const { return VRField; }
00216
00217     unsigned int GetSize() const;
00218
00219     bool Compatible(VR const &vr) const;
00220
00221     bool IsVRFile() const;
00222
00223     bool IsDual() const;
00224
00225 private:
00226     // Internal function that map a VRType to an index in the VRStrings table
00227     static unsigned int GetIndex(VRType vr);
00228     VRType VRField;
00229 };
00230 //-----
00231 inline std::ostream &operator<<(std::ostream &_os, const VR &val)

```

```

00232 {
00233     //_os « VR::GetVRStringFromFile(val.VRField);
00234     _os « VR::GetVRString(val.VRField);
00235     return _os;
00236 }
00237
00238 // Apparently SWIG is not happy with something, somewhere below...
00239 #ifndef SWIG
00240
00241 // Tells whether VR Type is ASCII or Binary
00242 template<long long T> struct VRToEncoding;
00243 // Convert from VR Type to real underlying type
00244 template<long long T> struct VRToType;
00245 #define TYPETOENCODING(type,rep, rtype) \
00246     template<> struct VRToEncoding<VR::type> \
00247     { enum:long long { Mode = VR::rep }; }; \
00248     template<> struct VRToType<VR::type> \
00249     { typedef rtype Type; };
00250
00251
00252 // Do not use me
00253 struct UI { char Internal[64+1];
00254     friend std::ostream& operator<(std::ostream &_os, const UI &_val);
00255 };
00256 inline std::ostream& operator<(std::ostream &_os, const UI &_val)
00257 {
00258     _os « _val.Internal;
00259     return _os;
00260 }
00261
00262 typedef String<'\\',16> AECComp;
00263 typedef String<'\\',64> ASCComp;
00264 typedef String<'\\',16> CSCComp;
00265 typedef String<'\\',64> DACComp;
00266 typedef String<'\\',64> DTCComp;
00267 typedef String<'\\',64> LOComp;
00268 typedef String<'\\',64> LTCComp;
00269 typedef String<'\\',64> PNComp;
00270 typedef String<'\\',64> SHComp;
00271 typedef String<'\\',64> STComp;
00272 typedef String<'\\',4294967294> UCCComp;
00273 typedef String<'\\',4294967294> URComp;
00274 typedef String<'\\',16> TMComp;
00275 typedef String<'\\',64,0> UICComp;
00276 typedef String<'\\',64> UTCComp;
00277
00278
00279 // TODO: Could be generated from XML file
00280 TYPETOENCODING(AE,VRASCII ,AECComp)
00281 TYPETOENCODING(AS,VRASCII ,ASCComp)
00282 TYPETOENCODING(AT,VRBINARY,Tag)
00283 TYPETOENCODING(CS,VRASCII ,CSCComp)
00284 TYPETOENCODING(DA,VRASCII ,DACComp)
00285 TYPETOENCODING(DS,VRASCII ,double)
00286 TYPETOENCODING(DT,VRASCII ,DTCComp)
00287 TYPETOENCODING(FL,VRBINARY,float)
00288 TYPETOENCODING(FD,VRBINARY,double)
00289 TYPETOENCODING(IS,VRASCII ,int32_t)
00290 TYPETOENCODING(LO,VRASCII ,LOComp)
00291 TYPETOENCODING(LT,VRASCII ,LTCComp)
00292 TYPETOENCODING(OB,VRBINARY,uint8_t)
00293 TYPETOENCODING(OD,VRBINARY,double)
00294 TYPETOENCODING(OF,VRBINARY,float)
00295 TYPETOENCODING(OL,VRBINARY,uint32_t)
00296 TYPETOENCODING(OV,VRBINARY,uint64_t)
00297 TYPETOENCODING(OW,VRBINARY,uint16_t)
00298 TYPETOENCODING(PN,VRASCII ,PNComp)
00299 TYPETOENCODING(SH,VRASCII ,SHComp)
00300 TYPETOENCODING(SL,VRBINARY,int32_t)
00301 TYPETOENCODING(SQ,VRBINARY,unsigned char) // FIXME
00302 TYPETOENCODING(SS,VRBINARY,int16_t)
00303 TYPETOENCODING(ST,VRASCII ,STComp)
00304 TYPETOENCODING(SV,VRBINARY,int64_t)
00305 TYPETOENCODING(TM,VRASCII ,TMComp)
00306 TYPETOENCODING(UC,VRASCII ,UCCComp)
00307 TYPETOENCODING(UI,VRASCII ,UICComp)
00308 TYPETOENCODING(UL,VRBINARY,uint32_t)
00309 TYPETOENCODING(UN,VRBINARY,uint8_t) // FIXME ?
00310 TYPETOENCODING(UR,VRASCII ,URComp)
00311 TYPETOENCODING(US,VRBINARY,uint16_t)
00312 TYPETOENCODING(UT,VRASCII ,UTCComp)

```

```

00313 TYPETOENCODING(UV,VRBINARY,uint64_t)
00314
00315 #define VRTypeTemplateCase(type) \
00316     case VR::type: \
00317         return sizeof ( VRToType<VR::type>::Type );
00318
00319 inline unsigned int VR::GetSize() const
00320 {
00321     switch(VRField)
00322     {
00323         VRTypeTemplateCase(AE)
00324         VRTypeTemplateCase(AS)
00325         VRTypeTemplateCase(AT)
00326         VRTypeTemplateCase(CS)
00327         VRTypeTemplateCase(DA)
00328         VRTypeTemplateCase(DS)
00329         VRTypeTemplateCase(DT)
00330         VRTypeTemplateCase(FL)
00331         VRTypeTemplateCase(FD)
00332         VRTypeTemplateCase(IS)
00333         VRTypeTemplateCase(LO)
00334         VRTypeTemplateCase(LT)
00335         VRTypeTemplateCase(OB)
00336         VRTypeTemplateCase(OD)
00337         VRTypeTemplateCase(OF)
00338         VRTypeTemplateCase(OL)
00339         VRTypeTemplateCase(OV)
00340         VRTypeTemplateCase(OW)
00341         VRTypeTemplateCase(PN)
00342         VRTypeTemplateCase(SH)
00343         VRTypeTemplateCase(SL)
00344         VRTypeTemplateCase(SQ)
00345         VRTypeTemplateCase(SS)
00346         VRTypeTemplateCase(ST)
00347         VRTypeTemplateCase(SV)
00348         VRTypeTemplateCase(TM)
00349         VRTypeTemplateCase(UC)
00350         VRTypeTemplateCase(UI)
00351         VRTypeTemplateCase(UL)
00352         VRTypeTemplateCase(UN)
00353         VRTypeTemplateCase(UR)
00354         VRTypeTemplateCase(US)
00355         VRTypeTemplateCase(UT)
00356         VRTypeTemplateCase(UV)
00357         case VR::US_SS:
00358             return 2;
00359
00360         case VR::INVALID:
00361         case VR::OB_OW:
00362         case VR::US_SS_OW:
00363         case VR::US_OW:
00364         case VR::VL16:
00365         case VR::VL32:
00366         case VR::VRASCII:
00367         case VR::VRBINARY:
00368         case VR::VR_VM1:
00369         case VR::VRALL:
00370         case VR::VR_END:
00371         default:
00372             assert( 0 && "should not" );
00373     }
00374     return 0;
00375 }
00376 #endif // SWIG
00377
00378
00379 } // end namespace gdcm
00380
00381 #endif //GDCMVR_H

```

## 11.195 gdcmVR16ExplicitDataElement.h File Reference

```

#include "gdcmDataElement.h"
#include "gdcmVR16ExplicitDataElement.txx"

```



- class `gdcm::VR16ExplicitDataElement`  
*Class to read/write a `DataElement` as Explicit Data `Element`.*

- namespace **gdcm**

[Go to the documentation of this file.](#)

Generated by Doxygen



**Classes**

- class `gdcm::Writer`  
*Writer ala DOM (Document Object Model)*

**Namespaces**

- namespace `gdcm`

**11.198 gdcmWriter.h**

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014
00015 #ifndef GDCMWRITER_H
00016 #define GDCMWRITER_H
00017
00018 #include "gdcmFile.h"
00019
00020 namespace gdcm
00021 {
00022
00023   class FileMetaInformation;
00024   class GDCM_EXPORT Writer
00025   {
00026   public:
00027     Writer();
00028     virtual ~Writer();
00029
00030     virtual bool Write(); // Execute()
00031
00032     void SetFileName(const char *filename_native);
00033
00034     void SetStream(std::ostream &output_stream) {
00035       Stream = &output_stream;
00036     }
00037
00038     void SetFile(const File& f) { F = f; }
00039     File &GetFile() { return *F; }
00040
00041     void SetCheckFileMetaInformation(bool b) { CheckFileMetaInformation = b; }
00042     void CheckFileMetaInformationOff() { CheckFileMetaInformation = false; }
00043     void CheckFileMetaInformationOn() { CheckFileMetaInformation = true; }
00044
00045 protected:
00046     void SetWriteDataSetOnly(bool b) { WriteDataSetOnly = b; }
00047
00048 protected:
00049     friend class StreamImageWriter;
00050     //this function is added for the StreamImageWriter, which needs to write
00051     //up to the pixel data and then stops right before writing the pixel data.
00052     //after that, for the raw codec at least, zeros are written for the length of the data
00053     std::ostream* GetStreamPtr() const { return Stream; }
00054
00055 protected:
00056     std::ostream *Stream;
00057     std::ofstream *Ofstream;

```

```

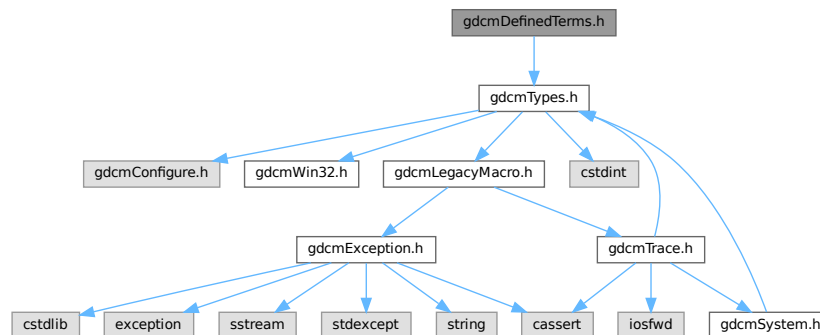
00087  bool GetCheckFileMetaInformation() const { return CheckFileMetaInformation; }
00088
00089 private:
00090  SmartPointer<File> F;
00091  bool CheckFileMetaInformation;
00092  bool WriteDataSetOnly;
00093 };
00094
00095 } // end namespace gdcm
00096
00097 #endif //GDCMWRITER_H

```

## 11.199 gdcmDefinedTerms.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmDefinedTerms.h:



### Classes

- class [gdcm::DefinedTerms](#)

*Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data [Element](#) with Defined Terms that does not contain a [Value](#) equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation [Type ID](#) (4008,0210) is an example of a Data [Element](#) having Defined Terms. It is defined to have a [Value](#) that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data [Element](#) has Defined Terms other Interpretation [Type IDs](#) may be defined by the implementor.*

### Namespaces

- namespace [gdcm](#)

## 11.200 gdcmDefinedTerms.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMDEFINEDTERMS_H
00015 #define GDCMDEFINEDTERMS_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021   class GDCM_EXPORT DefinedTerms
00022   {
00023   public:
00024     DefinedTerms() = default;
00025   private:
00026   };
00027 } // end namespace gdcm
00028
00029 #endif //GDCMDEFINEDTERMS_H

```

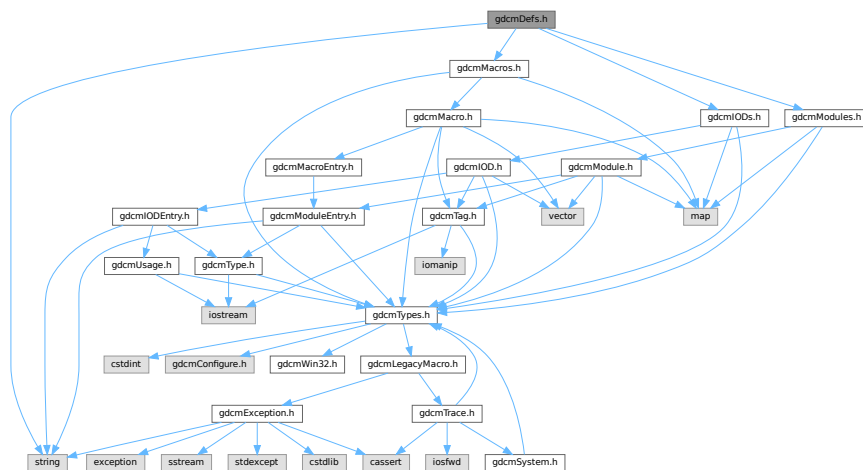
## 11.201 gdcmDefs.h File Reference

```

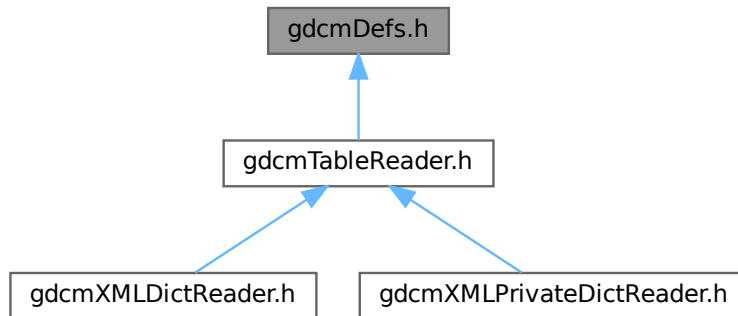
#include "gdcmModules.h"
#include "gdcmMacros.h"
#include "gdcmIODs.h"
#include <string>

```

Include dependency graph for gdcmDefs.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcM::Defs](#)  
*FIXME I do not like the name 'Defs'.*

## Namespaces

- namespace [gdcM](#)

## 11.202 gdcMDefs.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcM.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 #ifndef GDCMDEFS_H
00015 #define GDCMDEFS_H
00016
00017 #include "gdcMModules.h"
00018 #include "gdcMacros.h"
00019 #include "gdcIODs.h"
00020
00021 #include <string>
00022
00023 namespace gdcM
00024 {
00025     class DataSet;
00026     class File;
  
```

```

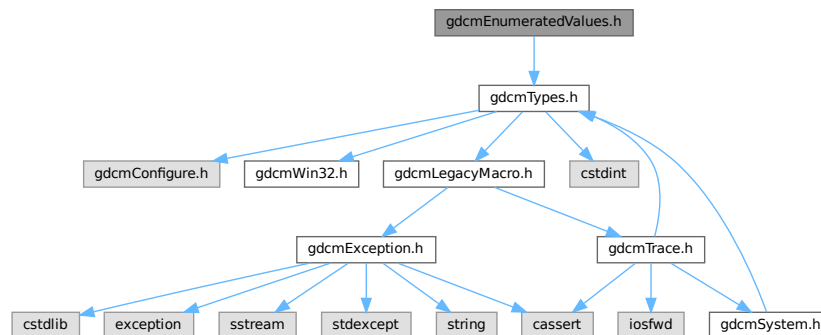
00027 class MediaStorage;
00032 class GDCM_EXPORT Defs
00033 {
00034 public:
00035     Defs();
00036     ~Defs();
00037     Defs &operator=(const Defs &val) = delete;
00038     Defs(const Defs &val) = delete;
00039
00040     const Modules &GetModules() const { return Part3Modules; }
00041     Modules &GetModules() { return Part3Modules; }
00042
00043     const Macros &GetMacros() const { return Part3Macros; }
00044     Macros &GetMacros() { return Part3Macros; }
00045
00046     const IODs & GetIODs() const { return Part3IODs; }
00047     IODs & GetIODs() { return Part3IODs; }
00048
00049     bool IsEmpty() const { return GetModules().IsEmpty(); }
00050
00051     bool Verify(const File& file) const;
00052
00053     // \deprecated DO NOT USE
00054     bool Verify(const DataSet& ds) const;
00055
00056     Type GetTypeFromTag(const File& file, const Tag& tag) const;
00057
00058     static const char *GetIODNameFromMediaStorage(MediaStorage const &ms);
00059
00060     const IOD& GetIODFromFile(const File& file) const;
00061
00062 protected:
00063     friend class Global;
00064     void LoadDefaults();
00065     void LoadFromFile(const char *filename);
00066 private:
00067     // Part 3 stuff:
00068     Macros Part3Macros;
00069     Modules Part3Modules;
00070     IODs Part3IODs;
00071 };
00072
00073 } // end namespace gdcM
00074
00075 #endif //GDCMDEFS_H

```

## 11.203 gdcMEnumeratedValues.h File Reference

```
#include "gdcMTypes.h"
```

Include dependency graph for gdcMEnumeratedValues.h:



## Classes

- class [gdcm::EnumeratedValues](#)

*Element.* A Data [Element](#) with Enumerated Values that does not have a [Value](#) equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:

## Namespaces

- namespace [gdcm](#)

## 11.204 gdcmEnumeratedValues.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMENUMERATEDVALUES_H
00015 #define GDCMENUMERATEDVALUES_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00034   class GDCM_EXPORT EnumeratedValues
00035   {
00036   public:
00037     EnumeratedValues() = default;
00038   private:
00039   };
00040
00041 } // end namespace gdcm
00042
00043 #endif //GDCMENUMERATEDVALUES_H

```

## 11.205 gdcmIOD.h File Reference

```

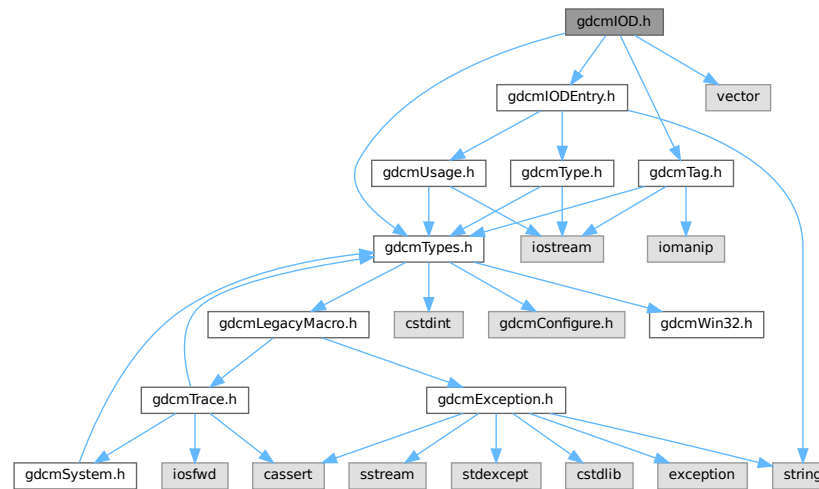
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmIODEntry.h"

```

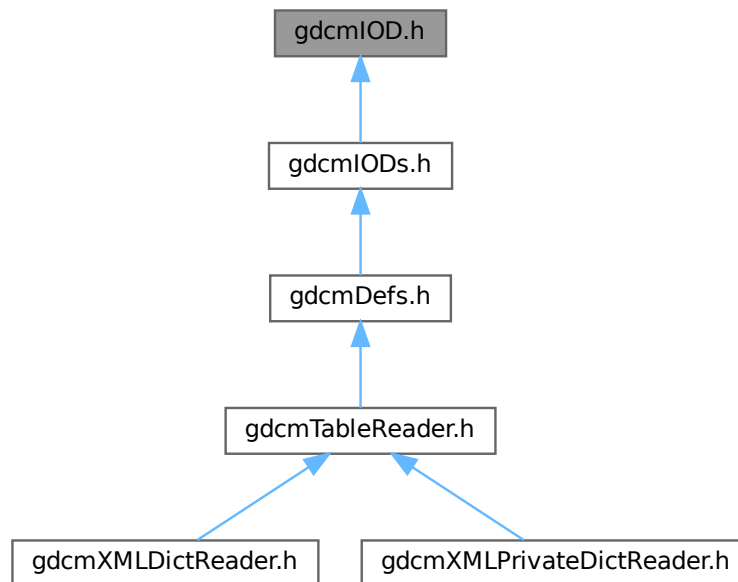


```
#include <vector>
```

Include dependency graph for gdcmIOD.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::IOD](#)

*Class for representing a [IOD](#).*

## Namespaces

- namespace [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const IOD &_val)`

## 11.206 gdcmIOD.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMIOD_H
00015 #define GDCMIOD_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmTag.h"
00019 #include "gdcmIODEntry.h"
00020
00021 #include <vector>
00022
00023 namespace gdcm
00024 {
00025 class DataSet;
00026 class Defs;
00027
00028 class GDCM_EXPORT IOD
00029 {
00030 public:
00031     typedef std::vector<IODEntry> MapIODEntry;
00032     typedef MapIODEntry::size_type SizeType;
00033
00034     IOD() = default;
00035     friend std::ostream& operator<<(std::ostream& _os, const IOD &_val);
00036
00037     void Clear() { IODInternal.clear(); }
00038
00039     void AddIODEntry(const IODEntry &iode)
00040     {
00041         IODInternal.push_back(iode);
00042     }
00043
00044     SizeType GetNumberOfIODs() const {
00045         return IODInternal.size();
00046     }
00047
00048     const IODEntry& GetIODEntry(SizeType idx) const
00049     {
00050         return IODInternal[idx];
00051     }
00052
00053     Type GetTypeFromTag(const Defs &defs, const Tag& tag) const;
00054
00055
00056
00057
00058
00059
00060

```

```

00061 private:
00062     //IOD &operator=(const IOD &_val); // purposely not implemented
00063     //IOD(const IOD &_val); // purposely not implemented
00064
00065     MapIODEntry IODInternal;
00066 };
00067 //-----
00068 inline std::ostream& operator<<(std::ostream& _os, const IOD &_val)
00069 {
00070     IOD::MapIODEntry::const_iterator it = _val.IODInternal.begin();
00071     for(; it != _val.IODInternal.end(); ++it)
00072     {
00073         _os << *it << '\n';
00074     }
00075
00076     return _os;
00077 }
00078
00079 } // end namespace gdcm
00080
00081 #endif //GDCMIOD_H

```

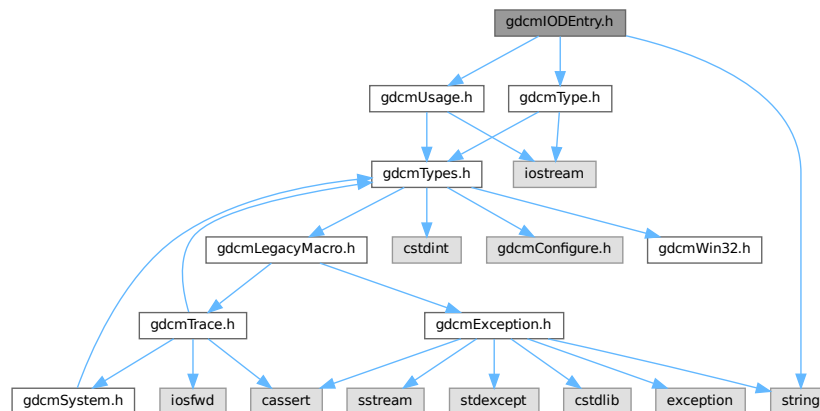
## 11.207 gdcmIODEntry.h File Reference

```

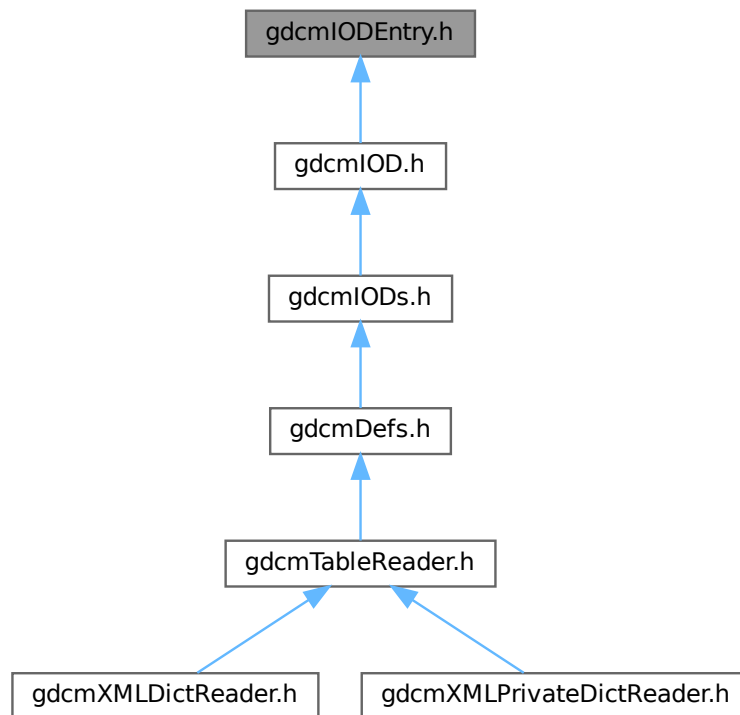
#include "gdcmUsage.h"
#include "gdcmType.h"
#include <string>

```

Include dependency graph for gdcmIODEntry.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcml::IODEntry](#)  
*Class for representing a [IODEntry](#).*

## Namespaces

- namespace [gdcml](#)

## Functions

- `std::ostream & gdcml::operator<< (std::ostream &_os, const IODEntry &_val)`

## 11.208 gdcmIODEntry.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMIODENTRY_H
00015 #define GDCMIODENTRY_H
00016
00017 #include "gdcmUsage.h"
00018 #include "gdcmType.h"
00019
00020 #include <string>
00021
00022 namespace gdcm
00023 {
00024
00051 class GDCM_EXPORT IODEntry
00052 {
00053 public:
00054   IODEntry(const char *name = "", const char *ref = "", const char *inUsage =
00055   ""):Name(name),Ref(ref),usage(inUsage) {
00056
00057   friend std::ostream& operator<<(std::ostream& _os, const IODEntry &_val);
00058
00059   void SetIE(const char *ie) { IE = ie; }
00060   const char *GetIE() const { return IE.c_str(); }
00061
00062   void SetName(const char *name) { Name = name; }
00063   const char *GetName() const { return Name.c_str(); }
00064
00065   void SetRef(const char *ref) { Ref = ref; }
00066   const char *GetRef() const { return Ref.c_str(); }
00067
00068   void SetUsage(const char *inUsage) { usage = inUsage; }
00069   const char *GetUsage() const { return usage.c_str(); }
00070   Usage::UsageType GetUsageType() const;
00071 private:
00072   std::string IE;
00073
00074   std::string Name;
00075
00076   std::string Ref;
00077
00078   std::string usage;
00079 };
00080 //-----
00081 inline std::ostream& operator<<(std::ostream& _os, const IODEntry &_val)
00082 {
00083   _os << _val.IE << "\t" << _val.Name << "\t" << _val.Ref << "\t" << _val.usage;
00084   return _os;
00085 }
00086
00087 } // end namespace gdcm
00088
00089 #endif //GDCMIODENTRY_H

```

## 11.209 gdcmIODs.h File Reference

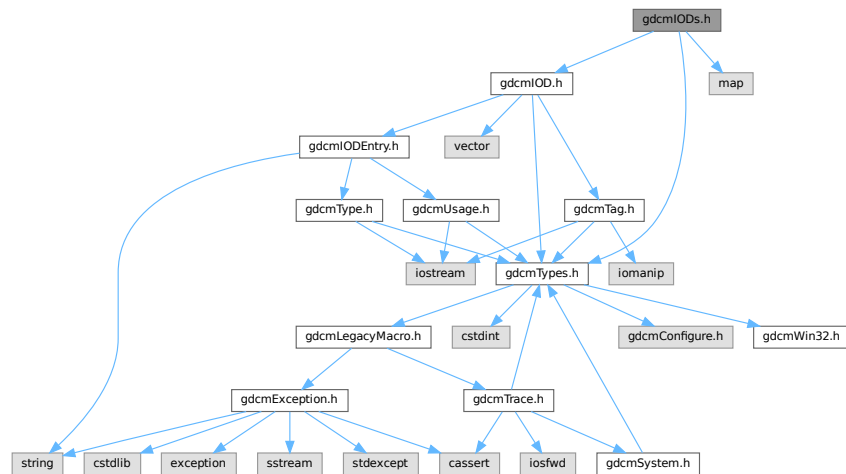
```

#include "gdcmTypes.h"
#include "gdcmIOD.h"

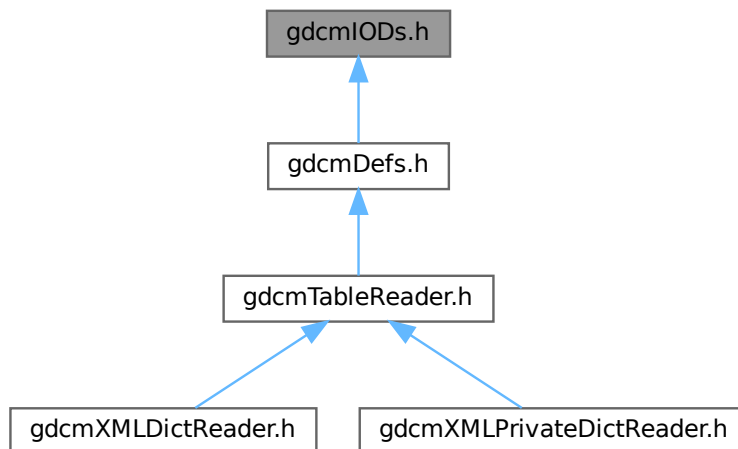
```

```
#include <map>
```

Include dependency graph for gdcmlODs.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcml::IODs](#)  
Class for representing a *IODs*.

## Namespaces

- namespace [gdcm](#)

## Functions

- [std::ostream & gdcm::operator<<](#) (std::ostream &\_os, const [IODs](#) &\_val)

## 11.210 gdcmIODs.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMIODS_H
00015 #define GDCMIODS_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmIOD.h"
00019
00020 #include <map>
00021
00022 namespace gdcm
00023 {
00024     class GDCM_EXPORT IODs
00025     {
00026     public:
00027         typedef std::string IODName;
00028         typedef std::map<IODName, IOD> IODMapType;
00029
00030         IODs() = default;
00031         friend std::ostream& operator<<(std::ostream& _os, const IODs &_val);
00032
00033         void Clear() { IODsInternal.clear(); }
00034
00035         void AddIOD(const char *name, const IOD & module)
00036         {
00037             IODsInternal.insert(
00038                 IODMapType::value_type(name, module));
00039         }
00040         const IOD &GetIOD(const char *name) const
00041         {
00042             //return IODsInternal[name];
00043             IODMapType::const_iterator it = IODsInternal.find( name );
00044             assert( it != IODsInternal.end() );
00045             assert( it->first == name );
00046             return it->second;
00047         }
00048
00049         typedef IODMapType::const_iterator IODMapTypeConstIterator;
00050         IODMapTypeConstIterator Begin() const { return IODsInternal.begin(); }
00051         IODMapTypeConstIterator End() const { return IODsInternal.end(); }
00052
00053     private:
00054         IODMapType IODsInternal;
00055     };
00056
00057 //-----
00058 inline std::ostream& operator<<(std::ostream& _os, const IODs &_val)
00059 {
00060     IODs::IODMapType::const_iterator it = _val.IODsInternal.begin();

```

```

00065     for(; it != _val.IODsInternal.end(); ++it)
00066     {
00067         const std::string &name = it->first;
00068         const IOD &m = it->second;
00069         _os << name << " " << m << '\n';
00070     }
00071
00072     return _os;
00073 }
00074
00075
00076 } // end namespace gdcM
00077
00078 #endif //GDCMIODS_H

```

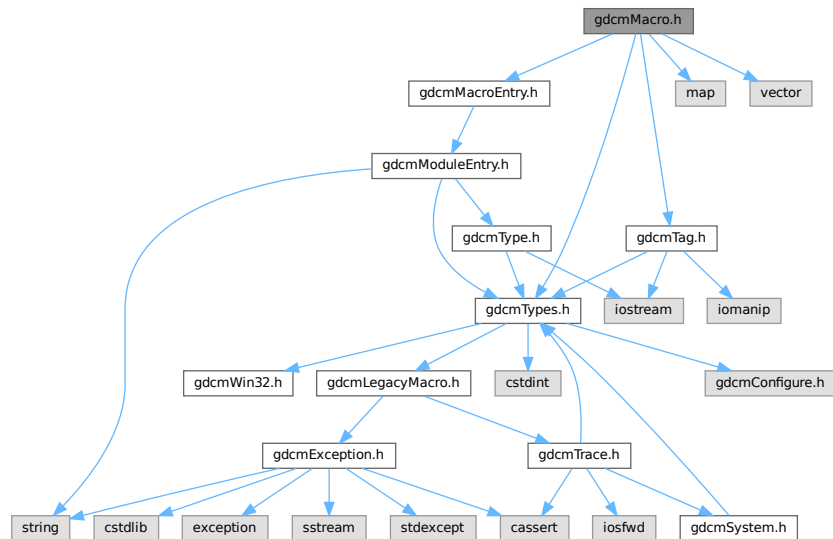
## 11.211 gdcMMacro.h File Reference

```

#include "gdcMTypes.h"
#include "gdcMTag.h"
#include "gdcMMacroEntry.h"
#include <map>
#include <vector>

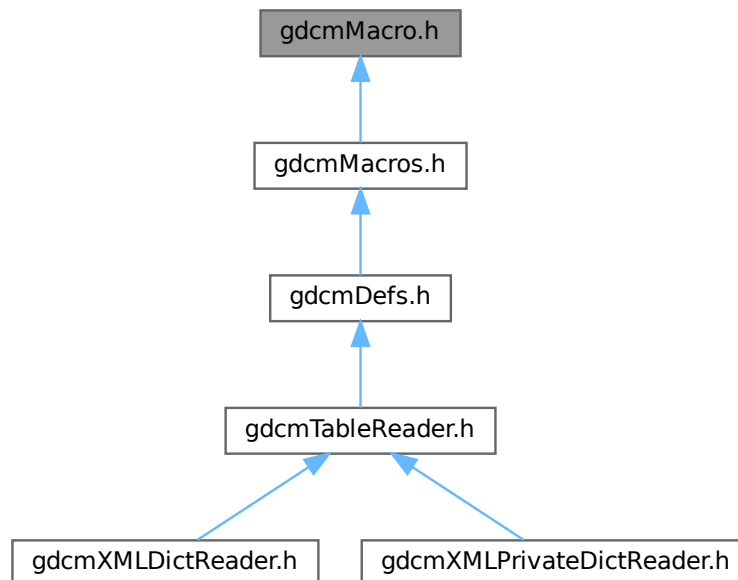
```

Include dependency graph for gdcMMacro.h:





This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::Macro](#)  
*Class for representing a [Macro](#).*

## Namespaces

- namespace [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Macro &_val)`

## 11.212 gdcmMacro.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
  
```

```

00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMMACRO_H
00015 #define GDCMMACRO_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmTag.h"
00019 #include "gdcmMacroEntry.h"
00020
00021 #include <map>
00022 #include <vector>
00023
00024 namespace gdcm
00025 {
00026
00027 class DataSet;
00028 class Usage;
00036 class GDCM_EXPORT Macro
00037 {
00038 public:
00039 typedef std::map<Tag, MacroEntry> MapModuleEntry;
00040 typedef std::vector<std::string> ArrayIncludeMacrosType;
00041
00042 //typedef MapModuleEntry::const_iterator ConstIterator;
00043 //typedef MapModuleEntry::iterator Iterator;
00044 //ConstIterator Begin() const { return ModuleInternal.begin(); }
00045 //Iterator Begin() { return ModuleInternal.begin(); }
00046 //ConstIterator End() const { return ModuleInternal.end(); }
00047 //Iterator End() { return ModuleInternal.end(); }
00048
00049 Macro() = default;
00050 friend std::ostream& operator<(std::ostream& _os, const Macro& _val);
00051
00052 void Clear() { ModuleInternal.clear(); }
00053
00055 void AddMacroEntry(const Tag& tag, const MacroEntry & module)
00056 {
00057     ModuleInternal.insert(
00058         MapModuleEntry::value_type(tag, module));
00059 }
00060
00063 bool FindMacroEntry(const Tag &tag) const;
00064 const MacroEntry& GetMacroEntry(const Tag &tag) const;
00065
00066 void SetName( const char *name) { Name = name; }
00067 const char *GetName() const { return Name.c_str(); }
00068
00069 // Verify will print on std::cerr for error
00070 // Upon success will return true, false otherwise
00071 bool Verify(const DataSet& ds, Usage const & usage) const;
00072
00073 private:
00074 //Module &operator=(const Module &_val); // purposely not implemented
00075 //Module(const Module &_val); // purposely not implemented
00076
00077 MapModuleEntry ModuleInternal;
00078 std::string Name;
00079 };
00080 //-----
00081 inline std::ostream& operator<(std::ostream& _os, const Macro &_val)
00082 {
00083     _os << _val.Name << '\n';
00084     Macro::MapModuleEntry::const_iterator it = _val.ModuleInternal.begin();
00085     for(; it != _val.ModuleInternal.end(); ++it)
00086     {
00087         const Tag &t = it->first;
00088         const MacroEntry &de = it->second;
00089         _os << t << " " << de << '\n';
00090     }
00091
00092     return _os;
00093 }
00094
00095 } // end namespace gdcm
00096

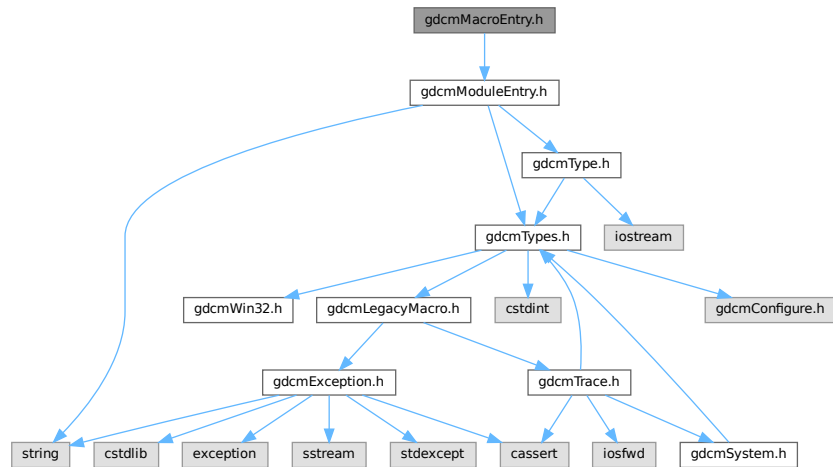
```

```
00097 #endif //GDCMMACRO_H
```

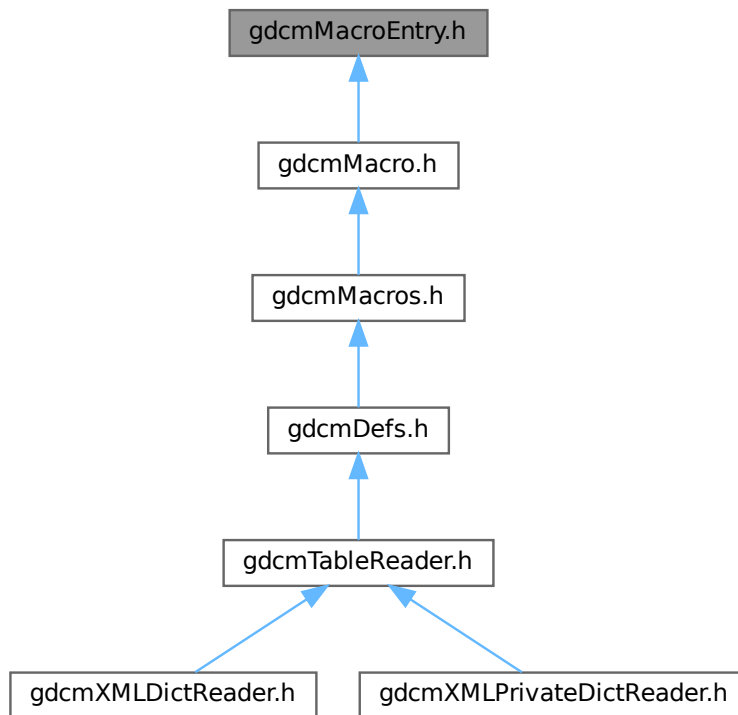
## 11.213 gdcmMacroEntry.h File Reference

```
#include "gdcmModuleEntry.h"
```

Include dependency graph for gdcmMacroEntry.h:



This graph shows which files directly or indirectly include this file:



## Macros

- `#define GDCMMACROENTRY_H`

## 11.213.1 Macro Definition Documentation

### 11.213.1.1 GDCMMACROENTRY\_H

```
#define GDCMMACROENTRY_H
```

## 11.214 gdcMacroEntry.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
```

```

00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdc.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #if 0
00015 #ifndef GDCMMACROENTRY_H
00016 #define GDCMMACROENTRY_H
00017
00018 #include "gdcTypes.h"
00019 #include "gdcType.h"
00020
00021 #include <string>
00022
00023 namespace gdc
00024 {
00025     class GDCM_EXPORT MacroEntry
00026     {
00027     public:
00028         MacroEntry(const char *name = "", const char *type = "3", const char *description =
00029         "") : Name(name) /*, Type(type)*/, DescriptionField(description) {
00030             DataElementType = Type::GetTypeType(type);
00031         }
00032         virtual ~MacroEntry() {} // important
00033         friend std::ostream& operator<(std::ostream& _os, const MacroEntry &_val);
00034
00035         void SetName(const char *name) { Name = name; }
00036         const char *GetName() const { return Name.c_str(); }
00037
00038         void SetType(const Type &type) { DataElementType = type; }
00039         const Type &GetType() const { return DataElementType; }
00040
00041         /*
00042          * WARNING: 'Description' is currently a std::string, but it might change in the future
00043          * do not expect it to remain the same, and always use the ModuleEntry::Description typedef
00044          * instead.
00045          */
00046         typedef std::string Description;
00047         void SetDescription(const char *d) { DescriptionField = d; }
00048         const Description & GetDescription() const { return DescriptionField; }
00049
00050     protected:
00051         // PS 3.3 repeats the name of an attribute, but often contains typos
00052         // for now we will not use this info, but instead access the DataDict instead
00053         std::string Name;
00054
00055         // An attribute, encoded as a Data Element, may or may not be required in a
00056         // Data Set, depending on that Attribute's Data Element Type.
00057         Type DataElementType;
00058
00059         // TODO: for now contains the raw description (with enumerated values, defined terms...)
00060         Description DescriptionField;
00061     };
00062
00063     //-----
00064     inline std::ostream& operator<(std::ostream& _os, const MacroEntry &_val)
00065     {
00066         _os << _val.Name << "\t" << _val.DataElementType << "\t" << _val.DescriptionField;
00067         return _os;
00068     }
00069
00070 } // end namespace gdc
00071
00072 #endif //GDCMMODULEENTRY_H
00073 #endif
00074
00075 #ifndef GDCMMACROENTRY_H
00076 #define GDCMMACROENTRY_H
00077 #include "gdcModuleEntry.h"
00078 #endif

```

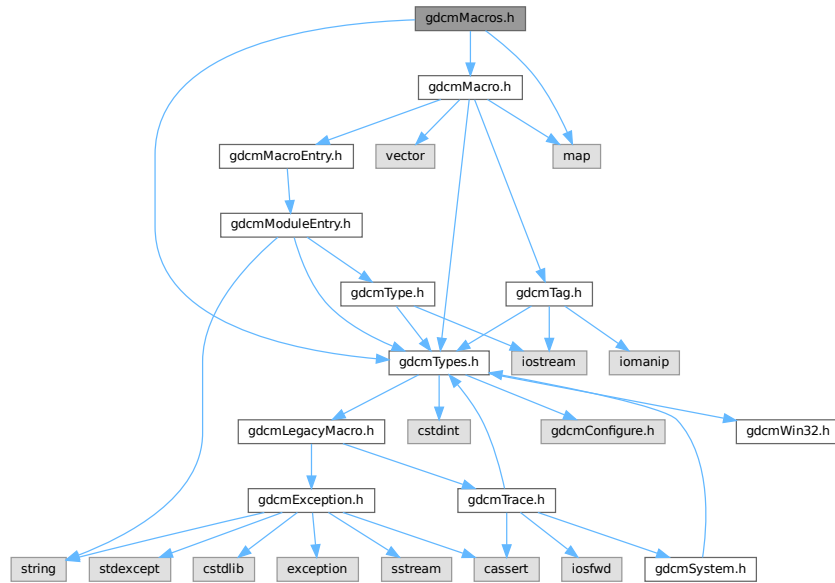
## 11.215 gdcmMacros.h File Reference

```
#include "gdcmTypes.h"
```

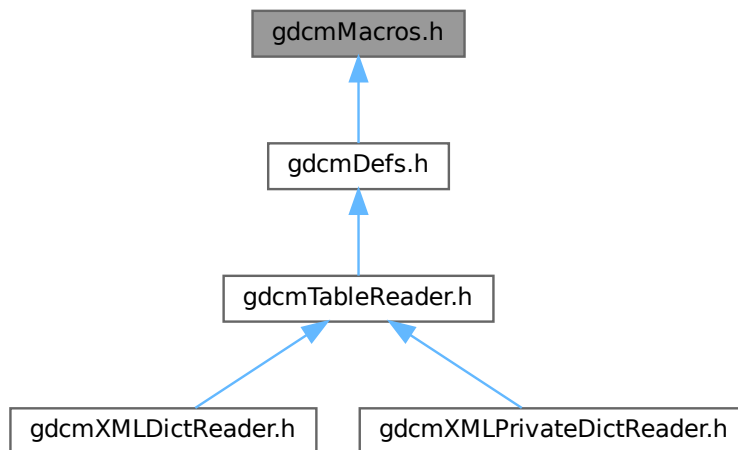
```
#include "gdcmMacro.h"
```

```
#include <map>
```

Include dependency graph for gdcmMacros.h:



This graph shows which files directly or indirectly include this file:



**Classes**

- class [gdcm::Macros](#)  
*Class for representing a [Modules](#).*

**Namespaces**

- namespace [gdcm](#)

**Functions**

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Macros &_val)`

**11.216 gdcmMacros.h**

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 #ifndef GDCMMACROS_H
00015 #define GDCMMACROS_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmMacro.h"
00019
00020 #include <map>
00021
00022 namespace gdcm
00023 {
00024     class GDCM_EXPORT Macros
00025     {
00026     public:
00027         typedef std::map<std::string, Macro> ModuleMapType;
00028
00029         Macros() = default;
00030         friend std::ostream& operator<<(std::ostream& _os, const Macros& _val);
00031
00032         void Clear() { ModulesInternal.clear(); }
00033
00034         // A Module is inserted based on it's ref
00035         void AddMacro(const char *ref, const Macro & module )
00036         {
00037             assert( ref && *ref );
00038             assert( ModulesInternal.find( ref ) == ModulesInternal.end() );
00039             ModulesInternal.insert (
00040                 ModuleMapType::value_type(ref, module));
00041         }
00042         const Macro &GetMacro(const char *name) const
00043         {
00044             assert( name && *name );
00045             ModuleMapType::const_iterator it = ModulesInternal.find( name );
00046             assert( it != ModulesInternal.end() );
00047             assert( it->first == name );
00048             return it->second;
00049         }
00050     }
00051
00052 }

```

```

00056     bool IsEmpty() const { return ModulesInternal.empty(); }
00057
00058 private:
00059     ModuleMapType ModulesInternal;
00060 };
00061 //-----
00062 inline std::ostream& operator<<(std::ostream& _os, const Macros &_val)
00063 {
00064     Macros::ModuleMapType::const_iterator it = _val.ModulesInternal.begin();
00065     for(; it != _val.ModulesInternal.end(); ++it)
00066     {
00067         const std::string &name = it->first;
00068         const Macro &m = it->second;
00069         _os << name << " " << m << '\n';
00070     }
00071
00072     return _os;
00073 }
00074
00075
00076 } // end namespace gdcmm
00077
00078 #endif //GDCMMODULES_H

```

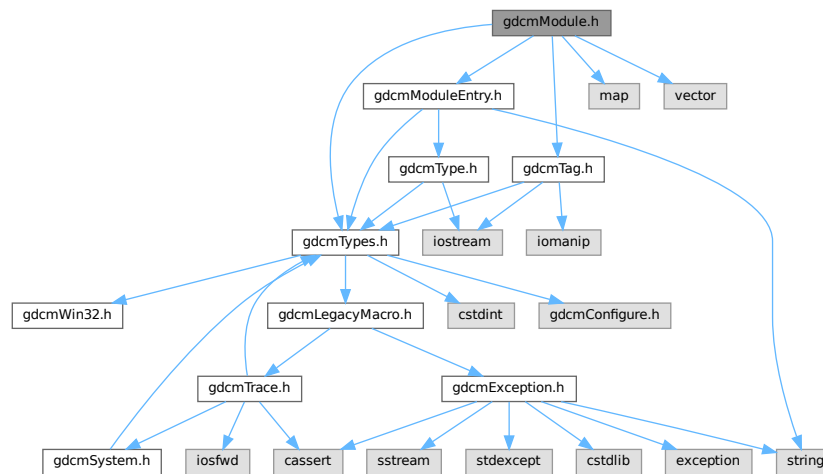
## 11.217 gdcmmModule.h File Reference

```

#include "gdcmmTypes.h"
#include "gdcmmTag.h"
#include "gdcmmModuleEntry.h"
#include <map>
#include <vector>

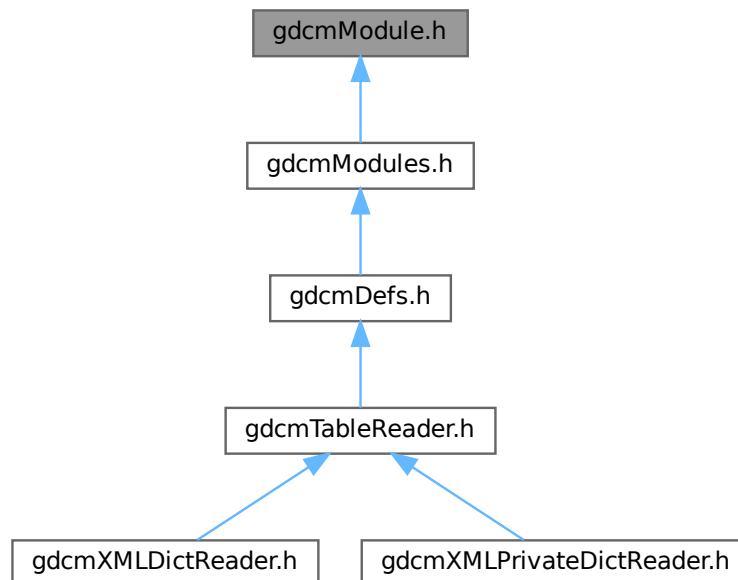
```

Include dependency graph for gdcmmModule.h:





This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::Module](#)  
*Class for representing a [Module](#).*

## Namespaces

- namespace [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Module &_val)`

## 11.218 gdcmModule.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
  
```

```

00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMMODULE_H
00015 #define GDCMMODULE_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmTag.h"
00019 #include "gdcmModuleEntry.h"
00020
00021 #include <map>
00022 #include <vector>
00023
00024 namespace gdcm
00025 {
00026
00027 class DataSet;
00028 class Usage;
00029 class Macros;
00037 class GDCM_EXPORT Module
00038 {
00039 public:
00040 typedef std::map<Tag, ModuleEntry> MapModuleEntry;
00041 typedef std::vector<std::string> ArrayIncludeMacrosType;
00042
00043 //typedef MapModuleEntry::const_iterator ConstIterator;
00044 //typedef MapModuleEntry::iterator Iterator;
00045 //ConstIterator Begin() const { return ModuleInternal.begin(); }
00046 //Iterator Begin() { return ModuleInternal.begin(); }
00047 //ConstIterator End() const { return ModuleInternal.end(); }
00048 //Iterator End() { return ModuleInternal.end(); }
00049
00050 Module() = default;
00051 friend std::ostream& operator<<(std::ostream& _os, const Module &_val);
00052
00053 void Clear() { ModuleInternal.clear(); }
00054
00055 void AddModuleEntry(const Tag& tag, const ModuleEntry & module)
00056 {
00057     ModuleInternal.insert(
00058         MapModuleEntry::value_type(tag, module));
00059 }
00060
00061 void AddMacro(const char *include)
00062 {
00063     ArrayIncludeMacros.push_back( include );
00064 }
00065
00066 bool FindModuleEntryInMacros(Macros const &macros, const Tag &tag) const;
00067 const ModuleEntry& GetModuleEntryInMacros(Macros const &macros, const Tag &tag) const;
00071
00072 void SetName( const char *name) { Name = name; }
00073 const char *GetName() const { return Name.c_str(); }
00074
00075 // Verify will print on std::cerr for error
00076 // Upon success will return true, false otherwise
00077 bool Verify(const DataSet& ds, Usage const & usage) const;
00078
00079 private:
00080 //Module &operator=(const Module &_val); // purposely not implemented
00081 //Module(const Module &_val); // purposely not implemented
00082
00083 MapModuleEntry ModuleInternal;
00084 std::string Name;
00085 ArrayIncludeMacrosType ArrayIncludeMacros;
00086 };
00087 //-----
00088 inline std::ostream& operator<<(std::ostream& _os, const Module &_val)
00089 {
00090     _os << _val.Name << '\n';
00091     Module::MapModuleEntry::const_iterator it = _val.ModuleInternal.begin();
00092     for(; it != _val.ModuleInternal.end(); ++it)
00093     {
00094         const Tag &t = it->first;
00095         const ModuleEntry &de = it->second;
00096         _os << t << " " << de << '\n';
00097     }
00098 }

```

```

00097     }
00098
00099     return _os;
00100 }
00101
00102 } // end namespace gdcm
00103
00104 #endif //GDCMMODULE_H

```

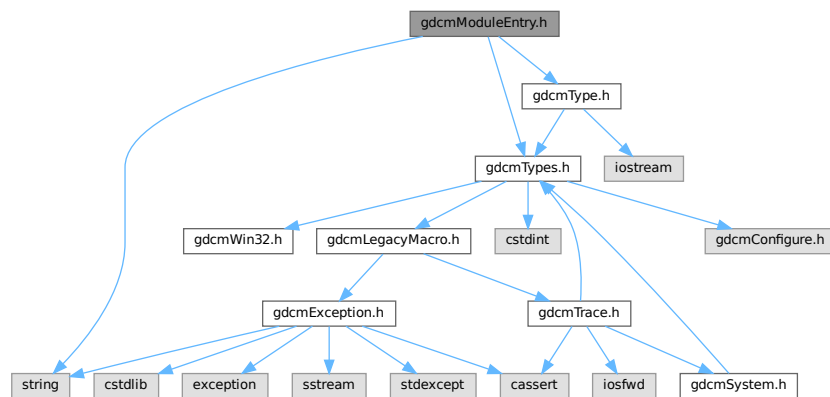
## 11.219 gdcmModuleEntry.h File Reference

```

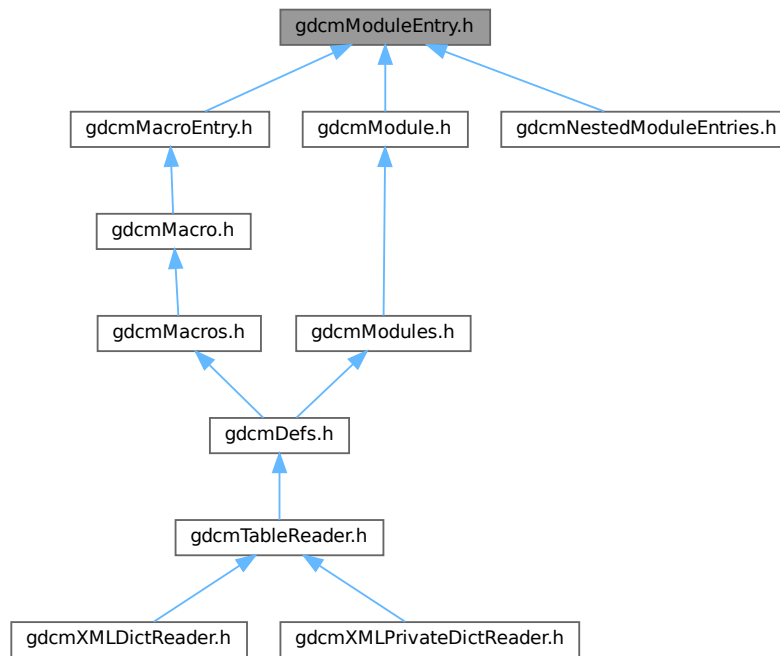
#include "gdcmTypes.h"
#include "gdcmType.h"
#include <string>

```

Include dependency graph for gdcmModuleEntry.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::ModuleEntry](#)  
Class for representing a *ModuleEntry*.

## Namespaces

- namespace [gdc](#)

## Typedefs

- typedef [ModuleEntry](#) [gdc::MacroEntry](#)

## Functions

- `std::ostream & gdc::operator<< (std::ostream &_os, const ModuleEntry &_val)`

## 11.220 gdcmModuleEntry.h

[Go to the documentation of this file.](#)

```

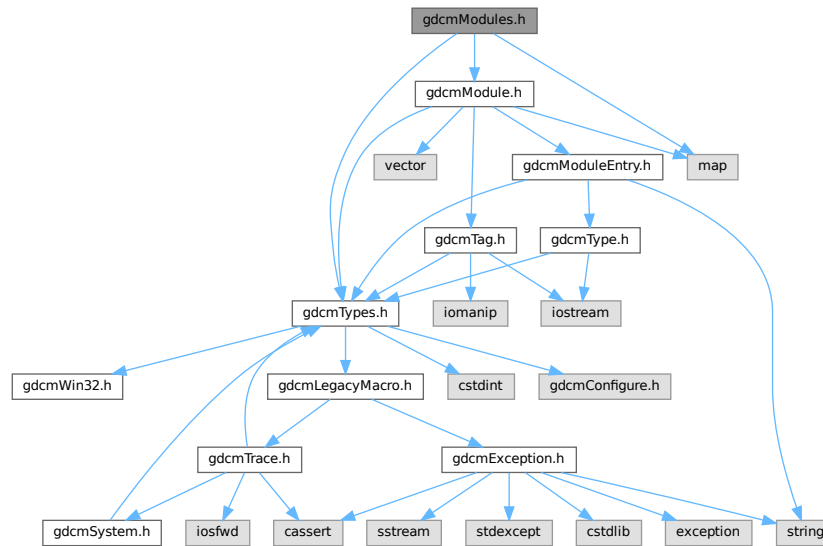
00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMMODULEENTRY_H
00015 #define GDCMMODULEENTRY_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmType.h"
00019
00020 #include <string>
00021
00022 namespace gdcm
00023 {
00024     class GDCM_EXPORT ModuleEntry
00025     {
00026     public:
00027         ModuleEntry(const char *name = "", const char *type = "3", const char *description =
00028             ""):Name(name)/*,Type(type)*/,DescriptionField(description) {
00029             DataElementType = Type::GetTypeType(type);
00030         }
00031         virtual ~ModuleEntry() = default; // important
00032         friend std::ostream& operator<(std::ostream& _os, const ModuleEntry &_val);
00033
00034         void SetName(const char *name) { Name = name; }
00035         const char *GetName() const { return Name.c_str(); }
00036
00037         void SetType(const Type &type) { DataElementType = type; }
00038         const Type &GetType() const { return DataElementType; }
00039
00040         /*
00041          * WARNING: 'Description' is currently a std::string, but it might change in the future
00042          * do not expect it to remain the same, and always use the ModuleEntry::Description typedef
00043          * instead.
00044          */
00045         typedef std::string Description;
00046         void SetDescription(const char *d) { DescriptionField = d; }
00047         const Description & GetDescription() const { return DescriptionField; }
00048
00049     protected:
00050         // PS 3.3 repeats the name of an attribute, but often contains typos
00051         // for now we will not use this info, but instead access the DataDict instead
00052         std::string Name;
00053
00054         // An attribute, encoded as a Data Element, may or may not be required in a
00055         // Data Set, depending on that Attribute's Data Element Type.
00056         Type DataElementType;
00057
00058         // TODO: for now contains the raw description (with enumerated values, defined terms...)
00059         Description DescriptionField;
00060     };
00061
00062 //-----
00063 inline std::ostream& operator<(std::ostream& _os, const ModuleEntry &_val)
00064 {
00065     _os << _val.Name << "\t" << _val.DataElementType << "\t" << _val.DescriptionField;
00066     return _os;
00067 }
00068
00069 typedef ModuleEntry MacroEntry;
00070
00071 } // end namespace gdcm
00072
00073 #endif //GDCMMODULEENTRY_H

```

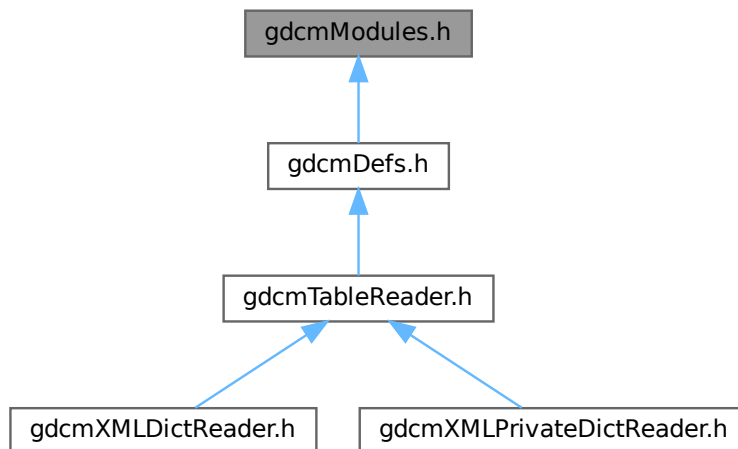
## 11.221 gdcmModules.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmModule.h"
#include <map>
```

Include dependency graph for gdcmModules.h:



This graph shows which files directly or indirectly include this file:



**Classes**

- class [gdcm::Modules](#)  
*Class for representing a [Modules](#).*

**Namespaces**

- namespace [gdcm](#)

**Functions**

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Modules &_val)`

**11.222 gdcmModules.h**

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 #ifndef GDCMMODULES_H
00015 #define GDCMMODULES_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmModule.h"
00019
00020 #include <map>
00021
00022 namespace gdcm
00023 {
00024     class GDCM_EXPORT Modules
00025     {
00026     public:
00027         typedef std::map<std::string, Module> ModuleMapType;
00028
00029         Modules() = default;
00030         friend std::ostream& operator<<(std::ostream& _os, const Modules &_val);
00031
00032         void Clear() { ModulesInternal.clear(); }
00033
00034         // A Module is inserted based on it's ref
00035         void AddModule(const char *ref, const Module & module )
00036         {
00037             assert( ref && *ref );
00038             assert( ModulesInternal.find( ref ) == ModulesInternal.end() );
00039             ModulesInternal.insert (
00040                 ModuleMapType::value_type(ref, module));
00041         }
00042         const Module &GetModule(const char *name) const
00043         {
00044             assert( name && *name );
00045             ModuleMapType::const_iterator it = ModulesInternal.find( name );
00046             assert( it != ModulesInternal.end() );
00047             assert( it->first == name );
00048             return it->second;
00049         }
00050     }
00051
00052 }

```

```

00056     bool IsEmpty() const { return ModulesInternal.empty(); }
00057
00058 private:
00059     ModuleMapType ModulesInternal;
00060 };
00061 //-----
00062 inline std::ostream& operator<(std::ostream& _os, const Modules &_val)
00063 {
00064     Modules::ModuleMapType::const_iterator it = _val.ModulesInternal.begin();
00065     for(; it != _val.ModulesInternal.end(); ++it)
00066     {
00067         const std::string &name = it->first;
00068         const Module &m = it->second;
00069         _os << name << " " << m << '\n';
00070     }
00071
00072     return _os;
00073 }
00074
00075
00076
00077 } // end namespace gdcmm
00078
00079 #endif //GDCMMODULES_H

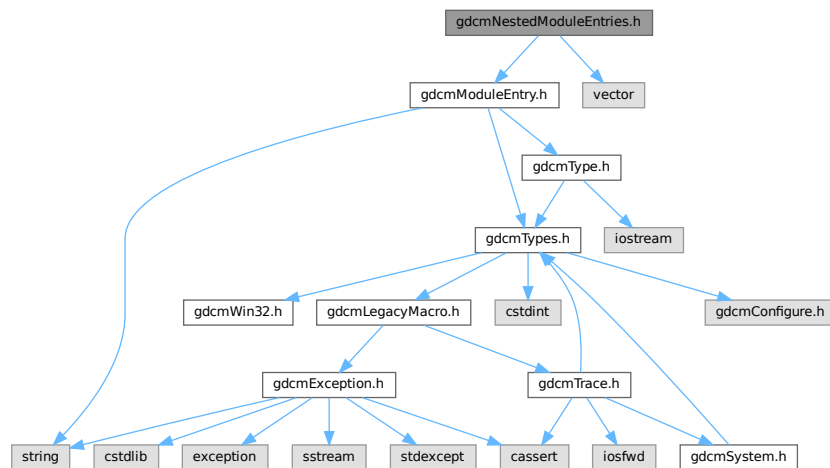
```

## 11.223 gdcmmNestedModuleEntries.h File Reference

```
#include "gdcmmModuleEntry.h"
```

```
#include <vector>
```

Include dependency graph for gdcmmNestedModuleEntries.h:



### Classes

- class [gdcmm::NestedModuleEntries](#)  
Class for representing a *NestedModuleEntries*.

### Namespaces

- namespace [gdcmm](#)



## Typedefs

- typedef [NestedModuleEntries](#) [gdcm::NestedMacroEntries](#)

## Functions

- [std::ostream & gdcm::operator<<](#) ([std::ostream &\\_os](#), const [NestedModuleEntries](#) &\_val)

## 11.224 gdcmNestedModuleEntries.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMNESTEDMODULEENTRIES_H
00015 #define GDCMNESTEDMODULEENTRIES_H
00016
00017 #include "gdcmModuleEntry.h"
00018 #include <vector>
00019
00020 namespace gdcm
00021 {
00022     class GDCM_EXPORT NestedModuleEntries : public ModuleEntry
00023     {
00024     public:
00025         NestedModuleEntries(const char *name = "", const char *type = "3", const char *description =
00026             ""):ModuleEntry(name,type,description) { }
00027         friend std::ostream& operator<<(std::ostream& _os, const NestedModuleEntries &_val);
00028
00029         typedef std::vector<ModuleEntry>::size_type SizeType;
00030         SizeType GetNumberOfModuleEntries() { return ModuleEntriesList.size(); }
00031
00032         const ModuleEntry &GetModuleEntry(SizeType idx) const { return ModuleEntriesList[idx]; }
00033         ModuleEntry &GetModuleEntry(SizeType idx) { return ModuleEntriesList[idx]; }
00034
00035         void AddModuleEntry(const ModuleEntry &me) { ModuleEntriesList.push_back( me ); }
00036
00037     private:
00038         std::vector<ModuleEntry> ModuleEntriesList;
00039     };
00040
00041 //-----
00042 inline std::ostream& operator<<(std::ostream& _os, const NestedModuleEntries &_val)
00043 {
00044     _os << "Nested:" << _val.Name << "\t" << _val.DataElementType << "\t" << _val.DescriptionField;
00045     return _os;
00046 }
00047
00048 typedef NestedModuleEntries NestedMacroEntries;
00049
00050 } // end namespace gdcm
00051
00052 #endif //GDCMNESTEDMODULEENTRIES_H

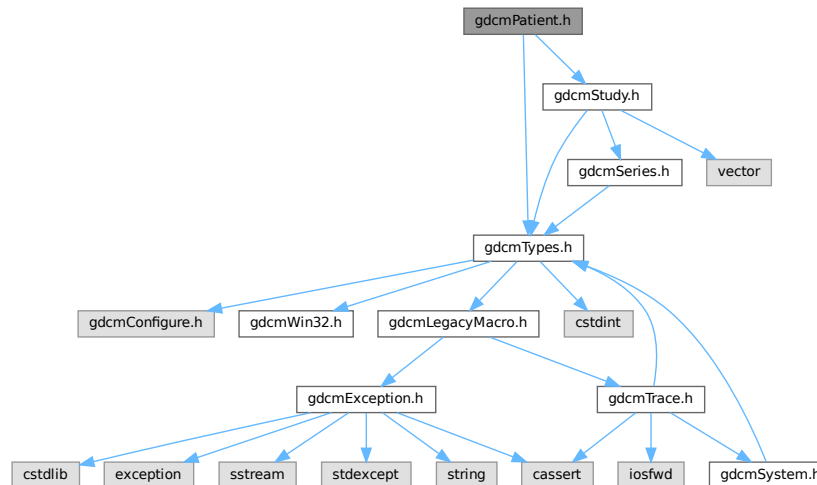
```

## 11.225 gdcmPatient.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmStudy.h"
```

Include dependency graph for gdcmPatient.h:



### Classes

- class [gdcm::Patient](#)

See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

### Namespaces

- namespace [gdcm](#)

## 11.226 gdcmPatient.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMPATIENT_H

```

```

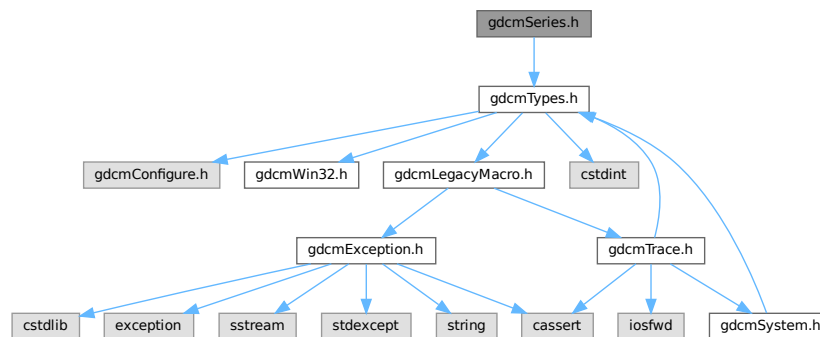
00015 #define GDCMPATIENT_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmStudy.h"
00019
00020 namespace gdcm
00021 {
00022     class GDCM_EXPORT Patient
00023     {
00024     public:
00025         Patient() = default;
00026
00027     private:
00028         std::vector<Study> StudyList;
00029     };
00030 } // end namespace gdcm
00031 #endif //GDCMPATIENT_H

```

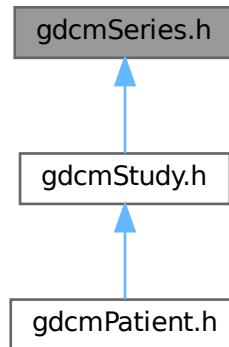
## 11.227 gdcmSeries.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmSeries.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcms::Series](#)  
*Series.*

## Namespaces

- namespace [gdcms](#)

## 11.228 gdcmsSeries.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMSERIES_H
00015 #define GDCMSERIES_H
00016
00017 #include "gdcmsTypes.h"
00018
00019 namespace gdcms
00020 {
00024   class GDCM_EXPORT Series
00025   {
00026   public:

```

```

00027     Series() = default;
00028 private:
00029     // Image, Waveform...
00030 };
00031
00032 } // end namespace gdcm
00033
00034 #endif //GDCMSERIES_H

```

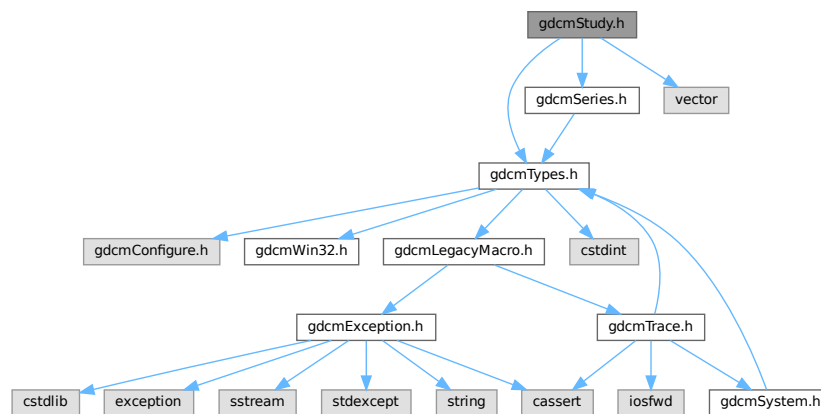
## 11.229 gdcmStudy.h File Reference

```

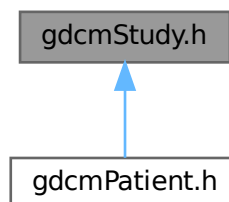
#include "gdcmTypes.h"
#include "gdcmSeries.h"
#include <vector>

```

Include dependency graph for gdcmStudy.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::Study`  
*Study.*

## Namespaces

- namespace `gdcm`

## 11.230 gdcmStudy.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMSTUDY_H
00015 #define GDCMSTUDY_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmSeries.h"
00019
00020 #include <vector>
00021
00022 namespace gdcm
00023 {
00024     class GDCM_EXPORT Study
00025     {
00026     public:
00027         Study() = default;
00028     private:
00029         std::vector<Series> SeriesList;
00030     };
00031
00032 } // end namespace gdcm
00033
00034 #endif //GDCMSTUDY_H

```

## 11.231 gdcmTable.h File Reference

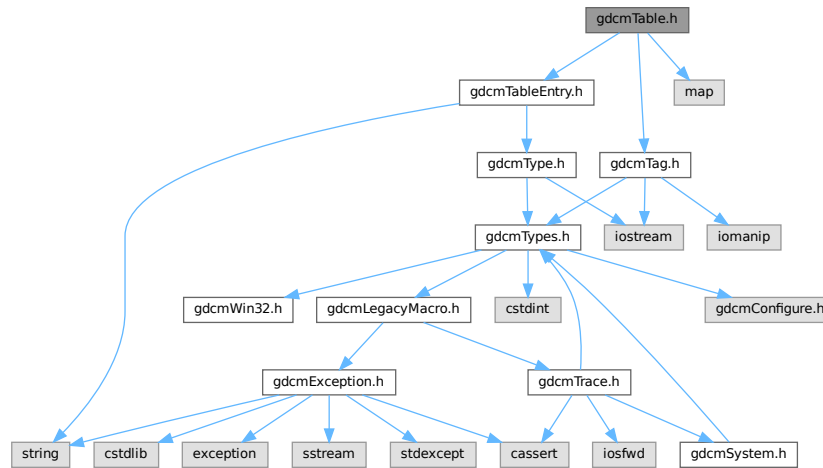
```

#include "gdcmTableEntry.h"
#include "gdcmTag.h"

```

```
#include <map>
```

Include dependency graph for gdcmTable.h:



## Classes

- class [gdcm::Table](#)  
*Table.*

## Namespaces

- namespace [gdcm](#)

## 11.232 gdcmTable.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMTABLE_H
00015  #define GDCMTABLE_H
00016
00017  #include "gdcmTableEntry.h"
00018  #include "gdcmTag.h"
00019
00020  #include <map>
00021

```

```

00022 namespace gdcM
00023 {
00024
00028 class Table
00029 {
00030 public:
00031     typedef std::map<Tag, TableEntry> MapTableEntry;
00032     Table() = default;
00033     ~Table() = default;
00034     Table &operator=(const Table &_val) = delete;
00035     Table(const Table&_val) = delete;
00036
00037     friend std::ostream& operator<<(std::ostream& _os, const Table &_val);
00038
00039     void InsertEntry(Tag const &tag, TableEntry const &te)
00040     {
00041 #ifndef NDEBUG
00042         MapTableEntry::size_type s = TableInternal.size();
00043 #endif
00044         TableInternal.insert(
00045             MapTableEntry::value_type(tag, te));
00046         assert( s < TableInternal.size() );
00047     }
00048
00049     const TableEntry &GetTableEntry(const Tag &tag) const
00050     {
00051         MapTableEntry::const_iterator it =
00052             TableInternal.find(tag);
00053         if (it == TableInternal.end())
00054         {
00055             assert( 0 && "Impossible" );
00056             return GetTableEntry(Tag(0,0));
00057         }
00058         return it->second;
00059     }
00060
00061     MapTableEntry TableInternal;
00062 };
00063
00064 } // end namespace gdcM
00065
00066 #endif //GDCMTABLE_H

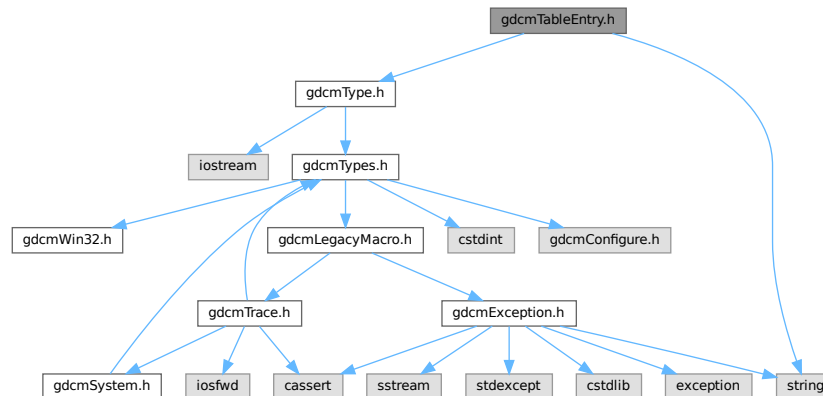
```

## 11.233 gdcMTableEntry.h File Reference

```
#include "gdcMType.h"
```

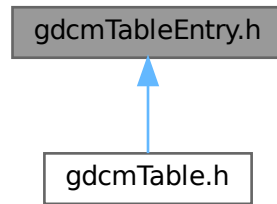
```
#include <string>
```

Include dependency graph for gdcMTableEntry.h:





This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::TableEntry`  
*TableEntry.*

## Namespaces

- namespace `gdcm`

## 11.234 gdcmTableEntry.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMTABLEENTRY_H
00015 #define GDCMTABLEENTRY_H
00016
00017 #include "gdcmType.h"
00018
00019 #include <string>
00020
00021 namespace gdcm
00022 {
00023
00027   class TableEntry
00028   {
00029   public:
00030     TableEntry(const char *attribute = nullptr,
00031               Type const &type = Type(), const char *des = nullptr ) :
00032       Attribute(attribute ? attribute : ""), TypeField(type), Description(des ? des : "") {}
00033     ~TableEntry() = default;
  
```



**Classes**

- class [gdcm::TableReader](#)  
Class for representing a *TableReader*.

**Namespaces**

- namespace [gdcm](#)

**11.236 gdcmTableReader.h**

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMTABLEREADER_H
00015  #define GDCMTABLEREADER_H
00016
00017  #include "gdcmTypes.h"
00018  #include "gdcmDefs.h"
00019  // #include "gdcmModule.h"
00020  // #include "gdcmIOD.h"
00021  // #include "gdcmIODs.h"
00022  // #include "gdcmModules.h"
00023
00024  #include <string>
00025  #include <vector>
00026  #include <map>
00027
00028  namespace gdcm
00029  {
00030  class GDCM_EXPORT TableReader
00031  {
00032  public:
00033      TableReader(Defs &defs):CurrentDefs(defs),ParsingModule(false),ParsingModuleEntry(false),
00034      ParsingModuleEntryDescription(false),
00035      ParsingMacro(false),
00036      ParsingMacroEntry(false),
00037      ParsingMacroEntryDescription(false),
00038      ParsingIOD(false),
00039      ParsingIODEntry(false),
00040      Description() {}
00041      virtual ~TableReader() = default;
00042
00043      // Set/Get filename
00044      void SetFilename(const char *filename) { Filename = filename; }
00045      const char *GetFilename() { return Filename.c_str(); }
00046
00047      int Read();
00048
00049  protected:
00050      // You need to override those function in your subclasses:
00051      virtual void StartElement(const char *name, const char **atts);
00052      virtual void EndElement(const char *name);
00053      virtual void CharacterDataHandler(const char *data, int length);
00054
00055      void HandleModuleEntry(const char **atts);
00056      void HandleModule(const char **atts);
00057      void HandleModuleEntryDescription(const char **atts);

```

```

00062 void HandleMacroEntry(const char **atts);
00063 void HandleMacro(const char **atts);
00064 void HandleMacroEntryDescription(const char **atts);
00065 void HandleModuleInclude(const char **atts);
00066 void HandleIODEntry(const char **atts);
00067 void HandleIOD(const char **atts);
00068
00069 //const Modules & GetModules() const { return CurrentModules; }
00070 //const Macros & GetMacros() const { return CurrentMacros; }
00071 //const IODs & GetIODs() const { return CurrentIODs; }
00072 const Defs & GetDefs() const { return CurrentDefs; }
00073
00074 private:
00075     std::string Filename;
00076     Defs &CurrentDefs;
00077     //Macros CurrentMacros;
00078     //Modules CurrentModules;
00079     //IODs CurrentIODs;
00080     Macro CurrentMacro;
00081     Module CurrentModule;
00082     IOD CurrentIOD;
00083     MacroEntry CurrentMacroEntry;
00084     ModuleEntry CurrentModuleEntry;
00085     IODEntry CurrentIODEntry;
00086     std::string CurrentModuleName;
00087     std::string CurrentModuleRef;
00088     std::string CurrentMacroRef;
00089     bool ParsingModule;
00090     bool ParsingModuleEntry;
00091     bool ParsingModuleEntryDescription;
00092     bool ParsingMacro;
00093     bool ParsingMacroEntry;
00094     bool ParsingMacroEntryDescription;
00095     bool ParsingIOD;
00096     bool ParsingIODEntry;
00097     Tag CurrentTag;
00098     std::string Description;
00099 };
00100
00101 } // end namespace gdcm
00102
00103 #endif //GDCMTABLEREADER_H

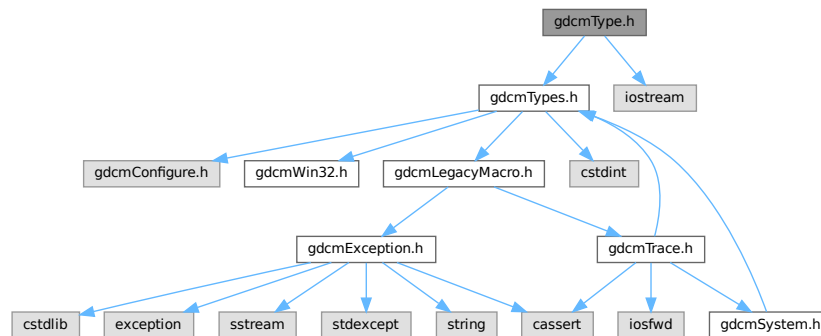
```

## 11.237 gdcmType.h File Reference

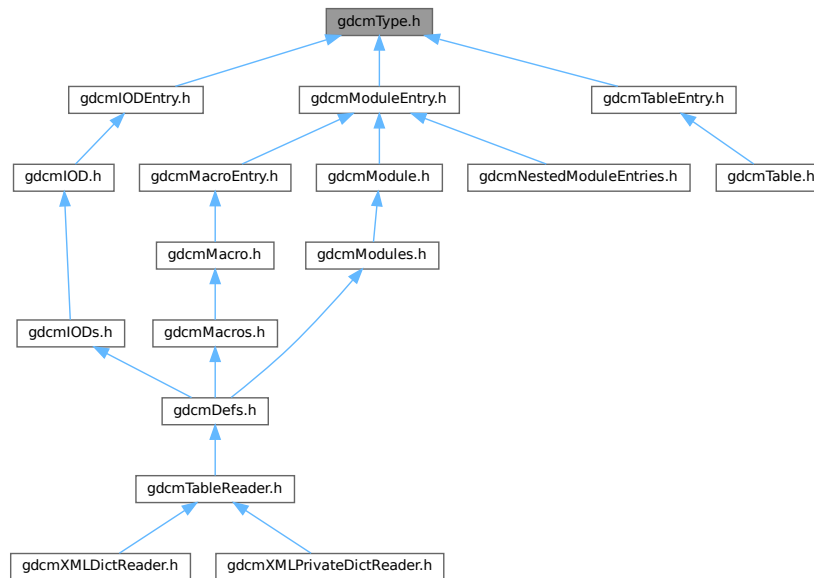
```
#include "gdcmTypes.h"
```

```
#include <iostream>
```

Include dependency graph for gdcmType.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::Type`  
*Type.*

## Namespaces

- namespace `gdcm`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Type &val)`

## 11.238 gdcmType.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
  
```

```

00010         the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011         PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014
00015 #ifndef GDCMTYPE_H
00016 #define GDCMTYPE_H
00017
00018 #include "gdcmTypes.h"
00019
00020 #include <iostream>
00021
00022 namespace gdcm
00023 {
00024
00041 class GDCM_EXPORT Type
00042 {
00043 public:
00044     typedef enum {
00045         T1 = 0,
00046         T1C,
00047         T2,
00048         T2C,
00049         T3,
00050         UNKNOWN
00051     } TypeType;
00052
00053     Type(TypeType type = UNKNOWN) : TypeField(type) { }
00054
00055     operator TypeType () const { return TypeField; }
00056     friend std::ostream &operator<<(std::ostream &os, const Type &vr);
00057
00058     static const char *GetTypeString(TypeType type);
00059     static TypeType GetTypeType(const char *type);
00060 private:
00061     TypeType TypeField;
00062 };
00063
00064 //-----
00065 inline std::ostream &operator<<(std::ostream &_os, const Type &val)
00066 {
00067     _os << Type::GetTypeString(val.TypeField);
00068     return _os;
00069 }
00070
00071 } // end namespace gdcm
00072
00073 #endif //GDCMTYPE_H

```

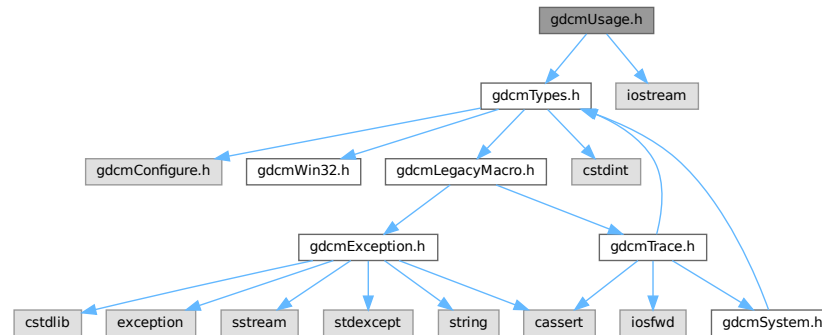
## 11.239 gdcmUsage.h File Reference

```

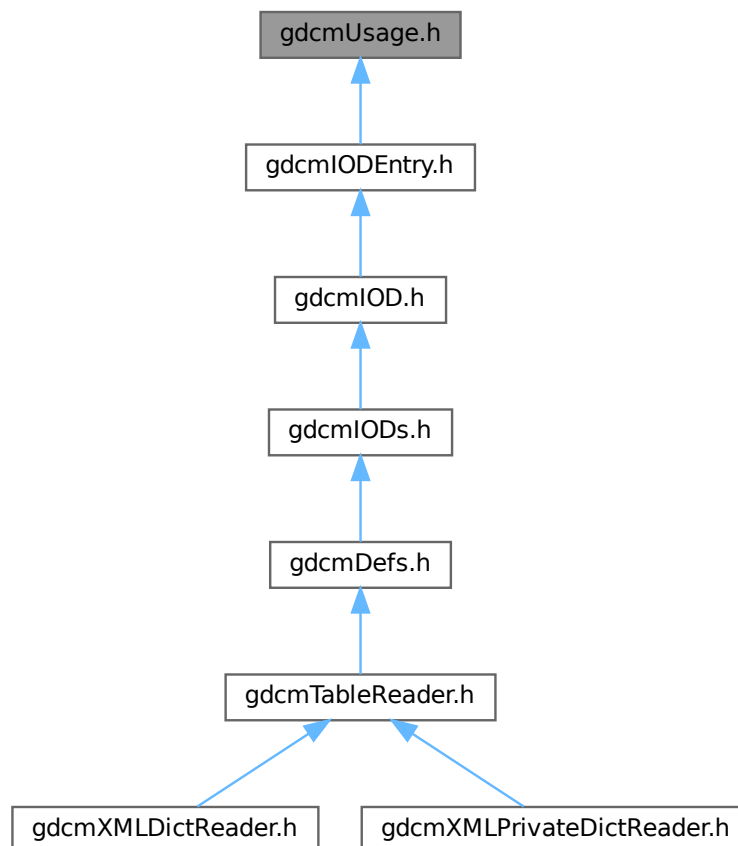
#include "gdcmTypes.h"
#include <iostream>

```

Include dependency graph for gdcmUsage.h:



This graph shows which files directly or indirectly include this file:



**Classes**

- class [gdcmm::Usage](#)  
*Usage.*

**Namespaces**

- namespace [gdcmm](#)

**Functions**

- `std::ostream & gdcmm::operator<< (std::ostream &_os, const Usage &val)`

**11.240 gdcmmUsage.h**

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMUSAGE_H
00015  #define GDCMUSAGE_H
00016
00017  #include "gdcmmTypes.h"
00018
00019  #include <iostream>
00020
00021  namespace gdcmm
00022  {
00023
00024  class GDCM_EXPORT Usage
00025  {
00026  public:
00027      typedef enum {
00028          Mandatory, // (see A.1.3.1) , abbreviated M
00029          Conditional, // (see A.1.3.2) , abbreviated C
00030          UserOption, // (see A.1.3.3) , abbreviated U
00031          Invalid
00032      } UsageType;
00033
00034      Usage(UsageType type = Invalid) : UsageField(type) { }
00035
00036      operator UsageType () const { return UsageField; }
00037      friend std::ostream &operator<<(std::ostream &os, const Usage &vr);
00038
00039      static const char *GetUsageString(UsageType type);
00040      static UsageType GetUsageType(const char *type);
00041
00042  private:
00043      UsageType UsageField;
00044  };
00045  //-----
00046  inline std::ostream &operator<<(std::ostream &_os, const Usage &val)
00047  {
00048      _os << Usage::GetUsageString(val.UsageField);
00049      return _os;
00050  }
00051
00052 } // end namespace gdcmm
00053
00054 #endif //GDCMUSAGE_H

```







## 11.244 gdcmXMLPrivateDictReader.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMXMLPRIVATEDICTREADER_H
00015  #define GDCMXMLPRIVATEDICTREADER_H
00016
00017  #include "gdcmTableReader.h"
00018  #include "gdcmDict.h"
00019  #include "gdcmDictEntry.h"
00020  #include "gdcmTag.h"
00021
00022  namespace gdcm
00023  {
00024
00025  class GDCM_EXPORT XMLPrivateDictReader : public TableReader
00026  {
00027  public:
00028      XMLPrivateDictReader();
00029      ~XMLPrivateDictReader() {}
00030
00031      void StartElement(const char *name, const char **atts);
00032      void EndElement(const char *name);
00033      void CharacterDataHandler(const char *data, int length);
00034
00035      const PrivateDict & GetPrivateDict() { return PDict; }
00036
00037  protected:
00038      void HandleEntry(const char **atts);
00039      void HandleDescription(const char **atts);
00040
00041  private:
00042      PrivateDict PDict;
00043      PrivateTag CurrentTag;
00044      DictEntry CurrentDE;
00045      bool ParsingDescription;
00046      std::string Description;
00047  };
00048
00049  } // end namespace gdcm
00050
00051  #endif //GDCMXMLPRIVATEDICTREADER_H

```

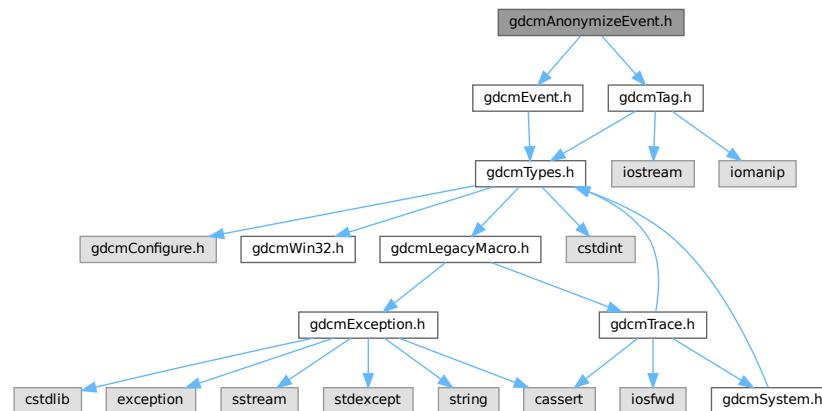
## 11.245 gdcmAnonymizeEvent.h File Reference

```

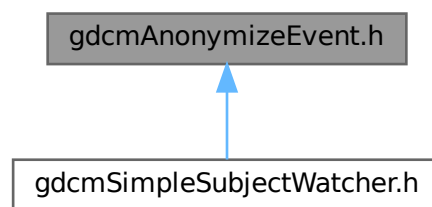
#include "gdcmEvent.h"
#include "gdcmTag.h"

```

Include dependency graph for `gdcmAnonymizeEvent.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::AnonymizeEvent`  
*AnonymizeEvent.*

## Namespaces

- namespace `gdcm`

## 11.246 gdcmAnonymizeEvent.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMANONYMIZEEVENT_H
00015  #define GDCMANONYMIZEEVENT_H
00016
00017  #include "gdcmEvent.h"
00018  #include "gdcmTag.h"
00019
00020  namespace gdcm
00021  {
00022
00029  class AnonymizeEvent : public AnyEvent
00030  {
00031  public:
00032      typedef AnonymizeEvent Self;
00033      typedef AnyEvent Superclass;
00034      AnonymizeEvent(Tag const &tag = 0):m_Tag(tag) {}
00035      ~AnonymizeEvent() override = default;
00036      AnonymizeEvent(const Self&s) : AnyEvent(s){}
00037      void operator=(const Self&) = delete;
00038
00039      const char * GetEventName() const override { return "AnonymizeEvent"; }
00040      bool CheckEvent(const ::gdcm::Event* e) const override
00041      { return (dynamic_cast<const Self*>(e) == nullptr ? false : true) ; }
00042      ::gdcm::Event* MakeObject() const override
00043      { return new Self; }
00044
00045      void SetTag(const Tag& t) { m_Tag = t; }
00046      Tag const & GetTag() const { return m_Tag; }
00047  private:
00048      Tag m_Tag;
00049  };
00050
00051
00052  } // end namespace gdcm
00053
00054  #endif //GDCMANONYMIZEEVENT_H

```

## 11.247 gdcmAnonymizer.h File Reference

```

#include "gdcmFile.h"
#include "gdcmSubject.h"
#include "gdcmEvent.h"
#include "gdcmSmartPointer.h"
#include <map>

```



```

00078 {
00079 public:
00080     Anonymizer():F(new File),CMS(nullptr) {}
00081     ~Anonymizer() override;
00082
00084     bool Empty( Tag const &t );
00085
00090     bool Empty( PrivateTag const &pt );
00091
00093     bool Clear( Tag const &t );
00094     bool Clear( PrivateTag const &pt );
00095
00097     bool Remove( Tag const &t );
00098
00104     bool Remove( PrivateTag const &pt );
00105
00108     bool Replace( Tag const &t, const char *value );
00109     bool Replace( PrivateTag const &t, const char *value );
00110
00113     bool Replace( Tag const &t, const char *value, VL const &vl );
00114     bool Replace( PrivateTag const &t, const char *value, VL const &vl );
00115
00117     bool RemovePrivateTags();
00118
00120     bool RemoveGroupLength();
00121
00123     bool RemoveRetired();
00124
00126     void SetFile(const File& f) { F = f; }
00127     //const File &GetFile() const { return *F; }
00128     File &GetFile() { return *F; }
00129
00134     bool BasicApplicationLevelConfidentialityProfile(bool deidentify = true);
00135
00137     void SetCryptographicMessageSyntax( CryptographicMessageSyntax *cms );
00138     const CryptographicMessageSyntax *GetCryptographicMessageSyntax() const;
00139
00141     static SmartPointer<Anonymizer> New() { return new Anonymizer; }
00142
00144     static std::vector<Tag> GetBasicApplicationLevelConfidentialityProfileAttributes();
00145
00148     static void ClearInternalUIDs();
00149
00150 protected:
00151     // Internal function used to either empty a tag or set it's value to a dummy value (Type 1 vs Type 2)
00152     bool BALCPProtect(DataSet &ds, Tag const &tag, const IOD &ioid);
00153     bool CanEmptyTag(Tag const &tag, const IOD &ioid) const;
00154     void RecurseDataSet( DataSet &ds );
00155
00156 private:
00157     bool BasicApplicationLevelConfidentialityProfile1();
00158     bool BasicApplicationLevelConfidentialityProfile2();
00159     bool CheckIfSequenceContainsAttributeToAnonymize(File const &file, SequenceOfItems* sqi) const;
00160
00161 private:
00162     // I would prefer to have a smart pointer to DataSet but DataSet does not derive from Object...
00163     SmartPointer<File> F;
00164     CryptographicMessageSyntax *CMS;
00165
00166     typedef std::pair< Tag, std::string > TagValueKey;
00167     typedef std::map< TagValueKey, std::string > DummyMapNonUIDTags;
00168     typedef std::map< std::string, std::string > DummyMapUIDTags;
00169     static DummyMapNonUIDTags dummyMapNonUIDTags;
00170     static DummyMapUIDTags dummyMapUIDTags;
00171 };
00172
00179 } // end namespace gdcm
00180
00181 #endif //GDCMANONYMIZER_H

```

## 11.249 gdcmApplicationEntity.h File Reference

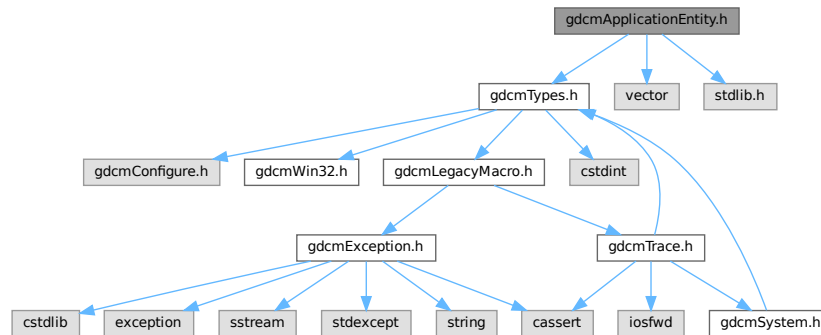
```

#include "gdcmTypes.h"
#include <vector>

```

```
#include <stdlib.h>
```

Include dependency graph for `gdcmApplicationEntity.h`:



## Classes

- class `gdcm::ApplicationEntity`  
*ApplicationEntity.*

## Namespaces

- namespace `gdcm`

## 11.250 gdcmApplicationEntity.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMAPPLICATIONENTITY_H
00015 #define GDCMAPPLICATIONENTITY_H
00016
00017 #include "gdcmTypes.h"
00018 #include <vector>
00019 #include <stdlib.h> // abort
00020
00021 namespace gdcm
00022 {
00023
00035   class GDCM_EXPORT ApplicationEntity
00036   {
00037   public:
00038     static const unsigned int MaxNumberOfComponents = 1;

```



```

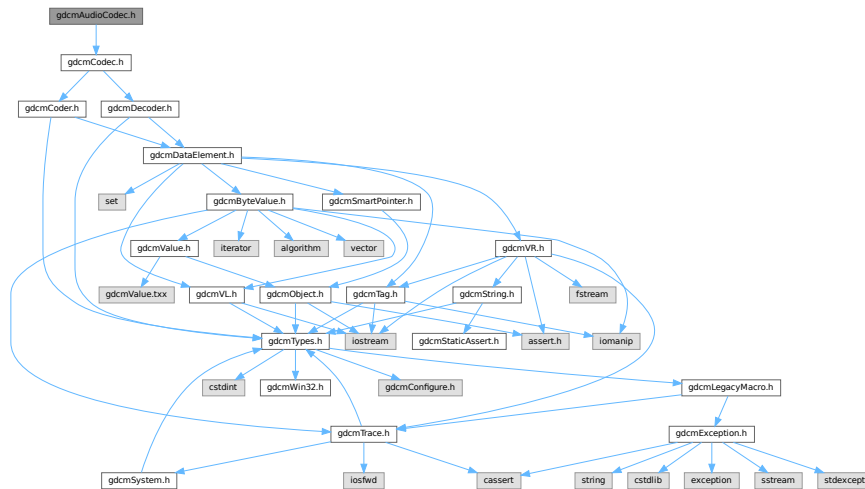
00039  static const unsigned int MaxLength = 16;
00040  std::string Internal;
00041  static const char Separator = ' ';
00042  static const char Padding = ' ';
00043  //static const char Excluded[5] = { '\\', /* 5CH */, '\n', /* LF */, '\f', /* FF */, '\r', /* CR */, 0x1b
/* ESC */};
00044
00045  bool IsValid() const {
00046      return true;
00047  }
00048  void Squeeze() {
00049      // trim leading and trailing white spaces
00050  }
00051  void SetBlob(const std::vector<char>& v) {
00052      (void)v;
00053      assert(0); //TODO
00054  }
00055  void Print(std::ostream &os) const {
00056      (void)os;
00057      assert(0); //TODO
00058  }
00059 };
00060
00061 } // end namespace gdcm
00062
00063 #endif //GDCMAPPLICATIONENTITY_H

```

## 11.251 gdcmAudioCodec.h File Reference

```
#include "gdcmCodec.h"
```

Include dependency graph for gdcmAudioCodec.h:



### Classes

- class [gdcm::AudioCodec](#)  
*AudioCodec.*

### Namespaces

- namespace [gdcm](#)

## 11.252 gdcmAudioCodec.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMAUDIOCODEC_H
00015 #define GDCMAUDIOCODEC_H
00016
00017 #include "gdcmCodec.h"
00018
00019 namespace gdcm
00020 {
00021
00022   class GDCM_EXPORT AudioCodec : public Codec
00023   {
00024   public:
00025     AudioCodec();
00026     ~AudioCodec() override;
00027     bool CanCode(TransferSyntax const &) const override { return false; }
00028     bool CanDecode(TransferSyntax const &) const override { return false; }
00029     bool Decode(DataElement const &is, DataElement &os) override;
00030   };
00031
00032 } // end namespace gdcm
00033
00034 #endif //GDCMAUDIOCODEC_H

```

## 11.253 gdcmBitmap.h File Reference

```

#include "gdcmObject.h"
#include "gdcmCurve.h"
#include "gdcmDataElement.h"
#include "gdcmLookupTable.h"
#include "gdcmOverlay.h"
#include "gdcmPhotometricInterpretation.h"
#include "gdcmPixelFormat.h"
#include "gdcmSmartPointer.h"
#include "gdcmTransferSyntax.h"
#include <vector>

```



```

00014 #ifndef GDCMBITMAP_H
00015 #define GDCMBITMAP_H
00016
00017 #include "gdcmObject.h"
00018 #include "gdcmCurve.h"
00019 #include "gdcmDataElement.h"
00020 // #include "gdcmIconImage.h"
00021 #include "gdcmLookupTable.h"
00022 #include "gdcmOverlay.h"
00023 #include "gdcmPhotometricInterpretation.h"
00024 #include "gdcmPixelFormat.h"
00025 #include "gdcmSmartPointer.h"
00026 #include "gdcmTransferSyntax.h"
00027
00028 #include <vector>
00029
00030 namespace gdcm
00031 {
00032
00033     class GDCM_EXPORT Bitmap : public Object
00034     {
00035     public:
00036         Bitmap();
00037         ~Bitmap() override;
00038         void Print(std::ostream &) const override;
00039
00040         virtual bool AreOverlaysInPixelData() const { return false; }
00041         virtual bool UnusedBitsPresentInPixelData() const { return false; }
00042
00043         unsigned int GetNumberOfDimensions() const;
00044         void SetNumberOfDimensions(unsigned int dim);
00045
00046         unsigned int GetPlanarConfiguration() const;
00047         void SetPlanarConfiguration(unsigned int pc);
00048
00049         bool GetNeedByteSwap() const
00050         {
00051             return NeedByteSwap;
00052         }
00053         void SetNeedByteSwap(bool b)
00054         {
00055             NeedByteSwap = b;
00056         }
00057
00058         void SetTransferSyntax(TransferSyntax const &ts) {
00059             TS = ts;
00060         }
00061         const TransferSyntax &GetTransferSyntax() const {
00062             return TS;
00063         }
00064         bool IsTransferSyntaxCompatible( TransferSyntax const & ts ) const;
00065         void SetDataElement(DataElement const &de) {
00066             PixelData = de;
00067         }
00068         const DataElement& GetDataElement() const { return PixelData; }
00069         DataElement& GetDataElement() { return PixelData; }
00070
00071         void SetLUT(LookupTable const &lut)
00072         {
00073             LUT = SmartPointer<LookupTable>( const_cast<LookupTable*>(&lut) );
00074         }
00075         const LookupTable &GetLUT() const
00076         {
00077             return *LUT;
00078         }
00079         LookupTable &GetLUT()
00080         {
00081             return *LUT;
00082         }
00083
00084         const unsigned int *GetDimensions() const;
00085         unsigned int GetDimension(unsigned int idx) const;
00086
00087         void SetColumns(unsigned int col) { SetDimension(0,col); }
00088         unsigned int GetColumns() const { return GetDimension(0); }
00089         void SetRows(unsigned int rows) { SetDimension(1,rows); }
00090         unsigned int GetRows() const { return GetDimension(1); }
00091         void SetDimensions(const unsigned int dims[3]);
00092         void SetDimension(unsigned int idx, unsigned int dim);
00093         const PixelFormat &GetPixelFormat() const

```

```

00108     {
00109         return PF;
00110     }
00111 PixelFormat &GetPixelFormat()
00112     {
00113         return PF;
00114     }
00115 void SetPixelFormat(PixelFormat const &pf)
00116     {
00117         PF = pf;
00118         PF.Validate();
00119     }
00120
00121 const PhotometricInterpretation &GetPhotometricInterpretation() const;
00122 void SetPhotometricInterpretation(PhotometricInterpretation const &pi);
00123
00124 bool IsEmpty() const { return Dimensions.empty(); }
00125 void Clear();
00126
00127 unsigned long GetBufferLength() const;
00128
00129 bool GetBuffer(char *buffer) const;
00130
00131 bool IsLossy() const;
00132
00133 void SetLossyFlag(bool f) { LossyFlag = f; }
00134
00135 protected:
00136 bool TryRAWCodec(char *buffer, bool &lossyflag) const;
00137 bool TryJPEGCodec(char *buffer, bool &lossyflag) const;
00138 bool TryPVRGCodec(char *buffer, bool &lossyflag) const;
00139 bool TryKAKADUCoDec(char *buffer, bool &lossyflag) const;
00140 bool TryJPEGLSCodec(char *buffer, bool &lossyflag) const;
00141 bool TryJPEG2000Codec(char *buffer, bool &lossyflag) const;
00142 bool TryRLECodec(char *buffer, bool &lossyflag) const;
00143
00144 bool TryJPEGCodec2(std::ostream &os) const;
00145 bool TryJPEG2000Codec2(std::ostream &os) const;
00146
00147 bool GetBuffer2(std::ostream &os) const;
00148
00149 friend class PixmapReader;
00150 friend class ImageChangeTransferSyntax;
00151 // Function to compute the lossy flag based only on the image buffer.
00152 // Watch out that image can be lossy but in implicit little endian format...
00153 bool ComputeLossyFlag();
00154
00155 //private:
00156 protected:
00157 unsigned int PlanarConfiguration;
00158 unsigned int NumberOfDimensions;
00159 TransferSyntax TS;
00160 PixelFormat PF; // SamplesPerPixel, BitsAllocated, BitsStored, HighBit, PixelRepresentation
00161 PhotometricInterpretation PI;
00162 // Mind dump: unsigned int is required here, since we are reading (0028,0008) Number Of Frames
00163 // which is VR::IS, so I cannot simply assumed that unsigned short is enough... :(
00164 std::vector<unsigned int> Dimensions; // Col/Row
00165 DataElement PixelData; // copied from 7fe0,0010
00166
00167 typedef SmartPointer<LookupTable> LUTPtr;
00168 LUTPtr LUT;
00169 // I believe the following 3 ivars can be derived from TS ...
00170 bool NeedByteSwap; // FIXME: remove me
00171 bool LossyFlag;
00172
00173 private:
00174 bool GetBufferInternal(char *buffer, bool &lossyflag) const;
00175 };
00176
00177 } // end namespace gdcm
00178
00179 #endif //GDCMBITMAP_H

```



## 11.257 qdcmCleaner.h File Reference

## Classes

- class `gdcm::Cleaner`  
*Cleaner.*

## Namespaces

- namespace `gdcm`

## 11.258 gdcmCleaner.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMCLEANER_H
00015  #define GDCMCLEANER_H
00016
00017  #include "gdcmDPath.h"
00018  #include "gdcmFile.h"
00019  #include "gdcmSmartPointer.h"
00020  #include "gdcmSubject.h"
00021
00022  namespace gdcm {
00030  class GDCM_EXPORT Cleaner : public Subject {
00031  public:
00032      Cleaner();
00033      ~Cleaner() override;
00034
00036      bool Empty(Tag const &t);
00037      bool Empty(PrivateTag const &pt);
00038      bool Empty(DPath const &dpath);
00039      bool Empty(VR const &vr);
00040
00041      bool Remove(Tag const &t);
00042      bool Remove(PrivateTag const &pt);
00043      bool Remove(DPath const &dpath);
00044      bool Remove(VR const &vr);
00045
00047      bool Scrub(Tag const &t);
00048      bool Scrub(PrivateTag const &pt);
00049      bool Scrub(DPath const &dpath);
00050      bool Scrub(VR const &vr);
00051
00052      bool Preserve(DPath const &dpath);
00053
00056      void RemoveAllMissingPrivateCreator(bool remove);
00057
00060      bool RemoveMissingPrivateCreator(Tag const &t);
00061
00063      void RemoveAllGroupLength(bool remove);
00064
00066      void RemoveAllIllegal(bool remove);
00067
00069      void EmptyWhenScrubFails(bool empty);
00070
00072      bool Clean();
00073
00075      void SetFile(const File &f) { F = f; }

```



```

00076 // const File &GetFile() const { return *F; }
00077 File &GetFile() { return *F; }
00078
00080 static SmartPointer<Cleaner> New() { return new Cleaner; }
00081
00082 private:
00083 // I would prefer to have a smart pointer to DataSet but DataSet does not
00084 // derive from Object...
00085 SmartPointer<File> F;
00086 struct impl;
00087 // PIMPL idiom
00088 impl *pimpl;
00089 };
00090
00091 } // end namespace gdcM
00092
00093 #endif // GDCMCLEANER_H

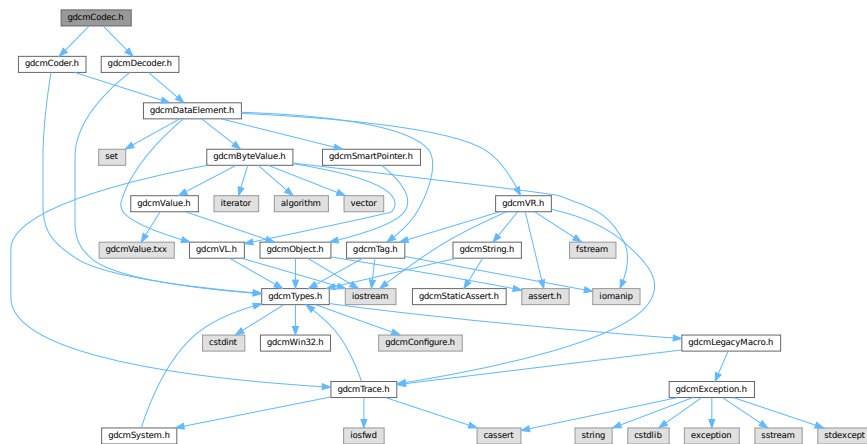
```

## 11.259 gdcMCodec.h File Reference

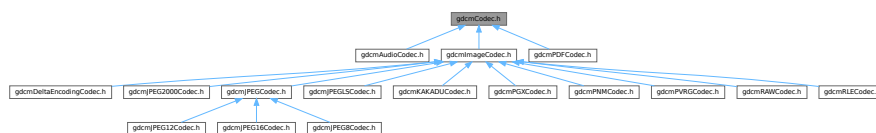
```
#include "gdcMCoder.h"
```

```
#include "gdcMDecoder.h"
```

Include dependency graph for gdcMCodec.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcM::Codec](#)  
*Codec* class.

## Namespaces

- namespace [gdcm](#)

## 11.260 gdcmCodec.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMCODEC_H
00015  #define GDCMCODEC_H
00016
00017  #include "gdcmCoder.h"
00018  #include "gdcmDecoder.h"
00019
00020  namespace gdcm
00021  {
00022
00026  class GDCM_EXPORT Codec : public Coder, public Decoder
00027  {
00028  };
00029
00030  } // end namespace gdcm
00031
00032  #endif //GDCMCODEC_H

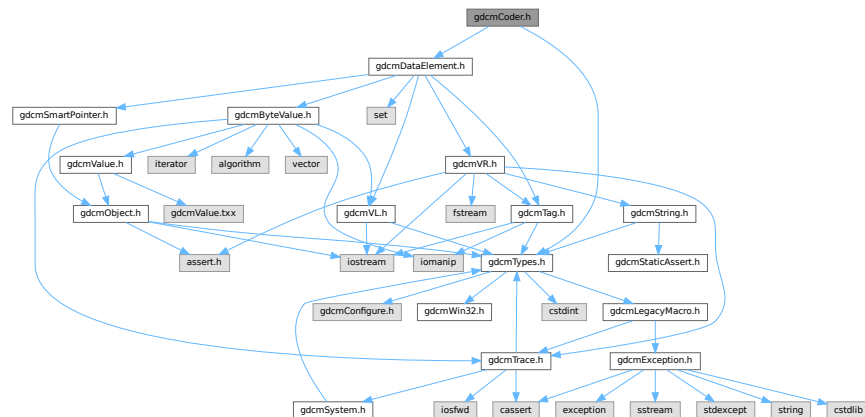
```

## 11.261 gdcmCoder.h File Reference

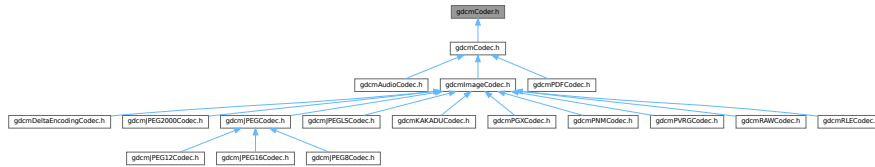
```
#include "gdcmTypes.h"
```

```
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmCoder.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::Coder`  
*Coder.*

## Namespaces

- namespace `gdcm`

## 11.262 gdcmCoder.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMCODER_H
00015  #define GDCMCODER_H
00016
00017  #include "gdcmTypes.h"
00018  #include "gdcmDataElement.h" // FIXME
00019
00020  namespace gdcm
00021  {
00022
00023  class TransferSyntax;
00024  class DataElement;
00028  class GDCM_EXPORT Coder
00029  {
00030  public:
00031    virtual ~Coder() = default;
00032
00033    virtual bool CanCode(TransferSyntax const &) const = 0;
00034
00035    // Note: in / out are reserved keyword in C#. Change to in_ / out_
00036
00037    virtual bool Code(DataElement const &in_, DataElement &out_) { (void)in_; (void)out_; return false; }
00039  protected:
00040    virtual bool InternalCode(const char *bv, unsigned long len, std::ostream &os) {
00041      (void)bv; (void)os; (void)len; return false; }
00042  };
00043
00044  } // end namespace gdcm
00045
00046  #endif //GDCMCODER_H

```

## 11.263 gdcmConstCharWrapper.h File Reference

### Classes

- class [gdcm::ConstCharWrapper](#)

*Do not use me.*

### Namespaces

- namespace [gdcm](#)

## 11.264 gdcmConstCharWrapper.h

[Go to the documentation of this file.](#)

```

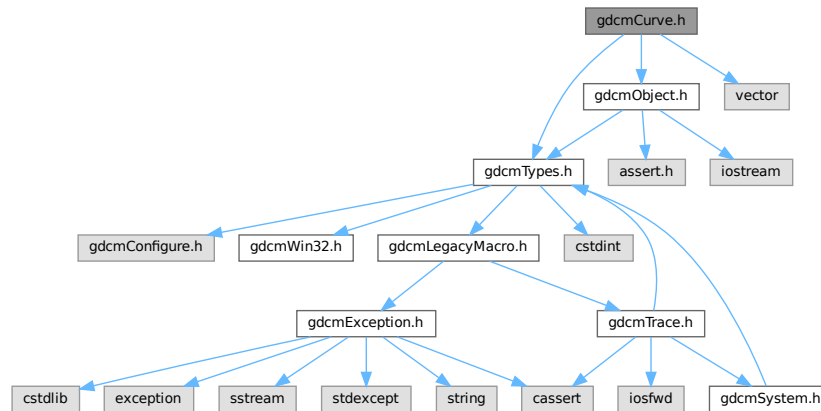
00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMCONSTCHARWRAPPER_H
00015 #define GDCMCONSTCHARWRAPPER_H
00016
00017 namespace gdcm
00018 {
00019
00020 #error
00021
00022 /*
00023  * This class is a pure hack. Its only goal is to work around a bad bug in :
00024  * $ swig -version
00025  * SWIG Version 1.3.31
00026  *
00027  * See
00028  * -
00029  * http://sourceforge.net/mailarchive/forum.php?thread_name=bf0c3b3f0802290552y5163989t76572b80a044ce28%40mail.gmail.com&forum=
00030  * As a side note there is also a problem with const reference to enum type:
00031  * -
00032  * http://sourceforge.net/mailarchive/forum.php?thread_name=bf0c3b3f0802290552y5163989t76572b80a044ce28%40mail.gmail.com&forum=
00033  * And to keep track of an issue with swig here is the last one:
00034  *
00035  * -
00036  * http://sourceforge.net/mailarchive/forum.php?thread_name=bf0c3b3f0802290552y5163989t76572b80a044ce28%40mail.gmail.com&forum=
00037  */
00038
00042 class ConstCharWrapper
00043 {
00044 public:
00045     ConstCharWrapper(const char *i=0):Internal(i) {}
00046     operator const char * () const { return Internal; }
00047 private:
00048     const char *Internal;
00049 };
00050
00051 } // end namespace gdcm
00052
00053 #endif //GDCMCONSTCHARWRAPPER_H

```

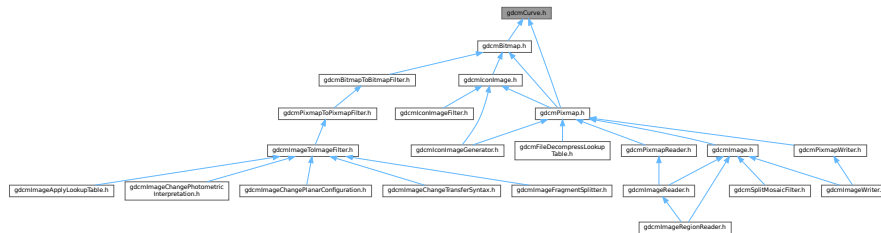
## 11.265 gdcmCurve.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmObject.h"
#include <vector>
```

Include dependency graph for gdcmCurve.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [gdcm::Curve](#)  
*Curve class to handle element 50xx,3000 Curve Data.*

### Namespaces

- namespace [gdcm](#)

## 11.266 gdcmCurve.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMCURVE_H
00015 #define GDCMCURVE_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmObject.h"
00019
00020 #include <vector>
00021
00022 namespace gdcm
00023 {
00024
00025   class CurveInternal;
00026   class ByteValue;
00027   class DataSet;
00028   class DataElement;
00040   class GDCM_EXPORT Curve : public Object
00041   {
00042   public:
00043     Curve();
00044     ~Curve() override;
00045     void Print(std::ostream &) const override;
00046
00047     void GetAsPoints(float *array) const;
00048
00049     static unsigned int GetNumberOfCurves(DataSet const & ds);
00050
00051     // Update curve data from dataelement de:
00052     void Update(const DataElement & de);
00053
00054     void SetGroup(unsigned short group);
00055     unsigned short GetGroup() const;
00056     void SetDimensions(unsigned short dimensions);
00057     unsigned short GetDimensions() const;
00058     void SetNumberOfPoints(unsigned short numberofpoints);
00059     unsigned short GetNumberOfPoints() const;
00060     void SetTypeOfData(const char *typeofdata);
00061     const char *GetTypeOfData() const;
00062     // See PS 3.3 - 2004 - C.10.2.1.1 Type of data
00063     const char *GetTypeOfDataDescription() const;
00064     void SetCurveDescription(const char *curvedescription);
00065     void SetDataValueRepresentation(unsigned short datavaluerepresentation);
00066     unsigned short GetDataValueRepresentation() const;
00067     void SetCurveDataDescriptor(const uint16_t * values, size_t num);
00068     std::vector<unsigned short> const &GetCurveDataDescriptor() const;
00069     void SetCoordinateStartValue( unsigned short v );
00070     void SetCoordinateStepValue( unsigned short v );
00071
00072     void SetCurve(const char *array, unsigned int length);
00073
00074     bool IsEmpty() const;
00075
00076     void Decode(std::istream &is, std::ostream &os);
00077
00078     Curve(Curve const &ov);
00079   private:
00080     double ComputeValueFromStartAndStep(unsigned int idx) const;
00081     CurveInternal *Internal;
00082   };
00083
00084 } // end namespace gdcm
00085
00086 #endif //GDCMCURVE_H

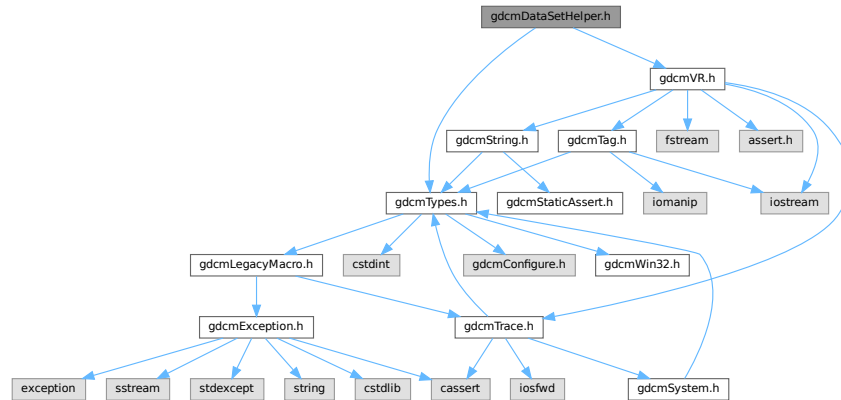
```

## 11.267 gdcmDataSetHelper.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmVR.h"
```

Include dependency graph for gdcmDataSetHelper.h:



### Classes

- class [gdcm::DataSetHelper](#)  
*DataSetHelper (internal class, not intended for user level)*

### Namespaces

- namespace [gdcm](#)

## 11.268 gdcmDataSetHelper.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012  =====*/
00013
00014  #ifndef GDCMDATASETHelper_H
00015  #define GDCMDATASETHelper_H
00016
00017  #include "gdcmTypes.h"
00018  #include "gdcmVR.h"
00019

```

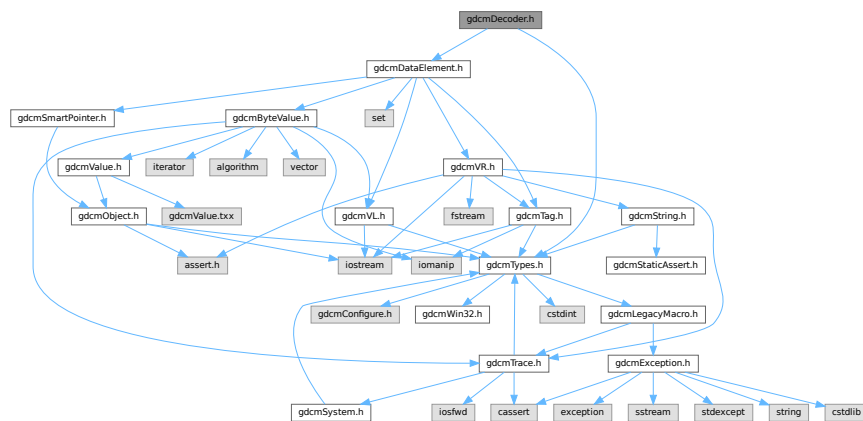
```
00020 namespace gdcm
00021 {
00022     class DataSet;
00023     class File;
00024     class Tag;
00025     class SequenceOfItems;
00026
00030     class GDCM_EXPORT DataSetHelper
00031     {
00032     public:
00035         static VR ComputeVR(File const & file, DataSet const &ds, const Tag& tag);
00036
00037         //static SequenceOfItems* ComputeSQFromByteValue(File const & file, DataSet const &ds, const Tag &tag);
00038
00039     protected:
00040     };
00041
00042 } // end namespace gdcm
00043
00044 #endif // GDCMDATASETHelper_H
```

## 11.269 gdcmDecoder.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmDecoder.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::Decoder`  
*Decoder*.



## Namespaces

- namespace `gdcm`

## 11.270 gdcmDecoder.h

[Go to the documentation of this file.](#)

```
00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014
00015 #ifndef GDCMDECODER_H
00016 #define GDCMDECODER_H
00017
00018 #include "gdcmTypes.h"
00019 #include "gdcmDataElement.h" // FIXME
00020
00021 namespace gdcm
00022 {
00023
00024   class TransferSyntax;
00025   class DataElement;
00029   class GDCM_EXPORT Decoder
00030   {
00031   public:
00032     virtual ~Decoder() = default;
00033
00035     virtual bool CanDecode(TransferSyntax const &) const = 0;
00036
00038     virtual bool Decode(DataElement const &, DataElement &) { return false; }
00039   protected:
00040     virtual bool DecodeByStreams(std::istream &, std::ostream &) { return false; }
00041   };
00042
00043 } // end namespace gdcm
00044
00045 #endif //GDCMDECODER_H
```



```

00021 {
00022
00027 class DeltaEncodingCodec : public ImageCodec
00028 {
00029 public:
00030     DeltaEncodingCodec();
00031     ~DeltaEncodingCodec();
00032     bool CanDecode(TransferSyntax const &ts);
00033     bool Decode(DataElement const &is, DataElement &os);
00034 protected:
00035     bool Decode(std::istream &is, std::ostream &os);
00036 };
00037
00038 } // end namespace gdcm
00039
00040 #endif //GDCMDELTAENCODINGCODEC_H

```

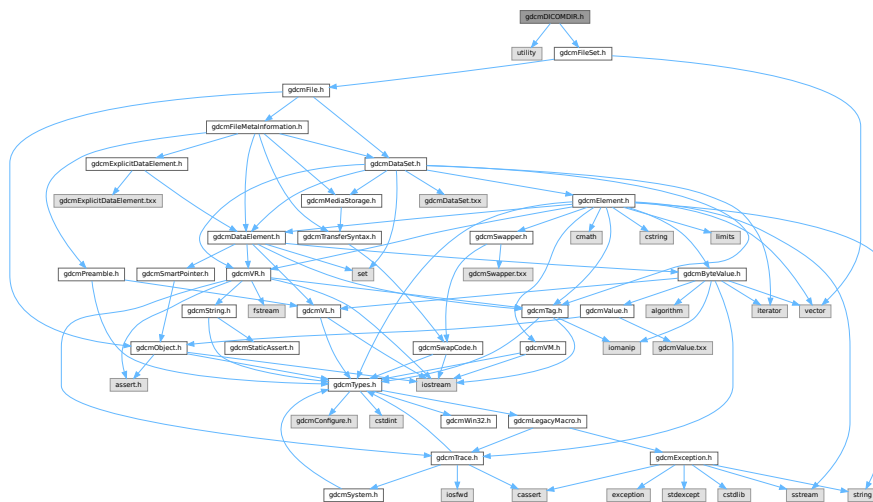
## 11.273 gdcmDICOMDIR.h File Reference

```

#include <utility>
#include "gdcmFileSet.h"

```

Include dependency graph for gdcmDICOMDIR.h:



### Classes

- class [gdcm::DICOMDIR](#)  
*DICOMDIR* class.

### Namespaces

- namespace [gdcm](#)

## 11.274 gdcmDICOMDIR.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMDICOMDIR_H
00015 #define GDCMDICOMDIR_H
00016
00017 #include <utility>
00018 #include "gdcmFileSet.h"
00019
00020 namespace gdcm
00021 {
00022     class GDCM_EXPORT DICOMDIR
00023     {
00024     public:
00025         DICOMDIR() = default;
00026         DICOMDIR(FileSet fs):_FS(std::move(std::move(fs))) {}
00027     private:
00028         FileSet _FS;
00029         //13 sept 2010 mmr-- added the underscore to FS to compile under Sunos gcc
00030     };
00031 } // end namespace gdcm
00032
00033 #endif //GDCMDICOMDIR_H

```

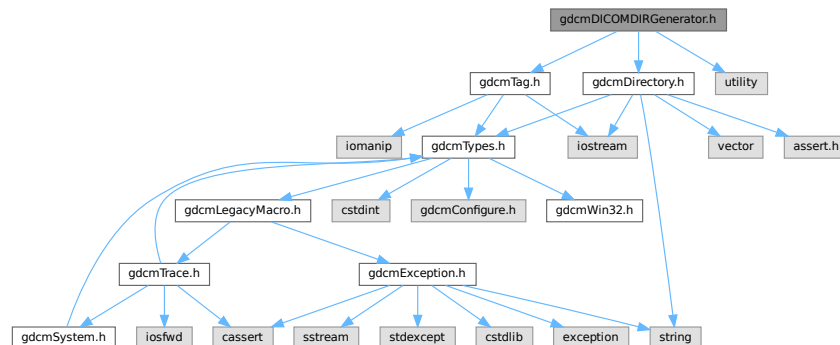
## 11.275 gdcmDICOMDIRGenerator.h File Reference

```

#include "gdcmDirectory.h"
#include "gdcmTag.h"
#include <utility>

```

Include dependency graph for gdcmDICOMDIRGenerator.h:



**Classes**

- class [gdcm::DICOMDIRGenerator](#)  
*DICOMDIRGenerator* class.

**Namespaces**

- namespace [gdcm](#)

**11.276 gdcmDICOMDIRGenerator.h**

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMDICOMDIRGENERATOR_H
00015  #define GDCMDICOMDIRGENERATOR_H
00016
00017  #include "gdcmDirectory.h"
00018  #include "gdcmTag.h"
00019  #include <utility> // std::pair
00020
00021  namespace gdcm
00022  {
00023  class File;
00024  class Scanner;
00025  class SequenceOfItems;
00026  class VL;
00027  class DICOMDIRGeneratorInternal;
00028
00056  class GDCM_EXPORT DICOMDIRGenerator
00057  {
00058  public:
00059      typedef Directory::FilenameType FilenameType;
00060      typedef Directory::FilenameType FilenameType;
00061      DICOMDIRGenerator();
00062      ~DICOMDIRGenerator();
00063
00065      void SetFilenames( FilenameType const & fns );
00066
00068      void SetRootDirectory( FilenameType const & root );
00069
00072      void SetDescriptor( const char *d );
00073
00075      bool Generate();
00076
00078      void SetFile(const File& f);
00079      File &GetFile();
00080
00081  protected:
00082      Scanner &GetScanner();
00083      bool AddPatientDirectoryRecord();
00084      bool AddStudyDirectoryRecord();
00085      bool AddSeriesDirectoryRecord();
00086      bool AddImageDirectoryRecord();
00087
00088  private:
00089      const char *ComputeFileID(const char *);
00090      bool TraverseDirectoryRecords(VL start );

```



## 11.278 gdcmDictPrinter.h

[Go to the documentation of this file.](#)

```

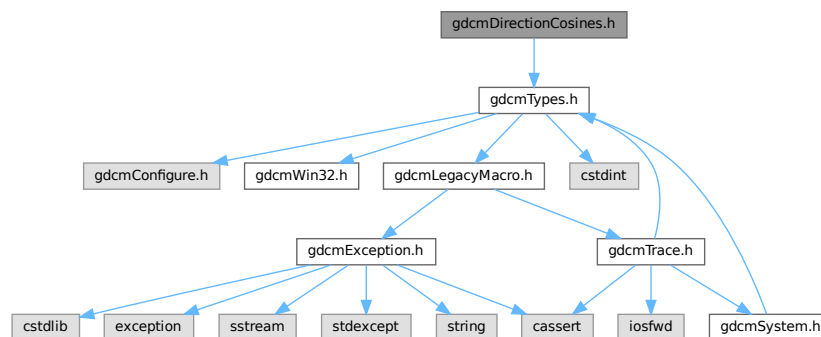
00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMDICTPRINTER_H
00015  #define GDCMDICTPRINTER_H
00016
00017  #include "gdcmPrinter.h"
00018
00019  namespace gdcm
00020  {
00021
00022  // It's a sink there is no output
00023  class GDCM_EXPORT DictPrinter : public Printer
00024  {
00025  public:
00026    DictPrinter();
00027    ~DictPrinter() = default;
00028
00029    void Print(std::ostream& os);
00030
00031  protected:
00032    void PrintDataElement2(std::ostream& os, const DataSet &ds, const DataElement &ide);
00033    void PrintDataSet2(std::ostream& os, const DataSet &ds);
00034  };
00035
00036  } // end namespace gdcm
00037
00038  #endif //GDCMDICTPRINTER_H

```

## 11.279 gdcmDirectionCosines.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmDirectionCosines.h:



## Classes

- class [gdcm::DirectionCosines](#)  
class to handle *DirectionCosines*

## Namespaces

- namespace [gdcm](#)

## 11.280 gdcmDirectionCosines.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMDIRECTIONCOSINES_H
00015 #define GDCMDIRECTIONCOSINES_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021
00022   class GDCM_EXPORT DirectionCosines
00023   {
00024   public:
00025     DirectionCosines();
00026     DirectionCosines(const double dircos[6]);
00027     // Cannot get the following signature to be wrapped with swig...
00028     //DirectionCosines(const double *dircos = 0 );
00029     ~DirectionCosines();
00030
00031     void Print(std::ostream &) const;
00032
00033     void Cross(double z[3]) const;
00034
00035     double Dot() const;
00036
00037     static double Dot(const double x[3], const double y[3]);
00038
00039     void Normalize();
00040
00041     static void Normalize(double v[3]);
00042
00043     static double Norm(const double v[3]);
00044
00045     operator const double* () const { return Values; }
00046
00047     bool IsValid() const;
00048
00049     bool SetFromString(const char *str);
00050
00051     double CrossDot(DirectionCosines const &dc) const;
00052
00053     double ComputeDistAlongNormal(const double ipp[3]) const;
00054
00055   private:
00056     double Values[6];
00057   };
00058
00059 } // end namespace gdcm
00060
00061 #endif //GDCMDIRECTIONCOSINES_H

```





```

00036 {
00037 public:
00038 //returns all series UUIDs in a given directory that match a particular SOP Instance UID
00039 static Directory::FilenameType GetSeriesUUIDsBySOPClassUID(const std::string& inDirectory,
00040     const std::string& inSOPClassUID);
00041
00042 //specific implementations of the SOPClassUID grabber, so you don't have to
00043 //remember the SOP Class UUIDs of CT or MR images.
00044 static Directory::FilenameType GetCTImageSeriesUUIDs(const std::string& inDirectory);
00045 static Directory::FilenameType GetMRImageSeriesUUIDs(const std::string& inDirectory);
00046 static Directory::FilenameType GetRTStructSeriesUUIDs(const std::string& inDirectory);
00047
00048 //given a directory and a series UID, provide all filenames with that series UID.
00049 static Directory::FilenameType GetFilenamesFromSeriesUUIDs(const std::string& inDirectory,
00050     const std::string& inSeriesUID);
00051
00052 //given a series UID, load all the images associated with that series UID
00053 //these images will be IPP sorted, so that they can be used for gathering all
00054 //the necessary information for generating an RTStruct
00055 //this function should be called by the writer once, if the writer's dataset
00056 //vector is empty. Make sure to have a new writer for new rtstructs.
00057 static std::vector<DataSet> LoadImageFromFiles(const std::string& inDirectory,
00058     const std::string& inSeriesUID);
00059
00060 //When writing RTStructs, each contour will have z position defined.
00061 //use that z position to determine the SOPInstanceUID for that plane.
00062 static std::string RetrieveSOPInstanceUIDFromZPosition(double inZPos,
00063     const std::vector<DataSet>& inDS);
00064
00065 //When writing RTStructs, the frame of reference is done by planes to start with
00066 static std::string RetrieveSOPInstanceUIDFromIndex(int inIndex,
00067     const std::vector<DataSet>& inDS);
00068
00069 //each plane needs to know the SOPClassUID, and that won't change from image to image
00070 //so, retrieve this once at the start of writing.
00071 static std::string GetSOPClassUID(const std::vector<DataSet>& inDS);
00072
00073 //retrieve the frame of reference from the set of datasets
00074 static std::string GetFrameOfReference(const std::vector<DataSet>& inDS);
00075
00076 //both the image and polydata readers use these functions to get std::strings
00077 static std::string GetStringValueFromTag(const Tag& t, const DataSet& ds);
00078 };
00079
00080 }

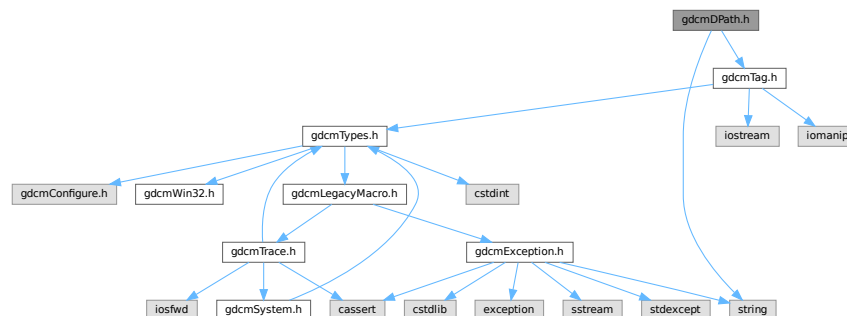
```

## 11.283 gdcmDPath.h File Reference

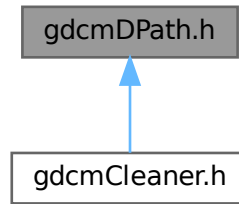
```
#include "gdcmTag.h"
```

```
#include <string>
```

Include dependency graph for gdcmDPath.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::DPath](#)

*class to handle a DICOM path While supp 118 did introduced a notion of XPath for XML Native model this convention is too XML-centric. Instead prefer DCMTK style notation <https://groups.google.com/g/comp.protocols.dicom/c/IyIH0IOBMPA>*

## Namespaces

- namespace [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const DPath &val)`

## 11.284 gdcmDPath.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMDPATH_H
00015 #define GDCMDPATH_H
00016
00017 #include "gdcmTag.h"
00018 #include <string>
00019
00020 namespace gdcm {

```



## 11.286 gdcmDumper.h

[Go to the documentation of this file.](#)

```

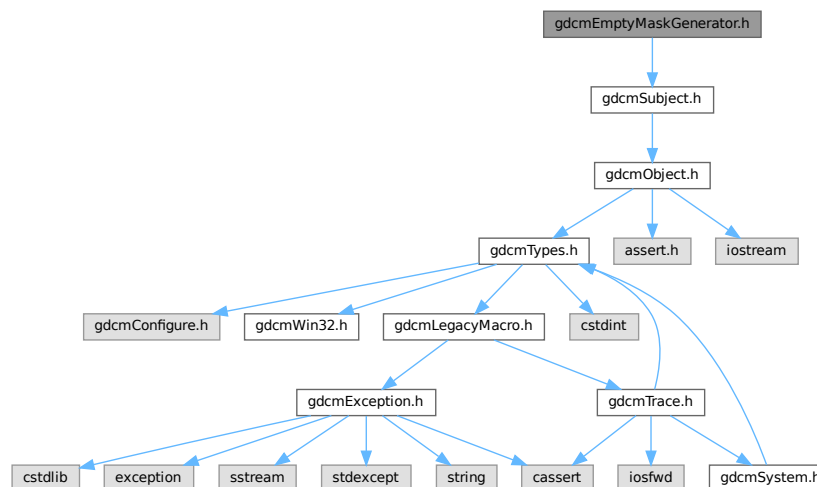
00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMDUMPER_H
00015 #define GDCMDUMPER_H
00016
00017 #include "gdcmPrinter.h"
00018
00019 namespace gdcm
00020 {
00021
00022   // It's a sink there is no output
00023   class GDCM_EXPORT Dumper : public Printer
00024   {
00025   public:
00026     Dumper() { PrintStyle = CONDENSED_STYLE; }
00027     ~Dumper() = default;
00028   };
00029
00030 } // end namespace gdcm
00031
00032 #endif //GDCMDUMPER_H

```

## 11.287 gdcmEmptyMaskGenerator.h File Reference

```
#include "gdcmSubject.h"
```

Include dependency graph for gdcmEmptyMaskGenerator.h:



## Classes

- class [gdcm::EmptyMaskGenerator](#)

*[EmptyMaskGenerator](#) Main class to generate an Empty Mask [Series](#) from an input [Series](#). This class takes an input folder and generates a series of DICOM files in the specified output directory. This class handles multiples DICOM [Series](#) within the same input directory.*

## Namespaces

- namespace [gdcm](#)

## 11.288 gdcmEmptyMaskGenerator.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMEMPTYMASKGENERATOR_H
00015  #define GDCMEMPTYMASKGENERATOR_H
00016
00017  #include "gdcmSubject.h"
00018
00019  namespace gdcm {
00050  class GDCM_EXPORT EmptyMaskGenerator
00051  {
00052  public:
00053      EmptyMaskGenerator();
00054      ~EmptyMaskGenerator();
00055
00056      enum SOPClassUIDMode {
00057          UseOriginalSOPClassUID = 0, // default
00058          UseGrayscaleSecondaryImageStorage
00059      };
00060
00063      void SetSOPClassUIDMode( SOPClassUIDMode mode );
00064
00066      void SetInputDirectory( const char * dirname );
00067
00069      void SetOutputDirectory( const char * dirname );
00070
00072      bool Execute();
00073
00074  private:
00075      struct impl;
00076      // PIMPL idiom
00077      impl* pimpl;
00078  };
00079  } // end namespace gdcm
00080  #endif //GDCMEMPTYMASKGENERATOR_H

```



```

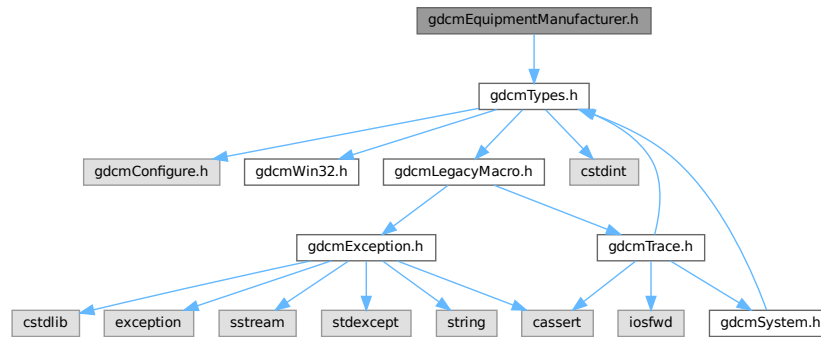
00025 {
00026 public:
00027     EncapsulatedDocument() = default;
00028
00029 private:
00030 };
00031
00032 } // end namespace gdcmm
00033
00034 #endif //GDCMENCAPSULATEDDOCUMENT_H

```

## 11.291 gdcmmEquipmentManufacturer.h File Reference

```
#include "gdcmmTypes.h"
```

Include dependency graph for gdcmmEquipmentManufacturer.h:



### Classes

- class `gdcmm::EquipmentManufacturer`

### Namespaces

- namespace `gdcmm`

## 11.292 gdcmmEquipmentManufacturer.h

[Go to the documentation of this file.](#)

```

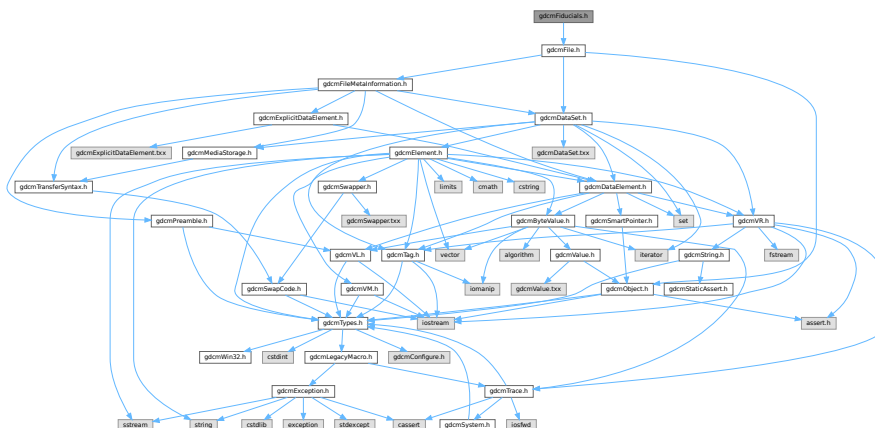
00001 /*=====
00002
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR

```



## 11.293 gdcMfiducials.h File Reference

Include dependency graph for qdcmFiducials.h:



**Classes**

- class `gdcm::Fiducials`  
*Fiducials.*

**Namespaces**

- namespace `gdcm`

**11.294 gdcmFiducials.h**

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMFIDUCIALS_H
00015 #define GDCMFIDUCIALS_H
00016
00017 #include "gdcmFile.h"
00018
00019 namespace gdcm
00020 {
00024   class GDCM_EXPORT Fiducials
00025   {
00026   public:
00027     Fiducials() = default;
00028
00029   private:
00030   };
00031
00032 } // end namespace gdcm
00033
00034 #endif //GDCMFIDUCIALS_H

```

**11.295 gdcmFileAnonymizer.h File Reference**

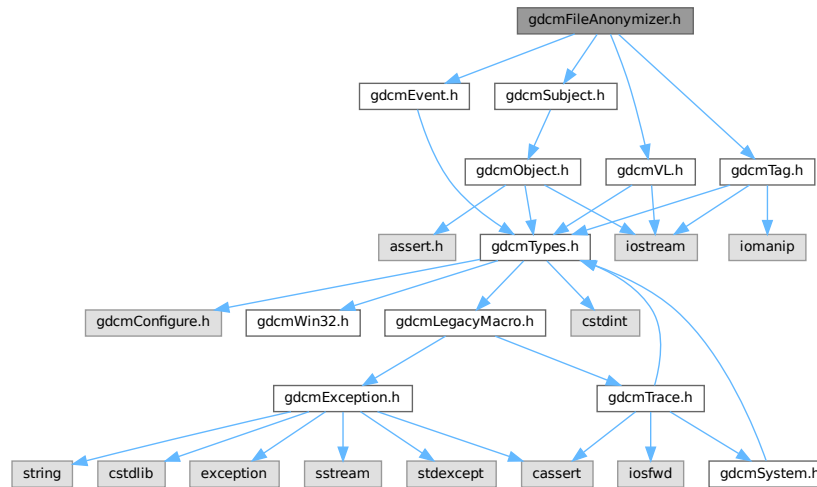
```

#include "gdcmSubject.h"
#include "gdcmEvent.h"
#include "gdcmTag.h"

```

```
#include "gdcmVL.h"
```

Include dependency graph for gdcmFileAnonymizer.h:



## Classes

- class [gdcm::FileAnonymizer](#)  
*FileAnonymizer.*

## Namespaces

- namespace [gdcm](#)

## 11.296 gdcmFileAnonymizer.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMFILEANONYMIZER_H
00015 #define GDCMFILEANONYMIZER_H
00016
00017 #include "gdcmSubject.h"
00018 #include "gdcmEvent.h"
00019 #include "gdcmTag.h"
00020 #include "gdcmVL.h"

```

```

00021
00022 namespace gdcM
00023 {
00024 class FileAnonymizerInternals;
00025
00047 class GDCM_EXPORT FileAnonymizer : public Subject
00048 {
00049 public:
00050     FileAnonymizer();
00051     ~FileAnonymizer() override;
00052
00055     void Empty( Tag const &t );
00056
00058     void Remove( Tag const &t );
00059
00063     void Replace( Tag const &t, const char *value_str );
00064
00067     void Replace( Tag const &t, const char *value_data, VL const &vl );
00068
00070     void SetInputFileName(const char *filename_native);
00071
00073     void SetOutputFileName(const char *filename_native);
00074
00076     bool Write();
00077
00078 private:
00079     bool ComputeEmptyTagPosition();
00080     bool ComputeRemoveTagPosition();
00081     bool ComputeReplaceTagPosition();
00082     FileAnonymizerInternals *Internals;
00083 };
00084
00085 } // end namespace gdcM
00086
00087 #endif //GDCMFILEANONYMIZER_H

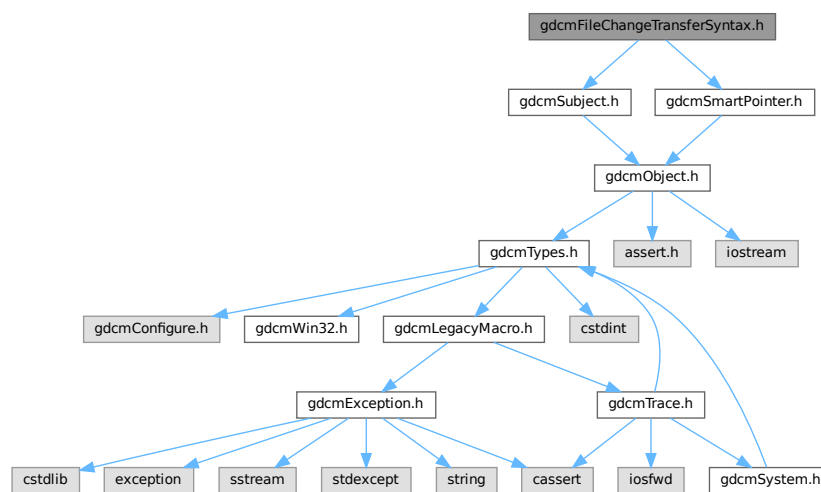
```

## 11.297 gdcMFileChangeTransferSyntax.h File Reference

```
#include "gdcMSubject.h"
```

```
#include "gdcMSmartPointer.h"
```

Include dependency graph for gdcMFileChangeTransferSyntax.h:



**Classes**

- class `gdcm::FileChangeTransferSyntax`  
*FileChangeTransferSyntax.*

**Namespaces**

- namespace `gdcm`

**11.298 gdcmFileChangeTransferSyntax.h**

[Go to the documentation of this file.](#)

```

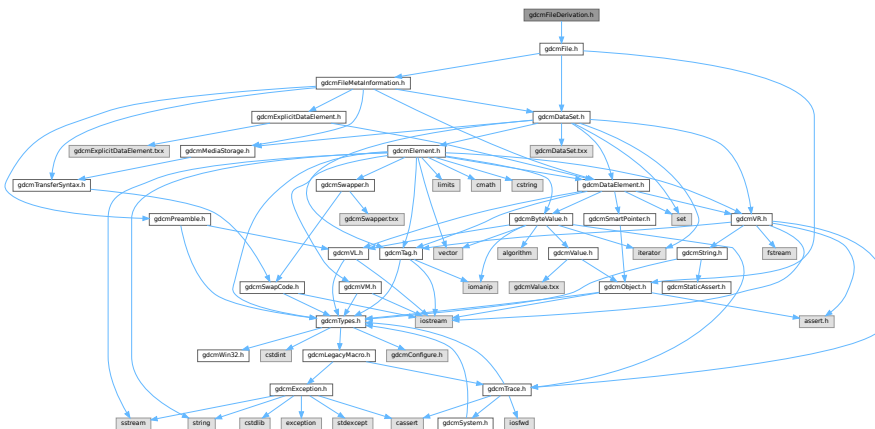
00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMFILECHANGETRANSFERSYNTAX_H
00015 #define GDCMFILECHANGETRANSFERSYNTAX_H
00016
00017 #include "gdcmSubject.h"
00018 #include "gdcmSmartPointer.h"
00019
00020 namespace gdcm
00021 {
00022     class FileChangeTransferSyntaxInternals;
00023     class ImageCodec;
00024     class TransferSyntax;
00025
00039     class GDCM_EXPORT FileChangeTransferSyntax : public Subject
00040     {
00041     public:
00042         FileChangeTransferSyntax();
00043         ~FileChangeTransferSyntax() override;
00044
00046         void SetInputFileName(const char *filename_native);
00047
00049         void SetOutputFileName(const char *filename_native);
00050
00052         bool Change();
00053
00055         void SetTransferSyntax( TransferSyntax const & ts );
00056
00059         ImageCodec * GetCodec();
00060
00062         static SmartPointer<FileChangeTransferSyntax> New() { return new FileChangeTransferSyntax; }
00063
00064     private:
00065         bool InitializeCopy();
00066         bool UpdateCompressionLevel(double level);
00067         FileChangeTransferSyntaxInternals *Internals;
00068     };
00069
00070 } // end namespace gdcm
00071
00072 #endif //GDCMFILEANONYMIZER_H

```



## 11.301 gdcmFileDerivation.h File Reference

Include dependency graph for gdcMFileDerivation.h:



- class `gdcm::FileDerivation`  
*FileDerivation* class.

- namespace **gdcm**

## 11.302 gdcmFileDerivation.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMFILEDERIVATION_H
00015  #define GDCMFILEDERIVATION_H
00016
00017  #include "gdcmFile.h"
00018
00019  namespace gdcm
00020  {
00021
00022  class FileDerivationInternals;
00023  class DataSet;
00039  class GDCM_EXPORT FileDerivation
00040  {
00041  public:
00042    FileDerivation();
00043    ~FileDerivation();
00044
00049    bool AddReference(const char *referencedsopclassuid, const char *referencedsopinstanceuid);
00050
00051    // CID 7202 Source Image Purposes of Reference
00052    // {"DCM",121320,"Uncompressed predecessor"},
00053
00055    void SetPurposeOfReferenceCodeSequenceCodeValue(unsigned int codevalue);
00056
00057    // CID 7203 Image Derivation
00058    // { "DCM",113040,"Lossy Compression" },
00059
00061    void SetDerivationCodeSequenceCodeValue(unsigned int codevalue);
00062
00064    void SetDerivationDescription( const char *dd );
00065
00069    void SetAppendDerivationHistory(bool b);
00070
00072    bool Derive();
00073
00075    void SetFile(const File& f) { F = f; }
00076    File &GetFile() { return *F; }
00077    const File &GetFile() const { return *F; }
00078
00079  protected:
00080    bool AddDerivationDescription();
00081    bool AddSourceImageSequence();
00082    bool AddPurposeOfReferenceCodeSequence(DataSet &ds);
00083
00084  private:
00085    SmartPointer<File> F;
00086    FileDerivationInternals *Internals;
00087  };
00088
00096 } // end namespace gdcm
00097
00098 #endif //GDCMFILEDERIVATION_H

```





```

00022
00038 class GDCM_EXPORT FileExplicitFilter
00039 {
00040 public:
00041     FileExplicitFilter():F(new
00042         File), ChangePrivateTags(false), UseVRUN(true), RecomputeItemLength(false), RecomputeSequenceLength(false) {}
00043     ~FileExplicitFilter() = default;
00044     void SetChangePrivateTags(bool b) { ChangePrivateTags = b; }
00045     void SetUseVRUN(bool b) { UseVRUN = b; }
00046     void SetRecomputeItemLength(bool b);
00047     void SetRecomputeSequenceLength(bool b);
00048
00049     bool Change();
00050
00051     void SetFile(const File& f) { F = f; }
00052     File &GetFile() { return *F; }
00053
00054 protected:
00055     bool ProcessDataSet(DataSet &ds, Dicts const &dicts);
00056     bool ChangeFMI();
00057
00058 private:
00059     SmartPointer<File> F;
00060     bool ChangePrivateTags;
00061     bool UseVRUN;
00062     bool RecomputeItemLength;
00063     bool RecomputeSequenceLength;
00064 };
00065
00066 } // end namespace gdcms
00067
00068 #endif //GDCMFILEEXPLICITFILTER_H

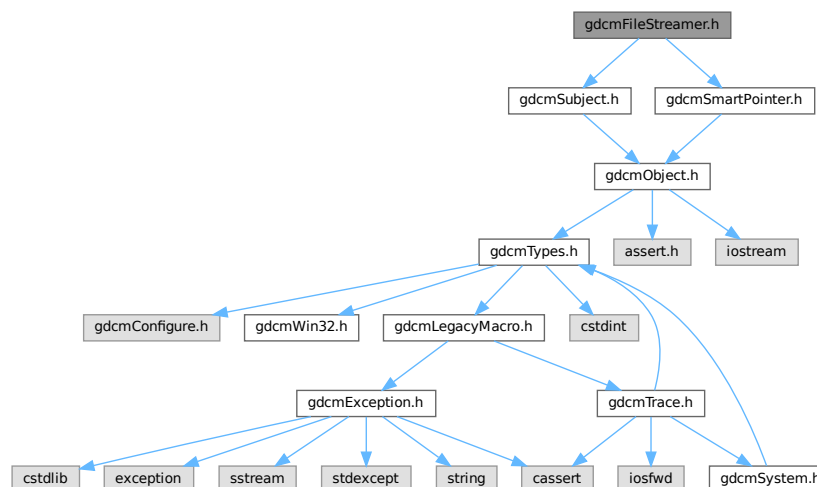
```

## 11.305 gdcmsFileStreamer.h File Reference

```
#include "gdcmsSubject.h"
```

```
#include "gdcmsSmartPointer.h"
```

Include dependency graph for gdcmsFileStreamer.h:



**Classes**

- class `gdcm::FileStreamer`  
*FileStreamer.*

**Namespaces**

- namespace `gdcm`

**11.306 gdcmFileStreamer.h**

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMFILESTREAMER_H
00015  #define GDCMFILESTREAMER_H
00016
00017  #include "gdcmSubject.h"
00018  #include "gdcmSmartPointer.h"
00019
00020  namespace gdcm
00021  {
00022  class FileStreamerInternals;
00023
00024  class Tag;
00025  class PrivateTag;
00041  class GDCM_EXPORT FileStreamer : public Subject
00042  {
00043  public:
00044  FileStreamer();
00045  ~FileStreamer() override;
00046
00048  void SetTemplateFileName(const char *filename_native);
00049
00050  // Decide to check template or not (default: false)
00055  void CheckTemplateFileName(bool check);
00056
00058  void SetOutputFileName(const char *filename_native);
00059
00064  bool CheckDataElement( const Tag & t );
00065
00068  bool StartDataElement( const Tag & t );
00070  bool AppendToDataElement( const Tag & t, const char *array, size_t len );
00072  bool StopDataElement( const Tag & t );
00076  bool ReserveDataElement( size_t len );
00077
00085  bool StartGroupDataElement( const PrivateTag & pt, size_t maxsize = 0, uint8_t startoffset = 0 );
00087  bool AppendToGroupDataElement( const PrivateTag & pt, const char *array, size_t len );
00089  bool StopGroupDataElement( const PrivateTag & pt );
00092  bool ReserveGroupDataElement( unsigned short ndataelement );
00093
00095  static SmartPointer<FileStreamer> New() { return new FileStreamer; }
00096
00097  private:
00098  bool InitializeCopy();
00099  FileStreamerInternals *Internals;
00100  };
00101
00102 } // end namespace gdcm
00103
00104 #endif //GDCMFILESTREAMER_H

```



## 11.308 gdcmlconImage.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMLCONIMAGE_H
00015 #define GDCMLCONIMAGE_H
00016
00017 #if 0
00018 #include "gdcmlObject.h"
00019 #include "gdcmlDataElement.h"
00020 #include "gdcmlPhotometricInterpretation.h"
00021 #include "gdcmlPixelFormat.h"
00022 #include "gdcmlTransferSyntax.h"
00023
00024 #include <vector>
00025
00026 namespace gdcml
00027 {
00028
00029   class GDCML_EXPORT IconImage : public Object
00030   {
00031   public:
00032     IconImage();
00033     ~IconImage();
00034     void Print(std::ostream &) const {}
00035
00036     void SetTransferSyntax(TransferSyntax const &ts) {
00037       TS = ts;
00038     }
00039     const TransferSyntax &GetTransferSyntax() const {
00040       return TS;
00041     }
00042     void SetDataElement(DataElement const &de) {
00043       PixelData = de;
00044     }
00045     const DataElement& GetDataElement() const { return PixelData; }
00046
00047     void SetColumns(unsigned int col) { SetDimension(0,col); }
00048     void SetRows(unsigned int rows) { SetDimension(1,rows); }
00049     void SetDimension(unsigned int idx, unsigned int dim);
00050     int GetColumns() const { return Dimensions[0]; }
00051     int GetRows() const { return Dimensions[1]; }
00052     // Get/Set PixelFormat
00053     const PixelFormat &GetPixelFormat() const
00054     {
00055       return PF;
00056     }
00057     void SetPixelFormat(PixelFormat const &pf)
00058     {
00059       PF = pf;
00060     }
00061     const PhotometricInterpretation &GetPhotometricInterpretation() const;
00062     void SetPhotometricInterpretation(PhotometricInterpretation const &pi);
00063
00064     bool IsEmpty() const { return Dimensions.size() == 0; }
00065     void Clear();
00066
00067     bool GetBuffer(char *buffer) const;
00068
00069   private:
00070     TransferSyntax TS;
00071     PixelFormat PF; // SamplesPerPixel, BitsAllocated, BitsStored, HighBit, PixelRepresentation
00072     PhotometricInterpretation PI;
00073     std::vector<unsigned int> Dimensions; // Col/Row
00074     std::vector<double> Spacing; // PixelAspectRatio ?
00075     DataElement PixelData; // copied from 7fe0,0010

```



### 11.311 gdcmlconImageGenerator.h File Reference

The diagram is a complex dependency graph for the gdcm library. It shows numerous header files and their interdependencies. The graph is organized into layers, with some headers like 'gdcmImage.h' and 'gdcmImageReader.h' at the top, and more fundamental headers like 'gdcm.h' and 'gdcmImage.h' at the bottom. The graph illustrates the intricate relationships between different components of the library, such as data elements, maps, pointers, and various utility classes.

## Classes

- class `gdcm::IconImageGenerator`  
*IconImageGenerator.*

## Namespaces

- namespace `gdcm`

## 11.312 gdcmIconImageGenerator.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMICONIMAGEGENERATOR_H
00015 #define GDCMICONIMAGEGENERATOR_H
00016
00017 #include "gdcmPixmap.h"
00018 #include "gdcmIconImage.h"
00019
00020 namespace gdcm
00021 {
00022     class IconImageGeneratorInternals;
00041     class GDCM_EXPORT IconImageGenerator
00042     {
00043     public:
00044         IconImageGenerator();
00045         ~IconImageGenerator();
00046
00048         void SetPixmap(const Pixmap& p) { P = p; }
00049         Pixmap &GetPixmap() { return *P; }
00050         const Pixmap &GetPixmap() const { return *P; }
00051
00053         void SetOutputDimensions(const unsigned int dims[2]);
00054
00058         void SetPixelMinMax(double min, double max);
00059
00063         void AutoPixelMinMax(bool b);
00064
00069         void ConvertRGBToPaletteColor(bool b);
00070
00074         void SetOutsideValuePixel(double v);
00075
00077         bool Generate();
00078
00080         const IconImage& GetIconImage() const { return *I; }
00081
00082     protected:
00083
00084     private:
00085         void BuildLUT( Bitmap & bitmap, unsigned int maxcolor );
00086
00087         SmartPointer<Pixmap> P;
00088         SmartPointer<IconImage> I;
00089         IconImageGeneratorInternals *Internals;
00090     };
00091
00092 } // end namespace gdcm
00093
00094 #endif //GDCMICONIMAGEGENERATOR_H

```





## 11.314 gdcmlImage.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMIMAGE_H
00015 #define GDCMIMAGE_H
00016
00017 #include "gdcmlPixmap.h"
00018
00019 #include <vector>
00020
00021 namespace gdcml
00022 {
00023
00024   class GDCM_EXPORT Image : public Pixmap
00025   {
00026   public:
00027     Image () : Spacing(), SC(), Intercept(0), Slope(1) {
00028       //DirectionCosines.resize(6);
00029       Origin.resize( 3 /*NumberOfDimensions*/ ); // fill with 0
00030       DirectionCosines.resize( 6 ); // fill with 0
00031       DirectionCosines[0] = 1;
00032       DirectionCosines[4] = 1;
00033       Spacing.resize( 3 /*NumberOfDimensions*/, 1 ); // fill with 1
00034     }
00035     ~Image() override = default;
00036
00037     const double *GetSpacing() const;
00038     double GetSpacing(unsigned int idx) const;
00039     void SetSpacing(const double spacing[3]);
00040     void SetSpacing(unsigned int idx, double spacing);
00041
00042     const double *GetOrigin() const;
00043     double GetOrigin(unsigned int idx) const;
00044     void SetOrigin(const float origin[3]);
00045     void SetOrigin(const double origin[3]);
00046     void SetOrigin(unsigned int idx, double ori);
00047
00048     const double *GetDirectionCosines() const;
00049     double GetDirectionCosines(unsigned int idx) const;
00050     void SetDirectionCosines(const float dircos[6]);
00051     void SetDirectionCosines(const double dircos[6]);
00052     void SetDirectionCosines(unsigned int idx, double dircos);
00053
00054     void Print(std::ostream &os) const override;
00055
00056     void SetIntercept(double intercept) { Intercept = intercept; }
00057     double GetIntercept() const { return Intercept; }
00058
00059     void SetSlope(double slope) { Slope = slope; }
00060     double GetSlope() const { return Slope; }
00061
00062   private:
00063     std::vector<double> Spacing;
00064     std::vector<double> Origin;
00065     std::vector<double> DirectionCosines;
00066
00067     // I believe the following 3 ivars can be derived from TS ...
00068     SwapCode SC;
00069     double Intercept;
00070     double Slope;
00071   };
00072
00073 } // end namespace gdcml
00074
00075 #endif //GDCMIMAGE_H

```





## Functions

- template<typename T >  
static T **gdcm::Clamp** (int v)
- template<typename T >  
static int **gdcm::Round** (T x)

## 11.318 gdcmImageChangePhotometricInterpretation.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMIMAGECHANGEPHOTOMETRICINTERPRETATION_H
00015 #define GDCMIMAGECHANGEPHOTOMETRICINTERPRETATION_H
00016
00017 #include "gdcmImageToImageFilter.h"
00018 #include "gdcmPhotometricInterpretation.h"
00019 #include <limits>
00020
00021 namespace gdcm
00022 {
00023
00024   class DataElement;
00025   class GDCM_EXPORT ImageChangePhotometricInterpretation : public ImageToImageFilter
00026   {
00027   public:
00028     ImageChangePhotometricInterpretation():PI() {}
00029     ~ImageChangePhotometricInterpretation() = default;
00030
00031     void SetPhotometricInterpretation(PhotometricInterpretation const &pi) { PI = pi; }
00032     const PhotometricInterpretation &GetPhotometricInterpretation() const { return PI; }
00033
00034     bool Change();
00035
00036     template <typename T>
00037     static void RGB2YBR(T ybr[3], const T rgb[3], unsigned short storedbits = 8);
00038     template <typename T>
00039     static void YBR2RGB(T rgb[3], const T ybr[3], unsigned short storedbits = 8);
00040
00041   protected:
00042     bool ChangeMonochrome();
00043     bool ChangeYBR2RGB();
00044     bool ChangeRGB2YBR();
00045
00046   private:
00047     PhotometricInterpretation PI;
00048   };
00049
00050   template <typename T>
00051   static inline int Round(T x)
00052   {
00053     return (int) (x+0.5);
00054   }
00055
00056   template <typename T>
00057   static inline T Clamp(int v)
00058   {
00059     assert( std::numeric_limits<T>::min() == 0 );
00060     return v < 0 ? 0 : (v > std::numeric_limits<T>::max() ? std::numeric_limits<T>::max() : v);
00061   }
00062 }
00063

```



**Classes**

- class [gdcm::ImageChangePlanarConfiguration](#)  
*ImageChangePlanarConfiguration* class.

**Namespaces**

- namespace [gdcm](#)

**11.320 gdcmImageChangePlanarConfiguration.h**

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMIMAGECHANGEPLANARCONFIGURATION_H
00015  #define GDCMIMAGECHANGEPLANARCONFIGURATION_H
00016
00017  #include "gdcmImageToImageFilter.h"
00018
00019  namespace gdcm
00020  {
00021
00022  class DataElement;
00028  class GDCM_EXPORT ImageChangePlanarConfiguration : public ImageToImageFilter
00029  {
00030  public:
00031  ImageChangePlanarConfiguration():PlanarConfiguration(0) {}
00032  ~ImageChangePlanarConfiguration() = default;
00033
00035  void SetPlanarConfiguration(unsigned int pc) { PlanarConfiguration = pc; }
00036  unsigned int GetPlanarConfiguration() const { return PlanarConfiguration; }
00037
00040  template <typename T>
00041  static size_t RGBPlanesToRGBPixels(T *out, const T *r, const T *g, const T *b, size_t s);
00042
00046  template <typename T>
00047  static size_t RGBPixelsToRGBPlanes(T *r, T *g, T *b, const T* rgb, size_t s);
00048
00050  bool Change();
00051
00052  protected:
00053
00054  private:
00055  unsigned int PlanarConfiguration;
00056  };
00057
00058  template <typename T>
00059  size_t ImageChangePlanarConfiguration::RGBPlanesToRGBPixels(T *out, const T *r, const T *g, const T *b,
00060  size_t s)
00060  {
00061  T *pout = out;
00062  for(size_t i = 0; i < s; ++i )
00063  {
00064  *pout++ = *r++;
00065  *pout++ = *g++;
00066  *pout++ = *b++;
00067  }
00068  }

```





## 11.322 gdcmImageChangeTransferSyntax.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMIMAGECHANGETRANSFERSYNTAX_H
00015 #define GDCMIMAGECHANGETRANSFERSYNTAX_H
00016
00017 #include "gdcmImageToImageFilter.h"
00018 #include "gdcmTransferSyntax.h"
00019
00020 namespace gdcm
00021 {
00022
00023   class DataElement;
00024   class ImageCodec;
00039   class GDCM_EXPORT ImageChangeTransferSyntax : public ImageToImageFilter
00040   {
00041   public:
00042     ImageChangeTransferSyntax():TS(TransferSyntax::TS_END),Force(false),CompressIconImage(false),UserCodec(nullptr)
00043     {}
00044     ~ImageChangeTransferSyntax() = default;
00046     void SetTransferSyntax(const TransferSyntax &ts) { TS = ts; }
00048     const TransferSyntax &GetTransferSyntax() const { return TS; }
00049
00051     bool Change();
00052
00055     void SetCompressIconImage(bool b) { CompressIconImage = b; }
00056
00061     void SetForce( bool f ) { Force = f; }
00062
00068     void SetUserCodec(ImageCodec *ic) { UserCodec = ic; }
00069
00070   protected:
00071     bool TryJPEGCodec(const DataElement &pixelde, Bitmap const &input, Bitmap &output);
00072     bool TryJPEG2000Codec(const DataElement &pixelde, Bitmap const &input, Bitmap &output);
00073     bool TryJPEGLSCodec(const DataElement &pixelde, Bitmap const &input, Bitmap &output);
00074     bool TryRAWCodec(const DataElement &pixelde, Bitmap const &input, Bitmap &output);
00075     bool TryRLECodec(const DataElement &pixelde, Bitmap const &input, Bitmap &output);
00076
00077   private:
00078     TransferSyntax TS;
00079     bool Force;
00080     bool CompressIconImage;
00081
00082     ImageCodec *UserCodec;
00083   };
00084
00090 } // end namespace gdcm
00091
00092 #endif //GDCMIMAGECHANGETRANSFERSYNTAX_H

```

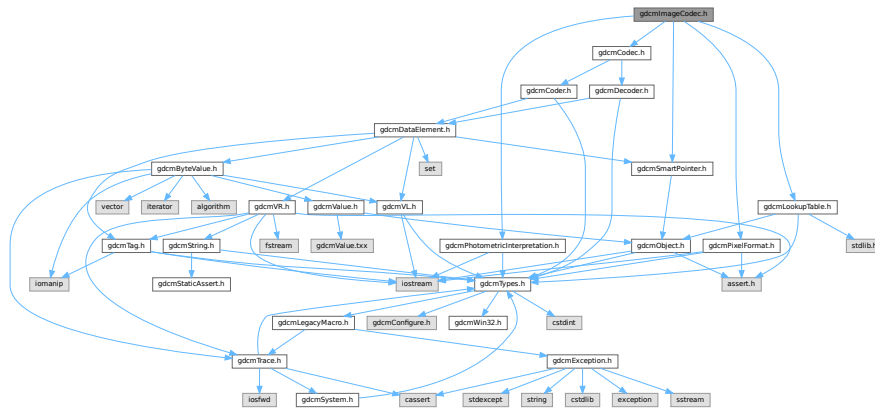
## 11.323 gdcmImageCodec.h File Reference

```

#include "gdcmCodec.h"
#include "gdcmPhotometricInterpretation.h"
#include "gdcmLookupTable.h"
#include "gdcmSmartPointer.h"

```

```
#include "gdcmPixelFormat.h"
Include dependency graph for gdcmImageCodec.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::ImageCodec](#)  
*ImageCodec.*

## Namespaces

- namespace [gdcm](#)

## 11.324 gdcmImageCodec.h

[Go to the documentation of this file.](#)

```
00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
```

```

00013 =====*/
00014 #ifndef GDCMIMAGECODEC_H
00015 #define GDCMIMAGECODEC_H
00016
00017 #include "gdcmCodec.h"
00018 #include "gdcmPhotometricInterpretation.h"
00019 #include "gdcmLookupTable.h"
00020 #include "gdcmSmartPointer.h"
00021 #include "gdcmPixelFormat.h"
00022
00023 namespace gdcm
00024 {
00025
00026     class GDCM_EXPORT ImageCodec : public Codec
00027     {
00028     public:
00029         ImageCodec();
00030         ~ImageCodec() override;
00031         bool CanCode(TransferSyntax const &) const override { return false; }
00032         bool CanDecode(TransferSyntax const &) const override { return false; }
00033         bool Decode(DataElement const &is_, DataElement &os) override;
00034         bool IsLossy() const;
00035         void SetLossyFlag(bool l);
00036         bool GetLossyFlag() const;
00037
00038         virtual bool GetHeaderInfo(std::istream &is_, TransferSyntax &ts);
00039         virtual ImageCodec * Clone() const = 0;
00040
00041     protected:
00042         bool DecodeByStreams(std::istream &is_, std::ostream &os) override;
00043         virtual bool IsValid(PhotometricInterpretation const &pi);
00044     public:
00045         unsigned int GetPlanarConfiguration() const
00046         {
00047             return PlanarConfiguration;
00048         }
00049         void SetPlanarConfiguration(unsigned int pc)
00050         {
00051             assert( pc == 0 || pc == 1 );
00052             PlanarConfiguration = pc;
00053         }
00054
00055         PixelFormat &GetPixelFormat()
00056         {
00057             return PF;
00058         }
00059         const PixelFormat &GetPixelFormat() const
00060         {
00061             return PF;
00062         }
00063         virtual void SetPixelFormat(PixelFormat const &pf)
00064         {
00065             PF = pf;
00066         }
00067         const PhotometricInterpretation &GetPhotometricInterpretation() const;
00068         void SetPhotometricInterpretation(PhotometricInterpretation const &pi);
00069
00070         bool GetNeedByteSwap() const
00071         {
00072             return NeedByteSwap;
00073         }
00074         void SetNeedByteSwap(bool b)
00075         {
00076             NeedByteSwap = b;
00077         }
00078         void SetNeedOverlayCleanup(bool b)
00079         {
00080             NeedOverlayCleanup = b;
00081         }
00082         void SetLUT(LookupTable const &lut)
00083         {
00084             LUT = SmartPointer<LookupTable>( const_cast<LookupTable*>(&lut) );
00085         }
00086         const LookupTable &GetLUT() const
00087         {
00088             return *LUT;
00089         }
00090     };
00091
00092 #endif

```

```

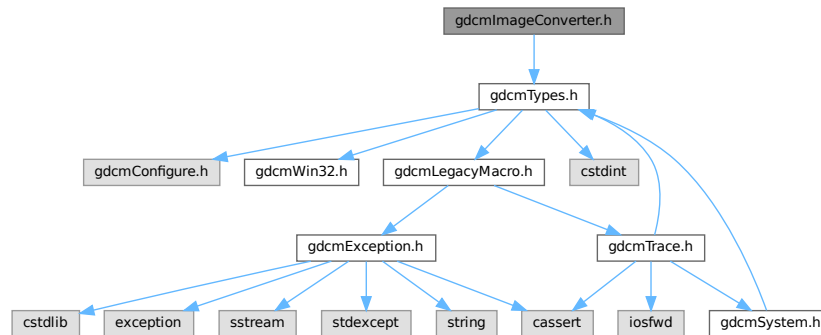
00098 void SetDimensions(const unsigned int d[3]);
00099 void SetDimensions(const std::vector<unsigned int> & d);
00100 const unsigned int *GetDimensions() const { return Dimensions; }
00101 void SetNumberOfDimensions(unsigned int dim);
00102 unsigned int GetNumberOfDimensions() const;
00103
00104 bool CleanupUnusedBits(char * data, size_t datalen);
00105
00106 protected:
00107     // Streaming (write) API:
00114     friend class FileChangeTransferSyntax;
00115     virtual bool StartEncode( std::ostream & os );
00116     virtual bool IsRowEncoder();
00117     virtual bool IsFrameEncoder();
00118     virtual bool AppendRowEncode( std::ostream & out, const char * data, size_t datalen );
00119     virtual bool AppendFrameEncode( std::ostream & out, const char * data, size_t datalen );
00120     virtual bool StopEncode( std::ostream & os);
00121
00122 protected:
00123     bool RequestPlanarConfiguration;
00124     bool RequestPaddedCompositePixelCode;
00125 //private:
00126     unsigned int PlanarConfiguration;
00127     PhotometricInterpretation PI;
00128     PixelFormat PF;
00129     bool NeedByteSwap;
00130     bool NeedOverlayCleanup;
00131
00132     typedef SmartPointer<LookupTable> LUTPtr;
00133     LUTPtr LUT;
00134     unsigned int Dimensions[3]; // FIXME
00135     unsigned int NumberOfDimensions;
00136     bool LossyFlag;
00137
00138     bool DoOverlayCleanup(std::istream &is_, std::ostream &os);
00139     bool DoByteSwap(std::istream &is_, std::ostream &os);
00140     bool DoYBR(std::istream &is_, std::ostream &os);
00141     bool DoYBRFull422(std::istream &is_, std::ostream &os);
00142     bool DoPlanarConfiguration(std::istream &is_, std::ostream &os);
00143     bool DoSimpleCopy(std::istream &is_, std::ostream &os);
00144     bool DoPaddedCompositePixelCode(std::istream &is_, std::ostream &os);
00145     bool DoInvertMonochrome(std::istream &is_, std::ostream &os);
00146
00147     //template <typename T>
00148     //bool DoInvertPlanarConfiguration(T *output, const T *input, uint32_t length);
00149 };
00150
00151 } // end namespace gdcm
00152
00153 #endif //GDCMIMAGECODEC_H

```

## 11.325 gdcmImageConverter.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmImageConverter.h:



### Classes

- class [gdcm::ImageConverter](#)  
*Image Converter.*

### Namespaces

- namespace [gdcm](#)

## 11.326 gdcmImageConverter.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014
00015 #ifndef GDCMIMAGECONVERTER_H
00016 #define GDCMIMAGECONVERTER_H
00017
00018 #include "gdcmTypes.h"
00019
00020 namespace gdcm
00021 {
00022
00023   class Image;

```



## 11.328 gdcmImageFragmentSplitter.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMIMAGEFRAGMENTSPLITTER_H
00015  #define GDCMIMAGEFRAGMENTSPLITTER_H
00016
00017  #include "gdcmImageToImageFilter.h"
00018
00019  namespace gdcm
00020  {
00021
00022  class DataElement;
00027  class GDCM_EXPORT ImageFragmentSplitter : public ImageToImageFilter
00028  {
00029  public:
00030      ImageFragmentSplitter():FragmentSizeMax(0),Force(false) {}
00031      ~ImageFragmentSplitter() = default;
00032
00033      bool Split();
00034
00035      void SetFragmentSizeMax(unsigned int fragsize);
00038      unsigned int GetFragmentSizeMax() const { return FragmentSizeMax; }
00039
00042      void SetForce( bool f ) { Force = f; }
00043
00044  protected:
00045
00046  private:
00047      unsigned int FragmentSizeMax;
00048      bool Force;
00049  };
00050
00051  } // end namespace gdcm
00052
00053  #endif //GDCMIMAGEFRAGMENTSPLITTER_H

```

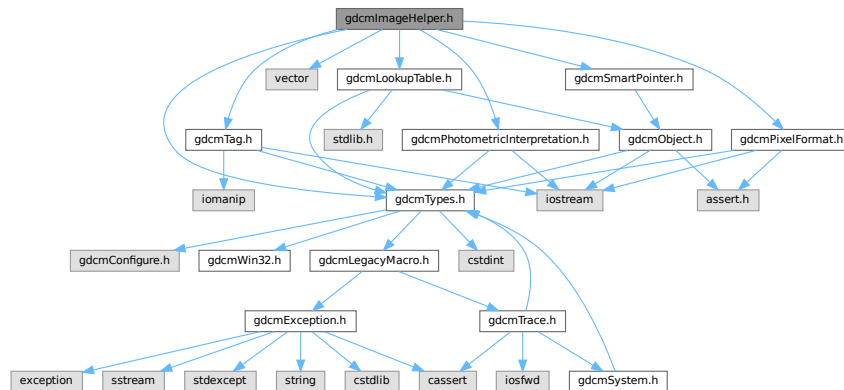
## 11.329 gdcmImageHelper.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmTag.h"
#include <vector>
#include "gdcmPixelFormat.h"
#include "gdcmPhotometricInterpretation.h"
#include "gdcmSmartPointer.h"
#include "gdcmLookupTable.h"

```

Include dependency graph for `gdcmImageHelper.h`:



## Classes

- class `gdcm::ImageHelper`  
*ImageHelper* (internal class, not intended for user level)
- struct `gdcm::RealWorldValueMappingContent`

## Namespaces

- namespace `gdcm`

## 11.330 gdcmImageHelper.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 #ifndef GDCMIMAGEHELPER_H
00015 #define GDCMIMAGEHELPER_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmTag.h"
00019 #include <vector>
00020 #include "gdcmPixelFormat.h"
00021 #include "gdcmPhotometricInterpretation.h"
00022 #include "gdcmSmartPointer.h"
00023 #include "gdcmLookupTable.h"
00024
00025 namespace gdcm
00026 {

```



```

00027
00028 class MediaStorage;
00029 class DataSet;
00030 class File;
00031 class Image;
00032 class Pixmap;
00033 class ByteValue;
00034
00035 // minimal struct:
00036 struct RealWorldValueMappingContent {
00037     double RealWorldValueIntercept;
00038     double RealWorldValueSlope;
00039     // http://dicom.nema.org/MEDICAL/DICOM/2014c/output/chtml/part16/sect_CID_7181.html
00040     std::string CodeValue;
00041     std::string CodeMeaning;
00042 };
00043
00059 class GDCM_EXPORT ImageHelper
00060 {
00061 public:
00068     static void SetForceRescaleInterceptSlope(bool);
00069     static bool GetForceRescaleInterceptSlope();
00070
00076     static void SetPMSRescaleInterceptSlope(bool);
00077     static bool GetPMSRescaleInterceptSlope();
00078
00085     static void SetForcePixelSpacing(bool);
00086     static bool GetForcePixelSpacing();
00087
00092     static void SetSecondaryCaptureImagePlaneModule(bool);
00093     static bool GetSecondaryCaptureImagePlaneModule();
00094
00098     static std::vector<unsigned int> GetDimensionsValue(const File& f);
00099     static void SetDimensionsValue(File& f, const Pixmap & img);
00100
00103     static PixelFormat GetPixelFormatValue(const File& f);
00104
00109     static std::vector<double> GetRescaleInterceptSlopeValue(File const & f);
00110     static void SetRescaleInterceptSlopeValue(File & f, const Image & img);
00111
00112     // read only for now
00113     static bool GetRealWorldValueMappingContent(File const & f, RealWorldValueMappingContent & rwwmc);
00114
00116     static std::vector<double> GetOriginValue(File const & f);
00117     static void SetOriginValue(DataSet & ds, const Image & img);
00118
00121     static std::vector<double> GetDirectionCosinesValue(File const & f);
00122     // FIXME: There is a major issue for image with multiple IOP (eg. Enhanced * Image Storage).
00123     static void SetDirectionCosinesValue(DataSet & ds, const std::vector<double> & dircos);
00124
00131     static std::vector<double> GetSpacingValue(File const & f);
00132     static void SetSpacingValue(DataSet & ds, const std::vector<double> & spacing);
00133
00136     static bool ComputeSpacingFromImagePositionPatient(const std::vector<double> & imageposition,
std::vector<double> & spacing);
00137
00138     static bool GetDirectionCosinesFromDataSet(DataSet const & ds, std::vector<double> & dircos);
00139
00140     //functions to get more information from a file
00141     //useful for the stream image reader, which fills in necessary image information
00142     //distinctly from the reader-style data input
00143     static PhotometricInterpretation GetPhotometricInterpretationValue(File const& f);
00144     //returns the configuration of colors in a plane, either RGB RGB RGB or RRR GGG BBB
00145     static unsigned int GetPlanarConfigurationValue(const File& f);
00146
00148     static SmartPointer<LookupTable> GetLUT(File const& f);
00149
00150     // Moved from PixampReader to here. Generally used for photometric interpretation.
00151     static const ByteValue* GetPointerFromElement(Tag const &tag, File const& f);
00152
00154     static MediaStorage ComputeMediaStorageFromModality(const char *modality,
    unsigned int dimension = 2, PixelFormat const & pf = PixelFormat(),
    PhotometricInterpretation const & pi = PhotometricInterpretation(),
    double rescaleintercept = 0, double rescaleslope = 1 );
00155
00159 protected:
00160     static Tag GetSpacingTagFromMediaStorage(MediaStorage const &ms);
00161     static Tag GetZSpacingTagFromMediaStorage(MediaStorage const &ms);
00162
00163 private:
00164     static bool ForceRescaleInterceptSlope;

```



## Namespaces

- namespace `gdcm`

## 11.332 gdcmImageReader.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMIMAGEREADER_H
00015 #define GDCMIMAGEREADER_H
00016
00017 #include "gdcmPixmapReader.h"
00018 #include "gdcmImage.h"
00019
00020 namespace gdcm
00021 {
00022
00023   class MediaStorage;
00024   class GDCM_EXPORT ImageReader : public PixmapReader
00025   {
00026   public:
00027     ImageReader();
00028     ~ImageReader() override; //needs to be virtual to ensure lack of memory leaks
00029
00030     bool Read() override;
00031
00032     // Following methods are valid only after a call to 'Read'
00033     const Image& GetImage() const;
00034     Image& GetImage();
00035     //void SetImage(Image const &img);
00036
00037   protected:
00038     bool ReadImage(MediaStorage const &ms) override;
00039     bool ReadACRNEMAImage() override;
00040   };
00041
00042 } // end namespace gdcm
00043
00044 #endif //GDCMIMAGEREADER_H

```

## 11.333 gdcmImageRegionReader.h File Reference

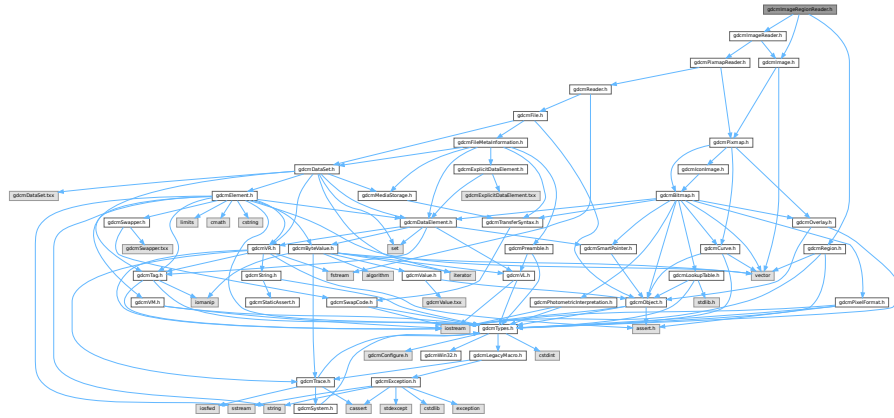
```

#include "gdcmImageReader.h"
#include "gdcmImage.h"

```

```
#include "gdcmRegion.h"
```

Include dependency graph for gdcmImageRegionReader.h:



## Classes

- class `gdcm::ImageRegionReader`  
*ImageRegionReader.*

## Namespaces

- namespace `gdcm`

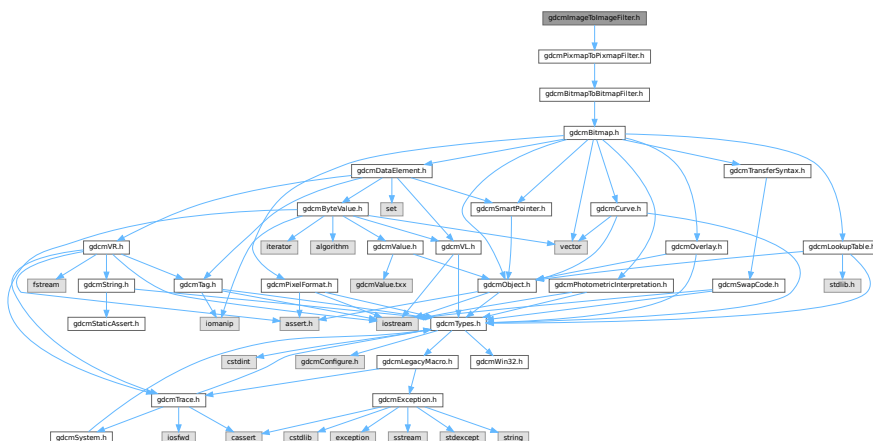
## 11.334 gdcmImageRegionReader.h

[Go to the documentation of this file.](#)

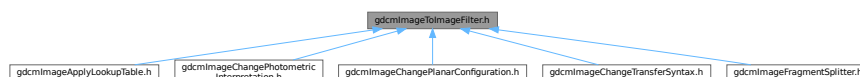
```
00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMIMAGEEXTENTREADER_H
00015 #define GDCMIMAGEEXTENTREADER_H
00016
00017 #include "gdcmImageReader.h"
00018 #include "gdcmImage.h"
00019 #include "gdcmRegion.h"
00020
00021 namespace gdcm
00022 {
00023
00024   class ImageRegionReaderInternals;
00034   class GDCM_EXPORT ImageRegionReader : public ImageReader
00035   {
00036   public:
```

### 11.335 gdcmlImageToImageFilter.h File Reference

Include dependency graph for `gdcmImageToImageFilter.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::ImageToImageFilter](#)  
*ImageToImageFilter* class.

## Namespaces

- namespace [gdcm](#)

## 11.336 gdcmImageToImageFilter.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMIMAGETOIMAGEFILTER_H
00015 #define GDCMIMAGETOIMAGEFILTER_H
00016
00017 #include "gdcmPixmapToPixmapFilter.h"
00018
00019 namespace gdcm
00020 {
00021
00022   class Image;
00027   class GDCM_EXPORT ImageToImageFilter : public PixmapToPixmapFilter
00028   {
00029   public:
00030     ImageToImageFilter();
00031     ~ImageToImageFilter() = default;
00032
00033     Image &GetInput();
00034
00035     // NOTE: covariant return-type to preserve backward compatible API
00037     const Image &GetOutput() const;
00038
00039   protected:
00040   };
00041
00042 } // end namespace gdcm
00043
00044 #endif //GDCMIMAGETOIMAGEFILTER_H

```

## 11.337 gdcmImageWriter.h File Reference

```

#include "gdcmPixmapWriter.h"
#include "gdcmImage.h"

```

- class `gdcm::ImageWriter`  
*ImageWriter*.

- namespace **gdcm**

[Go to the documentation of this file.](#)

Generated by Doxygen

```

00037
00041  const Image& GetImage() const override { return dynamic_cast<const Image*>(*PixelData); }
00042  Image& GetImage() override { return dynamic_cast<Image*>(*PixelData); } // FIXME
00043  //void SetImage(Image const &img);
00044
00046  bool Write() override; // Execute()
00047
00050  MediaStorage ComputeTargetMediaStorage();
00051 protected:
00052
00053 private:
00054 };
00055
00056 } // end namespace gdcmm
00057
00058 #endif //GDCMIMAGEWRITER_H

```

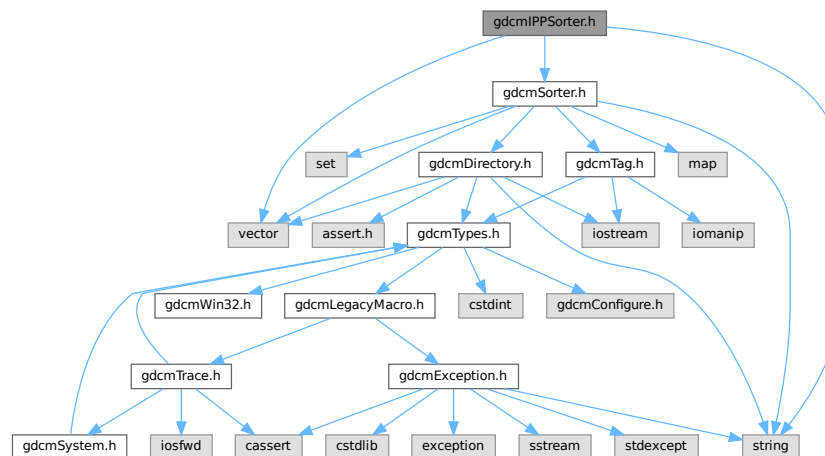
## 11.339 gdcmmIPPSorter.h File Reference

```

#include "gdcmmSorter.h"
#include <vector>
#include <string>

```

Include dependency graph for gdcmmIPPSorter.h:



### Classes

- class [gdcmm::IPPSorter](#)  
*IPPSorter.*

### Namespaces

- namespace [gdcmm](#)



## 11.340 gdcmlPPSorter.h

[Go to the documentation of this file.](#)

```

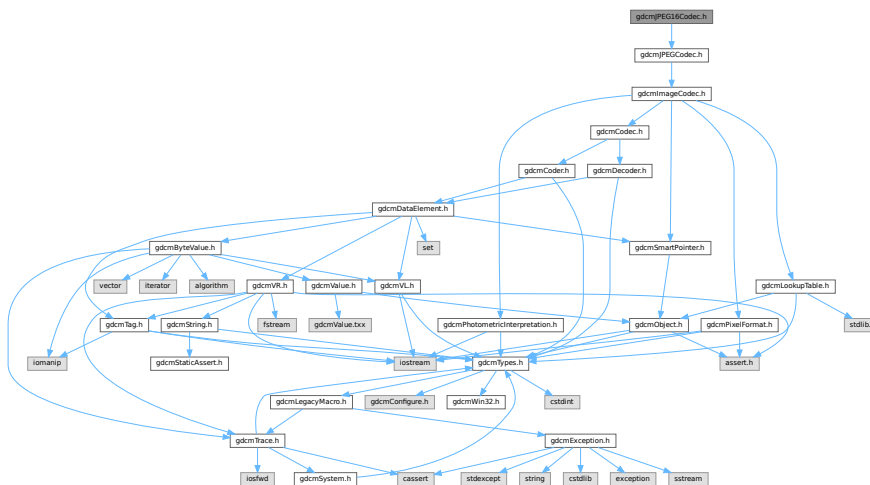
00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMIPPSORTER_H
00015  #define GDCMIPPSORTER_H
00016
00017  #include "gdcmlSorter.h"
00018
00019  #include <vector>
00020  #include <string>
00021
00022  namespace gdcml
00023  {
00024  class GDCM_EXPORT IPPSorter : public Sorter
00025  {
00026  public:
00027      IPPSorter();
00028
00029      // FIXME: I do not like public virtual function...
00030      bool Sort(std::vector<std::string> const & filenames) override;
00031
00032      void SetComputeZSpacing(bool b) { ComputeZSpacing = b; }
00033      void SetZSpacingTolerance(double tol) { ZTolerance = tol; }
00034      double GetZSpacingTolerance() const { return ZTolerance; }
00035
00036      void SetDirectionCosinesTolerance(double tol) { DirCosTolerance = tol; }
00037      double GetDirectionCosinesTolerance() const { return DirCosTolerance; }
00038
00039      void SetDropDuplicatePositions(bool b) { DropDuplicatePositions = b; }
00040
00041      double GetZSpacing() const { return ZSpacing; }
00042
00043  protected:
00044      bool ComputeZSpacing;
00045      bool DropDuplicatePositions;
00046      double ZSpacing;
00047      double ZTolerance;
00048      double DirCosTolerance;
00049
00050  private:
00051      GDCM_LEGACY(bool ComputeSpacing(std::vector<std::string> const & filenames))
00052      {};
00053
00054  } // end namespace gdcml
00055
00056  #endif //GDCMIPPSORTER_H

```



## 11.343 gdcMJPEG16Codec.h File Reference

Include dependency graph for qdcmJPEG16Codec.h:



- class `gdcm::JPEG16Codec`  
*Class to do JPEG 16bits (lossless)*

- namespace **gdcm**

## 11.344 gdcmJPEG16Codec.h

[Go to the documentation of this file.](#)

```
00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMJPEG16CODEC_H
00015 #define GDCMJPEG16CODEC_H
00016
00017 #include "gdcmJPEGCodec.h"
00018
00019 namespace gdcm
00020 {
00021
00022   class JPEGInternals_16BIT;
00023   class ByteValue;
00024
00025   class JPEG16Codec : public JPEGCodec
00026   {
00027   public:
00028     JPEG16Codec();
00029     ~JPEG16Codec() override;
00030
00031     bool DecodeByStreams(std::istream &is, std::ostream &os) override;
00032     bool InternalCode(const char *input, unsigned long len, std::ostream &os) override;
00033
00034     bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
00035
00036   protected:
00037     bool IsStateSuspension() const override;
00038     bool EncodeBuffer(std::ostream &os, const char *data, size_t datalen) override;
00039
00040   private:
00041     JPEGInternals_16BIT *Internals;
00042   };
00043
00044 } // end namespace gdcm
00045
00046 #endif //GDCMJPEG16CODEC_H
```



```

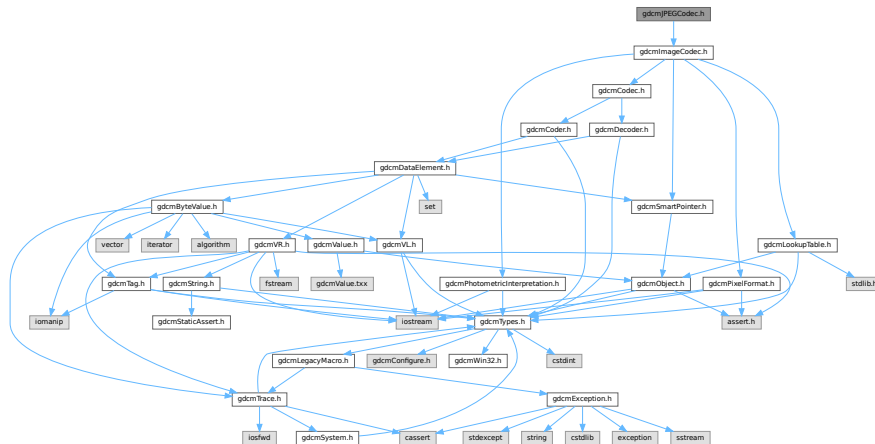
00021
00022 class JPEG2000Internals;
00030 class GDCM_EXPORT JPEG2000Codec : public ImageCodec
00031 {
00032 friend class ImageRegionReader;
00033 friend class Bitmap;
00034 public:
00035     JPEG2000Codec();
00036     ~JPEG2000Codec() override;
00037
00038     bool CanDecode(TransferSyntax const &ts) const override;
00039     bool CanCode(TransferSyntax const &ts) const override;
00040
00041     bool Decode(DataElement const &is, DataElement &os) override;
00042     bool Code(DataElement const &in, DataElement &out) override;
00043
00044     bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
00045     ImageCodec * Clone() const override;
00046
00047     // JPEG-2000 / OpenJPEG specific way of encoding lossy-ness
00048     // ref: http://www.openjpeg.org/index.php?menu=doc#encoder
00049     void SetRate(unsigned int idx, double rate);
00050     double GetRate(unsigned int idx = 0) const;
00051
00052     void SetQuality(unsigned int idx, double q);
00053     double GetQuality(unsigned int idx = 0) const;
00054
00055     void SetTileSize(unsigned int tx, unsigned int ty);
00056
00057     void SetNumberOfResolutions(unsigned int nres);
00058
00061     void SetNumberOfThreadsForDecompression(int nThreads);
00062
00063     void SetReversible(bool res);
00064     void SetMCT(unsigned int mct);
00065
00066 protected:
00067     bool DecodeExtent(
00068         char *buffer,
00069         unsigned int xmin, unsigned int xmax,
00070         unsigned int ymin, unsigned int ymax,
00071         unsigned int zmin, unsigned int zmax,
00072         std::istream & is
00073     );
00074
00075     bool DecodeByStreams(std::istream &is, std::ostream &os) override;
00076
00077     bool StartEncode( std::ostream & ) override;
00078     bool IsRowEncoder() override;
00079     bool IsFrameEncoder() override;
00080     bool AppendRowEncode( std::ostream & out, const char * data, size_t datalen ) override;
00081     bool AppendFrameEncode( std::ostream & out, const char * data, size_t datalen ) override;
00082     bool StopEncode( std::ostream & ) override;
00083
00084 private:
00085     std::pair<char *, size_t> DecodeByStreamsCommon(char *dummy_buffer, size_t buf_size);
00086     bool CodeFrameIntoBuffer(char * outdata, size_t outlen, size_t & complen, const char * indata, size_t
inlen );
00087     bool GetHeaderInfo(const char * dummy_buffer, size_t len, TransferSyntax &ts);
00088     JPEG2000Internals *Internals;
00089 };
00090
00091 } // end namespace gdcms
00092
00093 #endif //GDCMJPEG2000CODEC_H

```



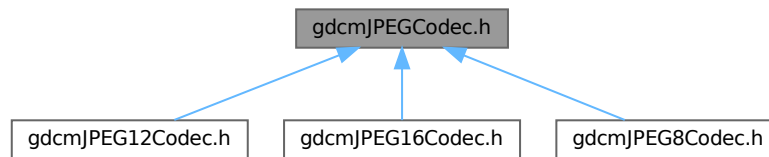
## 11.349 gdcmJPEGCodec.h File Reference

Include dependency graph for gdcMJPEGCodec.h:





This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcMJPEGCodec](#)  
*JPEG codec.*

## Namespaces

- namespace [gdcM](#)

## 11.350 gdcMJPEGCodec.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcM.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMJPEGCODEC_H
00015 #define GDCMJPEGCODEC_H
00016
00017 #include "gdcMImageCodec.h"
00018
00019 namespace gdcM
00020 {
00021
00022     class PixelFormat;
00023     class TransferSyntax;
00040     class GDCM_EXPORT JPEGCodec : public ImageCodec
00041     {
00042     friend class ImageRegionReader;
00043     public:
00044         JPEGCodec();
00045         ~JPEGCodec() override;
00046         bool CanDecode(TransferSyntax const &ts) const override;
00047         bool CanCode(TransferSyntax const &ts) const override;
00048         bool Decode(DataElement const &is, DataElement &os) override;
00049         void SetPixelFormat(PixelFormat const &pf) override;
00050
00052         void ComputeOffsetTable(bool b);
  
```

```

00053
00055     bool Code(DataElement const &in, DataElement &out) override;
00056
00057     bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
00058     ImageCodec * Clone() const override;
00059
00060     //void SetReversible(bool res);
00061
00062     void SetQuality(double q);
00063     double GetQuality() const;
00064
00065     void SetLossless(bool l);
00066     bool GetLossless() const;
00067
00068     virtual bool EncodeBuffer( std::ostream & out,
00069         const char *inbuffer, size_t inlen);
00070
00071 protected:
00072     bool DecodeExtent(
00073         char *buffer,
00074         unsigned int xmin, unsigned int xmax,
00075         unsigned int ymin, unsigned int ymax,
00076         unsigned int zmin, unsigned int zmax,
00077         std::istream & is
00078     );
00079
00080     bool DecodeByStreams(std::istream &is, std::ostream &os) override;
00081     bool IsValid(PhotometricInterpretation const &pi) override;
00082
00083     bool StartEncode( std::ostream & ) override;
00084     bool IsRowEncoder() override;
00085     bool IsFrameEncoder() override;
00086     bool AppendRowEncode( std::ostream & out, const char * data, size_t datalen ) override;
00087     bool AppendFrameEncode( std::ostream & out, const char * data, size_t datalen ) override;
00088     bool StopEncode( std::ostream & ) override;
00089
00090 protected:
00091     // Internal method called by SetPixelFormat
00092     // Instantiate the right jpeg codec (8, 12 or 16)
00093     void SetBitSample(int bit);
00094
00095     virtual bool IsStateSuspension() const;
00096
00097 protected:
00098     int BitSample;
00099     //bool Lossless;
00100     int Quality;
00101
00102 private:
00103     void SetupJPEGBitCodec(int bit);
00104     JPEGCodec *Internal;
00105 };
00106
00107 } // end namespace gdcm
00108
00109 #endif //GDCMJPEGCODEC_H

```



```

00021
00022 class JPEGGLSInternals;
00030 class GDCM_EXPORT JPEGGLSCodec : public ImageCodec
00031 {
00032 friend class ImageRegionReader;
00033 public:
00034     JPEGGLSCodec();
00035     ~JPEGGLSCodec() override;
00036     bool CanDecode(TransferSyntax const &ts) const override;
00037     bool CanCode(TransferSyntax const &ts) const override;
00038
00039     unsigned long GetBufferLength() const { return BufferLength; }
00040     void SetBufferLength(unsigned long l) { BufferLength = l; }
00041
00042     bool Decode(DataElement const &is, DataElement &os) override;
00043     bool Decode(DataElement const &in, char* outBuffer, size_t inBufferLength,
00044                 uint32_t inXMin, uint32_t inXMax, uint32_t inYMin,
00045                 uint32_t inYMax, uint32_t inZMin, uint32_t inZMax);
00046     bool Code(DataElement const &in, DataElement &out) override;
00047
00048     bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
00049     ImageCodec * Clone() const override;
00050
00051     void SetLossless(bool l);
00052     bool GetLossless() const;
00053
00054     /*
00055     * test.acr can look pretty bad, even with a lossy error of 2. Explanation follows:
00056     * I agree that the test image looks ugly. In this particular case I can
00057     * explain though.
00058     *
00059     * The image is 8 bit, but it does not use the full 8 bit dynamic range. The
00060     * black pixels have value 234 and the white 255. If you set allowed lossy
00061     * error to 2, you allow an error of about 10% of the actual dynamic range.
00062     * That is of course very visible.
00063     */
00065     void SetLossyError(int error);
00066
00067 protected:
00068     bool DecodeExtent(
00069         char *buffer,
00070         unsigned int xmin, unsigned int xmax,
00071         unsigned int ymin, unsigned int ymax,
00072         unsigned int zmin, unsigned int zmax,
00073         std::istream & is
00074     );
00075
00076     bool StartEncode( std::ostream & ) override;
00077     bool IsRowEncoder() override;
00078     bool IsFrameEncoder() override;
00079     bool AppendRowEncode( std::ostream & out, const char * data, size_t datalen ) override;
00080     bool AppendFrameEncode( std::ostream & out, const char * data, size_t datalen ) override;
00081     bool StopEncode( std::ostream & ) override;
00082
00083 private:
00084     bool DecodeByStreamsCommon(const char *buffer, size_t totalLen, std::vector<unsigned char> &rgbyteOut);
00085     bool CodeFrameIntoBuffer(char * outdata, size_t outlen, size_t & complen, const char * indata, size_t
inlen );
00086
00087     unsigned long BufferLength;
00088     int LossyError;
00089 };
00090
00091 } // end namespace gdcm
00092
00093 #endif //GDCMJPEGGLSCODEC_H

```

## 11.353 gdcmJSON.h File Reference

```

#include "gdcmFile.h"
#include "gdcmDataElement.h"

```



```

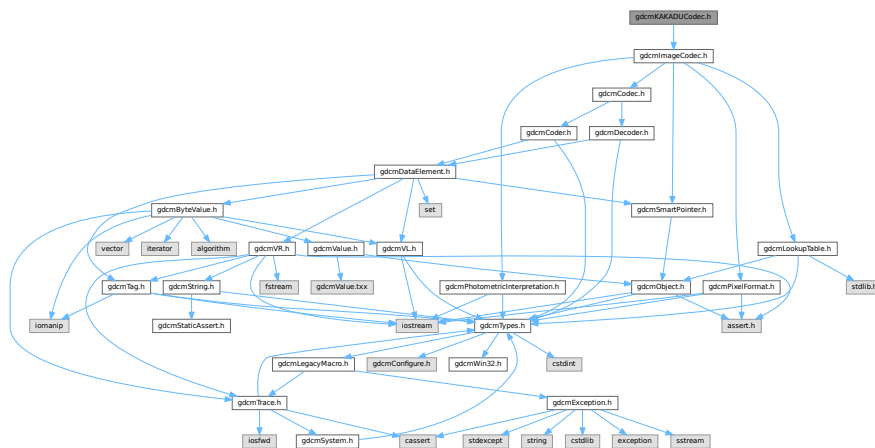
00030 {
00031 public:
00032     JSON();
00033     ~JSON();
00034
00035     bool GetPrettyPrint() const;
00036     void SetPrettyPrint(bool onoff);
00037     void PrettyPrintOn();
00038     void PrettyPrintOff();
00039
00040     bool Code(DataSet const & in, std::ostream & os);
00041     bool Decode(std::istream & is, DataSet & out);
00042
00043 private:
00044     JSONInternal *Internals;
00045 };
00046
00047 } // end namespace gdcm
00048
00049 #endif //GDCMXMLPRINTER_H

```

## 11.355 gdcmKAKADUCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmKAKADUCodec.h:



### Classes

- class [gdcm::KAKADUCodec](#)  
*KAKADUCodec.*

### Namespaces

- namespace [gdcm](#)

## 11.356 gdcmKAKADUCodec.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMKAKADUCODEC_H
00015  #define GDCMKAKADUCODEC_H
00016
00017  #include "gdcmImageCodec.h"
00018
00019  namespace gdcm
00020  {
00021
00022  class KAKADUCodec : public ImageCodec
00023  {
00024  public:
00025    KAKADUCodec();
00026    ~KAKADUCodec() override;
00027    bool CanDecode(TransferSyntax const &ts) const override;
00028    bool CanCode(TransferSyntax const &ts) const override;
00029
00030    bool Decode(DataElement const &is, DataElement &os) override;
00031    bool Code(DataElement const &in, DataElement &out) override;
00032
00033    ImageCodec * Clone() const override;
00034  private:
00035  };
00036
00037  } // end namespace gdcm
00038
00039  #endif //GDCMKAKADUCODEC_H

```

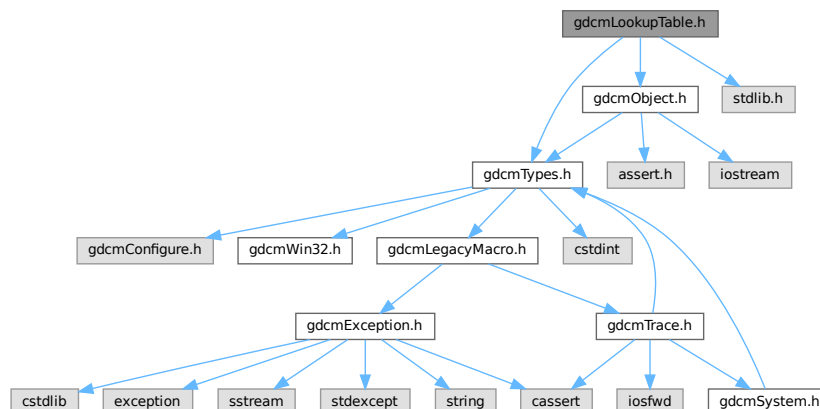
## 11.357 gdcmLookupTable.h File Reference

```

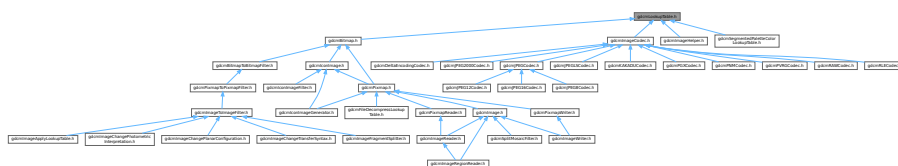
#include "gdcmTypes.h"
#include "gdcmObject.h"
#include <stdlib.h>

```

Include dependency graph for `gdcmLookupTable.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::LookupTable`  
*LookupTable* class.

## Namespaces

- namespace `gdcm`

## 11.358 gdcmLookupTable.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even

```



```

00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014
00015 #ifndef GDCMLOOKUPTABLE_H
00016 #define GDCMLOOKUPTABLE_H
00017
00018 #include "gdcmTypes.h"
00019 #include "gdcmObject.h"
00020 #include <stdlib.h>
00021
00022 namespace gdcm
00023 {
00024
00025 class LookupTableInternal;
00029 class GDCM_EXPORT LookupTable : public Object
00030 {
00031 public:
00032     typedef enum {
00033         RED = 0, // Keep RED == 0
00034         GREEN,
00035         BLUE,
00036         GRAY,
00037         UNKNOWN
00038     } LookupTableType;
00039
00040     LookupTable();
00041     ~LookupTable() override;
00042     void Print(std::ostream &) const override;
00043
00044     void Allocate( unsigned short bitsample = 8 );
00045     //TODO: check to see if length should be unsigned short, unsigned int, or whatever
00046     void InitializeLUT(LookupTableType type, unsigned short length,
00047         unsigned short subscript, unsigned short bitsize);
00048     unsigned int GetLUTLength(LookupTableType type) const;
00049     virtual void SetLUT(LookupTableType type, const unsigned char *array,
00050         unsigned int length);
00051     void GetLUT(LookupTableType type, unsigned char *array, unsigned int &length) const;
00052     void GetLUTDescriptor(LookupTableType type, unsigned short &length,
00053         unsigned short &subscript, unsigned short &bitsize) const;
00054
00055     void InitializeRedLUT(unsigned short length, unsigned short subscript,
00056         unsigned short bitsize);
00057     void SetRedLUT(const unsigned char *red, unsigned int length);
00058     void InitializeGreenLUT(unsigned short length, unsigned short subscript,
00059         unsigned short bitsize);
00060     void SetGreenLUT(const unsigned char *green, unsigned int length);
00061     void InitializeBlueLUT(unsigned short length, unsigned short subscript,
00062         unsigned short bitsize);
00063     void SetBlueLUT(const unsigned char *blue, unsigned int length);
00064
00065     void Clear();
00066
00067     void Decode(std::istream &is, std::ostream &os) const;
00068
00069     bool Decode(char *outputbuffer, size_t outlen, const char *inputbuffer, size_t inlen) const;
00070
00071     bool IsRGB8() const;
00072
00073     bool Decode8(char *outputbuffer, size_t outlen, const char *inputbuffer, size_t inlen) const;
00074
00075     LookupTable(LookupTable const &lut):Object(lut)
00076     {
00077         assert(0);
00078     }
00079
00080     bool GetBufferAsRGBA(unsigned char *rgba) const;
00081
00082     const unsigned char *GetPointer() const;
00083
00084     bool WriteBufferAsRGBA(const unsigned char *rgba);
00085
00086     unsigned short GetBitSample() const { return BitSample; }
00087
00088     bool Initialized() const;
00089
00090 private:
00091     void Encode(std::istream &is, std::ostream &os);
00092
00093 protected:

```

```

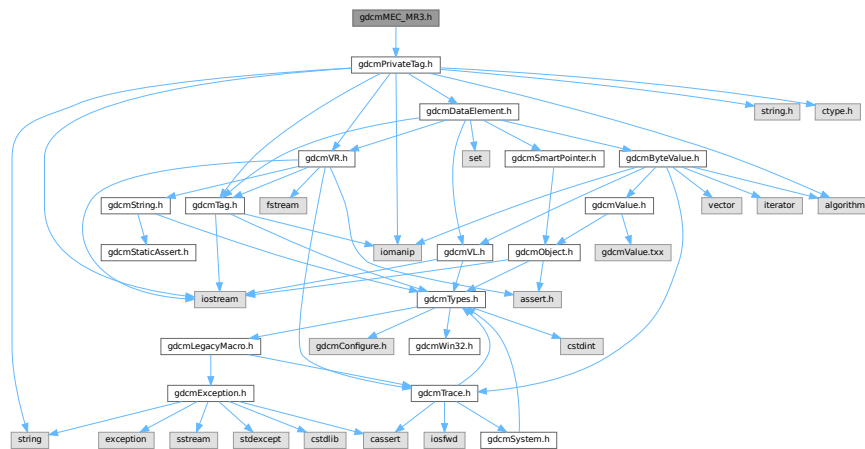
00110     LookupTableInternal *Internal;
00111     unsigned short BitSample; // refer to the pixel type (not the bit size of LUT)
00112     bool IncompleteLUT:1;
00113 };
00114
00115 } // end namespace gdcm
00116
00117 #endif //GDCMLOOKUPTABLE_H

```

## 11.359 gdcmMEC\_MR3.h File Reference

```
#include "gdcmPrivateTag.h"
```

Include dependency graph for gdcmMEC\_MR3.h:



### Classes

- class [gdcm::MEC\\_MR3](#)  
Class for [MEC\\_MR3](#).

### Namespaces

- namespace [gdcm](#)

## 11.360 gdcmMEC\_MR3.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

```

```

00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 #ifndef GDCMMEC_MR3_H
00015 #define GDCMMEC_MR3_H
00016
00017 #include "gdcmPrivateTag.h"
00018
00019 namespace gdcm {
00024 class GDCM_EXPORT MEC_MR3 {
00025 public:
00026     static bool Print(const char *src, size_t srclen);
00027
00030     static const PrivateTag &GetPMTFInformationDataTag();
00031
00034     static const PrivateTag &GetCanonMECMR3Tag();
00035
00038     static const PrivateTag &GetToshibaMECMR3Tag();
00039 };
00040
00041 } // end namespace gdcm
00042
00043 #endif // GDCMMEC_MR3_H

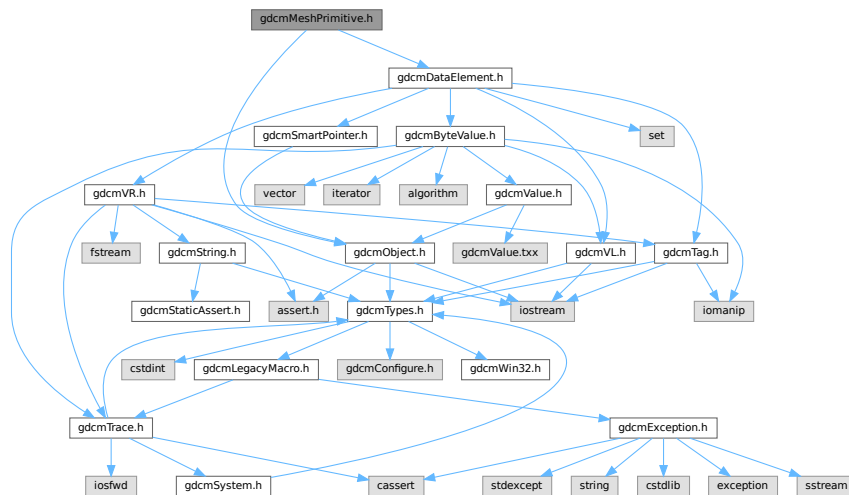
```

## 11.361 gdcmMeshPrimitive.h File Reference

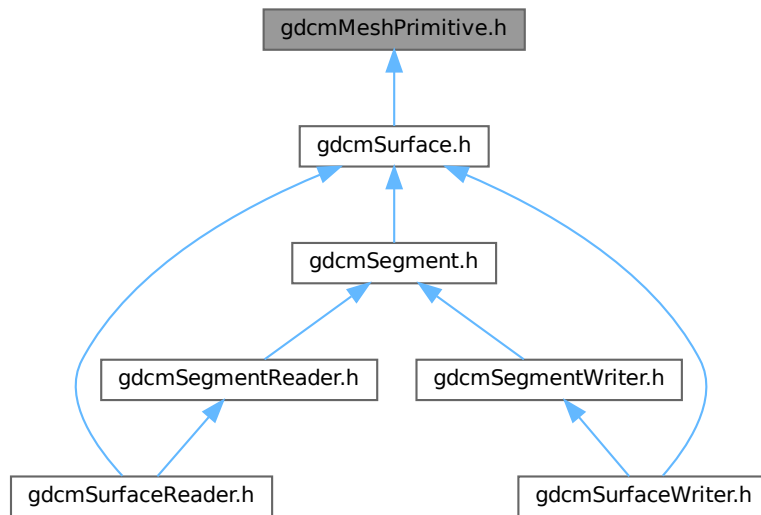
```
#include <gdcmObject.h>
```

```
#include <gdcmDataElement.h>
```

Include dependency graph for gdcmMeshPrimitive.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::MeshPrimitive](#)  
*This class defines surface mesh primitives.*

## Namespaces

- namespace [gdcm](#)

## 11.362 gdcmMeshPrimitive.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014
00015 #ifndef GDCMMESHPRIMITIVE_H
00016 #define GDCMMESHPRIMITIVE_H
00017
00018 #include <gdcmObject.h>

```

```

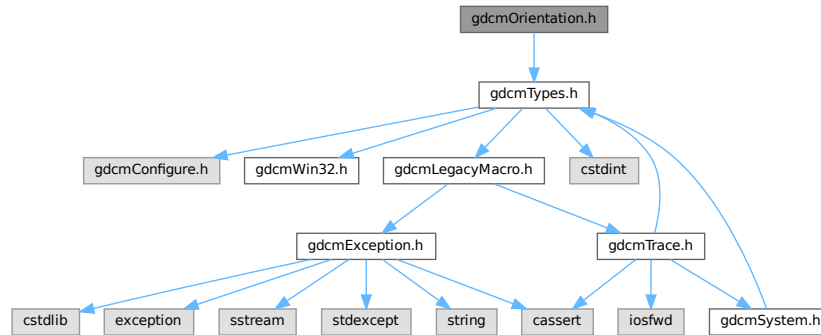
00019 #include <gdcmDataElement.h>
00020
00021 namespace gdcm
00022 {
00023
00030 class GDCM_EXPORT MeshPrimitive : public Object
00031 {
00032 public:
00033
00034     typedef std::vector< DataElement > PrimitivesData;
00035
00041     typedef enum {
00042         VERTEX = 0,
00043         EDGE,
00044         TRIANGLE,
00045         TRIANGLE_STRIP,
00046         TRIANGLE_FAN,
00047         LINE,
00048         FACET,
00049         MPType_END
00050     } MPType;
00051
00052     static const char * GetMPTypeString(const MPType type);
00053
00054     static MPType GetMPType(const char * type);
00055
00056     MeshPrimitive();
00057
00058     ~MeshPrimitive() override;
00059
00060     MPType GetPrimitiveType() const;
00061     void SetPrimitiveType(const MPType type);
00062
00063     const DataElement & GetPrimitiveData() const;
00064     DataElement & GetPrimitiveData();
00065     void SetPrimitiveData(DataElement const & de);
00066
00067     const PrimitivesData & GetPrimitivesData() const;
00068     PrimitivesData & GetPrimitivesData();
00069     void SetPrimitivesData(PrimitivesData const & DEs);
00070
00071     const DataElement & GetPrimitiveData(const unsigned int idx) const;
00072     DataElement & GetPrimitiveData(const unsigned int idx);
00073     void SetPrimitiveData(const unsigned int idx, DataElement const & de);
00074     void AddPrimitiveData(DataElement const & de);
00075
00076     unsigned int GetNumberOfPrimitivesData() const;
00077
00078 protected:
00079
00080     // Use to define tag where PrimitiveData will be put.
00081     MPType PrimitiveType;
00082
00083     // PrimitiveData contains point index list.
00084     // It shall have 1 or 1-n DataElement following PrimitiveType.
00085     PrimitivesData PrimitiveData;
00086 };
00087
00088 }
00089
00090 #endif // GDCMMESHPRIMITIVE_H

```

## 11.363 gdcmOrientation.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmOrientation.h:



### Classes

- class [gdcm::Orientation](#)  
*class to handle [Orientation](#)*

### Namespaces

- namespace [gdcm](#)

### Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Orientation &o)`

## 11.364 gdcmOrientation.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMORIENTATION_H
00015 #define GDCMORIENTATION_H

```

```

00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021
00025 class GDCM_EXPORT Orientation
00026 {
00027     friend std::ostream& operator<<(std::ostream &_os, const Orientation &o);
00028 public:
00029     Orientation();
00030     ~Orientation() = default;
00031
00033     void Print(std::ostream &) const;
00034
00035     typedef enum {
00036         UNKNOWN,
00037         AXIAL,
00038         CORONAL,
00039         SAGITTAL,
00040         OBLIQUE
00041     } OrientationType;
00042
00045     static OrientationType GetType(const double dircos[6]);
00046
00048     static void SetObliquityThresholdCosineValue(double val);
00049     static double GetObliquityThresholdCosineValue();
00050
00052     static const char *GetLabel(OrientationType type);
00053
00054 protected:
00055     static char GetMajorAxisFromPatientRelativeDirectionCosine(double x, double y, double z);
00056
00057 private:
00058     static double ObliquityThresholdCosineValue;
00059 };
00060 //-----
00061 inline std::ostream& operator<<(std::ostream &os, const Orientation &o)
00062 {
00063     o.Print( os );
00064     return os;
00065 }
00066
00067 } // end namespace gdcm
00068
00069 #endif //GDCMORIENTATION_H

```

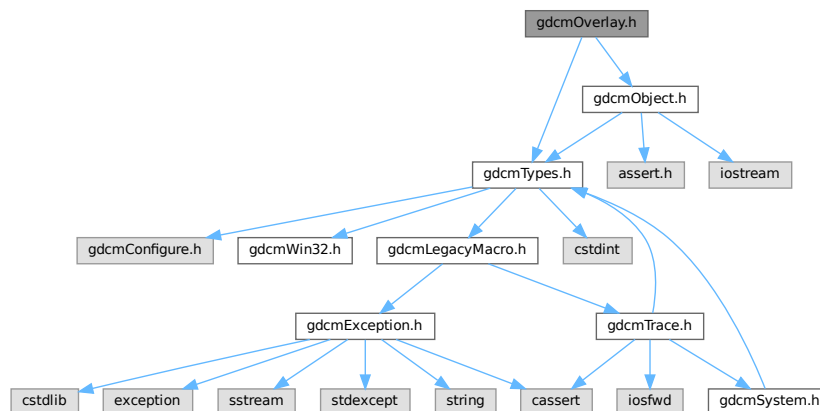
## 11.365 gdcmOverlay.h File Reference

```

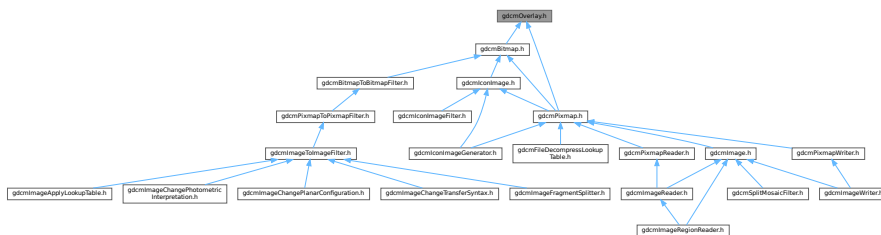
#include "gdcmTypes.h"
#include "gdcmObject.h"

```

Include dependency graph for `gdcmOverlay.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::Overlay`  
*Overlay* class.

## Namespaces

- namespace `gdcm`

## 11.366 gdcmOverlay.h

[Go to the documentation of this file.](#)

```

00001  /*****
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre

```



```

00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMOVERLAY_H
00015 #define GDCMOVERLAY_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmObject.h"
00019
00020 namespace gdcm
00021 {
00022
00023 class OverlayInternal;
00024 class ByteValue;
00025 class DataSet;
00026 class DataElement;
00038 class GDCM_EXPORT Overlay : public Object
00039 {
00040 public:
00041 Overlay();
00042 ~Overlay() override;
00044 void Print(std::ostream &) const override;
00045
00047 void Update(const DataElement & de);
00048
00050 void SetGroup(unsigned short group);
00052 unsigned short GetGroup() const;
00054 void SetRows(unsigned short rows);
00056 unsigned short GetRows() const;
00058 void SetColumns(unsigned short columns);
00060 unsigned short GetColumns() const;
00062 void SetNumberOfFrames(unsigned int numberofframes);
00064 void SetDescription(const char* description);
00066 const char *GetDescription() const;
00067 typedef enum {
00068 Invalid = 0,
00069 Graphics = 1,
00070 ROI = 2
00071 } OverlayType;
00073 void SetType(const char* type);
00075 const char *GetType() const;
00076 OverlayType GetTypeAsEnum() const;
00077 static const char *GetOverlayTypeAsString(OverlayType ot);
00078 static OverlayType GetOverlayTypeFromString(const char *);
00080 void SetOrigin(const signed short origin[2]);
00082 const signed short * GetOrigin() const;
00084 void SetFrameOrigin(unsigned short frameorigin);
00086 void SetBitsAllocated(unsigned short bitsallocated);
00088 unsigned short GetBitsAllocated() const;
00090 void SetBitPosition(unsigned short bitposition);
00092 unsigned short GetBitPosition() const;
00093
00095 void SetOverlay(const char *array, size_t length);
00097 bool GrabOverlayFromPixelData(DataSet const &ds);
00098
00101 const ByteValue &GetOverlayData() const;
00102
00104 bool IsEmpty() const;
00105
00107 bool IsZero() const;
00108
00110 bool IsInPixelData() const;
00111
00113 void IsInPixelData(bool b);
00114
00116 void Decompress(std::ostream &os) const;
00117
00120 size_t GetUnpackBufferLength() const;
00121
00124 bool GetUnpackBuffer(char *buffer, size_t len) const;
00125
00126 Overlay(Overlay const &ov);
00127 Overlay &operator=(Overlay const &ov);
00128
00129 private:
00130 OverlayInternal *Internal;

```

```

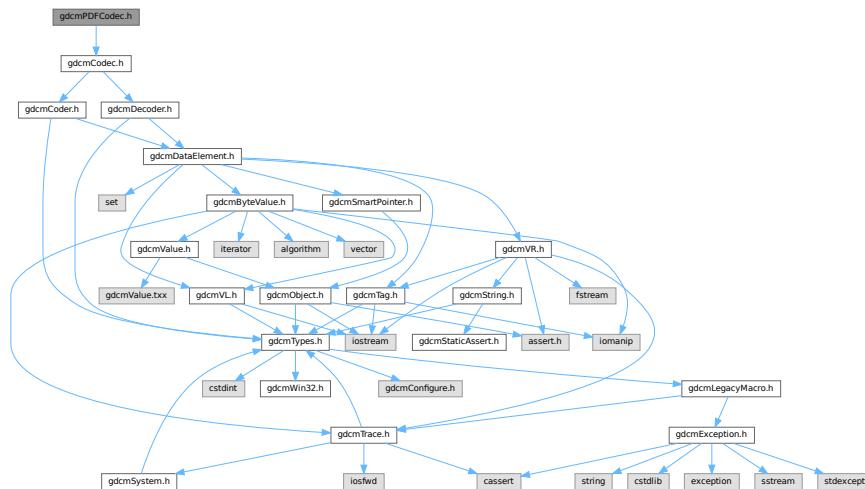
00131 };
00132
00133 } // end namespace gdcm
00134
00135 #endif //GDCMOVERLAY_H

```

## 11.367 gdcmPDFCodec.h File Reference

```
#include "gdcmCodec.h"
```

Include dependency graph for gdcmPDFCodec.h:



### Classes

- class [gdcm::PDFCodec](#)  
*PDFCodec* class.

### Namespaces

- namespace [gdcm](#)

## 11.368 gdcmPDFCodec.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008

```

```

00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE. See the above copyright notice for more information.
00012
00013      =====*/
00014      #ifndef GDCMPDFCODEC_H
00015      #define GDCMPDFCODEC_H
00016
00017      #include "gdcmCodec.h"
00018
00019      namespace gdcm
00020      {
00021
00022      class GDCM_EXPORT PDFCodec : public Codec
00023      {
00024      public:
00025          PDFCodec();
00026          ~PDFCodec() override;
00027          bool CanCode(TransferSyntax const &) const override { return false; }
00028          bool CanDecode(TransferSyntax const &) const override { return false; }
00029          bool Decode(DataElement const &is, DataElement &os) override;
00030      };
00031
00032      } // end namespace gdcm
00033
00034      #endif //GDCMPDFCODEC_H

```

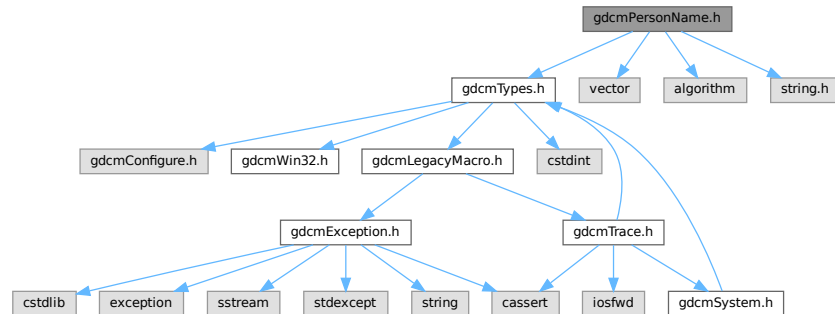
## 11.369 gdcmPersonName.h File Reference

```

#include "gdcmTypes.h"
#include <vector>
#include <algorithm>
#include <string.h>

```

Include dependency graph for gdcmPersonName.h:



### Classes

- class [gdcm::PersonName](#)  
*PersonName* class.

### Namespaces

- namespace [gdcm](#)

## 11.370 gdcmPersonName.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014
00015 #ifndef GDCMPERSONNAME_H
00016 #define GDCMPERSONNAME_H
00017
00018 #include "gdcmTypes.h"
00019 #include <vector>
00020 #include <algorithm> // std::min
00021 #include <string.h> // strlen
00022
00023 namespace gdcm
00024 {
00025
00026   class GDCM_EXPORT PersonName
00027   {
00028   public:
00029     static const unsigned int MaxNumberOfComponents = 5;
00030     static const unsigned int MaxLength = 64;
00031     char Component[MaxNumberOfComponents][MaxLength+1];
00032     static const char Separator = '^';
00033     static const char Padding = ' ';
00034
00035     unsigned int GetNumberOfComponents() const {
00036       unsigned int r = 0;
00037       for(unsigned int i = 0; i < 5; ++i) {
00038         if( *Component[i] != '\0' ) r = i;
00039       }
00040       return r+1;
00041     }
00042
00043     unsigned int GetMaxLength() const { return MaxLength; }
00044     void SetBlob(const std::vector<char>& v) {
00045       (void)v;
00046       //assert(0); //TODO
00047     }
00048     void SetComponents(const char *comp1 = "",
00049                       const char *comp2 = "",
00050                       const char *comp3 = "",
00051                       const char *comp4 = "",
00052                       const char *comp5 = "") {
00053       const char *components[5] = { comp1, comp2, comp3, comp4, comp5 };
00054       SetComponents( components );
00055     }
00056     void SetComponents(const char *components[]) {
00057       if( components )
00058         for(unsigned int i = 0; i < 5; ++i) {
00059           if( components[i] && strlen(components[i]) < GetMaxLength() )
00060             strcpy(Component[i], components[i]);
00061           assert( strlen(Component[i]) < GetMaxLength() );
00062         }
00063     }
00064
00065     void Print(std::ostream &os) const
00066     {
00067       //os << "Family Name Complex: " << Component[0] << std::endl;
00068       //os << "Given Name Complex: " << Component[1] << std::endl;
00069       //os << "Middle Name      : " << Component[2] << std::endl;
00070       //os << "Name Suffix       : " << Component[3] << std::endl;
00071       //os << "Name Prefix        : " << Component[4] << std::endl;
00072       os << Component[0] << '^';
00073       os << Component[1] << '^';
00074       os << Component[2] << '^';
00075       os << Component[3] << '^';
00076       os << Component[4];
00077     }
00078   };
00079 };

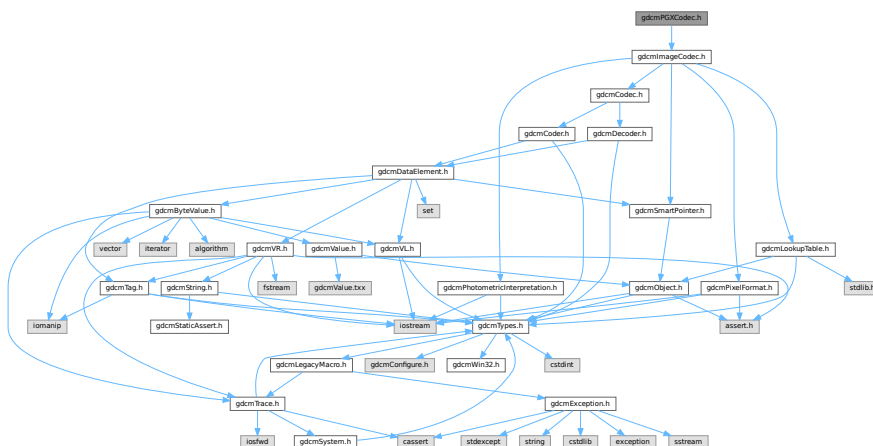
```

```
00080
00081 } // end namespace gdcmm
00082
00083 #endif //GDCMPERSONNAME_H
```

## 11.371 gdcMPEGXCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcnP GXCodec.h:



## Classes

- class `gdcm::PGXCodec`  
*Class to do PGX.*

## Namespaces

- namespace **gdcm**

## 11.372 gdcmPGXCodec.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012

```

```

00013 =====*/
00014 #ifndef GDCMPGXCODEC_H
00015 #define GDCMPGXCODEC_H
00016
00017 #include "gdcmImageCodec.h"
00018
00019 namespace gdcm
00020 {
00021
00026 class GDCM_EXPORT PGXCodec : public ImageCodec
00027 {
00028 public:
00029     PGXCodec();
00030     ~PGXCodec() override;
00031     bool CanDecode(TransferSyntax const &ts) const override;
00032     bool CanCode(TransferSyntax const &ts) const override;
00033
00034     bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
00035     ImageCodec * Clone() const override;
00036
00037     bool Read(const char *filename, DataElement &out) const;
00038     bool Write(const char *filename, const DataElement &out) const;
00039 private:
00040 };
00041
00042 } // end namespace gdcm
00043
00044 #endif //GDCMPGXCODEC_H

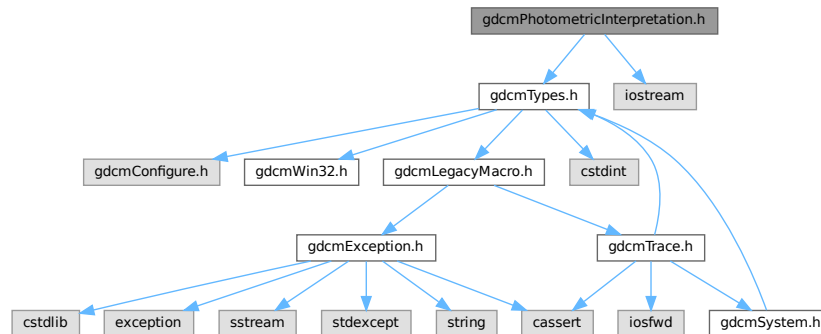
```

## 11.373 gdcmPhotometricInterpretation.h File Reference

```
#include "gdcmTypes.h"
```

```
#include <iostream>
```

Include dependency graph for gdcmPhotometricInterpretation.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::PhotometricInterpretation](#)  
Class to represent an *PhotometricInterpretation*.

## Namespaces

- namespace [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const PhotometricInterpretation &val)`

## 11.374 gdcmPhotometricInterpretation.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014
00015 #ifndef GDCMPHOTOMETRICINTERPRETATION_H
00016 #define GDCMPHOTOMETRICINTERPRETATION_H
00017
00018 #include "gdcmTypes.h"
00019 #include <iostream>
00020
00021 namespace gdcm
00022 {
00023
00024     class TransferSyntax;
00028     class GDCM_EXPORT PhotometricInterpretation
00029     {
00030     public:
00031         typedef enum {
00032             UNKNOWN = 0,
00033             MONOCHROME1,
00034             MONOCHROME2,
00035             PALETTE_COLOR,
00036             RGB,
00037             HSV,
00038             ARGB, // retired
00039             CMYK,
00040             YBR_FULL,
00041             YBR_FULL_422,
00042             YBR_PARTIAL_422,
00043             YBR_PARTIAL_420,
00044             YBR_ICT,
00045             YBR_RCT,
00046             // PALETTE_COLOR ?
00047             // MONOCHROME = MONOCHROME1 | MONOCHROME2,
00048             // COLOR = RGB | HSV | ARGB | CMYK | YBR_FULL | YBR_FULL_422 | YBR_PARTIAL_422 | YBR_PARTIAL_420 |
00049             YBR_ICT | YBR_RCT,
00049             PI_END // Helpful for internal implementation
00050         } PType; // PhotometricInterpretationType
00051
00052         PhotometricInterpretation(PType pi = UNKNOWN):PIField(pi) {}

```

```

00053
00054     static const char *GetPIString(PIType pi);
00055
00056     const char *GetString() const;
00057
00058     // You need to make sure end of string is \0
00059     static PIType GetPIType(const char *pi);
00060
00061     static bool IsRetired(PIType pi);
00062
00063     bool IsLossy() const;
00064     bool IsLossless() const;
00065
00066     unsigned short GetSamplesPerPixel() const;
00067
00068
00069     // TODO
00070     // not all PhotometricInterpretation are allowed for compressed Transfer
00071     // syntax
00072     // static bool IsAllowedForCompressedTS(PIType pi);
00073
00074     friend std::ostream& operator<<(std::ostream& os, const PhotometricInterpretation& pi);
00075
00076     operator PIType () const { return PIField; }
00077
00078     PIType GetType () const { return PIField; }
00079
00080     // Will return whether current PhotometricInterpretation is the same Color Space as input:
00081     // eg. RGB and YBR_RCT are
00082     bool IsSameColorSpace( PhotometricInterpretation const &pi ) const;
00083
00084     //static PIType GetEquivalent(TransferSyntax const &ts);
00085
00086 private:
00087     PIType PIField;
00088 };
00089 //-----
00090 inline std::ostream& operator<<(std::ostream& os, const PhotometricInterpretation &val)
00091 {
00092     const char *s = PhotometricInterpretation::GetPIString(val.PIField);
00093     os << (s ? s : "");
00094     return os;
00095 }
00096
00097
00098 } // end namespace gdcm
00099
00100 #endif //GDCMPHOTOMETRICINTERPRETATION_H

```

## 11.375 gdcmPixelFormat.h File Reference

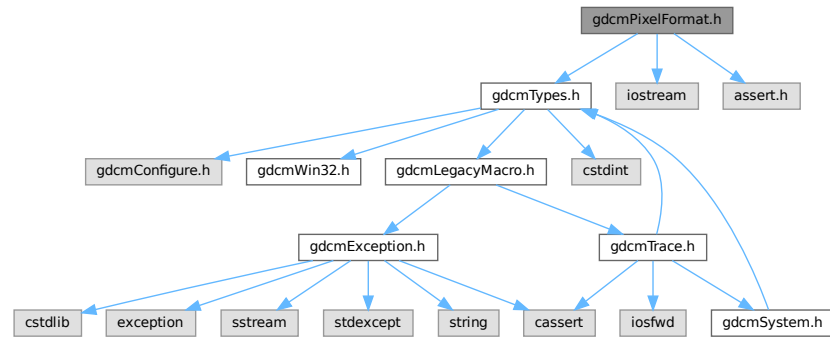
```

#include "gdcmTypes.h"
#include <iostream>
#include <assert.h>

```



Include dependency graph for gdcmPixelFormat.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::PixelFormat`  
*PixelFormat.*

## Namespaces

- namespace **gdcm**

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const PixelFormat &pf)`

## 11.376 gdcmPixelFormat.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014
00015  #ifndef GDCMPIXELFORMAT_H
00016  #define GDCMPIXELFORMAT_H
00017
00018  #include "gdcmTypes.h"
00019  #include <iostream>
00020  #include <assert.h>
00021
00022  namespace gdcm
00023  {
00024
00025  class TransferSyntax;
00026
00045  class GDCM_EXPORT PixelFormat
00046  {
00047  friend class Bitmap;
00048  friend std::ostream& operator<<(std::ostream &_os, const PixelFormat &pf);
00049  public:
00050  // When adding a type please add its dual type (its unsigned counterpart)
00051  typedef enum {
00052      UINT8,
00053      INT8,
00054      UINT12,
00055      INT12,
00056      UINT16,
00057      INT16,
00058      UINT32, // For some DICOM files (RT or SC)
00059      INT32,  //
00060      UINT64, // Needed when input is 32bits + intercept/slope (incomplete support)
00061      INT64,  //
00062      FLOAT16, // sure why not...
00063      FLOAT32, // good ol' 'float'
00064      FLOAT64, // aka 'double'
00065      SINGLEBIT, // bool / monochrome
00066      UNKNOWN // aka BitsAllocated == 0 && PixelRepresentation == 0
00067  } ScalarType;
00068
00069  // default ctor:
00070  PixelFormat () : PixelFormat(1, 8, 8, 7, 0) {}
00071
00072  explicit PixelFormat (
00073      unsigned short samplesperpixel,
00074      unsigned short bitsallocated = 8,
00075      unsigned short bitsstored = 8,
00076      unsigned short highbit = 7,
00077      unsigned short pixelrepresentation = 0 ) :
00078      SamplesPerPixel(samplesperpixel),
00079      BitsAllocated(bitsallocated),
00080      BitsStored(bitsstored),
00081      HighBit(highbit),
00082      PixelRepresentation(pixelrepresentation) {}
00083  // helper, for the common case
00084  PixelFormat(ScalarType st);
00085
00086  // For transparency of use
00087  operator ScalarType() const { return GetScalarType(); }
00088
00091  unsigned short GetSamplesPerPixel() const;
00092  void SetSamplesPerPixel(unsigned short spp)
00093  {
00094      gdcmAssertMacro( spp <= 4 );
00095      SamplesPerPixel = spp;
00096      assert( SamplesPerPixel == 1 || SamplesPerPixel == 3 || SamplesPerPixel == 4 );

```

```

00097     }
00098
00099 unsigned short GetBitsAllocated() const
00100 {
00101     return BitsAllocated;
00102 }
00103
00104 void SetBitsAllocated(unsigned short ba)
00105 {
00106     if( ba )
00107     {
00108         switch( ba )
00109         {
00110             /* some devices (FUJIFILM CR + MONO1) incorrectly set BitsAllocated/BitsStored
00111              * as bitmask instead of value. Do what they mean instead of what they say.
00112              */
00113             case 0xffff: ba = 16; break;
00114             case 0x0fff: ba = 12; break;
00115             case 0x00ff: ba = 8; break;
00116         }
00117         BitsAllocated = ba;
00118         BitsStored = ba;
00119         HighBit = (unsigned short)(ba - 1);
00120     }
00121     else // Make the PixelFormat as UNKNOWN
00122     {
00123         BitsAllocated = 0;
00124         PixelRepresentation = 0;
00125     }
00126 }
00127
00128 unsigned short GetBitsStored() const
00129 {
00130     assert( BitsStored <= BitsAllocated );
00131     return BitsStored;
00132 }
00133
00134 void SetBitsStored(unsigned short bs)
00135 {
00136     switch( bs )
00137     {
00138         /* see SetBitsAllocated for explanation
00139         */
00140         case 0xffff: bs = 16; break;
00141         case 0x0fff: bs = 12; break;
00142         case 0x00ff: bs = 8; break;
00143     }
00144     if( bs <= BitsAllocated && bs )
00145     {
00146         BitsStored = bs;
00147         SetHighBit( (unsigned short) (bs - 1) );
00148     }
00149 }
00150
00151 unsigned short GetHighBit() const
00152 {
00153     assert( HighBit < BitsStored );
00154     return HighBit;
00155 }
00156
00157 void SetHighBit(unsigned short hb)
00158 {
00159     switch( hb )
00160     {
00161         /* broken implementations that use bitmask for BitsAllocated/Stored
00162          * nonetheless use (BitsStored-1) for HighBit. correct for this here.
00163          */
00164         case 0xfffe: hb = 15; break;
00165         case 0x0ffe: hb = 11; break;
00166         case 0x00fe: hb = 7; break;
00167     }
00168     if( hb < BitsStored )
00169         HighBit = hb;
00170 }
00171
00172 unsigned short GetPixelRepresentation() const
00173 {
00174     return (unsigned short)(PixelRepresentation ? 1 : 0);
00175 }
00176
00177 void SetPixelRepresentation(unsigned short pr)
00178 {
00179     PixelRepresentation = (unsigned short)(pr ? 1 : 0);
00180 }
00181

```

```

00183     ScalarType GetScalarType() const;
00184
00187     void SetScalarType(ScalarType st);
00188     const char *GetScalarTypeAsString() const;
00189
00195     uint8_t GetPixelSize() const;
00196
00198     void Print(std::ostream &os) const;
00199
00201     int64_t GetMin() const;
00202
00204     int64_t GetMax() const;
00205
00207     bool IsValid() const;
00208
00209     bool operator==(ScalarType st) const
00210     {
00211         return GetScalarType() == st;
00212     }
00213     bool operator!=(ScalarType st) const
00214     {
00215         return GetScalarType() != st;
00216     }
00217     bool operator==(const PixelFormat &pf) const
00218     {
00219         return
00220             SamplesPerPixel == pf.SamplesPerPixel &&
00221             BitsAllocated == pf.BitsAllocated &&
00222             BitsStored == pf.BitsStored &&
00223             HighBit == pf.HighBit &&
00224             PixelRepresentation == pf.PixelRepresentation;
00225     }
00226     bool operator!=(const PixelFormat &pf) const
00227     {
00228         return
00229             SamplesPerPixel != pf.SamplesPerPixel ||
00230             BitsAllocated != pf.BitsAllocated ||
00231             BitsStored != pf.BitsStored ||
00232             HighBit != pf.HighBit ||
00233             PixelRepresentation != pf.PixelRepresentation;
00234     }
00235
00236     bool IsCompatible(const TransferSyntax &ts) const;
00237 protected:
00239     bool Validate();
00240
00241 private:
00242     // D 0028|0002 [US] [Samples per Pixel] [1]
00243     unsigned short SamplesPerPixel;
00244     // D 0028|0100 [US] [Bits Allocated] [8]
00245     unsigned short BitsAllocated;
00246     // D 0028|0101 [US] [Bits Stored] [8]
00247     unsigned short BitsStored;
00248     // D 0028|0102 [US] [High Bit] [7]
00249     unsigned short HighBit;
00250     // D 0028|0103 [US] [Pixel Representation] [0]
00251     unsigned short PixelRepresentation;
00252 };
00253 //-----
00254 inline std::ostream& operator<(std::ostream &os, const PixelFormat &pf)
00255 {
00256     pf.Print( os );
00257     return os;
00258 }
00259
00260 } // end namespace gdcm
00261
00262 #endif //GDCMPIXELFORMAT_H

```

## 11.377 gdcmPixmap.h File Reference

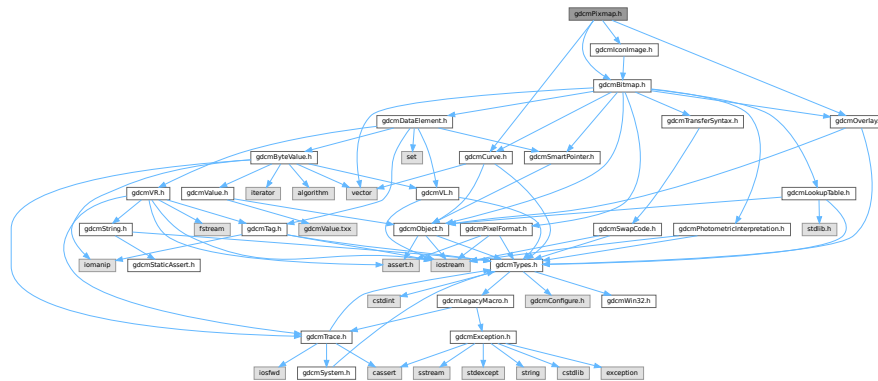
```

#include "gdcmBitmap.h"
#include "gdcmCurve.h"
#include "gdcmIconImage.h"

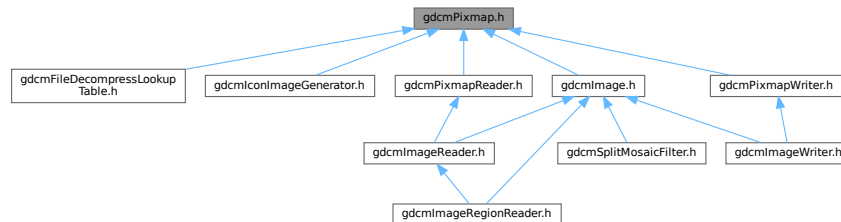
```

```
#include "gdcmOverlay.h"
```

Include dependency graph for gdcmPixmap.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::Pixmap`  
*Pixmap* class.

## Namespaces

- namespace **gdcm**

## 11.378 gdcmPixmap.h

[Go to the documentation of this file.](#)

```
00001  /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
```

```

00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMPIXMAP_H
00015 #define GDCMPIXMAP_H
00016
00017 #include "gdcmBitmap.h"
00018 #include "gdcmCurve.h"
00019 #include "gdcmIconImage.h"
00020 #include "gdcmOverlay.h"
00021
00022 namespace gdcm
00023 {
00024
00025 class GDCM_EXPORT Pixmap : public Bitmap
00026 {
00027 public:
00028     Pixmap();
00029     ~Pixmap() override;
00030     void Print(std::ostream &) const override;
00031
00032     bool AreOverlaysInPixelData() const override;
00033     bool UnusedBitsPresentInPixelData() const override;
00034
00035     Curve& GetCurve(size_t i = 0) {
00036         assert( i < Curves.size() );
00037         return Curves[i];
00038     }
00039     const Curve& GetCurve(size_t i = 0) const {
00040         assert( i < Curves.size() );
00041         return Curves[i];
00042     }
00043     size_t GetNumberOfCurves() const { return Curves.size(); }
00044     void SetNumberOfCurves(size_t n) { Curves.resize(n); }
00045
00046     Overlay& GetOverlay(size_t i = 0) {
00047         assert( i < Overlays.size() );
00048         return Overlays[i];
00049     }
00050     const Overlay& GetOverlay(size_t i = 0) const {
00051         assert( i < Overlays.size() );
00052         return Overlays[i];
00053     }
00054     size_t GetNumberOfOverlays() const { return Overlays.size(); }
00055     void SetNumberOfOverlays(size_t n) { Overlays.resize(n); }
00056     void RemoveOverlay(size_t i) {
00057         assert( i < Overlays.size() );
00058         Overlays.erase( Overlays.begin() + i );
00059     }
00060
00061     const IconImage &GetIconImage() const { return *Icon; }
00062     IconImage &GetIconImage() { return *Icon; }
00063     void SetIconImage(IconImage const &ii) { Icon = ii; }
00064
00065 private:
00066 protected:
00067     std::vector<Overlay> Overlays;
00068     std::vector<Curve> Curves;
00069     SmartPointer<IconImage> Icon;
00070 };
00071
00072 } // end namespace gdcm
00073
00074 #endif //GDCMPIXMAP_H

```

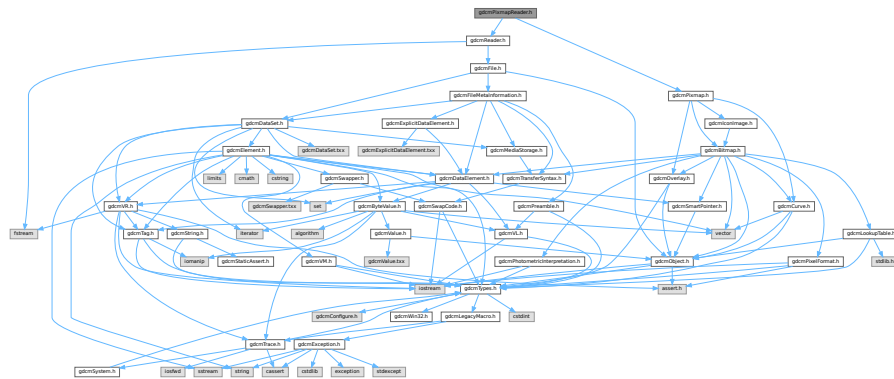
## 11.379 gdcmPixmapReader.h File Reference

```

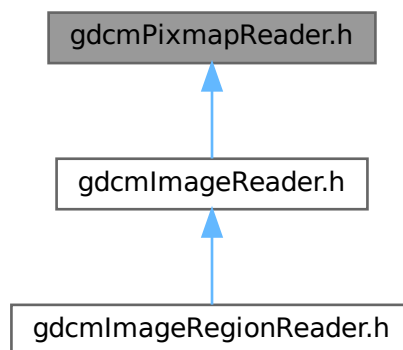
#include "gdcmReader.h"
#include "gdcmPixmap.h"

```

Include dependency graph for `gdcmPixmapReader.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::PixmapReader`  
*PixmapReader*.

## Namespaces

- namespace **gdcm**

## 11.380 gdcmPixmapReader.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMPIXMAPREADER_H
00015  #define GDCMPIXMAPREADER_H
00016
00017  #include "gdcmReader.h"
00018  #include "gdcmPixmap.h"
00019
00020  namespace gdcm
00021  {
00022
00023  class ByteValue;
00024  class MediaStorage;
00039  class GDCM_EXPORT PixmapReader : public Reader
00040  {
00041  public:
00042    PixmapReader();
00043    ~PixmapReader() override; //needs to be virtual to ensure lack of memory leaks
00044
00048    bool Read() override;
00049
00050    // Following methods are valid only after a call to 'Read'
00051
00052    const Pixmap& GetPixmap() const;
00053    Pixmap& GetPixmap();
00054    //void SetPixamp(Pixmap const &pix);
00055
00056  protected:
00057    bool ReadImageInternal(MediaStorage const &ms, bool handlepixeldata = true);
00058    virtual bool ReadImage(MediaStorage const &ms);
00059    virtual bool ReadACRNEMAImage();
00060
00061    SmartPointer<Pixmap> PixelData;
00062  };
00063
00064  } // end namespace gdcm
00065
00071  } // end namespace gdcm
00072
00073  #endif //GDCMPIXMAPREADER_H

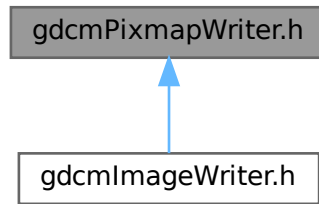
```







This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::PixmapWriter`  
*PixmapWriter.*

## Namespaces

- namespace `gdcm`

## 11.384 gdcmPixmapWriter.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMPIXMAPWRITER_H
00015  #define GDCMPIXMAPWRITER_H
00016
00017  #include "gdcmWriter.h"
00018  #include "gdcmPixmap.h"
00019
00020  namespace gdcm
00021  {
00022
00023  class StreamImageWriter;
00024  class Pixmap;
00025
00026  class GDCM_EXPORT PixmapWriter : public Writer
00027  {
00028  public:
00029    PixmapWriter();
00030    ~PixmapWriter() override;
00031
00032  };
00033
00034  }
00035
00036  #endif
00037
00038  #endif
00039
00040  #endif
00041

```

```

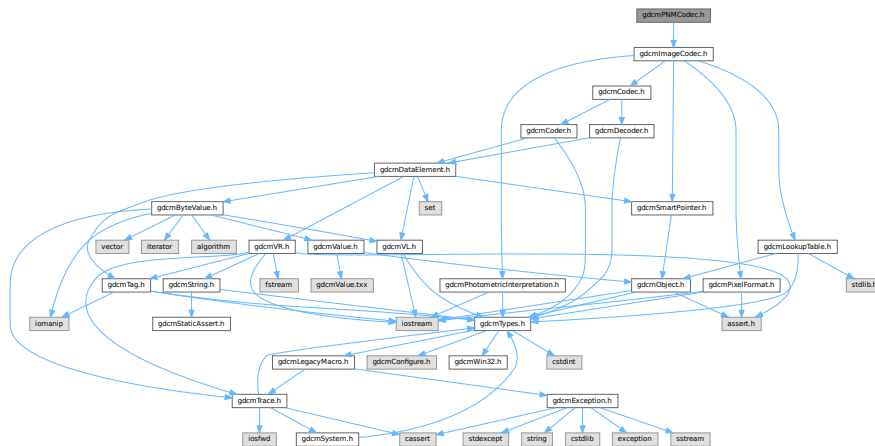
00042     const Pixmap& GetPixmap() const { return *PixelData; }
00043     Pixmap& GetPixmap() { return *PixelData; } // FIXME
00044     void SetPixmap(Pixmap const &img);
00045
00046     virtual const Pixmap& GetImage() const { return *PixelData; }
00047     virtual Pixmap& GetImage() { return *PixelData; } // FIXME
00051     virtual void SetImage(Pixmap const &img);
00052
00054     bool Write() override; // Execute()
00055
00056 protected:
00057     void DoIconImage(DataSet & ds, Pixmap const & image);
00058     bool PrepareWrite( MediaStorage const & refms );
00059
00060     SmartPointer<Pixmap> PixelData;
00061 };
00062
00068 } // end namespace gdcm
00069
00070 #endif //GDCMPIXMAPWRITER_H

```

## 11.385 gdcmPNMCodec.h File Reference

#include "gdcmImageCodec.h"

Include dependency graph for gdcmPNMCodec.h:



### Classes

- class [gdcm::PNMCodec](#)  
Class to do PNM.

### Namespaces

- namespace [gdcm](#)

## 11.386 gdcmPNMCodec.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMPNMCODEC_H
00015 #define GDCMPNMCODEC_H
00016
00017 #include "gdcmImageCodec.h"
00018
00019 namespace gdcm
00020 {
00021
00022   class GDCM_EXPORT PNMCodec : public ImageCodec
00023   {
00024   public:
00025     PNMCodec();
00026     ~PNMCodec() override;
00027     bool CanDecode(TransferSyntax const &ts) const override;
00028     bool CanCode(TransferSyntax const &ts) const override;
00029
00030     unsigned long GetBufferLength() const { return BufferLength; }
00031     void SetBufferLength(unsigned long l) { BufferLength = l; }
00032
00033     bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
00034     ImageCodec * Clone() const override;
00035
00036     bool Read(const char *filename, DataElement &out) const;
00037     bool Write(const char *filename, const DataElement &out) const;
00038     //bool Write(const char *filename);
00039   private:
00040     unsigned long BufferLength;
00041   };
00042
00043 } // end namespace gdcm
00044
00045 #endif //GDCMPNMCODEC_H

```

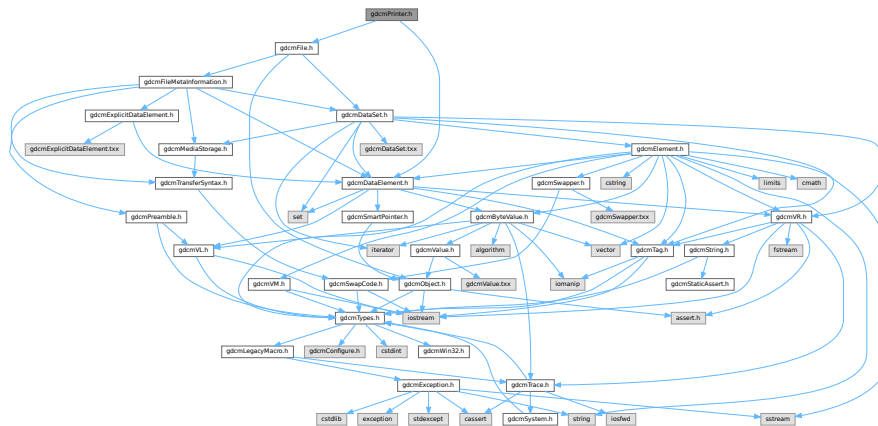
## 11.387 gdcmPrinter.h File Reference

```

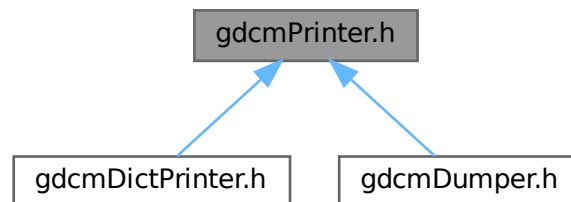
#include "gdcmFile.h"
#include "gdcmDataElement.h"

```

Include dependency graph for `gdcmPrinter.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::Printer`  
*Printer* class.

## Namespaces

- namespace `gdcm`

## 11.388 gdcmPrinter.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMPRINTER_H
00015  #define GDCMPRINTER_H
00016
00017  // TODO Class to implement printing
00018  // Since DICOM does printing ?
00019  // Also I would like to encapsulate the IsCharacterPrintable thing
00020  // (to avoid printing \0 and other weird characters)
00021  // \todo I still need to implement skipping of group (shadow)
00022  // need to implement longer field to read
00023
00024  /*
00025   * Output:
00026   * For ASCII:
00027   * Typically will look like:
00028   * [ORIGINAL\PRIMARY\OTHER]
00029   * If a non printable character is found: RED and INVERSE is used:
00030   * [
00031   *
00032   * when the VR is not found (file or dict), we check if we can print the output:
00033   * on success ASCII mode is used, on failure the output is printed a series of bytes
00034   *
00035   * Special case when the data element is empty:
00036   * INVERSE « (no value)
00037   *
00038   * retired public element are printed in red and underline
00039   * unknown private element are printed in RED followed by 'UNKNOWN'
00040   *
00041   * Correct VR is printed in green just after the found VR
00042   *
00043   * length of data element is printed in bytes, followed by the VM, a green VM is appended
00044   * if this is not compatible
00045   */
00046  #include "gdcmFile.h"
00047  #include "gdcmDataElement.h"
00048
00049  namespace gdcm
00050  {
00051
00052  class DataSet;
00053  class DictEntry;
00054  class Dicts;
00055
00056  // It's a sink there is no output
00057  class GDCM_EXPORT Printer
00058  {
00059  public:
00060    Printer();
00061    ~Printer() = default;
00062
00063    void SetFile(File const &f) { F = &f; }
00064
00065    void SetColor(bool c);
00066
00067    typedef enum {
00068      VERBOSE_STYLE = 0, // GDCM Legacy VERBOSE one
00069      CONDENSED_STYLE, //
00070      // Ok I am missing voc here ...better naming would be nice
00071      XML, //
00072      CXX
00073    } PrintStyles;
00074
00075    void SetStyle(PrintStyles ps) {
00076      PrintStyle = ps;
00077    }
00078
00079  };
00080
00081  }
00082

```

```

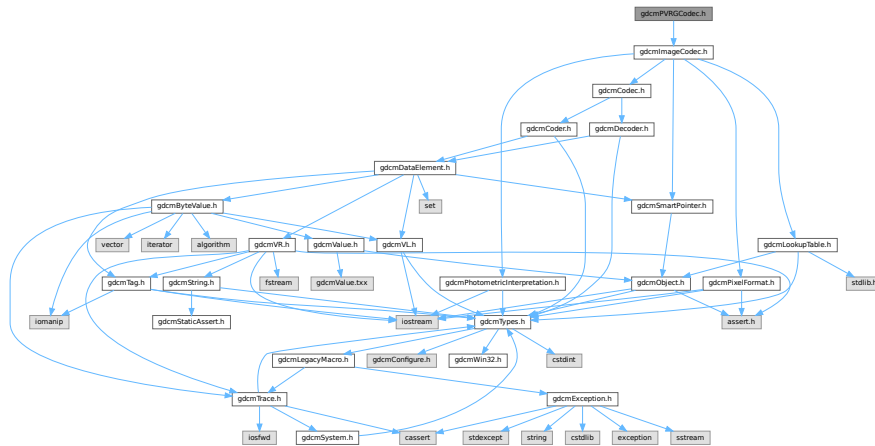
00084     PrintStyles GetPrintStyle() const {
00085         return PrintStyle;
00086     }
00087
00088     void Print(std::ostream& os);
00089
00090     void PrintDataSet(const DataSet &ds, std::ostream& os, const std::string &s = "");
00091
00092 protected:
00093     VR PrintDataElement(std::ostream& os, const Dicts &dicts, const DataSet &ds, const DataElement
00094     &de, std::ostream& out, std::string const &indent);
00095 void PrintSQ(const SequenceOfItems *sqi, std::ostream &os, std::string const &indent);
00096
00097     PrintStyles PrintStyle;
00098     const File *F;
00099     VL MaxPrintLength;
00100 };
00101
00102 // end namespace gdcm
00103 #endif //GDCMPRINTER_H

```

## 11.389 gdcmPVRGCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmPVRGCodec.h:



### Classes

- class [gdcm::PVRGCodec](#)  
*PVRGCodec.*

### Namespaces

- namespace [gdcm](#)





## Classes

- class [gdcm::RAWCodec](#)  
*RAWCodec* class.

## Namespaces

- namespace [gdcm](#)

## 11.392 gdcmRAWCodec.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMRAWCODEC_H
00015 #define GDCMRAWCODEC_H
00016
00017 #include "gdcmImageCodec.h"
00018
00019 namespace gdcm
00020 {
00021
00022     class RAWInternals;
00026     class GDCM_EXPORT RAWCodec : public ImageCodec
00027     {
00028     public:
00029         RAWCodec();
00030         ~RAWCodec() override;
00031         bool CanCode(TransferSyntax const &ts) const override;
00032         bool CanDecode(TransferSyntax const &ts) const override;
00033         bool Decode(DataElement const &is, DataElement &os) override;
00034         bool Code(DataElement const &in, DataElement &out) override;
00035
00036         bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
00037         ImageCodec * Clone() const override;
00038
00041         bool DecodeBytes(const char* inBytes, size_t inBufferLength,
00042             char* outBytes, size_t inOutBufferLength);
00043
00044     protected:
00045         bool DecodeByStreams(std::istream &is, std::ostream &os) override;
00046
00047     private:
00048         RAWInternals *Internals;
00049     };
00050
00051 } // end namespace gdcm
00052
00053 #endif // GDCMRAWCODEC_H

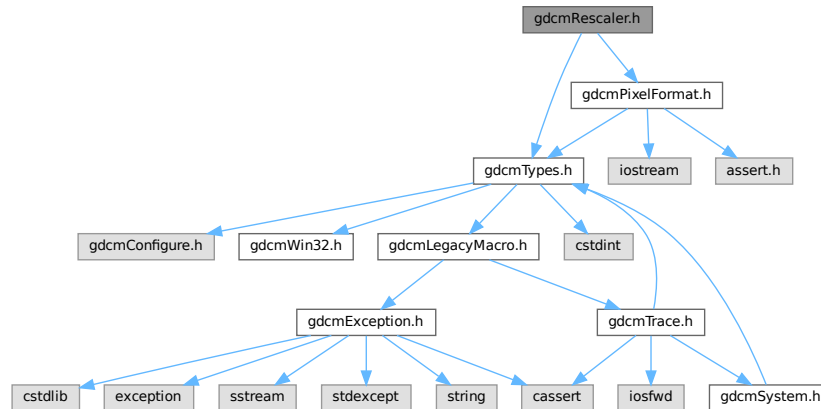
```

## 11.393 gdcmRescaler.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmPixelFormat.h"
```

Include dependency graph for gdcmRescaler.h:



### Classes

- class [gdcm::Rescaler](#)  
*Rescale class.*

### Namespaces

- namespace [gdcm](#)

## 11.394 gdcmRescaler.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012  =====*/
00013  #ifndef GDCMRESCALER_H
00014  #define GDCMRESCALER_H
00015
00016  #include "gdcmTypes.h"
00017  #include "gdcmPixelFormat.h"

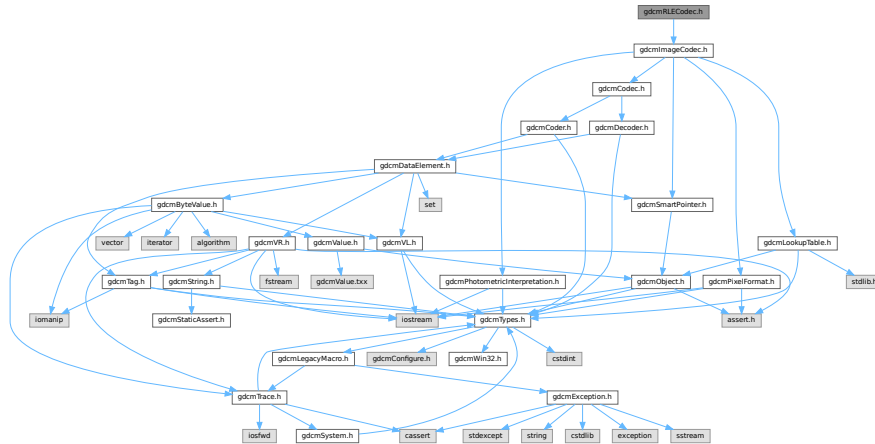
```

```
00019
00020 namespace gdcm
00021 {
00022
00068 class GDCM_EXPORT Rescaler
00069 {
00070 public:
00071     Rescaler(): Intercept(0), Slope(1), PF(PixelFormat::UNKNOWN), TargetScalarType(PixelFormat::UNKNOWN),
        ScalarRangeMin(0), ScalarRangeMax(0), UseTargetPixelType(false) {}
00072     ~Rescaler() = default;
00073
00075     bool Rescale(char *out, const char *in, size_t n);
00076
00078     bool InverseRescale(char *out, const char *in, size_t n);
00079
00081     void SetIntercept(double i) { Intercept = i; }
00082     double GetIntercept() const { return Intercept; }
00083
00085     void SetSlope(double s) { Slope = s; }
00086     double GetSlope() const { return Slope; }
00087
00092     void SetTargetPixelType( PixelFormat const & targetst );
00093
00095     void SetUseTargetPixelType(bool b);
00096
00098     void SetPixelFormat(PixelFormat const & pf) { PF = pf; }
00099
00102     PixelFormat::ScalarType ComputeInterceptSlopePixelType();
00103
00106     void SetMinMaxForPixelType(double min, double max);
00107
00110     PixelFormat ComputePixelTypeFromMinMax();
00111
00112 protected:
00113     template <typename TIn>
00114         void RescaleFunctionIntoBestFit(char *out, const TIn *in, size_t n);
00115     template <typename TIn>
00116         void InverseRescaleFunctionIntoBestFit(char *out, const TIn *in, size_t n);
00117
00118 private:
00119     double Intercept; // 0028,1052
00120     double Slope; // 0028,1053
00121     PixelFormat PF;
00122     PixelFormat::ScalarType TargetScalarType;
00123     double ScalarRangeMin;
00124     double ScalarRangeMax;
00125     bool UseTargetPixelType;
00126 };
00127
00128 } // end namespace gdcm
00129
00130 #endif //GDCMRESCALER_H
```

## 11.395 gdcmRLECodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmRLECodec.h:



### Classes

- class [gdcm::RLECodec](#)  
Class to do RLE.

### Namespaces

- namespace [gdcm](#)

## 11.396 gdcmRLECodec.h

[Go to the documentation of this file.](#)

```
00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMRLECODEC_H
00015 #define GDCMRLECODEC_H
00016
00017 #include "gdcmImageCodec.h"
00018
00019 namespace gdcm
00020 {
```

```

00021
00022 class Fragment;
00023 class RLEInternals;
00036 class GDCM_EXPORT RLECodec : public ImageCodec
00037 {
00038     friend class ImageRegionReader;
00039 public:
00040     RLECodec();
00041     ~RLECodec() override;
00042     bool CanCode(TransferSyntax const &ts) const override;
00043     bool CanDecode(TransferSyntax const &ts) const override;
00044     bool Decode(DataElement const &is, DataElement &os) override;
00045     unsigned long GetBufferLength() const { return BufferLength; }
00046     void SetBufferLength(unsigned long l) { BufferLength = l; }
00047
00048     bool Code(DataElement const &in, DataElement &out) override;
00049     bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
00050     ImageCodec * Clone() const override;
00051
00052 protected:
00053     bool DecodeExtent(
00054         char *buffer,
00055         unsigned int XMin, unsigned int XMax,
00056         unsigned int YMin, unsigned int YMax,
00057         unsigned int ZMin, unsigned int ZMax,
00058         std::istream & is
00059     );
00060
00061     bool DecodeByStreams(std::istream &is, std::ostream &os) override;
00062 public:
00063
00064     void SetLength(unsigned long l)
00065     {
00066         Length = l;
00067     }
00068
00069 protected:
00070     bool StartEncode( std::ostream & ) override;
00071     bool IsRowEncoder() override;
00072     bool IsFrameEncoder() override;
00073     bool AppendRowEncode( std::ostream & out, const char * data, size_t datalen ) override;
00074     bool AppendFrameEncode( std::ostream & out, const char * data, size_t datalen ) override;
00075     bool StopEncode( std::ostream & ) override;
00076
00077 private:
00078     bool DecodeByStreamsCommon(std::istream &is, std::ostream &os);
00079     RLEInternals *Internals;
00080     unsigned long Length;
00081     unsigned long BufferLength;
00082     size_t DecodeFragment(Fragment const & frag, char *buffer, size_t llen);
00083 };
00084
00085 } // end namespace gdcm
00086
00087 #endif //GDCMRLECODEC_H

```

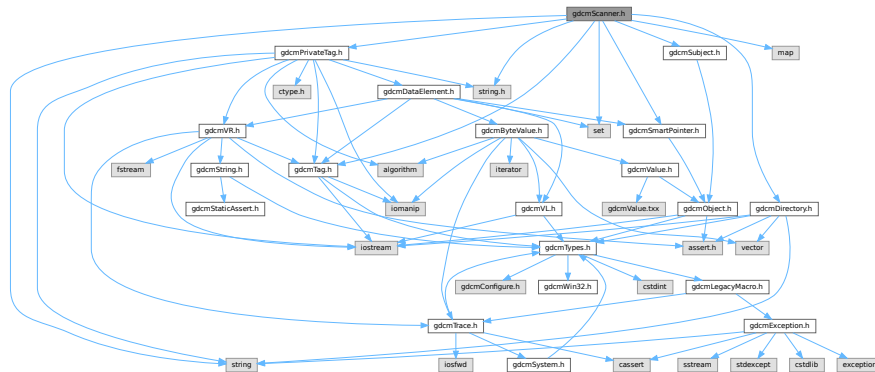
## 11.397 gdcmScanner.h File Reference

```

#include "gdcmDirectory.h"
#include "gdcmSubject.h"
#include "gdcmTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmSmartPointer.h"
#include <map>
#include <set>
#include <string>
#include <string.h>

```

Include dependency graph for gdcmScanner.h:



## Classes

- struct `gdcm::Scanner::ltstr`
- class `gdcm::Scanner`  
*Scanner.*

## Namespaces

- namespace `gdcm`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Scanner &s)`

## 11.398 gdcmScanner.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMSCANNER_H
00015 #define GDCMSCANNER_H
00016
00017 #include "gdcmDirectory.h"
00018 #include "gdcmSubject.h"
00019 #include "gdcmTag.h"
00020 #include "gdcmPrivateTag.h"

```

```

00021 #include "gdcmSmartPointer.h"
00022
00023 #include <map>
00024 #include <set>
00025 #include <string>
00026
00027 #include <string.h> // strcmp
00028
00029 namespace gdcm
00030 {
00031     class StringFilter;
00032
00033     class GDCM_EXPORT Scanner : public Subject
00034     {
00035     public:
00036         friend std::ostream& operator<<(std::ostream &_os, const Scanner &s);
00037         Scanner():Values(),FileNames(),Mappings() {}
00038         ~Scanner() override;
00039
00040         typedef std::map<Tag, const char*> TagToValue;
00041         //typedef std::map<Tag, ConstCharWrapper> TagToValue; //StringMap;
00042         //typedef TagToStringMap TagToValue;
00043         typedef TagToValue::value_type TagToValueValueType;
00044
00045         void AddTag( Tag const & t );
00046         void ClearTags();
00047
00048         // Work in progress do not use:
00049         void AddPrivateTag( PrivateTag const & t );
00050
00051         void AddSkipTag( Tag const & t );
00052         void ClearSkipTags();
00053
00054         bool Scan( Directory::FileNamesType const & filenames );
00055
00056         Directory::FileNamesType const &GetFileNames() const { return FileNames; }
00057
00058         void Print( std::ostream & os ) const override;
00059
00060         void PrintTable( std::ostream & os ) const;
00061
00062         bool IsKey( const char * filename ) const;
00063
00064         Directory::FileNamesType GetKeys() const;
00065
00066         // struct to store all the values found:
00067         typedef std::set< std::string > ValueType;
00068
00069         ValueType const & GetValues() const { return Values; }
00070
00071         ValueType GetValues(Tag const &t) const;
00072
00073         Directory::FileNamesType GetOrderedValues(Tag const &t) const;
00074
00075         /* ltstr is CRITICAL, otherwise pointers value are used to do the key comparison */
00076         struct ltstr
00077         {
00078             bool operator()(const char* s1, const char* s2) const
00079             {
00080                 assert( s1 && s2 );
00081                 return strcmp(s1, s2) < 0;
00082             }
00083         };
00084
00085         typedef std::map<const char *,TagToValue, ltstr> MappingType;
00086         typedef MappingType::const_iterator ConstIterator;
00087         ConstIterator Begin() const { return Mappings.begin(); }
00088         ConstIterator End() const { return Mappings.end(); }
00089
00090         MappingType const & GetMappings() const { return Mappings; }
00091
00092         TagToValue const & GetMapping(const char *filename) const;
00093
00094         const char *GetFilenameFromTagToValue(Tag const &t, const char *valueref) const;
00095
00096         Directory::FileNamesType GetAllFileNamesFromTagToValue(Tag const &t, const char *valueref) const;
00097
00098         // by a call to GetMapping()
00099         TagToValue const & GetMappingFromTagToValue(Tag const &t, const char *value) const;
00100
00101         const char* GetValue(const char *filename, Tag const &t) const;
00102
00103     };
00104
00105     StringFilter
00106     {
00107     public:
00108         StringFilter() {}
00109         ~StringFilter() {}
00110         void AddTag( Tag const & t );
00111         void ClearTags();
00112         void AddPrivateTag( PrivateTag const & t );
00113         void AddSkipTag( Tag const & t );
00114         void ClearSkipTags();
00115         bool Scan( Directory::FileNamesType const & filenames );
00116         Directory::FileNamesType const &GetFileNames() const { return FileNames; }
00117         void Print( std::ostream & os ) const override;
00118         void PrintTable( std::ostream & os ) const;
00119         bool IsKey( const char * filename ) const;
00120         Directory::FileNamesType GetKeys() const;
00121         // struct to store all the values found:
00122         typedef std::set< std::string > ValueType;
00123         ValueType const & GetValues() const { return Values; }
00124         ValueType GetValues(Tag const &t) const;
00125         Directory::FileNamesType GetOrderedValues(Tag const &t) const;
00126         /* ltstr is CRITICAL, otherwise pointers value are used to do the key comparison */
00127         struct ltstr
00128         {
00129             bool operator()(const char* s1, const char* s2) const
00130             {
00131                 assert( s1 && s2 );
00132                 return strcmp(s1, s2) < 0;
00133             }
00134         };
00135         typedef std::map<const char *,TagToValue, ltstr> MappingType;
00136         typedef MappingType::const_iterator ConstIterator;
00137         ConstIterator Begin() const { return Mappings.begin(); }
00138         ConstIterator End() const { return Mappings.end(); }
00139         MappingType const & GetMappings() const { return Mappings; }
00140         TagToValue const & GetMapping(const char *filename) const;
00141         const char *GetFilenameFromTagToValue(Tag const &t, const char *valueref) const;
00142         Directory::FileNamesType GetAllFileNamesFromTagToValue(Tag const &t, const char *valueref) const;
00143         // by a call to GetMapping()
00144         TagToValue const & GetMappingFromTagToValue(Tag const &t, const char *value) const;
00145         const char* GetValue(const char *filename, Tag const &t) const;
00146     };
00147
00148     class GDCM_EXPORT Scanner : public Subject
00149     {
00150     public:
00151         friend std::ostream& operator<<(std::ostream &_os, const Scanner &s);
00152         Scanner():Values(),FileNames(),Mappings() {}
00153         ~Scanner() override;
00154
00155         typedef std::map<Tag, const char*> TagToValue;
00156         //typedef std::map<Tag, ConstCharWrapper> TagToValue; //StringMap;
00157         //typedef TagToStringMap TagToValue;
00158         typedef TagToValue::value_type TagToValueValueType;
00159
00160         void AddTag( Tag const & t );
00161         void ClearTags();
00162
00163         // Work in progress do not use:
00164         void AddPrivateTag( PrivateTag const & t );
00165
00166         void AddSkipTag( Tag const & t );
00167         void ClearSkipTags();
00168
00169         bool Scan( Directory::FileNamesType const & filenames );
00170
00171         Directory::FileNamesType const &GetFileNames() const { return FileNames; }
00172
00173         void Print( std::ostream & os ) const override;
00174
00175         void PrintTable( std::ostream & os ) const;
00176
00177         bool IsKey( const char * filename ) const;
00178
00179         Directory::FileNamesType GetKeys() const;
00180
00181         // struct to store all the values found:
00182         typedef std::set< std::string > ValueType;
00183
00184         ValueType const & GetValues() const { return Values; }
00185
00186         ValueType GetValues(Tag const &t) const;
00187
00188         Directory::FileNamesType GetOrderedValues(Tag const &t) const;
00189
00190         /* ltstr is CRITICAL, otherwise pointers value are used to do the key comparison */
00191         struct ltstr
00192         {
00193             bool operator()(const char* s1, const char* s2) const
00194             {
00195                 assert( s1 && s2 );
00196                 return strcmp(s1, s2) < 0;
00197             }
00198         };
00199
00200         typedef std::map<const char *,TagToValue, ltstr> MappingType;
00201         typedef MappingType::const_iterator ConstIterator;
00202         ConstIterator Begin() const { return Mappings.begin(); }
00203         ConstIterator End() const { return Mappings.end(); }
00204
00205         MappingType const & GetMappings() const { return Mappings; }
00206
00207         TagToValue const & GetMapping(const char *filename) const;
00208
00209         const char *GetFilenameFromTagToValue(Tag const &t, const char *valueref) const;
00210
00211         Directory::FileNamesType GetAllFileNamesFromTagToValue(Tag const &t, const char *valueref) const;
00212
00213         // by a call to GetMapping()
00214         TagToValue const & GetMappingFromTagToValue(Tag const &t, const char *value) const;
00215         const char* GetValue(const char *filename, Tag const &t) const;
00216     };

```



```

00157     static SmartPointer<Scanner> New() { return new Scanner; }
00158
00159 protected:
00160     void ProcessPublicTag(StringFilter &sf, const char *filename);
00161 private:
00162     // struct to store all uniq tags in ascending order:
00163     typedef std::set< Tag > TagsType;
00164     typedef std::set< PrivateTag > PrivateTagsType;
00165     std::set< Tag > Tags;
00166     std::set< PrivateTag > PrivateTags;
00167     std::set< Tag > SkipTags;
00168     ValueType Values;
00169     Directory::FileNamesType Filenames;
00170
00171     // Main struct that will hold all mapping:
00172     MappingType Mappings;
00173
00174     double Progress;
00175 };
00176 //-----
00177 inline std::ostream& operator<<(std::ostream &os, const Scanner &s)
00178 {
00179     s.Print( os );
00180     return os;
00181 }
00182
00183 #if defined(SWIGPYTHON) || defined(SWIGCSharp) || defined(SWIGJAVA) || defined(SWIGPHP)
00184 /*
00185  * HACK: I need this temp class to be able to manipulate a std::map from python,
00186  * swig does not support wrapping of simple class like std::map...
00187  */
00188 class SWIGTagToValue
00189 {
00190 public:
00191     SWIGTagToValue(Scanner::TagToValue const &t2v):Internal(t2v),it(t2v.begin()) {}
00192     const Scanner::TagToValueValueType& GetCurrent() const { return *it; }
00193     const Tag& GetCurrentTag() const { return it->first; }
00194     const char *GetCurrentValue() const { return it->second; }
00195     void Start() { it = Internal.begin(); }
00196     bool IsAtEnd() const { return it == Internal.end(); }
00197     void Next() { ++it; }
00198 private:
00199     const Scanner::TagToValue& Internal;
00200     Scanner::TagToValue::const_iterator it;
00201 };
00202 #endif /* SWIG */
00203
00209 } // end namespace gdcm
00210
00211 #endif //GDCMSCANNER_H

```

## 11.399 gdcmScanner2.h File Reference

```

#include "gdcmDirectory.h"
#include "gdcmSubject.h"
#include "gdcmTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmSmartPointer.h"
#include <map>
#include <set>
#include <string>
#include <string.h>

```



```

00021 #include "gdcmSmartPointer.h"
00022
00023 #include <map>
00024 #include <set>
00025 #include <string>
00026
00027 #include <string.h> // strcmp
00028
00029 namespace gdcm
00030 {
00031     class StringFilter;
00032
00033     class GDCM_EXPORT Scanner2 : public Subject
00034     {
00035     public:
00036         friend std::ostream& operator<<(std::ostream &_os, const Scanner2 &s);
00037     public:
00038         Scanner2():Values(),FileNames(),PublicMappings(),PrivateMappings(),Progress(0.0) {}
00039         ~Scanner2() override;
00040
00041         typedef std::map<Tag, const char*> PublicTagToValue;
00042         typedef PublicTagToValue::value_type PublicTagToValueValueType;
00043
00044         typedef std::map<PrivateTag, const char*> PrivateTagToValue;
00045         typedef PrivateTagToValue::value_type PrivateTagToValueValueType;
00046
00047         bool AddPublicTag( Tag const & t );
00048         void ClearPublicTags();
00049
00050         // Work in progress do not use:
00051         bool AddPrivateTag( PrivateTag const & pt );
00052         void ClearPrivateTags();
00053
00054         bool AddSkipTag( Tag const & t );
00055         void ClearSkipTags();
00056
00057         bool Scan( Directory::FileNamesType const & filenames );
00058
00059         Directory::FileNamesType const &GetFileNames() const { return FileNames; }
00060
00061         void Print( std::ostream & os ) const override;
00062
00063         void PrintTable( std::ostream & os, bool header = false ) const;
00064
00065         bool IsKey( const char * filename ) const;
00066
00067         Directory::FileNamesType GetKeys() const;
00068
00069         // struct to store all the values found:
00070         typedef std::set< std::string > ValueType;
00071
00072         ValueType const & GetValues() const { return Values; }
00073
00074         ValueType GetPublicValues(Tag const &t) const;
00075
00076         ValueType GetPrivateValues(PrivateTag const &pt) const;
00077
00078         Directory::FileNamesType GetPublicOrderedValues(Tag const &t) const;
00079
00080         Directory::FileNamesType GetPrivateOrderedValues(PrivateTag const &pt) const;
00081
00082         /* ltstr is CRITICAL, otherwise pointers value are used to do the key comparison */
00083         struct ltstr
00084         {
00085             bool operator()(const char* s1, const char* s2) const
00086             {
00087                 assert( s1 && s2 );
00088                 return strcmp(s1, s2) < 0;
00089             }
00090         };
00091
00092         typedef std::map<const char *,PublicTagToValue, ltstr> PublicMappingType;
00093         typedef PublicMappingType::const_iterator PublicConstIterator;
00094         PublicConstIterator Begin() const { return PublicMappings.begin(); }
00095         PublicConstIterator End() const { return PublicMappings.end(); }
00096
00097         typedef std::map<const char *,PrivateTagToValue, ltstr> PrivateMappingType;
00098         typedef PrivateMappingType::const_iterator PrivateConstIterator;
00099         PrivateConstIterator PrivateBegin() const { return PrivateMappings.begin(); }
00100         PrivateConstIterator PrivateEnd() const { return PrivateMappings.end(); }
00101
00102         PublicMappingType const & GetPublicMappings() const { return PublicMappings; }
00103         PrivateMappingType const & GetPrivateMappings() const { return PrivateMappings; }

```

```

00148
00150 PublicTagToValue const & GetPublicMapping(const char *filename) const;
00151 PrivateTagToValue const & GetPrivateMapping(const char *filename) const;
00152
00155 const char *GetFilenameFromPublicTagToValue(Tag const &t, const char *valueref) const;
00156 const char *GetFilenameFromPrivateTagToValue(PrivateTag const &pt, const char *valueref) const;
00157
00160 Directory::FileNamesType GetAllFileNamesFromPublicTagToValue(Tag const &t, const char *valueref) const;
00161 Directory::FileNamesType GetAllFileNamesFromPrivateTagToValue(PrivateTag const &pt, const char
*valueref) const;
00162
00164 // by a call to GetMapping()
00165 PublicTagToValue const & GetMappingFromPublicTagToValue(Tag const &t, const char *value) const;
00166 PrivateTagToValue const & GetMappingFromPrivateTagToValue(PrivateTag const &pt, const char *value)
const;
00167
00173 const char* GetPublicValue(const char *filename, Tag const &t) const;
00174 const char* GetPrivateValue(const char *filename, PrivateTag const &t) const;
00175
00177 static SmartPointer<Scanner2> New() { return new Scanner2; }
00178
00179 protected:
00180 void ProcessPublicTag(StringFilter &sf, const char *filename);
00181 void ProcessPrivateTag(StringFilter &sf, const char *filename);
00182 private:
00183 // struct to store all uniq tags in ascending order:
00184 typedef std::set< Tag > PublicTagsType;
00185 typedef std::set< PrivateTag > PrivateTagsType;
00186 std::set< Tag > PublicTags; // Public and Private Creator
00187 std::set< PrivateTag > PrivateTags; // Only Private (no Private Creator)
00188 std::set< Tag > SkipTags;
00189 ValuesType Values;
00190 Directory::FileNamesType FileNames;
00191
00192 // Main struct that will hold all public mapping:
00193 PublicMappingType PublicMappings;
00194 // Main struct that will hold all private mapping:
00195 PrivateMappingType PrivateMappings;
00196
00197 double Progress;
00198 };
00199 //-----
00200 inline std::ostream& operator<<(std::ostream &os, const Scanner2 &s)
00201 {
00202     s.Print( os );
00203     return os;
00204 }
00205
00206 } // end namespace gdcm
00207
00208 #endif //GDCMSCANNER2_H

```

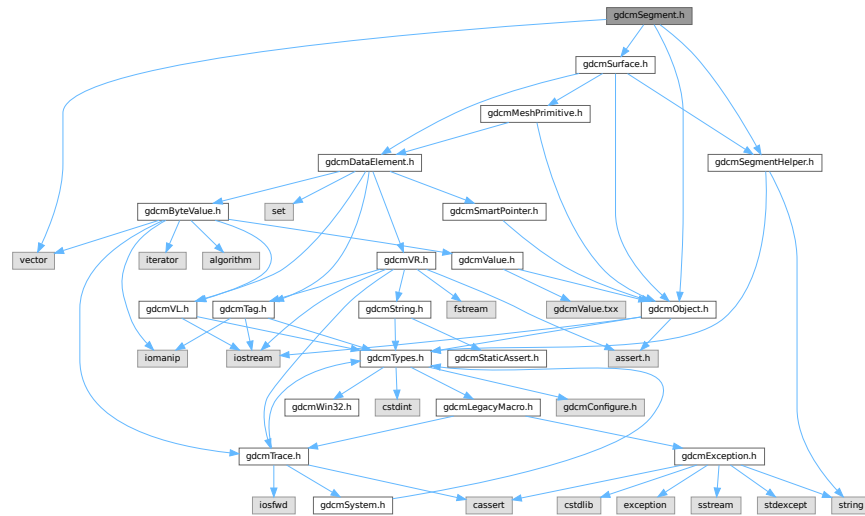
## 11.401 gdcmSegment.h File Reference

```

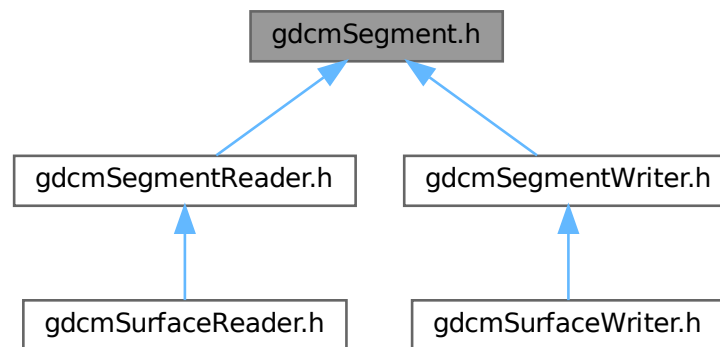
#include <vector>
#include <gdcmObject.h>
#include <gdcmSurface.h>
#include "gdcmSegmentHelper.h"

```

Include dependency graph for gdcmSegment.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::Segment](#)  
*This class defines a segment.*

## Namespaces

- namespace [gdcm](#)

## 11.402 gdcmSegment.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMSEGMENT_H
00015  #define GDCMSEGMENT_H
00016
00017  #include <vector>
00018
00019  #include <gdcmObject.h>
00020  #include <gdcmSurface.h>
00021  #include "gdcmSegmentHelper.h"
00022
00023  namespace gdcm
00024  {
00025
00033  class GDCM_EXPORT Segment : public Object
00034  {
00035  public:
00036
00037      typedef std::vector< SmartPointer< Surface > > SurfaceVector;
00038      typedef std::vector< SegmentHelper::BasicCodedEntry > BasicCodedEntryVector;
00039
00040      typedef enum {
00041          AUTOMATIC = 0,
00042          SEMIAUTOMATIC,
00043          MANUAL,
00044          ALGOType_END
00045      } ALGOType;
00046
00047      static const char * GetALGOTypeString(ALGOType type);
00048      static ALGOType GetALGOType(const char * type);
00049
00050
00051      Segment();
00052
00053      ~Segment() override;
00054
00055      /**      Segment getters/setters      */
00056      unsigned short GetSegmentNumber() const;
00057      void SetSegmentNumber(const unsigned short num);
00058
00059      const char * GetSegmentLabel() const;
00060      void SetSegmentLabel(const char * label);
00061
00062      const char * GetSegmentDescription() const;
00063      void SetSegmentDescription(const char * description);
00064
00065      SegmentHelper::BasicCodedEntry const & GetAnatomicRegion() const;
00066      SegmentHelper::BasicCodedEntry & GetAnatomicRegion();
00067      void SetAnatomicRegion(SegmentHelper::BasicCodedEntry const & BSE);
00068
00069      BasicCodedEntryVector const & GetAnatomicRegionModifiers() const;
00070      BasicCodedEntryVector & GetAnatomicRegionModifiers();
00071      void SetAnatomicRegionModifiers(BasicCodedEntryVector const & BSEV);
00072
00073      SegmentHelper::BasicCodedEntry const & GetPropertyCategory() const;
00074      SegmentHelper::BasicCodedEntry & GetPropertyCategory();
00075      void SetPropertyCategory(SegmentHelper::BasicCodedEntry const & BSE);
00076
00077      SegmentHelper::BasicCodedEntry const & GetPropertyType() const;
00078      SegmentHelper::BasicCodedEntry & GetPropertyType();
00079      void SetPropertyType(SegmentHelper::BasicCodedEntry const & BSE);
00080
00081      BasicCodedEntryVector const & GetPropertyTypeModifiers() const;
00082      BasicCodedEntryVector & GetPropertyTypeModifiers();
00083      void SetPropertyTypeModifiers(BasicCodedEntryVector const & BSEV);

```

```

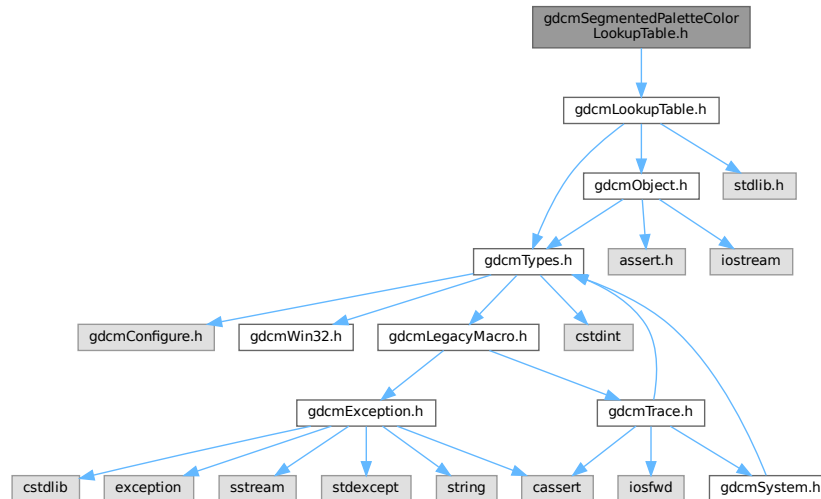
00084
00085     ALGOType GetSegmentAlgorithmType() const;
00086     void SetSegmentAlgorithmType(ALGOType type);
00087     void SetSegmentAlgorithmType(const char * typeStr);
00088
00089     const char * GetSegmentAlgorithmName() const;
00090     void SetSegmentAlgorithmName(const char * name);
00091
00092     /**          Surface getters/setters          **/
00093     unsigned long GetSurfaceCount();
00094     void SetSurfaceCount(const unsigned long nb);
00095
00096     SurfaceVector const & GetSurfaces() const;
00097     SurfaceVector & GetSurfaces();
00098
00099     SmartPointer< Surface > GetSurface(const unsigned int idx = 0) const;
00100
00101     void AddSurface(SmartPointer< Surface > surface);
00102
00103 protected :
00104     /**          Segment members          **/
00105     //0062 0004 US 1 Segment Number
00106     unsigned short SegmentNumber;
00107     //0062 0005 LO 1 Segment Label
00108     std::string SegmentLabel;
00109     //0062 0006 ST 1 Segment Description
00110     std::string SegmentDescription;
00111
00112     // General Anatomic Region
00113     SegmentHelper::BasicCodedEntry AnatomicRegion;
00114     // General Anatomic Region Modifier
00115     BasicCodedEntryVector AnatomicRegionModifiers;
00116     // Property Category Code
00117     SegmentHelper::BasicCodedEntry PropertyCategory;
00118     // Property Type Code
00119     SegmentHelper::BasicCodedEntry PropertyType;
00120     // Property Type Modifier Code
00121     BasicCodedEntryVector PropertyTypeModifiers;
00122
00123     //0062 0008 CS 1 Segment Algorithm Type
00124     ALGOType SegmentAlgorithmType;
00125     //0062 0009 LO 1 Segment Algorithm Name
00126     std::string SegmentAlgorithmName;
00127
00128     /**          Surface members          **/
00129     //0066 002a UL 1 Surface Count
00130     unsigned long SurfaceCount;
00131
00132     SurfaceVector Surfaces;
00133
00134 private :
00135     void ComputeSurfaceCount();
00136 };
00137
00138 }
00139
00140 #endif // GDCMSEGMENT_H

```

## 11.403 gdcmSegmentedPaletteColorLookupTable.h File Reference

```
#include "gdcmLookupTable.h"
```

Include dependency graph for gdcmSegmentedPaletteColorLookupTable.h:



### Classes

- class [gdcm::SegmentedPaletteColorLookupTable](#)  
*SegmentedPaletteColorLookupTable* class.

### Namespaces

- namespace [gdcm](#)

## 11.404 gdcmSegmentedPaletteColorLookupTable.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014
00015 #ifndef GDCMSEGMENTEDPALETTECOLORLOOKUPTABLE_H

```



```

00016 #define GDCMSEGMENTEDPALETTECOLORLOOKUPTABLE_H
00017
00018 #include "gdcmLookupTable.h"
00019
00020 namespace gdcm
00021 {
00022
00026 class GDCM_EXPORT SegmentedPaletteColorLookupTable : public LookupTable
00027 {
00028 public:
00029     SegmentedPaletteColorLookupTable();
00030     ~SegmentedPaletteColorLookupTable() override;
00031     void Print(std::ostream &) const override {}
00032
00034     void SetLUT(LookupTableType type, const unsigned char *array,
00035         unsigned int length) override;
00036
00037 };
00038
00039 } // end namespace gdcm
00040
00041 #endif //GDCMSEGMENTEDPALETTECOLORLOOKUPTABLE_H

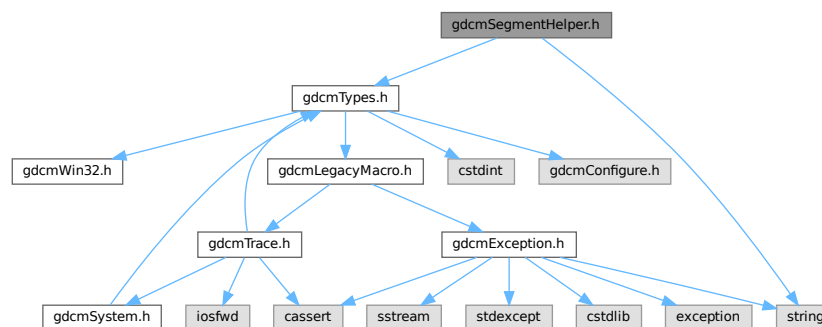
```

## 11.405 gdcmSegmentHelper.h File Reference

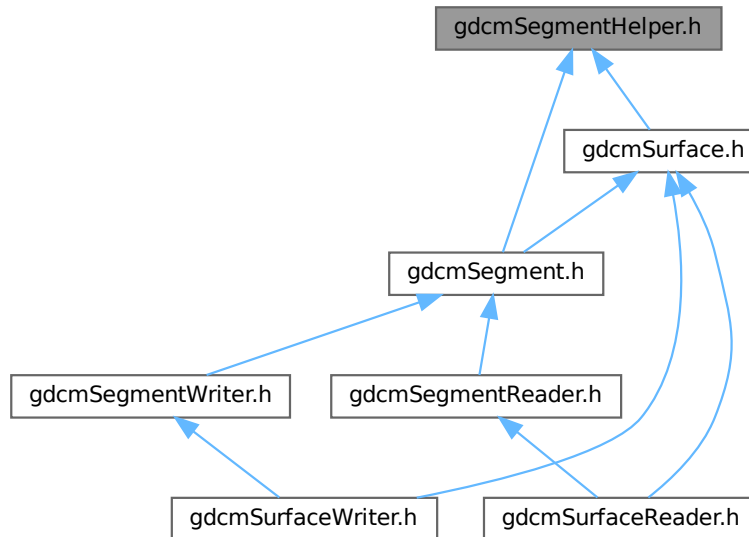
```
#include "gdcmTypes.h"
```

```
#include <string>
```

Include dependency graph for gdcmSegmentHelper.h:



This graph shows which files directly or indirectly include this file:



## Classes

- struct [gdcm::SegmentHelper::BasicCodedEntry](#)  
This structure defines a basic coded entry with all of its attributes.

## Namespaces

- namespace [gdcm](#)
- namespace [gdcm::SegmentHelper](#)

## 11.406 gdcmSegmentHelper.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMSEGMENTHELPER_H

```

```

00015 #define GDCMSEGMENTHELPER_H
00016
00017 #include "gdcmTypes.h"
00018
00019 #include <string>
00020
00021 namespace gdcm
00022 {
00023
00024 namespace SegmentHelper
00025 {
00026
00032 struct GDCM_EXPORT BasicCodedEntry
00033 {
00037     BasicCodedEntry():
00038         CV(""),
00039         CSD(""),
00040         CSV(""),
00041         CM("")
00042     {}
00043
00047     BasicCodedEntry(const char * a_CV,
00048                     const char * a_CSD,
00049                     const char * a_CM):
00050         CV(a_CV),
00051         CSD(a_CSD),
00052         CSV(""),
00053         CM(a_CM)
00054     {}
00055
00059     BasicCodedEntry(const char * a_CV,
00060                     const char * a_CSD,
00061                     const char * a_CSV,
00062                     const char * a_CM):
00063         CV(a_CV),
00064         CSD(a_CSD),
00065         CSV(a_CSV),
00066         CM(a_CM)
00067     {}
00068
00074     bool IsEmpty(const bool checkOptionalAttributes = false) const;
00075
00076
00077     /**      Members      */
00078     // 0008 0100 1   Code Value
00079     std::string CV;
00080     // 0008 0102 1   Coding Scheme Designator
00081     std::string CSD;
00082     // 0008 0103 1C  Coding Scheme Version
00083     std::string CSV;
00084     // 0008 0104 1   Code Meaning
00085     std::string CM;
00086 };
00087
00088 } // end of SegmentHelper namespace
00089
00090 } // end of gdcm namespace
00091
00092 #endif // GDCMSEGMENTHELPER_H

```

## 11.407 gdcmSegmentReader.h File Reference

```

#include <map>
#include <gdcmReader.h>
#include <gdcmSegment.h>

```



## 11.408 gdcmSegmentReader.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMSEGMENTREADER_H
00015 #define GDCMSEGMENTREADER_H
00016
00017 #include <map>
00018
00019 #include <gdcmReader.h>
00020 #include <gdcmSegment.h>
00021
00022 namespace gdcm
00023 {
00024
00025     class GDCM_EXPORT SegmentReader : public Reader
00026     {
00027     public:
00028         typedef std::vector<SmartPointer<Segment>> SegmentVector;
00029
00030         SegmentReader();
00031
00032         ~SegmentReader() override;
00033
00034         bool Read() override; // Set to protected ?
00035
00036         /**      Segment getters/setters      **/
00037         SegmentVector GetSegments() const;
00038         SegmentVector GetSegments();
00039
00040         // unsigned int GetNumberOfSegments();
00041
00042     protected:
00043
00044         typedef std::map<unsigned long, SmartPointer<Segment>> SegmentMap;
00045
00046         bool ReadSegments();
00047
00048         bool ReadSegment(const Item & segmentItem, const unsigned int idx);
00049
00050         SegmentMap Segments; // The key value is item number (in segment sequence)
00051                             // or the surface number (for a surface segmentation).
00052
00053     };
00054
00055 #endif // GDCMSEGMENTREADER_H

```

## 11.409 gdcmSegmentWriter.h File Reference

```

#include <gdcmWriter.h>
#include <gdcmSegment.h>

```



## 11.410 gdcmSegmentWriter.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMSEGMENTWRITER_H
00015 #define GDCMSEGMENTWRITER_H
00016
00017 #include <gdcmWriter.h>
00018 #include <gdcmSegment.h>
00019
00020 namespace gdcm
00021 {
00022
00023   class GDCM_EXPORT SegmentWriter : public Writer
00024   {
00025   public:
00026     typedef std::vector<SmartPointer<Segment>> SegmentVector;
00027
00028     SegmentWriter();
00029
00030     ~SegmentWriter() override;
00031
00032     bool Write() override; // Set to protected ?
00033
00034     /** Segment getters/setters */
00035     unsigned int GetNumberOfSegments() const;
00036     void SetNumberOfSegments(const unsigned int size);
00037
00038     const SegmentVector & GetSegments() const;
00039     SegmentVector & GetSegments();
00040     SmartPointer<Segment> GetSegment(const unsigned int idx = 0) const;
00041
00042     void AddSegment(SmartPointer<Segment> segment);
00043
00044     void SetSegments(SegmentVector & segments);
00045
00046   protected:
00047
00048     bool PrepareWrite();
00049
00050     SegmentVector Segments;
00051   };
00052
00053 #endif // GDCMSEGMENTWRITER_H

```

## 11.411 gdcmSerieHelper.h File Reference

```

#include "gdcmTag.h"
#include "gdcmSmartPointer.h"
#include "gdcmFile.h"
#include <vector>
#include <string>

```





## 11.412 gdcmSerieHelper.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMSERIEHELPER_H
00015  #define GDCMSERIEHELPER_H
00016
00017  #include "gdcmTag.h"
00018  #include "gdcmSmartPointer.h"
00019  #include "gdcmFile.h"
00020  #include <vector>
00021  #include <string>
00022  #include <map>
00023
00024  namespace gdcm
00025  {
00026
00027  enum CompOperators {
00028      GDCM_EQUAL = 0,
00029      GDCM_DIFFERENT,
00030      GDCM_GREATER,
00031      GDCM_GREATEROREQUAL,
00032      GDCM_LESS,
00033      GDCM_LESSOREQUAL
00034  };
00035  enum LodModeType
00036  {
00037      LD_ALL          = 0x00000000,
00038      LD_NOSEQ        = 0x00000001,
00039      LD_NOSHADOW     = 0x00000002,
00040      LD_NOSHADOWSEQ = 0x00000004
00041  };
00042
00043  class GDCM_EXPORT FileWithName : public File
00044  {
00045  public:
00046      FileWithName(File &f):File(f),filename(){}
00047      std::string filename;
00048  };
00049
00050  typedef std::vector< SmartPointer<FileWithName> > FileList;
00051  typedef bool (*BOOL_FUNCTION_PFILE_PFILE_POINTER) (File *, File *);
00052  class Scanner;
00053
00054  class GDCM_EXPORT SerieHelper
00055  {
00056  public:
00057      SerieHelper();
00058      ~SerieHelper();
00059
00060      void Clear();
00061      void SetLoadMode (int ) {}
00062      void SetDirectory(std::string const &dir, bool recursive=false);
00063
00064      void AddRestriction(const std::string & tag);
00065      void SetUseSeriesDetails( bool useSeriesDetails );
00066      void CreateDefaultUniqueSeriesIdentifier();
00067      FileList *GetFirstSingleSeriesUIDFileSet();
00068      FileList *GetNextSingleSeriesUIDFileSet();
00069      std::string CreateUniqueSeriesIdentifier( File * inFile );
00070      void OrderFileList(FileList *fileSet);
00071      void AddRestriction(uint16_t group, uint16_t elem, std::string const &value, int op);
00072
00073  protected:
00074      bool UserOrdering(FileList *fileSet);
00075      void AddFileName(std::string const &filename);

```

```

00090  bool AddFile(FileWithName &header);
00091  void AddRestriction(const Tag& tag);
00092  bool ImagePositionPatientOrdering(FileList *fileSet);
00093  bool ImageNumberOrdering( FileList *fileList );
00094  bool FileNameOrdering( FileList *fileList );
00095
00096  using Rule = struct RuleStructure{
00097      uint16_t group;
00098      uint16_t elem;
00099      std::string value;
00100      int op;
00101  };
00102  typedef std::vector<Rule> SerieRestrictions;
00103
00104  typedef std::map<std::string, FileList *> SingleSerieUIDFileSetmap;
00105  SingleSerieUIDFileSetmap SingleSerieUIDFileSetHT;
00106  SingleSerieUIDFileSetmap::iterator ItFileSetHT;
00107
00108 private:
00109     SerieRestrictions Restrictions;
00110     SerieRestrictions Refine;
00111
00112     bool UseSeriesDetails;
00113     bool DirectOrder;
00114
00115     BOOL_FUNCTION_PFILE_PFILE_POINTER UserLessThanFunction;
00116 };
00117
00118 // backward compat
00119 } // end namespace gdcm
00120
00121
00122 #endif //GDCMSERIEHELPER_H

```

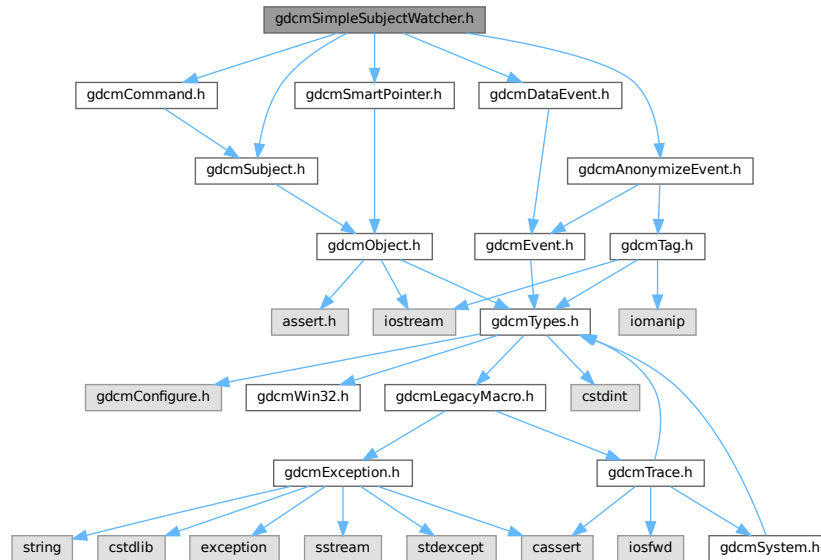
## 11.413 gdcmSimpleSubjectWatcher.h File Reference

```

#include "gdcmSubject.h"
#include "gdcmCommand.h"
#include "gdcmSmartPointer.h"
#include "gdcmAnonymizeEvent.h"
#include "gdcmDataEvent.h"

```

Include dependency graph for gdcmsimpleSubjectWatcher.h:



## Classes

- class [gdcmsimpleSubjectWatcher](#)  
*SimpleSubjectWatcher.*

## Namespaces

- namespace [gdcmsimple](#)

## 11.414 gdcmsimpleSubjectWatcher.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMSIMPLESUBJECTWATCHER_H
00015 #define GDCMSIMPLESUBJECTWATCHER_H
00016
00017 #include "gdcmsimpleSubject.h"
00018 #include "gdcmsimpleCommand.h"

```

```

00019 #include "gdcmSmartPointer.h"
00020 #include "gdcmAnonymizeEvent.h"
00021 #include "gdcmDataEvent.h"
00022
00023 namespace gdcm
00024 {
00025 //-----
00026 class Event;
00031 class GDCM_EXPORT SimpleSubjectWatcher
00032 {
00033 public:
00034     SimpleSubjectWatcher(Subject * s, const char *comment = "");
00035     virtual ~SimpleSubjectWatcher();
00036     SimpleSubjectWatcher(const SimpleSubjectWatcher&) = delete;
00037     void operator=(const SimpleSubjectWatcher&) = delete;
00038
00039 protected:
00040     virtual void StartFilter();
00041     virtual void EndFilter();
00042     virtual void ShowProgress(Subject *caller, const Event &evt);
00043     virtual void ShowFileName(Subject *caller, const Event &evt);
00044     virtual void ShowIteration();
00045     virtual void ShowAnonymization(Subject *caller, const Event &evt);
00046     virtual void ShowDataSet(Subject *caller, const Event &evt);
00047     virtual void ShowData(Subject *caller, const Event &evt);
00048     virtual void ShowAbort();
00049
00050 protected:
00051     // Custom API used for internal Testing do not use !
00052     void TestAbortOn();
00053     void TestAbortOff();
00054
00055 private:
00056     SmartPointer<Subject> m_Subject;
00057     std::string m_Comment;
00058
00059     typedef SimpleMemberCommand<SimpleSubjectWatcher> SimpleCommandType;
00060     typedef MemberCommand<SimpleSubjectWatcher> CommandType;
00061
00062     SmartPointer<SimpleCommandType> m_StartFilterCommand;
00063     SmartPointer<SimpleCommandType> m_EndFilterCommand;
00064     SmartPointer<CommandType> m_ProgressFilterCommand;
00065     SmartPointer<CommandType> m_FileNameFilterCommand;
00066     SmartPointer<SimpleCommandType> m_IterationFilterCommand;
00067     SmartPointer<SimpleCommandType> m_AbortFilterCommand;
00068     SmartPointer<CommandType> m_AnonymizeFilterCommand;
00069     SmartPointer<CommandType> m_DataFilterCommand;
00070     SmartPointer<CommandType> m_DataSetFilterCommand;
00071
00072     unsigned long m_StartTag;
00073     unsigned long m_EndTag;
00074     unsigned long m_ProgressTag;
00075     unsigned long m_FileNameTag;
00076     unsigned long m_IterationTag;
00077     unsigned long m_AbortTag;
00078     unsigned long m_AnonymizeTag;
00079     unsigned long m_DataTag;
00080     unsigned long m_DataSetTag;
00081
00082     bool m_TestAbort;
00083
00084 };
00085 } // end namespace gdcm
00086 //-----
00087 #endif //GDCMSIMPLESUBJECTWATCHER_H

```

## 11.415 gdcmSorter.h File Reference

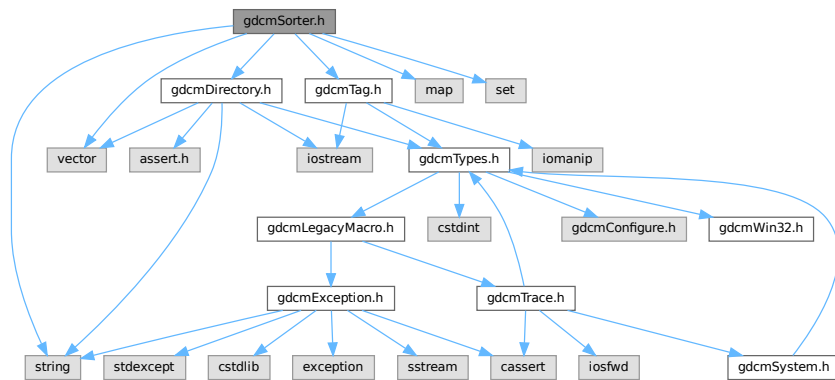
```

#include "gdcmDirectory.h"
#include "gdcmTag.h"
#include <vector>
#include <string>
#include <map>

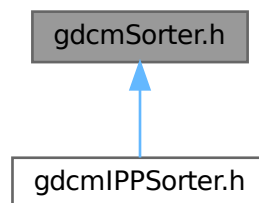
```

```
#include <set>
```

Include dependency graph for gdcmSorter.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::Sorter](#)  
*Sorter.*

## Namespaces

- namespace [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Sorter &s)`

## 11.416 gdcmSorter.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMSORTER_H
00015 #define GDCMSORTER_H
00016
00017 #include "gdcmDirectory.h"
00018 #include "gdcmTag.h"
00019
00020 #include <vector>
00021 #include <string>
00022 #include <map>
00023 #include <set>
00024
00025 namespace gdcm
00026 {
00027   class DataSet;
00028
00029   class GDCM_EXPORT Sorter
00030   {
00031   {
00041     friend std::ostream& operator<<(std::ostream &_os, const Sorter &s);
00042   public:
00043     Sorter();
00044     virtual ~Sorter();
00045
00047     virtual bool Sort(std::vector<std::string> const & filenames);
00048
00051     const std::vector<std::string> &GetFileNames() const { return FileNames; }
00052
00054     void Print(std::ostream &os) const;
00055
00057     bool AddSelect( Tag const &tag, const char *value );
00058
00062     void SetTagsToRead( std::set<Tag> const & tags );
00063
00065     typedef bool (*SortFunction)(DataSet const &, DataSet const &);
00066     void SetSortFunction( SortFunction f );
00067
00068     virtual bool StableSort(std::vector<std::string> const & filenames);
00069
00070   protected:
00071     std::vector<std::string> FileNames;
00072     typedef std::map<Tag, std::string> SelectionMap;
00073     std::map<Tag, std::string> Selection;
00074     SortFunction SortFunc;
00075     std::set<Tag> TagsToRead;
00076   };
00077   //-----
00078   inline std::ostream& operator<<(std::ostream &os, const Sorter &s)
00079   {
00080     s.Print( os );
00081     return os;
00082   }
00083
00084
00085 } // end namespace gdcm
00086
00087 #endif //GDCMSORTER_H

```







## Classes

- class [gdcm::Spectroscopy](#)  
*Spectroscopy* class.

## Namespaces

- namespace [gdcm](#)

## 11.420 gdcmSpectroscopy.h

[Go to the documentation of this file.](#)

```
00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMSPECTROSCOPY_H
00015 #define GDCMSPECTROSCOPY_H
00016
00017 #include "gdcmFile.h"
00018
00019 namespace gdcm
00020 {
00024   class GDCM_EXPORT Spectroscopy
00025   {
00026   public:
00027     Spectroscopy() = default;
00028
00029   private:
00030   };
00031
00032 } // end namespace gdcm
00033
00034 #endif //GDCMSPECTROSCOPY_H
```

## 11.421 gdcmSplitMosaicFilter.h File Reference

```
#include "gdcmFile.h"
#include "gdcmImage.h"
```



## 11.423 gdcmStreamImageReader.h File Reference

Include dependency graph for gdcmStreamImageReader.h:



## Classes

- class [gdcm::StreamImageReader](#)  
*StreamImageReader.*

## Namespaces

- namespace [gdcm](#)

## 11.424 gdcmStreamImageReader.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  * http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *=====*/
00017 #ifndef GDCMSTREAMIMAGEREADER_H
00018 #define GDCMSTREAMIMAGEREADER_H
00019
00020 #include "gdcmReader.h"
00021
00022 namespace gdcm
00023 {
00024     {
00025         class MediaStorage;
00026         class GDCM_EXPORT StreamImageReader
00027         {
00028         public:
00029             StreamImageReader();
00030             virtual ~StreamImageReader();
00031
00032             void SetFileName(const char* inFileName);
00033             void SetStream(std::istream& inStream);
00034
00035             std::vector<unsigned int> GetDimensionsValueForResolution( unsigned int );
00036
00037             void DefinePixelExtent(uint16_t inXMin, uint16_t inXMax,
00038                                     uint16_t inYMin, uint16_t inYMax, uint16_t inZMin = 0, uint16_t inZMax = 1);
00039
00040             uint32_t DefineProperBufferLength() const;
00041
00042             bool Read(char* inReadBuffer, const std::size_t& inBufferLength);
00043
00044             bool CanReadImage() const;
00045
00046             virtual bool ReadImageInformation();
00047
00048             File const & GetFile() const;
00049
00050 protected:
00051 private:
00052             //contains a reader for being able to ReadUpToTag
00053             //however, we don't want the user to be able to call Read
00054             //either directly or via a parent class call, so we hide the reader in here.
00055             Reader mReader;
00056         }
00057     }
00058 }

```

## 11.425 gdcmStreamImageWriter.h File Reference

[illegible]

- class `gdcm::StreamImageWriter`  
*StreamImageReader*.

- namespace **gdcm**

## 11.426 gdcmStreamImageWriter.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  * http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018
00019 #ifndef GDCMSTREAMIMAGEWRITER_H
00020 #define GDCMSTREAMIMAGEWRITER_H
00021
00022 #include "gdcmWriter.h"
00023 #include <iostream>
00024 #include "gdcmDataSet.h"
00025
00026 namespace gdcm
00027 {
00028
00029 class MediaStorage;
00030 class RAWCodec;
00042 class GDCM_EXPORT StreamImageWriter
00043 {
00044
00045 public:
00046     StreamImageWriter();
00047     virtual ~StreamImageWriter();
00048
00049
00053     void SetFileName(const char* inFileName);
00054     void SetStream(std::ostream& inStream);
00055
00064     void DefinePixelExtent(uint16_t inXMin, uint16_t inXMax,
00065         uint16_t inYMin, uint16_t inYMax, uint16_t inZMin = 0, uint16_t inZMax = 1);
00066
00067
00073     uint32_t DefineProperBufferLength();
00074
00082     bool Write(void* inWriteBuffer, const std::size_t inBufferLength);
00083
00087     virtual bool WriteImageInformation();
00088
00092     bool CanWriteFile() const;
00093
00094
00097     void SetFile(const File& inFile);
00098
00099 protected:
00100
00101     //contains the PrepareWrite function, which will get the given dataset ready
00102     //for writing to disk by manufacturing the header information.
00103     //note that if there is a pixel element in the given dataset, that will be removed
00104     //during the copy, so that the imagewriter can write everything else out
00105     Writer mWriter;
00106
00107     //is the offset necessary if we always append?
00108     //std::streamoff mFileOffset; //the fileoffset for getting header information
00109     SmartPointer<File> mspFile; //all the non-pixel information
00110
00111     //for thread safety, these should not be stored here, but should be used
00112     //for every read subregion operation.
00113     uint16_t mXMin, mYMin, mXMax, mYMax, mZMin, mZMax;
00114
00119     //virtual bool ReadImageSubregionRAW(std::ostream& os);
00120     virtual bool WriteImageSubregionRAW(char* inWriteBuffer, const std::size_t inBufferLength);
00121
00131     int WriteRawHeader(RAWCodec* inCodec, std::ostream* inStream);

```

```

00132
00137     int mElementOffsets;
00138     int mElementOffsets1;
00139
00140 };
00141
00142
00143 } // end namespace gdcm
00144
00145 #endif //GDCMSTREAMIMAGEWRITER_H

```

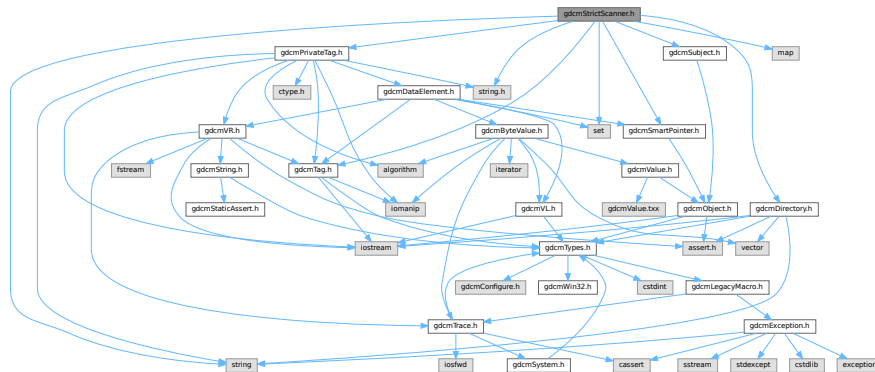
## 11.427 gdcmStrictScanner.h File Reference

```

#include "gdcmDirectory.h"
#include "gdcmSubject.h"
#include "gdcmTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmSmartPointer.h"
#include <map>
#include <set>
#include <string>
#include <string.h>

```

Include dependency graph for gdcmStrictScanner.h:



### Classes

- struct [gdcm::StrictScanner::Itstr](#)
- class [gdcm::StrictScanner](#)  
*StrictScanner.*

### Namespaces

- namespace [gdcm](#)

### Functions

- [std::ostream & gdcm::operator<<](#) ([std::ostream &os](#), const [StrictScanner](#) &s)

## 11.428 gdcmStrictScanner.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMSTRICTSCANNER_H
00015  #define GDCMSTRICTSCANNER_H
00016
00017  #include "gdcmDirectory.h"
00018  #include "gdcmSubject.h"
00019  #include "gdcmTag.h"
00020  #include "gdcmPrivateTag.h"
00021  #include "gdcmSmartPointer.h"
00022
00023  #include <map>
00024  #include <set>
00025  #include <string>
00026
00027  #include <string.h> // strcmp
00028
00029  namespace gdcm
00030  {
00031  class StringFilter;
00032
00033  class GDCM_EXPORT StrictScanner : public Subject
00034  {
00035  {
00036  friend std::ostream& operator<<(std::ostream &_os, const StrictScanner &s);
00037  public:
00038  StrictScanner():Values(),FileNames(),Mappings() {}
00039  ~StrictScanner() override;
00040
00041  typedef std::map<Tag, const char*> TagToValue;
00042  //typedef std::map<Tag, ConstCharWrapper> TagToValue; //StringMap;
00043  //typedef TagToStringMap TagToValue;
00044  typedef TagToValue::value_type TagToValueValueType;
00045
00046  void AddTag( Tag const & t );
00047  void ClearTags();
00048
00049  // Work in progress do not use:
00050  void AddPrivateTag( PrivateTag const & t );
00051
00052  void AddSkipTag( Tag const & t );
00053  void ClearSkipTags();
00054
00055  bool Scan( Directory::FileNamesType const & filenames );
00056
00057  Directory::FileNamesType const &GetFileNames() const { return FileNames; }
00058
00059  void Print( std::ostream & os ) const override;
00060
00061  void PrintTable( std::ostream & os ) const;
00062
00063  bool IsKey( const char * filename ) const;
00064
00065  Directory::FileNamesType GetKeys() const;
00066
00067  // struct to store all the values found:
00068  typedef std::set< std::string > ValueType;
00069
00070  ValueType const & GetValues() const { return Values; }
00071
00072  ValueType GetValues(Tag const &t) const;
00073
00074  Directory::FileNamesType GetOrderedValues(Tag const &t) const;
00075
00076  /* ltstr is CRITICAL, otherwise pointers value are used to do the key comparison */
00077  struct ltstr

```



```

00119     {
00120     bool operator()(const char* s1, const char* s2) const
00121     {
00122         assert( s1 && s2 );
00123         return strcmp(s1, s2) < 0;
00124     }
00125     };
00126     typedef std::map<const char *,TagToValue, ltstr> MappingType;
00127     typedef MappingType::const_iterator ConstIterator;
00128     ConstIterator Begin() const { return Mappings.begin(); }
00129     ConstIterator End() const { return Mappings.end(); }
00130
00131     MappingType const & GetMappings() const { return Mappings; }
00132
00133     TagToValue const & GetMapping(const char *filename) const;
00134
00135     const char *GetFilenameFromTagToValue(Tag const &t, const char *valueref) const;
00136
00137     Directory::FileNamesType GetAllFileNamesFromTagToValue(Tag const &t, const char *valueref) const;
00138
00139     // by a call to GetMapping()
00140     TagToValue const & GetMappingFromTagToValue(Tag const &t, const char *value) const;
00141
00142     const char* GetValue(const char *filename, Tag const &t) const;
00143
00144     static SmartPointer<StrictScanner> New() { return new StrictScanner; }
00145
00146 protected:
00147     void ProcessPublicTag(StringFilter &sf, const char *filename);
00148 private:
00149     // struct to store all uniq tags in ascending order:
00150     typedef std::set< Tag > TagsType;
00151     typedef std::set< PrivateTag > PrivateTagsType;
00152     std::set< Tag > Tags;
00153     std::set< PrivateTag > PrivateTags;
00154     std::set< Tag > SkipTags;
00155     ValuesType Values;
00156     Directory::FileNamesType FileNames;
00157
00158     // Main struct that will hold all mapping:
00159     MappingType Mappings;
00160
00161     double Progress;
00162 };
00163 //-----
00164 inline std::ostream& operator<<(std::ostream &os, const StrictScanner &s)
00165 {
00166     s.Print( os );
00167     return os;
00168 }
00169
00170 } // end namespace gdcm
00171
00172 #endif //GDCMSTRICTSCANNER_H

```

## 11.429 gdcmStrictScanner2.h File Reference

```

#include "gdcmDirectory.h"
#include "gdcmPrivateTag.h"
#include "gdcmSmartPointer.h"
#include "gdcmSubject.h"
#include "gdcmTag.h"
#include <map>
#include <set>
#include <string>
#include <string.h>

```



```

00019 #include "gdcmSmartPointer.h"
00020 #include "gdcmSubject.h"
00021 #include "gdcmTag.h"
00022
00023 #include <map>
00024 #include <set>
00025 #include <string>
00026
00027 #include <string.h> // strcmp
00028
00029 namespace gdcm {
00030 class StringFilter;
00031
00032 class GDCM_EXPORT StrictScanner2 : public Subject {
00033     friend std::ostream &operator<<(std::ostream &_os, const StrictScanner2 &s);
00034
00035 public:
00036     StrictScanner2() : Values(), Filenames(), PublicMappings(), PrivateMappings(), Progress(0.0) {}
00037     ~StrictScanner2() override;
00038
00039     typedef std::map<Tag, const char *> PublicTagToValue;
00040     typedef PublicTagToValue::value_type PublicTagToValueValueType;
00041
00042     typedef std::map<PrivateTag, const char *> PrivateTagToValue;
00043     typedef PrivateTagToValue::value_type PrivateTagToValueValueType;
00044
00045     bool AddPublicTag(Tag const &t);
00046     void ClearPublicTags();
00047
00048     // Work in progress do not use:
00049     bool AddPrivateTag(PrivateTag const &pt);
00050     void ClearPrivateTags();
00051
00052     bool AddSkipTag(Tag const &t);
00053     void ClearSkipTags();
00054
00055     bool Scan(Directory::FileNamesType const &filenames);
00056
00057     Directory::FileNamesType const &GetFilenames() const { return Filenames; }
00058
00059     void Print(std::ostream &os) const override;
00060
00061     void PrintTable(std::ostream &os, bool header = false) const;
00062
00063     bool IsKey(const char *filename) const;
00064
00065     Directory::FileNamesType GetKeys() const;
00066
00067     // struct to store all the values found:
00068     typedef std::set<std::string> ValueType;
00069
00070     ValueType const &GetValues() const { return Values; }
00071
00072     ValueType GetPublicValues(Tag const &t) const;
00073
00074     ValueType GetPrivateValues(PrivateTag const &pt) const;
00075
00076     Directory::FileNamesType GetPublicOrderedValues(Tag const &t) const;
00077
00078     Directory::FileNamesType GetPrivateOrderedValues(PrivateTag const &pt) const;
00079
00080     /* ltstr is CRITICAL, otherwise pointers value are used to do the key
00081     * comparison */
00082     struct ltstr {
00083         bool operator()(const char *s1, const char *s2) const {
00084             assert(s1 && s2);
00085             return strcmp(s1, s2) < 0;
00086         }
00087     };
00088
00089     typedef std::map<const char *, PublicTagToValue, ltstr> PublicMappingType;
00090     typedef PublicMappingType::const_iterator PublicConstIterator;
00091     PublicConstIterator Begin() const { return PublicMappings.begin(); }
00092     PublicConstIterator End() const { return PublicMappings.end(); }
00093
00094     typedef std::map<const char *, PrivateTagToValue, ltstr> PrivateMappingType;
00095     typedef PrivateMappingType::const_iterator PrivateConstIterator;
00096     PrivateConstIterator PrivateBegin() const { return PrivateMappings.begin(); }
00097     PrivateConstIterator PrivateEnd() const { return PrivateMappings.end(); }
00098
00099     PublicMappingType const &GetPublicMappings() const { return PublicMappings; }
00100     PrivateMappingType const &GetPrivateMappings() const {

```

```

00146     return PrivateMappings;
00147 }
00148
00150 PublicTagToValue const &GetPublicMapping(const char *filename) const;
00151 PrivateTagToValue const &GetPrivateMapping(const char *filename) const;
00152
00155 const char *GetFilenameFromPublicTagToValue(Tag const &t,
00156                                             const char *valueref) const;
00157 const char *GetFilenameFromPrivateTagToValue(PrivateTag const &pt,
00158                                             const char *valueref) const;
00159
00162 Directory::FileNamesType GetAllFileNamesFromPublicTagToValue(
00163     Tag const &t, const char *valueref) const;
00164 Directory::FileNamesType GetAllFileNamesFromPrivateTagToValue(
00165     PrivateTag const &pt, const char *valueref) const;
00166
00169 // by a call to GetMapping()
00170 PublicTagToValue const &GetMappingFromPublicTagToValue(
00171     Tag const &t, const char *value) const;
00172 PrivateTagToValue const &GetMappingFromPrivateTagToValue(
00173     PrivateTag const &pt, const char *value) const;
00174
00180 const char *GetPublicValue(const char *filename, Tag const &t) const;
00181 const char *GetPrivateValue(const char *filename, PrivateTag const &t) const;
00182
00184 static SmartPointer<StrictScanner2> New() { return new StrictScanner2; }
00185
00186 protected:
00187 void ProcessPublicTag(StringFilter &sf, const char *filename);
00188 void ProcessPrivateTag(StringFilter &sf, const char *filename);
00189
00190 private:
00191 // struct to store all uniq tags in ascending order:
00192 typedef std::set<Tag> PublicTagsType;
00193 typedef std::set<PrivateTag> PrivateTagsType;
00194 std::set<Tag> PublicTags; // Public and Private Creator
00195 std::set<PrivateTag> PrivateTags; // Only Private (no Private Creator)
00196 std::set<Tag> SkipTags;
00197 ValueType Values;
00198 Directory::FileNamesType FileNames;
00199
00200 // Main struct that will hold all public mapping:
00201 PublicMappingType PublicMappings;
00202 // Main struct that will hold all private mapping:
00203 PrivateMappingType PrivateMappings;
00204
00205 double Progress;
00206 };
00207 //-----
00208 inline std::ostream &operator<<(std::ostream &os, const StrictScanner2 &s) {
00209     s.Print(os);
00210     return os;
00211 }
00212
00213 } // end namespace gdcm
00214
00215 #endif // GDCMSTRICTSCANNER2_H

```

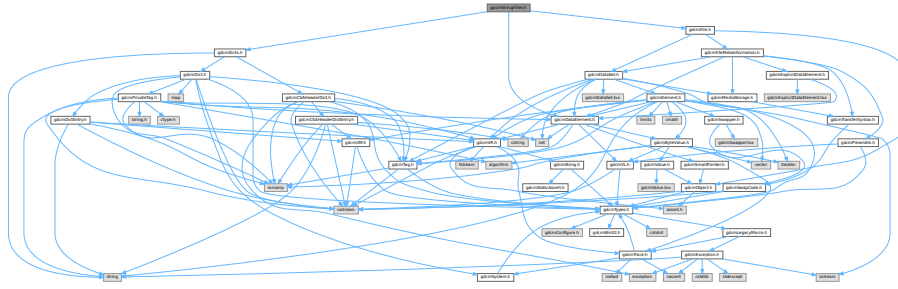
## 11.431 gdcmStringFilter.h File Reference

```

#include "gdcmDataElement.h"
#include "gdcmDicts.h"
#include "gdcmFile.h"

```

Include dependency graph for gdcmStringFilter.h:



## Classes

- class `gdcm::StringFilter`  
*StringFilter.*

## Namespaces

- namespace `gdcm`

## 11.432 gdcmStringFilter.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 #ifndef GDCMSTRINGFILTER_H
00015 #define GDCMSTRINGFILTER_H
00016
00017 #include "gdcmDataElement.h"
00018 #include "gdcmDicts.h"
00019 #include "gdcmFile.h"
00020
00021 namespace gdcm
00022 {
00023
00024     class GDCM_EXPORT StringFilter
00025     {
00026     public:
00027         StringFilter();
00028         ~StringFilter();
00029
00030         void UseDictAlways(bool) {}
00031
00032         void SetDicts(const Dicts &dicts);
00033
00034         std::string ToString(const DataElement& de) const;
00035
00036     };
00037
00038 }
00039
00040
00041
00042
00043
00044
00045

```

```

00047     std::string ToString(const Tag& t) const;
00048
00049     std::string ToString(const PrivateTag& t) const;
00050
00055     std::pair<std::string, std::string> ToStringPair(const DataElement& de) const;
00057     std::pair<std::string, std::string> ToStringPair(const Tag& t) const;
00058
00060     std::string FromString(const Tag&t, const char * value, size_t len);
00061
00063     void SetFile(const File& f) { F = f; }
00064     File &GetFile() { return *F; }
00065     const File &GetFile() const { return *F; }
00066
00070     bool ExecuteQuery(std::string const &query, std::string & value) const;
00071
00072 protected:
00073     std::pair<std::string, std::string> ToStringPair(const Tag& t, DataSet const &ds) const;
00074     bool ExecuteQuery(std::string const &query, DataSet const &ds, std::string & value) const;
00075
00076 private:
00077     std::pair<std::string, std::string> ToStringPairInternal(const DataElement& de, DataSet const &ds)
00078     const;
00078     SmartPointer<File> F;
00079 };
00080
00081 } // end namespace gdcmm
00082
00083 #endif //GDCMSTRINGFILTER_H

```

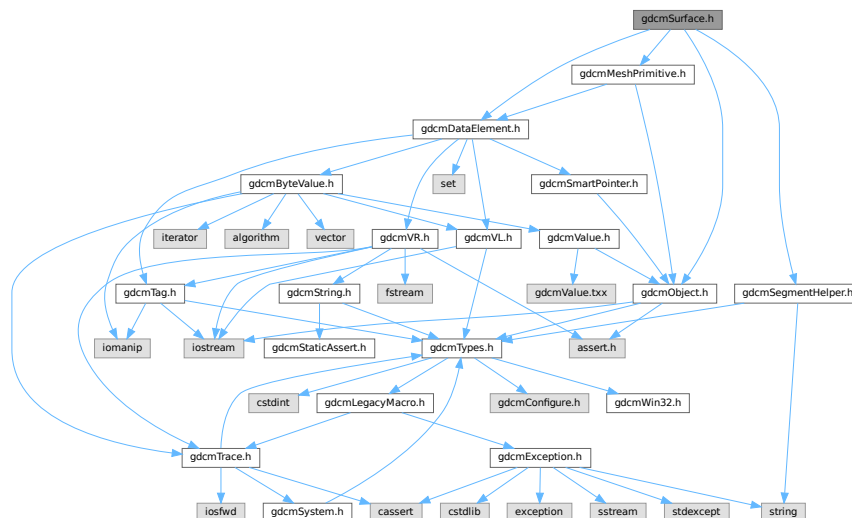
## 11.433 gdcmmSurface.h File Reference

```

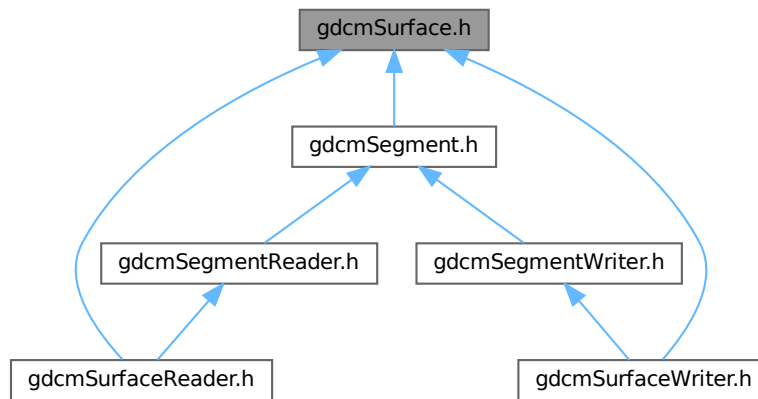
#include <gdcmmObject.h>
#include <gdcmmDataElement.h>
#include <gdcmmMeshPrimitive.h>
#include "gdcmmSegmentHelper.h"

```

Include dependency graph for gdcmmSurface.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::Surface](#)  
This class defines a SURFACE IE.

## Namespaces

- namespace [gdcm](#)

## 11.434 gdcmSurface.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMSURFACE_H
00015 #define GDCMSURFACE_H
00016
00017 #include <gdcmObject.h>
00018 #include <gdcmDataElement.h>
00019 #include <gdcmMeshPrimitive.h>
00020 #include "gdcmSegmentHelper.h" // for BasicCodedEntry
00021
00022 namespace gdcm
00023 {

```

```

00024
00031 class GDCM_EXPORT Surface : public Object
00032 {
00033 public:
00034
00035     typedef enum {
00036         NO = 0,
00037         YES,
00038         UNKNOWN,
00039         STATES_END
00040     } STATES;
00041
00042     static const char * GetSTATESString(STATES state);
00043     static STATES GetSTATES(const char * state);
00044
00045     typedef enum {
00046         SURFACE = 0,
00047         WIREFRAME,
00048         POINTS,
00049         VIEWType_END
00050     } VIEWType;
00051
00052     static const char * GetVIEWTypeString(VIEWType type);
00053     static VIEWType GetVIEWType(const char * type);
00054
00055     Surface();
00056
00057     ~Surface() override;
00058
00059     /** Common getters/setters */
00060     unsigned long GetSurfaceNumber() const;
00061     void SetSurfaceNumber(const unsigned long nb);
00062
00063     const char * GetSurfaceComments() const;
00064     void SetSurfaceComments(const char * comment);
00065
00066     bool GetSurfaceProcessing() const;
00067     void SetSurfaceProcessing(bool b);
00068
00069     float GetSurfaceProcessingRatio() const;
00070     void SetSurfaceProcessingRatio(const float ratio);
00071
00072     const char * GetSurfaceProcessingDescription() const;
00073     void SetSurfaceProcessingDescription(const char * description);
00074
00075     SegmentHelper::BasicCodedEntry const & GetProcessingAlgorithm() const;
00076     SegmentHelper::BasicCodedEntry & GetProcessingAlgorithm();
00077     void SetProcessingAlgorithm(SegmentHelper::BasicCodedEntry const & BSE);
00078
00079     unsigned short GetRecommendedDisplayGrayscaleValue() const;
00080     void SetRecommendedDisplayGrayscaleValue(const unsigned short vl);
00081
00082     const unsigned short * GetRecommendedDisplayCIELabValue() const;
00083     unsigned short GetRecommendedDisplayCIELabValue(const unsigned int idx) const;
00084     void SetRecommendedDisplayCIELabValue(const unsigned short vl[3]);
00085     void SetRecommendedDisplayCIELabValue(const unsigned short vl, const unsigned int idx = 0);
00086     void SetRecommendedDisplayCIELabValue(const std::vector< unsigned short > & vl);
00087
00088     float GetRecommendedPresentationOpacity() const;
00089     void SetRecommendedPresentationOpacity(const float opacity);
00090
00091     VIEWType GetRecommendedPresentationType() const;
00092     void SetRecommendedPresentationType(VIEWType type);
00093
00094     STATES GetFiniteVolume() const;
00095     void SetFiniteVolume(STATES state);
00096
00097     STATES GetManifold() const;
00098     void SetManifold(STATES state);
00099
00100     SegmentHelper::BasicCodedEntry const & GetAlgorithmFamily() const;
00101     SegmentHelper::BasicCodedEntry & GetAlgorithmFamily();
00102     void SetAlgorithmFamily(SegmentHelper::BasicCodedEntry const & BSE);
00103
00104     const char * GetAlgorithmVersion() const;
00105     void SetAlgorithmVersion(const char * str);
00106
00107     const char * GetAlgorithmName() const;
00108     void SetAlgorithmName(const char * str);
00109
00110     /** Points getters/setters */

```



```

00116 unsigned long GetNumberOfSurfacePoints() const;
00117 void SetNumberOfSurfacePoints(const unsigned long nb);
00118
00119 const DataElement & GetPointCoordinatesData() const;
00120 DataElement & GetPointCoordinatesData();
00121
00122 void SetPointCoordinatesData(DataElement const & de);
00123
00127 const float * GetPointPositionAccuracy() const;
00128 void SetPointPositionAccuracy(const float * accuracies);
00129
00130 float GetMeanPointDistance() const;
00131 void SetMeanPointDistance(float average);
00132
00133 float GetMaximumPointDistance() const;
00134 void SetMaximumPointDistance(float maximum);
00135
00139 const float * GetPointsBoundingBoxCoordinates() const;
00140 void SetPointsBoundingBoxCoordinates(const float * coordinates);
00141
00145 const float * GetAxisOfRotation() const;
00146 void SetAxisOfRotation(const float * axis);
00147
00151 const float * GetCenterOfRotation() const;
00152 void SetCenterOfRotation(const float * center);
00153
00154 /** Vectors getters/setters */
00155 unsigned long GetNumberOfVectors() const;
00156 void SetNumberOfVectors(const unsigned long nb);
00157
00158 unsigned short GetVectorDimensionality() const;
00159 void SetVectorDimensionality(const unsigned short dim);
00160
00161 const float * GetVectorAccuracy() const;
00162 void SetVectorAccuracy(const float * accuracy);
00163
00164 const DataElement & GetVectorCoordinateData() const;
00165 DataElement & GetVectorCoordinateData();
00166
00167 void SetVectorCoordinateData(DataElement const & de);
00168
00169 /** Primitive getters/setters */
00170 MeshPrimitive const & GetMeshPrimitive() const;
00171 MeshPrimitive & GetMeshPrimitive();
00172
00173 void SetMeshPrimitive(MeshPrimitive & mp);
00174
00175 private:
00176
00177 /** Common members */
00178
00179 //0066 0003 UL 1 Surface Number
00180 unsigned long SurfaceNumber;
00181 //0066 0004 LT 1 Surface Comments
00182 std::string SurfaceComments;
00183
00184 //0066 0009 CS 1 Surface Processing
00185 bool SurfaceProcessing;
00186 //0066 000a FL 1 Surface Processing Ratio
00187 float SurfaceProcessingRatio;
00188 //0066 000b LO 1 Surface Processing Description
00189 std::string SurfaceProcessingDescription;
00190 // Processing Algorithm Code
00191 SegmentHelper::BasicCodedEntry ProcessingAlgorithm;
00192
00193 //0062 000c US 1 Recommended Display Grayscale Value
00194 unsigned short RecommendedDisplayGrayscaleValue;
00195 //0062 000d US 3 Recommended Display CIELab Value
00196 unsigned short RecommendedDisplayCIELabValue[3];
00197
00198 // 0066 000c FL 1 Recommended Presentation Opacity
00199 float RecommendedPresentationOpacity;
00200 // 0066 000d CS 1 Recommended Presentation Type
00201 VIEWType RecommendedPresentationType;
00202
00203 //0066 000e CS 1 Finite Volume
00204 STATES FiniteVolume;
00205 //0066 0010 CS 1 Manifold
00206 STATES Manifold;
00207
00208 // Algorithm Family Code

```

```

00209 SegmentHelper::BasicCodedEntry AlgorithmFamily;
00210
00211 //0066 0031 LO 1 Algorithm Version
00212 std::string AlgorithmVersion;
00213 //0066 0032 LT 1 Algorithm Parameters
00214 //0066 0036 LO 1 Algorithm Name
00215 std::string AlgorithmName;
00216
00217
00218 /**      Point members      **/
00219
00220 //0066 0015 UL 1 Number of Surface Points
00221 unsigned long NumberOfSurfacePoints;
00222 //0066 0016 OF 1 Point Coordinates Data
00223 DataElement PointCoordinatesData;
00224 //0066 0017 FL 3 Point Position Accuracy
00225 float * PointPositionAccuracy;
00226 //0066 0018 FL 1 Mean Point Distance
00227 float MeanPointDistance;
00228 //0066 0019 FL 1 Maximum Point Distance
00229 float MaximumPointDistance;
00230 //0066 001a FL 6 Points Bounding Box Coordinates
00231 float * PointsBoundingBoxCoordinates;
00232 //0066 001b FL 3 Axis of Rotation
00233 float * AxisOfRotation;
00234 //0066 001c FL 3 Center of Rotation
00235 float * CenterOfRotation;
00236
00237
00238 /**      Normal members      **/
00239
00240 //0066 001e UL 1 Number of Vectors
00241 unsigned long NumberOfVectors;
00242 //0066 001f US 1 Vector Dimensionality
00243 unsigned short VectorDimensionality;
00244 //0066 0020 FL 1-n Vector Accuracy
00245 float * VectorAccuracy;
00246 //0066 0021 OF 1 Vector Coordinate Data
00247 DataElement VectorCoordinateData;
00248
00249
00250 /**      Primitive members      **/
00251 SmartPointer< MeshPrimitive > Primitive;
00252 };
00253
00254 }
00255
00256 #endif // GDCMSURFACE_H

```

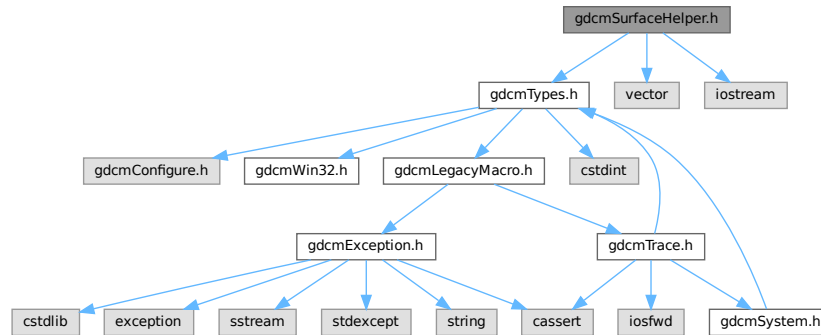
## 11.435 gdcmSurfaceHelper.h File Reference

```

#include "gdcmTypes.h"
#include <vector>
#include <iostream>

```

Include dependency graph for gdcmSurfaceHelper.h:



## Classes

- class [gdcm::SurfaceHelper](#)  
*SurfaceHelper.*

## Namespaces

- namespace [gdcm](#)

## 11.436 gdcmSurfaceHelper.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2017 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMSURFACEHELPER_H
00015 #define GDCMSURFACEHELPER_H
00016
00017 #include "gdcmTypes.h" // for GDCM_EXPORT
00018
00019 #include <vector>
00020 #include <iostream>
00021
00022 namespace gdcm
00023 {
00024
00025     class GDCM_EXPORT SurfaceHelper
00026     {
00027     public:
00028         typedef std::vector< unsigned short > ColorArray;

```

```

00034
00046 template <typename T, typename U>
00047 static unsigned short RGBToRecommendedDisplayGrayscale(const std::vector<T> & RGB,
00048                                                         const U rangeMax = 255);
00060 template <typename T, typename U>
00061 static ColorArray RGBToRecommendedDisplayCIELab(const std::vector<T> & RGB,
00062                                                  const U rangeMax = 255);
00074 template <typename T, typename U>
00075 static std::vector<T> RecommendedDisplayCIELabToRGB(const ColorArray & CIELab,
00076                                                     const U rangeMax = 255);
00087 template <typename U>
00088 static std::vector<float> RecommendedDisplayCIELabToRGB(const ColorArray & CIELab,
00089                                                         const U rangeMax = 255);
00090
00091 private:
00092
00093 static std::vector< float > RGBToXYZ(const std::vector<float> & RGB);
00094
00095 static std::vector< float > XYZToRGB(const std::vector<float> & XYZ);
00096
00097 static std::vector< float > XYZToCIELab(const std::vector<float> & XYZ);
00098
00099 static std::vector< float > CIELabToXYZ(const std::vector<float> & CIELab);
00100 };
00101
00102 template <typename T, typename U>
00103 unsigned short SurfaceHelper::RGBToRecommendedDisplayGrayscale(const std::vector<T> & RGB,
00104                                                                const U rangeMax/* = 255*/)
00105 {
00106     assert(RGB.size() > 2);
00107
00108     unsigned short Grayscale = 0;
00109
00110     const float inverseRangeMax = 1.0f / (float) rangeMax;
00111
00112     // 0xFFFF "=" 255 "=" white
00113     Grayscale = (unsigned short) ((0.2989 * RGB[0] + 0.5870 * RGB[1] + 0.1140 * RGB[2])
00114                                  * inverseRangeMax // Convert to range 0-1
00115                                  * 0xFFFF);        // Convert to range 0x0000-0xFFFF
00116
00117     return Grayscale;
00118 }
00119
00120 template <typename T, typename U>
00121 SurfaceHelper::ColorArray SurfaceHelper::RGBToRecommendedDisplayCIELab(const std::vector<T> & RGB,
00122                                                                           const U rangeMax/* = 255*/)
00123 {
00124     assert(RGB.size() > 2);
00125
00126     ColorArray CIELab(3);
00127     std::vector<float> tmp(3);
00128
00129     // Convert to range 0-1
00130     const float inverseRangeMax = 1.0f / (float) rangeMax;
00131     tmp[0] = (float) (RGB[0] * inverseRangeMax);
00132     tmp[1] = (float) (RGB[1] * inverseRangeMax);
00133     tmp[2] = (float) (RGB[2] * inverseRangeMax);
00134
00135     tmp = SurfaceHelper::XYZToCIELab( SurfaceHelper::RGBToXYZ( tmp ) );
00136
00137     // Convert to range 0x0000-0xFFFF
00138     // 0xFFFF "=" 127, 0x8080 "=" 0, 0x0000 "=" -128
00139     CIELab[0] = (unsigned short) ( 0xFFFF * (tmp[0]*0.01f));
00140     if(tmp[1] >= -128 && tmp[1] <= 0)
00141     {
00142         CIELab[1] = (unsigned short) (((float) (0x8080)/128.0f)*tmp[1] + ((float) 0x8080));
00143     }
00144     else if(tmp[1] <= 127 && tmp[1] > 0)
00145     {
00146         CIELab[1] = (unsigned short) (((float) (0xFFFF - 0x8080)/127.0f)*tmp[1] + (float) (0x8080));
00147     }
00148     if(tmp[2] >= -128 && tmp[2] <= 0)
00149     {
00150         CIELab[2] = (unsigned short) (((float) 0x8080/128.0f)*tmp[2] + ((float) 0x8080));
00151     }
00152     else if(tmp[2] <= 127 && tmp[2] > 0)
00153     {
00154         CIELab[2] = (unsigned short) (((float) (0xFFFF - 0x8080)/127.0f)*tmp[2] + (float) (0x8080));
00155     }
00156
00157     return CIELab;

```

```

00158 }
00159
00160 template <typename T, typename U>
00161 std::vector<T> SurfaceHelper::RecommendedDisplayCIELabToRGB(const ColorArray & CIELab,
00162                                                            const U rangeMax/* = 255*/)
00163 {
00164     assert(CIELab.size() > 2);
00165
00166     std::vector<T> RGB(3);
00167     std::vector<float> tmp(3);
00168
00169     // Convert to range 0-1
00170
00171     tmp[0] = 100.0f*CIELab[0] / (float)(0xFFFF);
00172     if(CIELab[1] <= 0x8080)
00173     {
00174         tmp[1] = (float)((CIELab[1] - 0x8080) * 128.0f) / (float)0x8080;
00175     }
00176     else
00177     {
00178         tmp[1] = (float)((CIELab[1]-0x8080)*127.0f / (float)(0xFFFF - 0x8080));
00179     }
00180     if(CIELab[2] <= 0x8080)
00181     {
00182         tmp[2] = (float)((CIELab[2] - 0x8080) * 128.0f) / (float)0x8080;
00183     }
00184     else
00185     {
00186         tmp[2] = (float)((CIELab[2]-0x8080)*127.0f / (float)(0xFFFF - 0x8080));
00187     }
00188
00189     tmp = SurfaceHelper::XYZToRGB( SurfaceHelper::CIELabToXYZ( tmp ) );
00190
00191     // Convert to range 0-rangeMax
00192     RGB[0] = (T) (tmp[0] * rangeMax);
00193     RGB[1] = (T) (tmp[1] * rangeMax);
00194     RGB[2] = (T) (tmp[2] * rangeMax);
00195
00196     return RGB;
00197 }
00198
00199 template <typename U>
00200 std::vector<float> SurfaceHelper::RecommendedDisplayCIELabToRGB(const ColorArray & CIELab,
00201                                                                const U rangeMax/* = 255*/)
00202 {
00203     return RecommendedDisplayCIELabToRGB<float>(CIELab, rangeMax);
00204 }
00205
00206 } // end namespace gdcm
00207
00208 #endif // GDCMSURFACEHELPER_H

```

## 11.437 gdcmSurfaceReader.h File Reference

```

#include <gdcmSegmentReader.h>
#include <gdcmSurface.h>

```



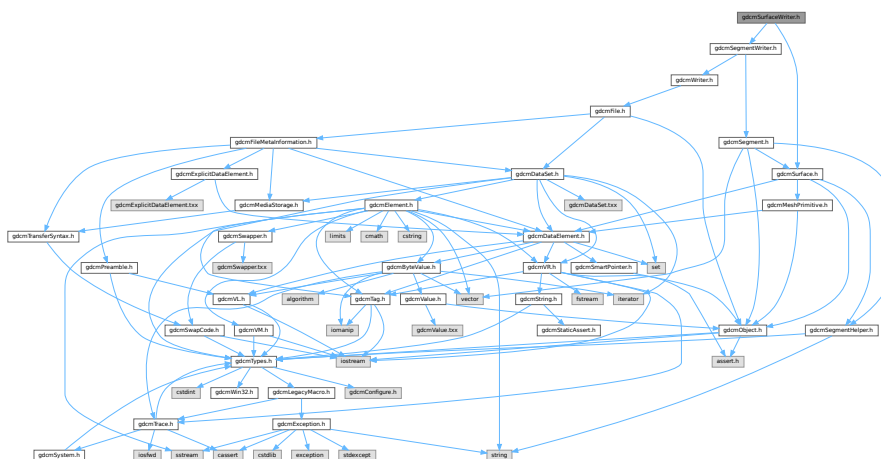
```
00038
00039     unsigned long GetNumberOfSurfaces() const;
00040
00041 protected:
00042
00043     bool ReadSurfaces();
00044
00045     bool ReadSurface(const Item & surfaceItem, const unsigned long idx);
00046
00047     bool ReadPointMacro(SmartPointer< Surface > surface, const DataSet & surfaceDS);
00048 };
00049
00050 }
00051
00052 #endif // GDCMSURFACEREADER_H
```

## 11.439 gdcmsurfacewriter.h File Reference

```
#include <gdcmSegmentWriter.h>
```

```
#include <qdcmsurface.h>
```

Include dependency graph for `gdcmSurfaceWriter.h`:



## Classes

- class `gdcm::SurfaceWriter`

*This class defines a SURFACE IE writer.*

## Namespaces

- namespace **gdcm**

## 11.440 gdcmSurfaceWriter.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMSURFACEWRITER_H
00015  #define GDCMSURFACEWRITER_H
00016
00017  #include <gdcmSegmentWriter.h>
00018  #include <gdcmSurface.h>
00019
00020  namespace gdcm
00021  {
00022
00029  class GDCM_EXPORT SurfaceWriter : public SegmentWriter
00030  {
00031  public:
00032      SurfaceWriter();
00033
00034      ~SurfaceWriter() override;
00035
00036      // const Surface & GetSurface() const { return *SurfaceData; }
00037      // Surface & GetSurface() { return *SurfaceData; }
00038      // void SetSurface(Surface const & segment);
00039
00041      bool Write() override; // Execute()
00042
00043      unsigned long GetNumberOfSurfaces();
00044      void SetNumberOfSurfaces(const unsigned long nb);
00045
00046  protected:
00047
00048      bool PrepareWrite();
00049
00050      void ComputeNumberOfSurfaces();
00051
00052      bool PrepareWritePointMacro(SmartPointer< Surface > surface,
00053                                  DataSet & surfaceDS,
00054                                  const TransferSyntax & ts);
00055
00056      //0066 0001 UL 1 Number of Surfaces
00057      unsigned long NumberOfSurfaces;
00058  };
00059
00060  }
00061
00062  #endif // GDCMSURFACEWRITER_H

```

## 11.441 gdcmTagPath.h File Reference

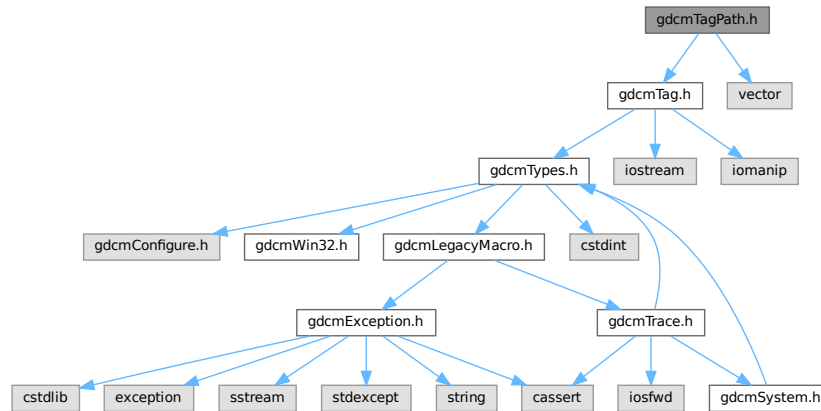
```

#include "gdcmTag.h"
#include <vector>

```



Include dependency graph for gdcmTagPath.h:



## Classes

- class [gdcm::TagPath](#)  
*class to handle a path of tag.*

## Namespaces

- namespace [gdcm](#)

## 11.442 gdcmTagPath.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMTAGPATH_H
00015 #define GDCMTAGPATH_H
00016
00017 #include "gdcmTag.h"
00018
00019 #include <vector>
00020
00021 namespace gdcm
00022 {
00023
00024   class GDCM_EXPORT TagPath
00025   {
00026
00027   
```

```

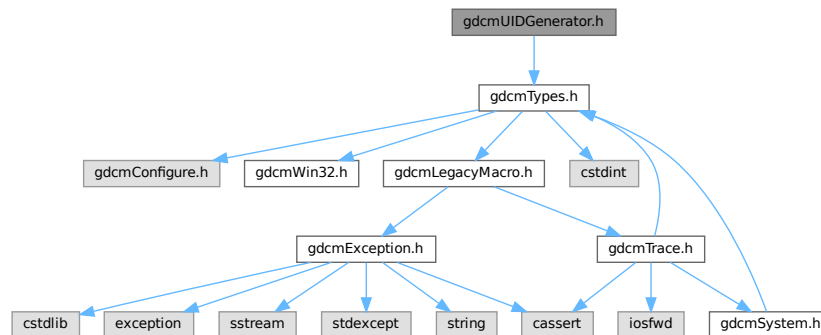
00032 public:
00033     TagPath();
00034     ~TagPath();
00035     void Print(std::ostream &) const;
00036
00041     bool ConstructFromString(const char *path);
00042
00044     static bool IsValid(const char *path);
00045
00047     bool ConstructFromTagList(Tag const *l, unsigned int n);
00048
00049     bool Push(Tag const & t);
00050     bool Push(unsigned int itemnum);
00051
00052 private:
00053     std::vector<Tag> Path;
00054 };
00055
00056 } // end namespace gdc
00057
00058 #endif //GDCMTAGPATH_H

```

## 11.443 gdcUIDGenerator.h File Reference

```
#include "gdcTypes.h"
```

Include dependency graph for gdcUIDGenerator.h:



### Classes

- class `gdc::UIDGenerator`  
Class for generating unique UID.

### Namespaces

- namespace `gdc`

## 11.444 gdcmUIDGenerator.h

[Go to the documentation of this file.](#)

```

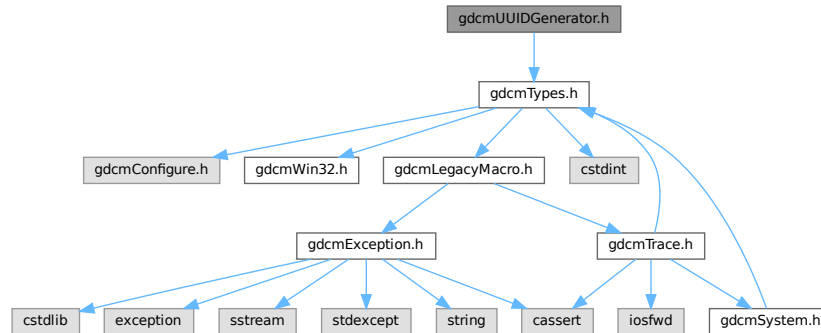
00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMUIDGENERATOR_H
00015 #define GDCMUIDGENERATOR_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021
00022   class GDCM_EXPORT UIDGenerator
00023   {
00024   public:
00025     UIDGenerator():Unique() {}
00026
00027     // Function to override the GDCM root with a user one:
00028     // WARNING: This need to be a valid root, otherwise call will fail
00029     // Implementation note. According to DICOM standard PS 3.5, Section 9 :
00030     // Unique Identifiers (UIDs), we have:
00031     /*
00032     ...
00033     The <org root> portion of the UID uniquely identifies an organization, (i.e., manufacturer, research
00034     organization, NEMA, etc.), and is composed of a number of numeric components as defined by ISO 8824.
00035     The <suffix> portion of the UID is also composed of a number of numeric components, and shall be
00036     unique within the scope of the <org root>. This implies that the organization identified in the <org
00037     root> is
00038     responsible for guaranteeing <suffix> uniqueness by providing registration policies. These policies
00039     shall
00040     guarantee <suffix> uniqueness for all UID's created by that organization. Unlike the <org root>, which
00041     may
00042     be common for UID's in an organization, the <suffix> shall take different unique values between
00043     different
00044     UID's that identify different objects.
00045     ...
00046     */
00047     static void SetRoot(const char * root);
00048     static const char *GetRoot();
00049
00050     const char* Generate();
00051
00052     static bool IsValid(const char *uid);
00053
00054     static const char *GetGDCMUID(); // who would want that in the public API ??
00055
00056 protected:
00057     static bool GenerateUUID(unsigned char *uuid_data);
00058
00059 private:
00060     static const char GDCM_UID[];
00061     static std::string Root;
00062     static std::string EncodedHardwareAddress;
00063     static std::string Unique; // Buffer
00064 };
00065
00066 } // end namespace gdcm
00067
00068 #endif //GDCMUIDGENERATOR_H

```

## 11.445 gdcmUUIDGenerator.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmUUIDGenerator.h:



### Classes

- class [gdcm::UUIDGenerator](#)  
Class for generating unique UUID.

### Namespaces

- namespace [gdcm](#)

## 11.446 gdcmUUIDGenerator.h

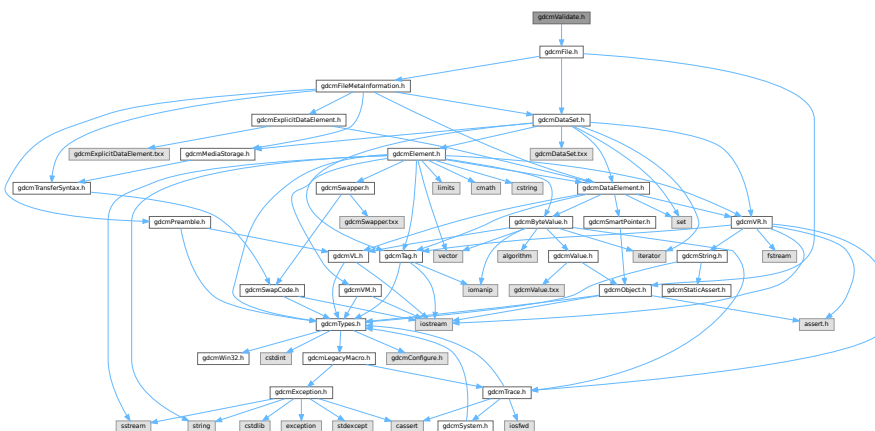
[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMUUUIDGENERATOR_H
00015 #define GDCMUUUIDGENERATOR_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021
00022   class GDCM_EXPORT UUIDGenerator
00023   {
00024
00025   
```

```
00028 public:
00031     const char* Generate();
00032
00034     static bool IsValid(const char *uid);
00035
00036 private:
00037     std::string Unique; // Buffer
00038 };
00039
00040 } // end namespace gdcmm
00041
00042 #endif //GDCMUUIDGENERATOR_H
```

```
#include "gdcmFile.h"
Include dependency graph for gdcmValidate.h:
```



- class `gdcm::Validate`  
*Validate* class.

- namespace **gdcm**

[Go to the documentation of this file.](#)

```
00001  /*****
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
```



- namespace `gdcm`

[Go to the documentation of this file.](#)

## 11.451 gdcmXMLPrinter.h File Reference

## Classes

- class [gdcm::XMLPrinter](#)

## Namespaces

- namespace [gdcm](#)

## 11.452 gdcmXMLPrinter.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMXMLPRINTER_H
00015 #define GDCMXMLPRINTER_H
00016
00017 /*
00018 The Normative version of the XML Schema for the Native DICOM Model follows:
00019
00020
00021
00022
00023 start = element NativeDicomModel { DicomDataSet }
00024
00025 # A DICOM Data Set is as defined in PS3.5. It does not appear
00026 # as an XML Element, since it does not appear in the binary encoded
00027 # DICOM objects. It exists here merely as a documentation aid.
00028
00029 DicomDataSet = DicomAttribute*
00030 DicomAttribute = element DicomAttribute {
00031   Tag, VR, Keyword?, PrivateCreator?,
00032   ( BulkData | Value+ | Item+ | PersonName+ )?
00033 }
00034
00035 BulkData = element BulkData{ UUID }
00036 Value = element Value { Number, xsd:string }
00037 Item = element Item { Number, DicomDataSet }
00038 PersonName = element PersonName {
00039   Number,
00040   element SingleByte { NameComponents }?,
00041   element Ideographic { NameComponents }?,
00042   element Phonetic
00043   { NameComponents }?
00044 }
00045
00046 NameComponents =
00047   element FamilyName {xsd:string}?,
00048   element GivenName {xsd:string}?,
00049   element MiddleName {xsd:string}?,
00050   element NamePrefix {xsd:string}?,
00051   element NameSuffix {xsd:string}?
00052
00053 # keyword is the attribute tag from PS3.6
00054 # (derived from the DICOM Attribute's name)
00055 Keyword = attribute keyword { xsd:token }
00056 # canonical XML definition of Hex, with lowercase letters disallowed
00057 Tag = attribute tag { xsd:string{ minLength="8" maxLength="8" pattern="[0-9A-F]{8}" } }
00058 VR = attribute vr { "AE" | "AS" | "AT" | "CS" | "DA" | "DS" | "DT" | "FL" | "FD"
00059 | "IS" | "LO" | "LT" | "OB" | "OF" | "OW" | "PN" | "SH" | "SL"

```



```

00060 | "SQ" | "SS" | "ST" | "TM" | "UI" | "UL" | "UN" | "US" | "UT" }
00061 PrivateCreator = attribute privateCreator{ xsd:string }
00062 UUID = attribute uuid { xsd:string }
00063 Number = attribute number { xsd:positiveInteger }
00064
00065
00066 */
00067
00068 #include "gdcmFile.h"
00069 #include "gdcmDataElement.h"
00070
00071 namespace gdcm
00072 {
00073
00074 class DataSet;
00075 class DictEntry;
00076 class Dicts;
00077
00078 class GDCM_EXPORT XMLPrinter
00079 {
00080 public:
00081     XMLPrinter();
00082     virtual ~XMLPrinter();
00083
00084     // Set file
00085     void SetFile(File const &f) { F = &f; }
00086
00087
00088     typedef enum {
00089         OnlyUUID = 0 ,
00090         LOADBULKDATA = 1
00091     } PrintStyles;
00092
00093     // Set PrintStyle value
00094     void SetStyle(PrintStyles ps)
00095     {
00096         PrintStyle = ps;
00097     }
00098
00099     // Get PrintStyle value
00100     PrintStyles GetPrintStyle() const
00101     {
00102         return PrintStyle;
00103     }
00104
00105     // Print
00106     void Print(std::ostream& os);
00107
00108     // Print an individual dataset
00109     void PrintDataSet(const DataSet &ds, const TransferSyntax &ts, std::ostream& os);
00110
00111     //void PrintUUID(std::ostream &os);
00112
00113     virtual void HandleBulkData(const char *uuid, const TransferSyntax &ts,
00114         const char *bulkdata, size_t bulklen);
00115
00116 protected:
00117
00118     VR PrintDataElement(std::ostream &os, const Dicts &dicts, const DataSet &ds, const DataElement &de,
00119         const TransferSyntax &ts);
00120
00121     void PrintSQ(const SequenceOfItems *sqi, const TransferSyntax &ts, std::ostream &os);
00122
00123     PrintStyles PrintStyle;
00124
00125     const File *F;
00126 };
00127
00128 } // end namespace gdcm
00129
00130 #endif //GDCMXMLPRINTER_H

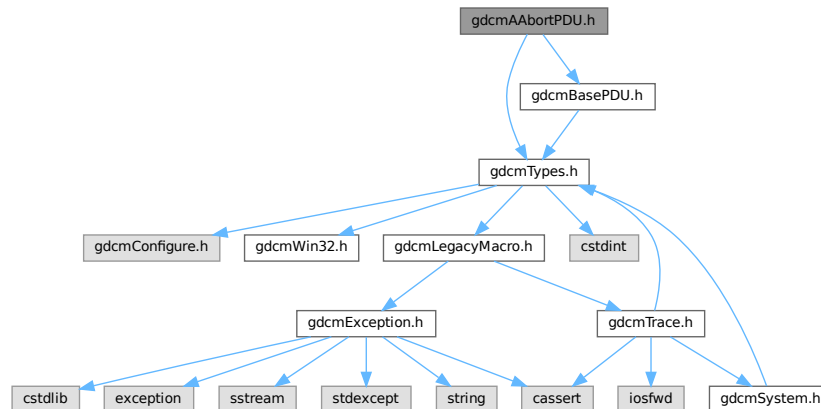
```

## 11.453 gdcmAAbortPDU.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAAbortPDU.h:



### Classes

- class `gdcm::network::AAbortPDU`  
*AAbortPDU*.

### Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

## 11.454 gdcmAAbortPDU.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMAABORTPDU_H
00015  #define GDCMAABORTPDU_H
00016

```

```

00017 #include "gdcmTypes.h"
00018 #include "gdcmBasePDU.h"
00019
00020 namespace gdcm
00021 {
00022
00023 namespace network
00024 {
00025
00030 class GDCM_EXPORT AAbortPDU : public BasePDU
00031 {
00032 public:
00033     AAbortPDU();
00034     std::istream &Read(std::istream &is) override;
00035     const std::ostream &Write(std::ostream &os) const override;
00036
00038     size_t Size() const override;
00039     void Print(std::ostream &os) const override;
00040
00041     bool IsLastFragment() const override { return true; }
00042
00043     void SetSource(const uint8_t s);
00044     void SetReason(const uint8_t r);
00045
00046 private:
00047     static const uint8_t ItemType; // PDUType ?
00048     static const uint8_t Reserved2;
00049     uint32_t ItemLength; // PDU Length
00050     static const uint8_t Reserved7;
00051     static const uint8_t Reserved8;
00052     uint8_t Source;
00053     uint8_t Reason; // diag
00054 };
00055
00056 } // end namespace network
00057
00058 } // end namespace gdcm
00059
00060 #endif //GDCMAABORTPDU_H

```

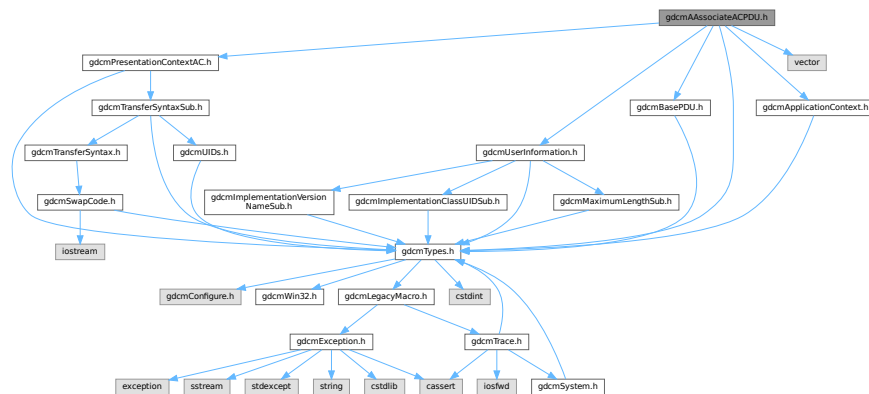
## 11.455 gdcmAAssociateACPDU.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmApplicationContext.h"
#include "gdcmPresentationContextAC.h"
#include "gdcmUserInformation.h"
#include "gdcmBasePDU.h"
#include <vector>

```

Include dependency graph for gdcmAAssociateACPDU.h:



## Classes

- class `gdcm::network::AAssociateACPDU`  
*AAssociateACPDU.*

## Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

## 11.456 gdcmAAssociateACPDU.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 #ifndef GDCMAASSOCIATEACPDU_H
00015 #define GDCMAASSOCIATEACPDU_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmApplicationContext.h"
00019 #include "gdcmPresentationContextAC.h"
00020 #include "gdcmUserInformation.h"
00021 #include "gdcmBasePDU.h"
00022
00023 #include <vector>
00024
00025 namespace gdcm
00026 {
00027
00028     namespace network
00029     {
00030         class AAssociateRQPDU;
00031
00032         class AAssociateACPDU : public BasePDU
00033         {
00034         public:
00035             AAssociateACPDU();
00036             std::istream &Read(std::istream &is) override;
00037             const std::ostream &Write(std::ostream &os) const override;
00038
00039             void AddPresentationContextAC( PresentationContextAC const &pcac );
00040
00041             typedef std::vector<PresentationContextAC>::size_type SizeType;
00042             const PresentationContextAC &GetPresentationContextAC( SizeType i ) {
00043                 assert( !PresContextAC.empty() && i < PresContextAC.size() );
00044                 return PresContextAC[i];
00045             }
00046             SizeType GetNumberOfPresentationContextAC() const {
00047                 return PresContextAC.size();
00048             }
00049             const UserInformation &GetUserInformation() const { return UserInfo; }
00050
00051             SizeType Size() const override;
00052
00053             void Print(std::ostream &os) const override;
00054             bool IsLastFragment() const override { return true; }
00055
00056             void InitFromRQ( AAssociateRQPDU const & rqpdu );

```

```

00062 protected:
00063     friend class AAssociateRQPDU;
00064     void SetCalledAETitle(const char calledaetitle[16]);
00065     void SetCallingAETitle(const char callingaetitle[16]);
00066
00067 private:
00068     void InitSimple( AAssociateRQPDU const & rqpdu );
00069
00070 private:
00071     static const uint8_t ItemType; // PDUType ?
00072     static const uint8_t Reserved2;
00073     uint32_t PDULength; // len of
00074     static const uint16_t ProtocolVersion;
00075     static const uint16_t Reserved9_10;
00076
00077     // This reserved field shall be sent with a value identical to the value
00078     // received in the same field of the A-ASSOCIATE-RQ PDU, but its value
00079     // shall not be tested when received.
00080     char Reserved11_26[16];
00081     // This reserved field shall be sent with a value identical to the value
00082     // received in the same field of the A-ASSOCIATE-RQ PDU, but its value
00083     // shall not be tested when received.
00084     char Reserved27_42[16];
00085     // This reserved field shall be sent with a value identical to the value
00086     // received in the same field of the A-ASSOCIATE-RQ PDU, but its value
00087     // shall not be tested when received.
00088     char Reserved43_74[32];
00089     /*
00090     75-xxx Variable items This variable field shall contain the following items: one Application
00091     Context Item, one or more Presentation Context Item(s) and one User
00092     Information Item. For a complete description of these items see Sections
00093     7.1.1.2, 7.1.1.14, and 7.1.1.6.
00094     */
00095     ApplicationContext AppContext;
00096     std::vector<PresentationContextAC> PresContextAC;
00097     UserInformation UserInfo;
00098 };
00099
00100 } // end namespace network
00101
00102 } // end namespace gdcm
00103
00104 #endif //GDCMAASSOCIATEACPDU_H

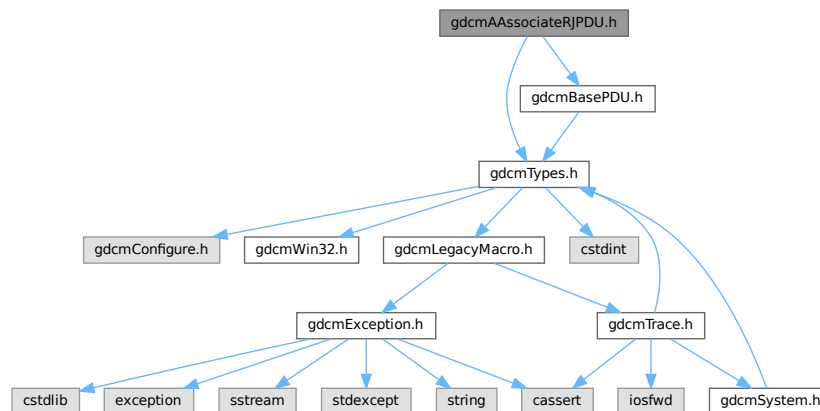
```

## 11.457 gdcmAAssociateRJPDU.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAAssociateRJPDU.h:



## Classes

- class `gdcm::network::AAssociateRJPDU`  
*AAssociateRJPDU.*

## Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

## 11.458 gdcmAAssociateRJPDU.h

[Go to the documentation of this file.](#)

```

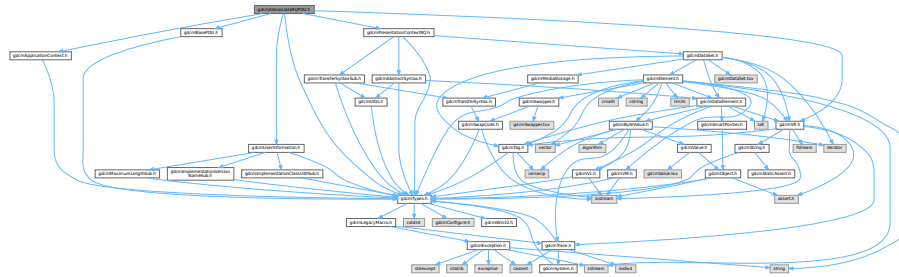
00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMAASSOCIATERJPDU_H
00015 #define GDCMAASSOCIATERJPDU_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmBasePDU.h"
00019
00020 namespace gdcm
00021 {
00022
00023     namespace network
00024     {
00025
00031         class AAssociateRJPDU : public BasePDU
00032         {
00033         public:
00034             AAssociateRJPDU();
00035             std::istream &Read(std::istream &is) override;
00036             const std::ostream &Write(std::ostream &os) const override;
00037             void Print(std::ostream &os) const override;
00038             size_t Size() const override;
00039             bool IsLastFragment() const override { return true; }
00040         private:
00041             static const uint8_t ItemType; // PDUType ?
00042             static const uint8_t Reserved2;
00043             uint32_t ItemLength; // PDU Length ?
00044             static const uint8_t Reserved8;
00045             uint8_t Result;
00046             uint8_t Source;
00047             uint8_t Reason; // diag ?
00048         };
00049
00050     } // end namespace network
00051
00052 } // end namespace gdcm
00053
00054 #endif //GDCMAASSOCIATERJPDU_H

```

## 11.459 gdcmAAssociateRQPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmApplicationContext.h"
#include "gdcmPresentationContextRQ.h"
#include "gdcmUserInformation.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAAssociateRQPDU.h:



## Classes

- class `gdcn::network::AAssociateRQPDU`  
`AAssociateRQPDU`.

## Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

## 11.460 gdcmAAssociateRQPDU.h

[Go to the documentation of this file.](#)

```
00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 #ifndef GDCMAASSOCIATERQPDU_H
00015 #define GDCMAASSOCIATERQPDU_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmVR.h" // AEComp
00019 #include "gdcmApplicationContext.h"
00020 #include "gdcmPresentationContextRQ.h"
```

```

00021 #include "gdcmUserInformation.h"
00022 #include "gdcmBasePDU.h"
00023
00024 namespace gdcm
00025 {
00026
00027 namespace network
00028 {
00029
00030 class AAssociateACPDU;
00035 class AAssociateRQPDU : public BasePDU
00036 {
00037 public:
00038     AAssociateRQPDU();
00039     std::istream &Read(std::istream &is) override;
00040     const std::ostream &Write(std::ostream &os) const override;
00041     size_t Size() const override;
00042     void AddPresentationContext( PresentationContextRQ const &pc );
00043
00044     void SetCalledAETitle(const char calledaetitle[16]);
00046     std::string GetCalledAETitle() const { return std::string(CalledAETitle,16); }
00047
00049     void SetCallingAETitle(const char callingaetitle[16]);
00050     std::string GetCallingAETitle() const { return std::string(CallingAETitle,16); }
00051
00053     static bool IsAETitleValid(const char title[16]);
00054
00057     //void InitFromRQ( AAssociateACPDU &acpdu );
00058
00059     void Print(std::ostream &os) const override;
00060
00061     AAssociateRQPDU(const AAssociateRQPDU &pdu):BasePDU(pdu)
00062     {
00063         assert( 0 );
00064     }
00065     //this function fails to compile on windows.
00066     // AAssociateRQPDU &operator=(const AAssociateRQPDU &_val)
00067     // {
00068     //     assert( 0 );
00069     // }
00070
00071     typedef std::vector<PresentationContextRQ>::size_type SizeType;
00072     SizeType GetNumberOfPresentationContext() const {
00073         return PresContext.size();
00074     }
00075     PresentationContextRQ const &GetPresentationContext(SizeType i) const {
00076         assert( !PresContext.empty() && i < PresContext.size() );
00077         return PresContext[i];
00078     }
00079     typedef std::vector<PresentationContextRQ> PresentationContextArrayType;
00080     PresentationContextArrayType const &GetPresentationContexts() { return PresContext; }
00081
00082     const PresentationContextRQ *GetPresentationContextByID(uint8_t i) const;
00083     const PresentationContextRQ *GetPresentationContextByAbstractSyntax(AbstractSyntax const &absyn )
const;
00084     bool IsLastFragment() const override { return true; }
00085
00086     const UserInformation & GetUserInfo() const { return UserInfo; }
00087     void SetUserInformation( UserInformation const &ui );
00088
00089 protected:
00090     friend class AAssociateACPDU;
00091     std::string GetReserved43_74() const;
00092
00093 private:
00094     // 1 PDU-type 01H
00095     static const uint8_t ItemType; // PDUType ?
00096     // 2 Reserved This reserved field shall be sent with a value 00H but not tested to this value when
received.
00097     static const uint8_t Reserved2;
00098     /* 3-6 PDU-length This PDU-length shall be the number of bytes from the first byte of the
00099     following field to the last byte of the variable field. It shall be encoded as
00100     an unsigned binary number
00101     */
00102     uint32_t ItemLength; // PDU Length
00103     /*
00104     7-8 Protocol-version This two byte field shall use one bit to identify each version of the
00105     DICOM UL protocol supported by the calling end-system. This is
00106     Version 1 and shall be identified with bit 0 set. A receiver of this PDU
00107     implementing only this version of the DICOM UL protocol shall only test
00108     that bit 0 is set.

```



```

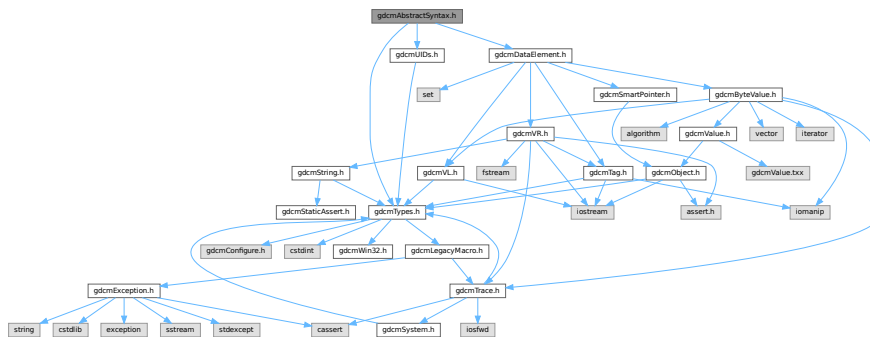
00109      */
00110      static const uint16_t ProtocolVersion;
00111      /*
00112      9-10 Reserved This reserved field shall be sent with a value 0000H but not tested to
00113      this value when received.
00114      */
00115      static const uint16_t Reserved9_10;
00116      /*
00117      11-26 Called-AE-title Destination DICOM Application Name. It shall be encoded as 16
00118      characters as defined by the ISO 646:1990-Basic G0 Set with leading
00119      and trailing spaces (20H) being non-significant. The value made of 16
00120      spaces (20H) meaning "no Application Name specified" shall not be
00121      used. For a complete description of the use of this field, see Section
00122      7.1.1.4.
00123      */
00124      char CalledAETitle[16];
00125      /*
00126      27-42 Calling-AE-title Source DICOM Application Name. It shall be encoded as 16
00127      characters as defined by the ISO 646:1990-Basic G0 Set with leading
00128      and trailing spaces (20H) being non-significant. The value made of 16
00129      spaces (20H) meaning "no Application Name specified" shall not be
00130      used. For a complete description of the use of this field, see Section
00131      7.1.1.3.
00132      */
00133      char CallingAETitle[16];
00134      /*
00135      43-74 Reserved This reserved field shall be sent with a value 00H for all bytes but not
00136      tested to this value when received
00137      */
00138      char Reserved43_74[32]; // { 0 }
00139      /*
00140      75-xxx Variable items This variable field shall contain the following items: one Application
00141      Context Item, one or more Presentation Context Items and one User
00142      Information Item. For a complete description of the use of these items
00143      see Sections 7.1.1.2, 7.1.1.13, and 7.1.1.6.
00144      */
00145      ApplicationContext AppContext;
00146      std::vector<PresentationContextRQ> PresContext;
00147      UserInformation UserInfo;
00148  };
00149
00150 } // end namespace network
00151 } // end namespace gdcml
00152
00153 #endif //GDCMAASSOCIATERQPDU_H

```

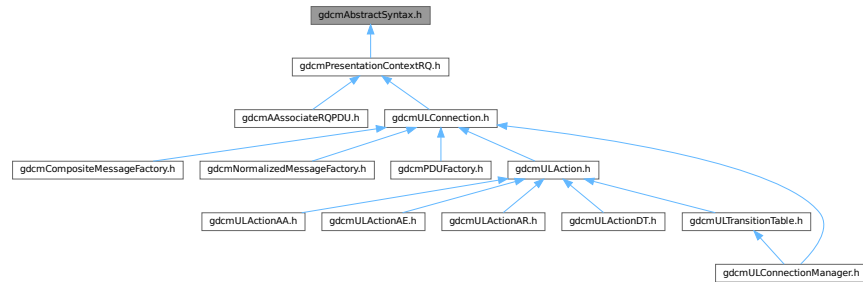
## 11.461 gdcAbstractSyntax.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmUIDs.h"
#include "gdcmDataElement.h"
```

Include dependency graph for `gdcMAbstractSyntax.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcml::network::AbstractSyntax](#)  
*AbstractSyntax.*

## Namespaces

- namespace [gdcml](#)
- namespace [gdcml::network](#)

## 11.462 gdcmlAbstractSyntax.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMABSTRACTSYNTAX_H
00015  #define GDCMABSTRACTSYNTAX_H
00016
00017  #include "gdcmlTypes.h"
00018  #include "gdcmlUIDs.h"
00019  #include "gdcmlDataElement.h"
00020
00021  namespace gdcml
00022  {
00023
00024  namespace network
00025  {
00026
00032  class AbstractSyntax
00033  {
00034  public:
00035      AbstractSyntax();
00036      std::istream &Read(std::istream &is);
  
```

```

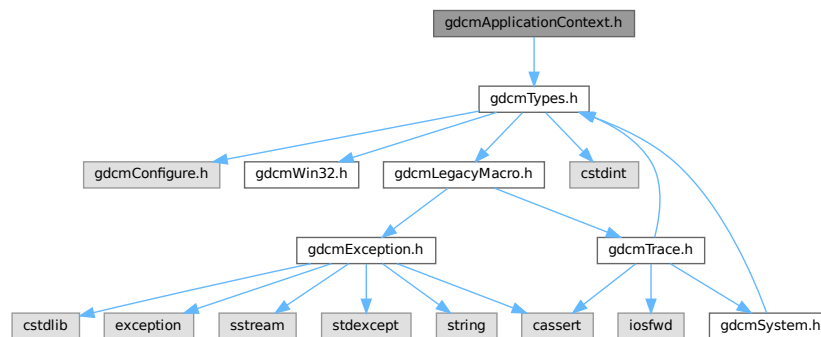
00037     const std::ostream &Write(std::ostream &os) const;
00038
00039 void SetName( const char *name ) { UpdateName( name ); }
00040 const char *GetName() const { return Name.c_str(); }
00041
00042 // accept a UIDs::TSType also...
00043 void SetNameFromUID( UIDs::TSName tsname );
00044 //now that the PresentationContext messes around with UIDs and returns a string
00045 //use that string as well.
00046 //void SetNameFromUIDString( const std::string& inUIDName );
00047
00048 size_t Size() const;
00049
00050 void Print(std::ostream &os) const;
00051
00052 bool operator==(const AbstractSyntax & as) const
00053 {
00054     return Name == as.Name;
00055 }
00056
00057 DataElement GetAsDataElement() const;
00058
00059 private:
00060 void UpdateName( const char *name );
00061 static const uint8_t ItemType;
00062 static const uint8_t Reserved2;
00063 uint16_t ItemLength; // len of
00064 std::string /*AbstractSyntax*/ Name; // UID
00065 };
00066
00067 } // end namespace network
00068 } // end namespace gdcm
00069
00070 #endif //GDCMABSTRACTSYNTAX_H

```

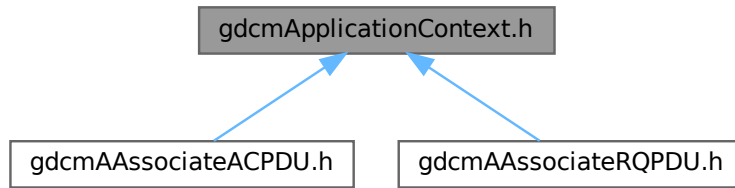
## 11.463 gdcmApplicationContext.h File Reference

#include "gdcmTypes.h"

Include dependency graph for gdcmApplicationContext.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcmap::network::ApplicationContext](#)  
*ApplicationContext.*

## Namespaces

- namespace [gdcmap](#)
- namespace [gdcmap::network](#)

## 11.464 gdcmapApplicationContext.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcmap.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012  =====*/
00013  #ifndef GDCMAPAPPLICATIONCONTEXT_H
00014  #define GDCMAPAPPLICATIONCONTEXT_H
00015
00016  #include "gdcmapTypes.h"
00017
00018  namespace gdcmap
00019  {
00020  {
00021  namespace network
00022  {
00023  {
00024
00025  class ApplicationContext
00026  {
00027  public:
00028    ApplicationContext();
00029    std::istream &Read(std::istream &is);
00030    const std::ostream &Write(std::ostream &os) const;
  
```

```

00038
00039 void SetName( const char *name ) { UpdateName( name ); }
00040 const char *GetName() const { return Name.c_str(); }
00041 size_t Size() const;
00042
00043 //static const uint8_t GetItemType() { return ItemType; }
00044 void Print(std::ostream &os) const;
00045
00046 private:
00047 void UpdateName( const char *name );
00048 static const uint8_t ItemType;
00049 static const uint8_t Reserved2;
00050 uint16_t ItemLength; // len of application context name
00051 std::string /*ApplicationContext*/ Name; // UID
00052 };
00053
00054 } // end namespace network
00055
00056 } // end namespace gdcm
00057
00058 #endif //GDCMAPPLICATIONCONTEXT_H

```

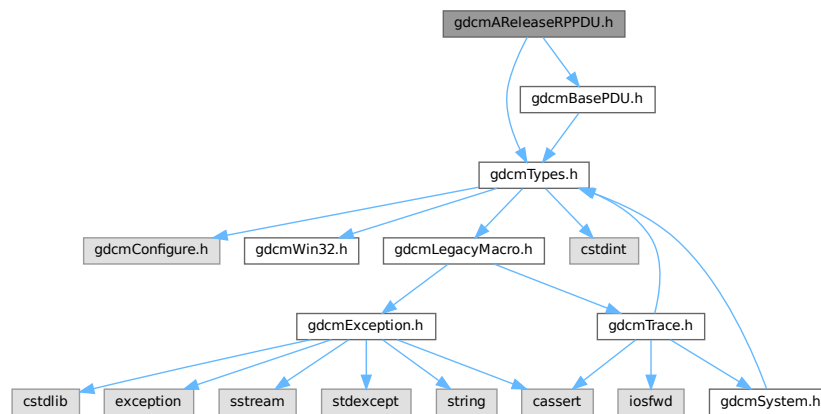
## 11.465 gdcmAReleaseRPPDU.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmBasePDU.h"

```

Include dependency graph for gdcmAReleaseRPPDU.h:



### Classes

- class `gdcm::network::AReleaseRPPDU`  
*AReleaseRPPDU*.

### Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

## 11.466 gdcmAReleaseRPPDU.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMARELEASERPPDU_H
00015  #define GDCMARELEASERPPDU_H
00016
00017  #include "gdcmTypes.h"
00018  #include "gdcmBasePDU.h"
00019
00020  namespace gdcm
00021  {
00022
00023  namespace network
00024  {
00025
00031  class AReleaseRPPDU : public BasePDU
00032  {
00033  public:
00034      AReleaseRPPDU();
00035      std::istream &Read(std::istream &is) override;
00036      const std::ostream &Write(std::ostream &os) const override;
00037      size_t Size() const override;
00038      void Print(std::ostream &os) const override;
00039      bool IsLastFragment() const override { return true; }
00040  private:
00041      static const uint8_t ItemType; // PDUType ?
00042      static const uint8_t Reserved2;
00043      uint32_t ItemLength; // PDU Length
00044      static const uint32_t Reserved7_10;
00045  };
00046
00047  } // end namespace network
00048
00049  } // end namespace gdcm
00050
00051  #endif //GDCMARELEASERPPDU_H

```

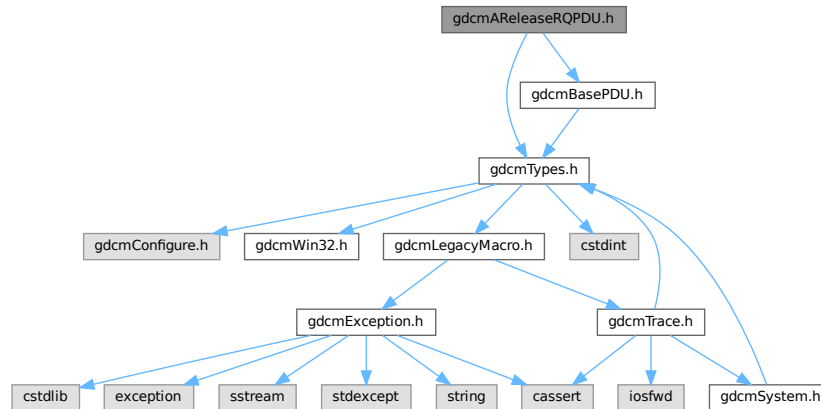
## 11.467 gdcmAReleaseRQPDU.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmBasePDU.h"

```

Include dependency graph for gdcmAReleaseRQPDU.h:



## Classes

- class `gdcm::network::AReleaseRQPDU`  
*AReleaseRQPDU.*

## Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

## 11.468 gdcmAReleaseRQPDU.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMARELEASERQPDU_H
00015  #define GDCMARELEASERQPDU_H
00016
00017  #include "gdcmTypes.h"
00018  #include "gdcmBasePDU.h"
00019
00020  namespace gdcm
00021  {
00022
00023  namespace network
00024  {
  
```

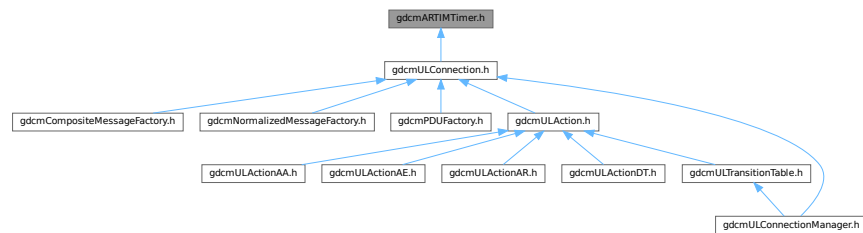
```

00025
00031 class AReleaseRQPDU : public BasePDU
00032 {
00033 public:
00034     AReleaseRQPDU();
00035     std::istream &Read(std::istream &is) override;
00036     const std::ostream &Write(std::ostream &os) const override;
00037     size_t Size() const override;
00038     void Print(std::ostream &os) const override;
00039     bool IsLastFragment() const override { return true; }
00040 private:
00041     static const uint8_t ItemType; // PDUType ?
00042     static const uint8_t Reserved2;
00043     uint32_t ItemLength; // PDU Length
00044     static const uint32_t Reserved7_10;
00045 };
00046
00047 } // end namespace network
00048
00049 } // end namespace gdcmm
00050
00051 #endif //GDCMARELEASERQPDU_H

```

## 11.469 gdcmmARTIMTimer.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class `gdcmm::network::ARTIMTimer`  
*ARTIMTimer.*

### Namespaces

- namespace `gdcmm`
- namespace `gdcmm::network`



## 11.470 gdcmARTIMTimer.h

[Go to the documentation of this file.](#)

```

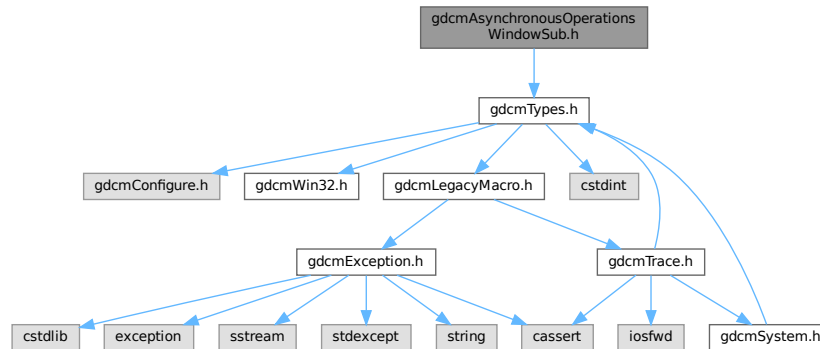
00001 /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  * http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMARTIMTIMER_H
00019 #define GDCMARTIMTIMER_H
00020
00021 namespace gdcm {
00022     namespace network{
00023     class ARTIMTimer
00024     {
00025     private:
00026         double mStartTime; //ms timing should be good enough, but there are also
00027         //high-resolution timing options. Those return doubles. For now,
00028         //go with integer timing solutions based on milliseconds (DWORD on windows),
00029         //but leave as doubles to ease transitions to other timing methods.
00030
00031         double mTimeout;
00032         //once GetCurrentTime() -mStartTime > mTimeout, GetHasExpired returns true.
00033
00034         double GetCurrentTime() const;//a platform-specific implementation of getting the
00035         //current time.
00036
00037     public:
00038         ARTIMTimer(); //initiates the start and timeout at -1;
00039         void Start(); // 'start' the timer by getting the current wall time
00040         void Stop(); // 'stop' the timer by resetting the 'start' to -1;
00041         void SetTimeout(double inTimeout);
00042         double GetTimeout() const;
00043
00044         double GetElapsedTime() const;
00045
00046         bool GetHasExpired() const;
00047
00048     };
00049 }
00050
00051 #endif //GDCMARTIMTIMER_H

```

## 11.471 gdcmAsynchronousOperationsWindowSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmAsynchronousOperationsWindowSub.h:



### Classes

- class [gdcm::network::AsynchronousOperationsWindowSub](#)  
*AsynchronousOperationsWindowSub.*

### Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.472 gdcmAsynchronousOperationsWindowSub.h

[Go to the documentation of this file.](#)

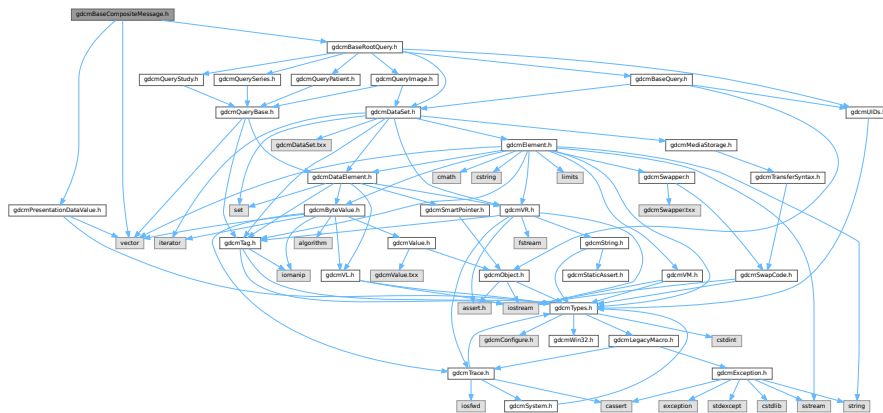
```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMASYNCHRONOUSOPERATIONSWINDOWSUB_H
00015  #define GDCMASYNCHRONOUSOPERATIONSWINDOWSUB_H
00016
00017  #include "gdcmTypes.h"
00018
00019  namespace gdcm
00020  {
00021

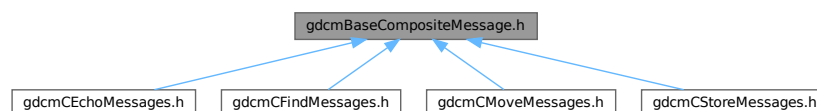
```

```
00022 namespace network
00023 {
00024
00032 class AsynchronousOperationsWindowSub
00033 {
00034 public:
00035     AsynchronousOperationsWindowSub();
00036     std::istream &Read(std::istream &is);
00037     const std::ostream &Write(std::ostream &os) const;
00038
00039     size_t Size() const;
00040     void Print(std::ostream &os) const;
00041
00042 private:
00043     static const uint8_t ItemType;
00044     static const uint8_t Reserved2;
00045     uint16_t ItemLength;
00046     uint16_t MaximumNumberOperationsInvoked;
00047     uint16_t MaximumNumberOperationsPerformed;
00048 };
00049
00050 } // end namespace network
00051
00052 } // end namespace gdcms
00053
00054 #endif // GDCMASYNCHRONOUSOPERATIONSWINDOWSUB_H
```

```
#include "gdcmPresentationDataValue.h"
#include "gdcmBaseRootQuery.h"
#include <vector>
Include dependency graph for gdcmBaseCompositeMessage.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::network::BaseCompositeMessage](#)  
*BaseCompositeMessage.*

## Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.474 gdcmBaseCompositeMessage.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002  *
00003  *   Copyright NumFOCUS
00004  *
00005  *   Licensed under the Apache License, Version 2.0 (the "License");
00006  *   you may not use this file except in compliance with the License.
00007  *   You may obtain a copy of the License at
00008  *
00009  *       http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  *   Unless required by applicable law or agreed to in writing, software
00012  *   distributed under the License is distributed on an "AS IS" BASIS,
00013  *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  *   See the License for the specific language governing permissions and
00015  *   limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMBASECOMPOSITEMESSAGE_H
00019 #define GDCMBASECOMPOSITEMESSAGE_H
00020
00021 #include "gdcmPresentationDataValue.h"
00022 #include "gdcmBaseRootQuery.h"
00023
00024 #include <vector>
00025
00026 namespace gdcm
00027 {
00028     namespace network
00029     {
00030         class ULConnection;
00052         class BaseCompositeMessage
00053         {
00054             public:
00055                 virtual ~BaseCompositeMessage() = default;
00056                 //construct the appropriate pdv and dataset for this message
00057                 //for instance, setting tag 0x0,0x100 to the appropriate value
00058                 //the pdv, as described in Annex E of 3.8-2009, is the first byte
00059                 //of the message (the MessageHeader), and then the subsequent dataset
00060                 //that describes the operation.
00061                 virtual std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
00062                     const BaseRootQuery * inRootQuery) = 0;
00063         };
00064     }
00065 }
00066 #endif //BASECOMPOSITEMESSAGE_H

```

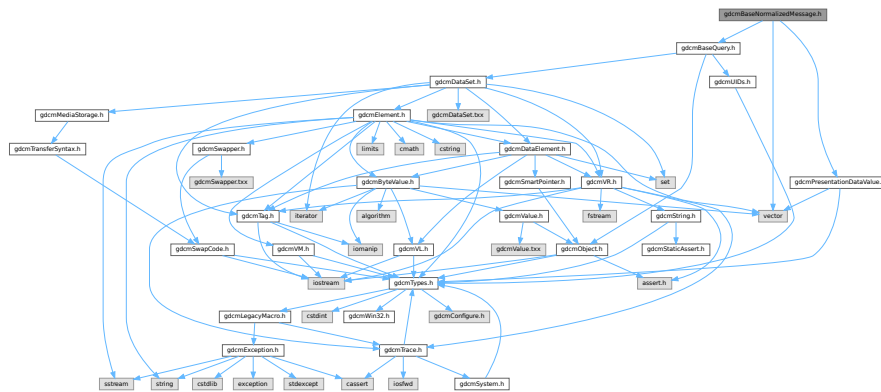
## 11.475 gdcmBaseNormalizedMessage.h File Reference

```
#include "gdcmPresentationDataValue.h"
```

```
#include "gdcmBaseQuery.h"
```

```
#include <vector>
```

Include dependency graph for gdcmBaseNormalizedMessage.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [gdcm::network::BaseNormalizedMessage](#)  
*BaseNormalizedMessage.*

### Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.476 gdcmBaseNormalizedMessage.h

[Go to the documentation of this file.](#)

```
00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2014 Mathieu Malaterre
```

```

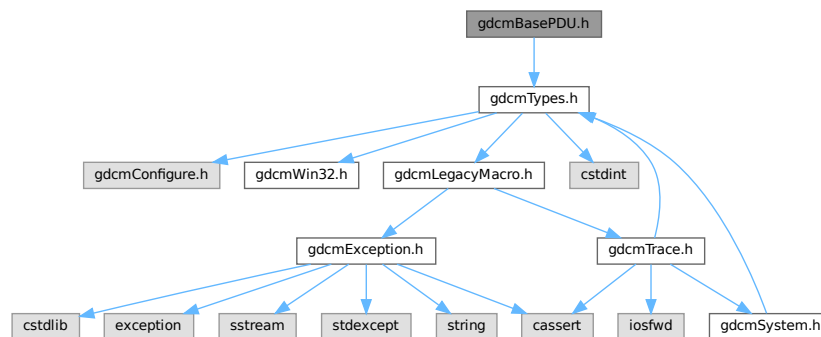
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014   #ifndef GDCMBASENORMALIZEDMESSAGE_H
00015   #define GDCMBASENORMALIZEDMESSAGE_H
00016
00017   #include "gdcmPresentationDataValue.h"
00018   #include "gdcmBaseQuery.h"
00019
00020   #include <vector>
00021
00022   namespace gdcm
00023   {
00024       namespace network
00025       {
00026           class ULConnection;
00049           class BaseNormalizedMessage
00050           {
00051           public:
00052               virtual ~BaseNormalizedMessage() = default;
00053               //construct the appropriate pdv and dataset for this message
00054               //for instance, setting tag 0x0,0x100 to the appropriate value
00055               //the pdv, as described in Annex E of 3.8-2009, is the first byte
00056               //of the message (the MessageHeader), and then the subsequent dataset
00057               //that describes the operation.
00058               virtual std::vector<PresentationDataValue> ConstructPDV( const ULConnection &inConnection,
00059                                                                       const BaseQuery * inQuery) = 0;
00060           };
00061       }
00062   }
00063   #endif //GDCMBASENORMALIZEDMESSAGE_H

```

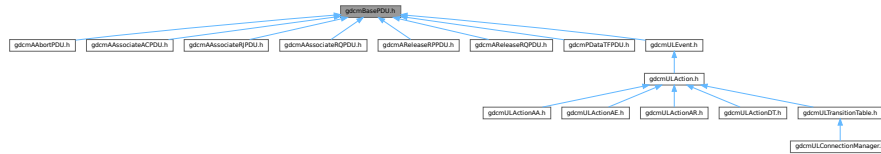
## 11.477 gdcmBasePDU.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmBasePDU.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::network::BasePDU`  
*BasePDU.*

## Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

## 11.478 gdcmBasePDU.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002  *
00003  *   Copyright NumFOCUS
00004  *
00005  *   Licensed under the Apache License, Version 2.0 (the "License");
00006  *   you may not use this file except in compliance with the License.
00007  *   You may obtain a copy of the License at
00008  *
00009  *       http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  *   Unless required by applicable law or agreed to in writing, software
00012  *   distributed under the License is distributed on an "AS IS" BASIS,
00013  *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  *   See the License for the specific language governing permissions and
00015  *   limitations under the License.
00016  *
00017  *===== */
00018 #ifndef GDCMBASEPDU_H
00019 #define GDCMBASEPDU_H
00020
00021 #include "gdcmTypes.h"
00022
00023 namespace gdcm
00024 {
00025     namespace network
00026     {
00027
00050         class BasePDU
00051         {
00052         public:
00053             virtual ~BasePDU() = default;
00054
00055             virtual std::istream &Read(std::istream &is) = 0;
00056             virtual const std::ostream &Write(std::ostream &os) const = 0;
00057
00058             virtual size_t Size() const = 0;
00059             virtual void Print(std::ostream &os) const = 0;
00060
00061             virtual bool IsLastFragment() const = 0;
00062         };
00063
00064     } // end namespace network
00065 } // end namespace gdcm
00066
00067 #endif // GDCMBASEPDU_H

```

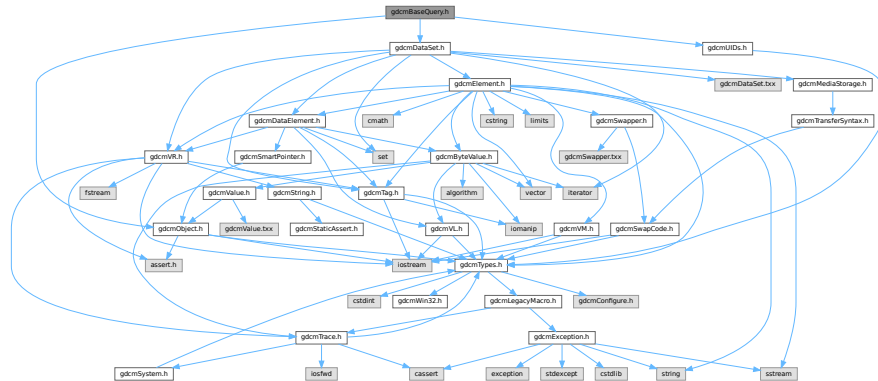
## 11.479 gdcmbaseQuery.h File Reference

```
#include "gdcmDataSet.h"
```

```
#include "gdcmUIDs.h"
```

```
#include "gdcmObject.h"
```

Include dependency graph for gdcmbaseQuery.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::BaseQuery`  
*BaseQuery*.

## Namespaces

- namespace **gdcm**

## Enumerations

- enum `gdcm::ENQueryType` {  
`gdcm::eCreateMMPS` = 0 ,  
`gdcm::eSetMMPS` }



## 11.480 gdcmBaseQuery.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  * http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMBASEQUERY_H
00019 #define GDCMBASEQUERY_H
00020
00021 #include "gdcmDataSet.h"
00022 #include "gdcmUIDs.h"
00023 #include "gdcmObject.h"
00024
00025 namespace gdcm
00026 {
00027     class QueryFactory;
00028     class DictEntry;
00029
00030     enum ENQueryType
00031     {
00032         eCreateMMPS = 0,
00033         eSetMMPS
00034     };
00041 class GDCM_EXPORT BaseQuery : public Object
00042 {
00043     //these four classes contain the required, unique, and optional tags from the standard.
00044     //used both to list the tags as well as to validate a dataset, if ever we were to do so.
00045 protected:
00046     DataSet mDataSet;
00047     friend class QueryFactory;
00048     BaseQuery();
00049
00050     std::string mSopInstanceUID;
00052
00053     void SetSearchParameter(const Tag& inTag, const DictEntry& inDictEntry, const std::string& inValue);
00054
00055     bool ValidDataSet( const DataSet & dataSetToValid, const DataSet & dataSetReference ) const ;
00056 public:
00057     ~BaseQuery() override;
00058
00059     void SetSearchParameter(const Tag& inTag, const std::string& inValue);
00060     void SetSearchParameter(const std::string& inKeyword, const std::string& inValue);
00061
00062     const std::ostream &WriteHelpFile(std::ostream &os);
00063
00064     //this function allows writing of the query to disk for storing for future use
00065     //virtual in case it needs to be overridden
00066     //returns false if the operation failed
00067     bool WriteQuery(const std::string& inFileName);
00068
00070     DataSet const & GetQueryDataSet() const;
00071     DataSet & GetQueryDataSet();
00072     void AddQueryDataSet(const DataSet & ds);
00073
00074     virtual bool ValidateQuery( bool inStrict = true ) const = 0;
00075
00076     virtual UIDs::TSName GetAbstractSyntaxUID() const = 0;
00077     std::string GetSOPInstanceUID() const { return mSopInstanceUID ; }
00078     void SetSOPInstanceUID( const std::string & iSopInstanceUID ) { mSopInstanceUID = iSopInstanceUID ; }
00079
00080
00081
00082     void Print(std::ostream &os) const override;
00083 };

```



## Enumerations

- enum `gdcm::EQueryLevel` {  
`gdcm::ePatient` = 0 ,  
`gdcm::eStudy` = 1 ,  
`gdcm::eSeries` = 2 ,  
`gdcm::eImage` = 3 }
- enum `gdcm::EQueryType` {  
`gdcm::eFind` = 0 ,  
`gdcm::eMove` ,  
`gdcm::eWLMFind` }

## 11.482 gdcmBaseRootQuery.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002  *
00003  *   Copyright NumFOCUS
00004  *
00005  *   Licensed under the Apache License, Version 2.0 (the "License");
00006  *   you may not use this file except in compliance with the License.
00007  *   You may obtain a copy of the License at
00008  *
00009  *       http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  *   Unless required by applicable law or agreed to in writing, software
00012  *   distributed under the License is distributed on an "AS IS" BASIS,
00013  *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  *   See the License for the specific language governing permissions and
00015  *   limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMBASEROOTQUERY_H
00019 #define GDCMBASEROOTQUERY_H
00020
00021 #include "gdcmDataSet.h"
00022 #include "gdcmUIDs.h"
00023 #include "gdcmBaseQuery.h"
00024 #include "gdcmQueryPatient.h"
00025 #include "gdcmQueryStudy.h"
00026 #include "gdcmQuerySeries.h"
00027 #include "gdcmQueryImage.h"
00028
00029 namespace gdcm
00030 {
00031     class QueryFactory;
00032     class DictEntry;
00033
00034     enum EQueryLevel
00035     {
00036         // -1 is reserved do not use
00037         ePatient = 0,
00038         eStudy = 1,
00039         eSeries = 2,
00040         eImage = 3
00041     };
00042     enum EQueryType
00043     {
00044         eFind = 0,
00045         eMove,
00046         eWLMFind
00047     };
00048
00066     class GDCM_EXPORT BaseRootQuery : public BaseQuery
00067     {
00068         //these four classes contain the required, unique, and optional tags from the standard.
00069         //used both to list the tags as well as to validate a dataset, if ever we were to do so.
00070     protected:
00071         QueryPatient mPatient;
00072         QueryStudy mStudy;

```

## 11.483 gdcmCEchoMessages.h File Reference

Include dependency graph for gdcmCEchoMessages.h:



- Generated by Doxygen

**Namespaces**

- namespace [gdcm](#)
- namespace [gdcm::network](#)

**11.484 gdcmCEchoMessages.h**

[Go to the documentation of this file.](#)

```

00001 /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  * http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMCECHOMESSAGES_H
00019 #define GDCMCECHOMESSAGES_H
00020
00021 #include "gdcmBaseCompositeMessage.h"
00022
00023 namespace gdcm{
00024     namespace network{
00025
00026     class ULConnection;
00027
00032     class CEchoRQ : public BaseCompositeMessage {
00033     public:
00034         std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
00035             const BaseRootQuery* inRootQuery) override;
00036     };
00037
00042     class CEchoRSP : public BaseCompositeMessage {
00043     public:
00044         std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
00045     };
00046     }
00047 }
00048 #endif // GDCMCECHOMESSAGES_H

```

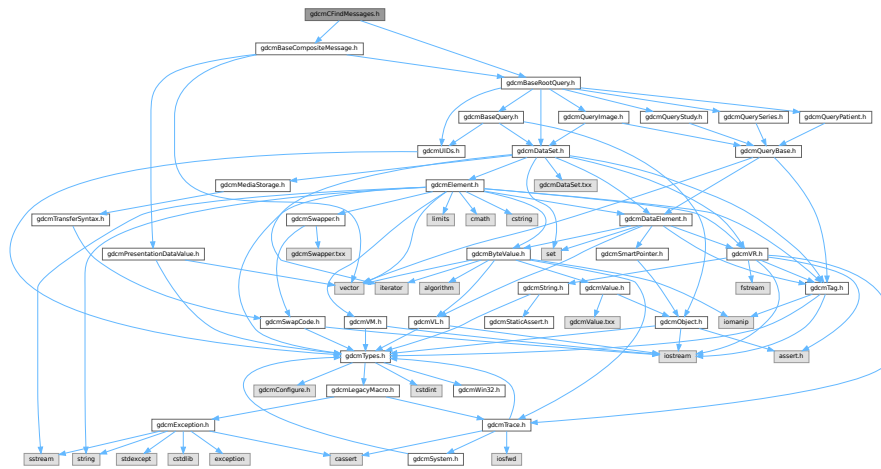
**11.485 gdcmCFindMessages.h File Reference**

```

#include "gdcmBaseCompositeMessage.h"
#include "gdcmBaseRootQuery.h"

```

Include dependency graph for `gdcmCFindMessages.h`:



## Classes

- class `gdcm::network::CFindCancelRQ`  
*CFindCancelRQ* this file defines the messages for the *cfind* action.
- class `gdcm::network::CFindRQ`  
*CFindRQ*.
- class `gdcm::network::CFindRSP`  
*CFindRSP* this file defines the messages for the *cfind* action.

## Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

## 11.486 gdcmCFindMessages.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  *
00003  *   Copyright NumFOCUS
00004  *
00005  *   Licensed under the Apache License, Version 2.0 (the "License");
00006  *   you may not use this file except in compliance with the License.
00007  *   You may obtain a copy of the License at
00008  *
00009  *       http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  *   Unless required by applicable law or agreed to in writing, software
00012  *   distributed under the License is distributed on an "AS IS" BASIS,
00013  *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  *   See the License for the specific language governing permissions and
00015  *   limitations under the License.
00016  *
00017  *===== */

```

## 11.487 qdcmCMoveMessages.h File Reference

Include dependency graph for `gdcMCMoveMessages.h`:



- Generated by Doxygen

## Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.488 gdcmCMoveMessages.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  * http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMCMOVEMESSAGES_H
00019 #define GDCMCMOVEMESSAGES_H
00020
00021 #include "gdcmBaseCompositeMessage.h"
00022 #include "gdcmBaseRootQuery.h"
00023
00024 namespace gdcm{
00025     namespace network{
00026         class ULConnection;
00027     class CMoveRQ : public BaseCompositeMessage {
00028         //this class will fulfill the inheritance,
00029         //but additional information is needed by cmovd
00030         //namely, the root type or the calling AE-TITLE
00031         std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
00032     public:
00033         std::vector<PresentationDataValue> ConstructPDV(
00034             const ULConnection &inConnection,
00035             const BaseRootQuery* inRootQuery) override;
00036     };
00037     class CMoveRSP : public BaseCompositeMessage {
00038     public:
00039         std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
00040     };
00041     class CMoveCancelRq : public BaseCompositeMessage {
00042     public:
00043         std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
00044     };
00045     }
00046 }
00047 #endif

```

## 11.489 gdcmCommandDataSet.h File Reference

```

#include "gdcmDataSet.h"
#include "gdcmDataElement.h"

```



- class `gdcm::CommandDataSet`  
*Class to represent a `Command DataSet`.*

- namespace **gdcm**

- `std::ostream & gdcm::operator<< (std::ostream &os, const CommandDataSet &val)`

[Go to the documentation of this file.](#)

```
00001 /*=====*/
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 #ifndef GDCMCOMMANDDATASET_H
00015 #define GDCMCOMMANDDATASET_H
00016
00017 #include "gdcmDataSet.h"
00018 #include "gdcmDataElement.h"
00019
00020 namespace gdcm
```



## Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.492 gdcmCompositeMessageFactory.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  * http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMCOMPOSITEMESSAGEFACTORY_H
00019 #define GDCMCOMPOSITEMESSAGEFACTORY_H
00020
00021 #include "gdcmPresentationDataValue.h"
00022 #include "gdcmULConnection.h"
00023
00024 namespace gdcm {
00025     class BaseRootQuery;
00026     class File;
00027     namespace network {
00028         class BasePDU;
00037     class CompositeMessageFactory
00038     {
00039     public:
00040         //the echo request only needs a properly constructed PDV.
00041         //find, move, etc, may need something more robust, but since those are
00042         //easily placed into the appropriate pdatapdu in the pdufactory,
00043         //this approach without a base class (but done internally) is useful.
00044         static std::vector<PresentationDataValue> ConstructCEchoRQ(const ULConnection& inConnection);
00045
00046         static std::vector<PresentationDataValue> ConstructCStoreRQ(const ULConnection& inConnection, const
00047         File &file, bool writeDataSet = true );
00048         static std::vector<PresentationDataValue> ConstructCStoreRSP(const DataSet *inDataSet, const
00049         BasePDU* inPC);
00050         static std::vector<PresentationDataValue> ConstructCFindRQ(const ULConnection& inConnection, const
00051         BaseRootQuery* inRootQuery);
00052         static std::vector<PresentationDataValue> ConstructCMoveRQ(const ULConnection& inConnection, const
00053         BaseRootQuery* inRootQuery);
00054     };
00055 }
00056
00057 #endif // GDCMCOMPOSITEMESSAGEFACTORY_H

```

## 11.493 gdcmCompositeNetworkFunctions.h File Reference

```

#include "gdcmDirectory.h"
#include "gdcmBaseRootQuery.h"

```

- class `gdcm::CompositeNetworkFunctions`  
*Composite Network Functions.*

- namespace **gdcm**

[Go to the documentation of this file.](#)

Generated by Doxygen

## 11.495 gdcmCStoreMessages.h File Reference

Include dependency graph for gdcnCStoreMessages.h:



- Generated by Doxygen

*CStoreRQ.*

- class `gdcm::network::CStoreRSP`

*CStoreRSP* this file defines the messages for the cecho action.

## Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

## 11.496 gdcmCStoreMessages.h

[Go to the documentation of this file.](#)

```

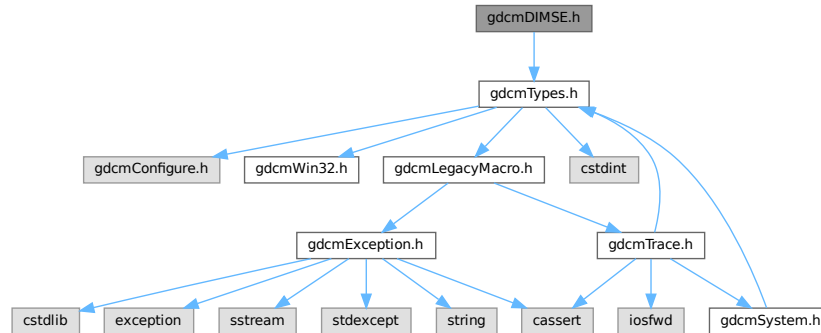
00001 /*=====
00002  *
00003  *   Copyright NumFOCUS
00004  *
00005  *   Licensed under the Apache License, Version 2.0 (the "License");
00006  *   you may not use this file except in compliance with the License.
00007  *   You may obtain a copy of the License at
00008  *
00009  *       http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  *   Unless required by applicable law or agreed to in writing, software
00012  *   distributed under the License is distributed on an "AS IS" BASIS,
00013  *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  *   See the License for the specific language governing permissions and
00015  *   limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMCSTOREMESSAGES_H
00019 #define GDCMCSTOREMESSAGES_H
00020
00021 #include "gdcmBaseCompositeMessage.h"
00022
00023 namespace gdcm{
00024 class File;
00025     namespace network{
00026     class BasePDU;
00031 class CStoreRQ : public BaseCompositeMessage {
00032     std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection, const
BaseRootQuery* inRootQuery) override;//to fulfill the virtual contract
00033     public:
00034         std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
00035             const File& file, bool writeDataSet = true );
00036     };
00037
00042     class CStoreRSP : public BaseCompositeMessage {
00043     std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection, const
BaseRootQuery* inRootQuery) override;//to fulfill the virtual contract
00044     public:
00045         std::vector<PresentationDataValue> ConstructPDV(const DataSet* inDataSet, const BasePDU* inPC);
00046     };
00047 }
00048 }
00049 #endif // GDCMCSTOREMESSAGES_H

```

## 11.497 gdcmDIMSE.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmDIMSE.h:



### Classes

- class [gdcm::network::CEchoRQ](#)  
*CEchoRQ.*
- class [gdcm::network::CEchoRSP](#)  
*CEchoRSP* this file defines the messages for the cecho action.
- class [gdcm::network::CFind](#)
- class [gdcm::network::DIMSE](#)  
*DIMSE.*

### Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.498 gdcmDIMSE.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/

```

```

00014 #ifndef GDCMDIMSE_H
00015 #define GDCMDIMSE_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021
00022 namespace network
00023 {
00024
00033 class DIMSE {
00034 public:
00035     typedef enum {
00036         C_STORE_RQ      = 0x0001,
00037         C_STORE_RSP     = 0x8001,
00038         C_GET_RQ        = 0x0010,
00039         C_GET_RSP       = 0x8010,
00040         C_FIND_RQ       = 0x0020,
00041         C_FIND_RSP     = 0x8020,
00042         C_MOVE_RQ       = 0x0021,
00043         C_MOVE_RSP     = 0x8021,
00044         C_ECHO_RQ       = 0x0030,
00045         C_ECHO_RSP     = 0x8030,
00046         N_EVENT_REPORT_RQ = 0x0100,
00047         N_EVENT_REPORT_RSP = 0x8100,
00048         N_GET_RQ        = 0x0110,
00049         N_GET_RSP       = 0x8110,
00050         N_SET_RQ        = 0x0120,
00051         N_SET_RSP       = 0x8120,
00052         N_ACTION_RQ     = 0x0130,
00053         N_ACTION_RSP    = 0x8130,
00054         N_CREATE_RQ     = 0x0140,
00055         N_CREATE_RSP    = 0x8140,
00056         N_DELETE_RQ     = 0x0150,
00057         N_DELETE_RSP    = 0x8150,
00058         C_CANCEL_RQ     = 0x0FFF
00059     } CommandTypes;
00060 };
00061
00062 /*
00063 9.1.5.1 C-ECHO parameters
00064 Table 9.1-5
00065 C-ECHO PARAMETERS
00066 */
00067 class CEchoRQ
00068 {
00069 public:
00070     uint16_t      MessageID;          /* M */
00071     UIComp        AffectedSOPClassUID; /* M */
00072 };
00073
00074 class CEchoRSP
00075 {
00076 public:
00077     /*
00078     Message ID M U
00079     Message ID Being Responded To M
00080     Affected SOP Class UID M U(=)
00081     Status M
00082     */
00083 };
00084
00089 class CFind
00090 {
00091     /*
00092     Failure Refused: Out of Resources A700 (0000,0902)
00093     Identifier does not match SOP Class A900 (0000,0901)
00094     (0000,0902)
00095     Unable to process Cxxx (0000,0901)
00096     (0000,0902)
00097     Cancel Matching terminated due to Cancel
00098     request
00099     FE00 None
00100     Success Matching is complete - No final Identifier
00101     is supplied.
00102     0000 None
00103     Pending Matches are continuing - Current Match
00104     is supplied and any Optional Keys were
00105     supported in the same manner as
00106     Required Keys.

```

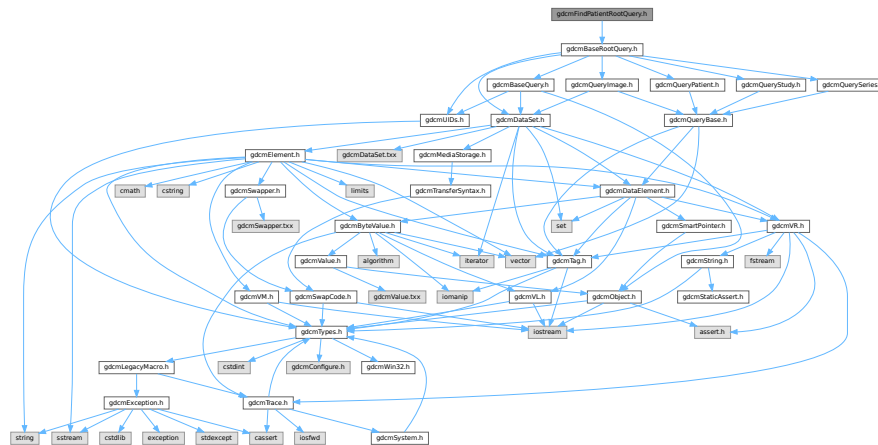


```
00107 FF00 Identifier
00108 Matches are continuing - Warning that
00109 one or more Optional Keys were not
00110 supported for existence and/or matching
00111 for this Identifier.
00112 FF01 Identifier
00113 */
00114 };
00115
00116
00117 } // end namespace network
00118
00119 } // end namespace gdcmm
00120
00121 #endif //GDCMDIMSE_H
```

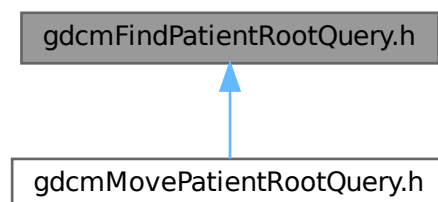
## 11.499 gdcMFindPatientRootQuery.h File Reference

```
#include "gdcmBaseRootQuery.h"
```

Include dependency graph for gdcmFindPatientRootQuery.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::FindPatientRootQuery](#)  
*PatientRootQuery.*

## Namespaces

- namespace [gdcm](#)

## 11.500 gdcmFindPatientRootQuery.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMFINDPATIENTROOTQUERY_H
00015 #define GDCMFINDPATIENTROOTQUERY_H
00016
00017 #include "gdcmBaseRootQuery.h"
00018
00019 namespace gdcm
00020 {
00021     class GDCM_EXPORT FindPatientRootQuery : public BaseRootQuery
00022     {
00023     public:
00024         FindPatientRootQuery();
00025
00026         void InitializeDataSet(const EQueryLevel& inQueryLevel) override;
00027
00028         std::vector<Tag> GetTagListByLevel(const EQueryLevel& inQueryLevel) override;
00029         bool ValidateQuery(bool inStrict = true) const override;
00030
00031         UIDs::TSName GetAbstractSyntaxUID() const override;
00032     };
00033 } // end namespace gdcm
00034
00035 #endif // GDCMFINDPATIENTROOTQUERY_H

```



```

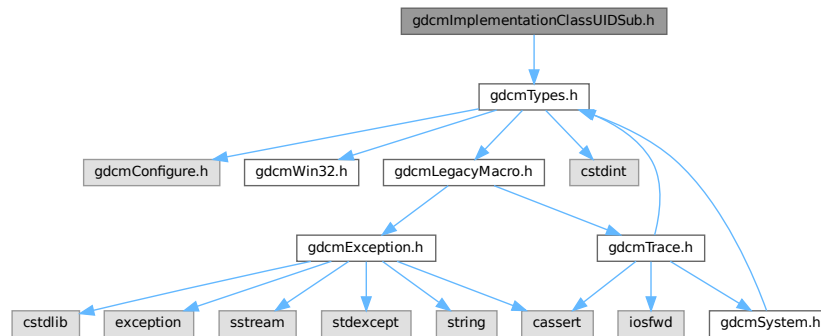
00026 {
00027     friend class QueryFactory;
00028 public:
00029     FindStudyRootQuery();
00030
00031     void InitializeDataSet(const EQueryLevel& inQueryLevel) override;
00032
00033     std::vector<Tag> GetTagListByLevel(const EQueryLevel& inQueryLevel) override;
00034
00038     bool ValidateQuery(bool inStrict = true) const override;
00039
00040     UIDs::TSName GetAbstractSyntaxUID() const override;
00041 };
00042
00043 } // end namespace gdcm
00044
00045 #endif // GDCMFINDSTUDYROOTQUERY_H

```

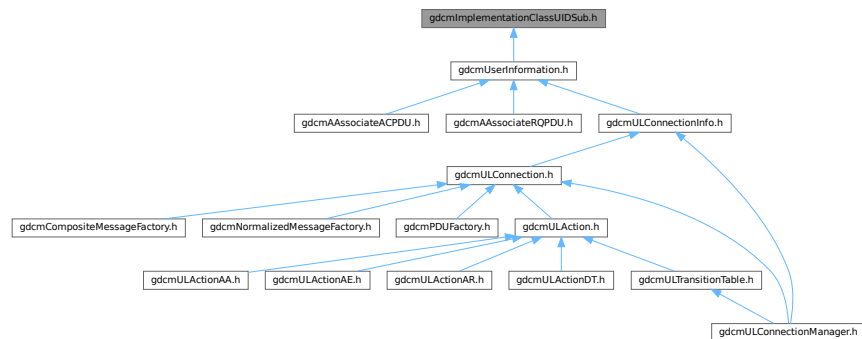
## 11.503 gdcmImplementationClassUIDSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmImplementationClassUIDSub.h:



This graph shows which files directly or indirectly include this file:



**Classes**

- class `gdcm::network::ImplementationClassUIDSub`  
*ImplementationClassUIDSub.*

**Namespaces**

- namespace `gdcm`
- namespace `gdcm::network`

**11.504 gdcmImplementationClassUIDSub.h**

[Go to the documentation of this file.](#)

```

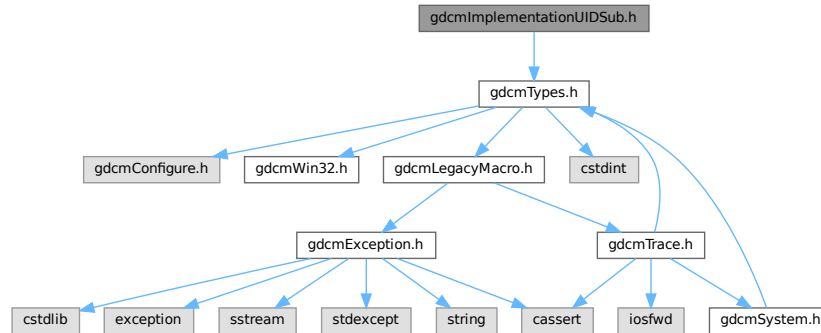
00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMIMPLEMENTATIONCLASSUIDSUB_H
00015  #define GDCMIMPLEMENTATIONCLASSUIDSUB_H
00016
00017  #include "gdcmTypes.h"
00018
00019  namespace gdcm
00020  {
00021
00022    namespace network
00023    {
00024
00025      class ImplementationClassUIDSub
00026      {
00027      public:
00028        ImplementationClassUIDSub();
00029        std::istream &Read(std::istream &is);
00030        const std::ostream &Write(std::ostream &os) const;
00031
00032        size_t Size() const;
00033
00034        void Print(std::ostream &os) const;
00035
00036      private:
00037        static const uint8_t ItemType;
00038        static const uint8_t Reserved2;
00039        uint16_t ItemLength;
00040        std::string ImplementationClassUID;
00041      };
00042    } // end namespace network
00043  } // end namespace gdcm
00044
00045  #endif //GDCMMAXIMUMLENGTHSUB_H

```

## 11.505 gdcmImplementationUIDSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmImplementationUIDSub.h:



### Classes

- class [gdcm::network::ImplementationUIDSub](#)  
*ImplementationUIDSub.*

### Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.506 gdcmImplementationUIDSub.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMIMPLEMENTATIONUIDSUB_H
00015 #define GDCMIMPLEMENTATIONUIDSUB_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021

```

```

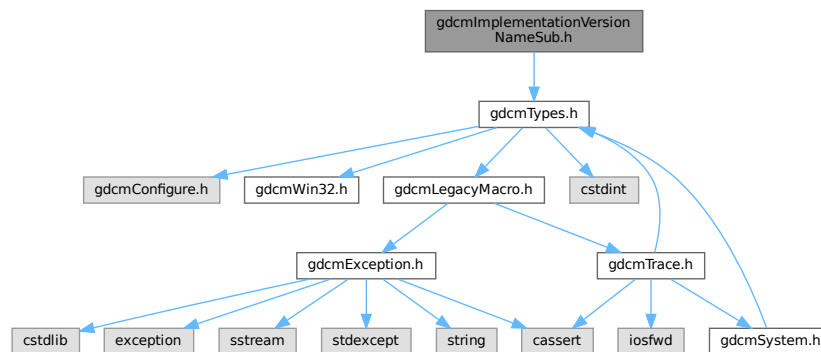
00022 namespace network
00023 {
00024
00030 class GDCM_EXPORT ImplementationUIDSub
00031 {
00032 public:
00033     ImplementationUIDSub();
00034     const std::ostream &Write(std::ostream &os) const;
00035 private:
00036     static const uint8_t ItemType;
00037     static const uint8_t Reserved2;
00038     uint16_t ItemLength;
00039     std::string ImplementationClassUID;
00040 };
00041
00042 } // end namespace network
00043
00044 } // end namespace gdcm
00045
00046 #endif //GDCMMAXIMUMLENGTHSUB_H

```

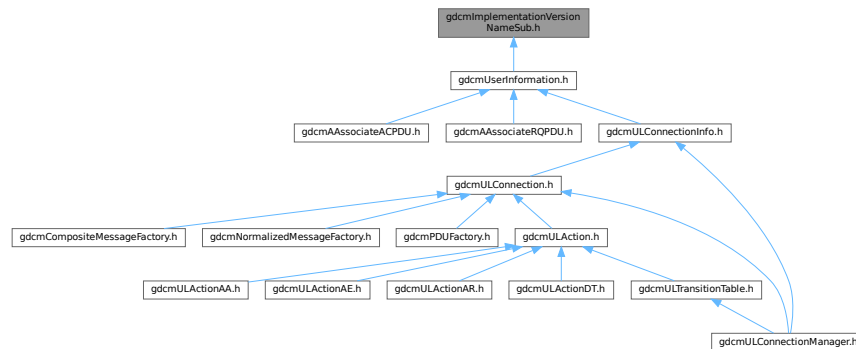
## 11.507 gdcmImplementationVersionNameSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmImplementationVersionNameSub.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::network::ImplementationVersionNameSub](#)  
*ImplementationVersionNameSub.*

## Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.508 gdcmImplementationVersionNameSub.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMIMPLEMENTATIONVERSIONNAMESUB_H
00015  #define GDCMIMPLEMENTATIONVERSIONNAMESUB_H
00016
00017  #include "gdcmTypes.h"
00018
00019  namespace gdcm
00020  {
00021
00022    namespace network
00023    {
00024
00030      class ImplementationVersionNameSub
00031      {
00032      public:
00033        ImplementationVersionNameSub();
00034        std::istream &Read(std::istream &is);
00035        const std::ostream &Write(std::ostream &os) const;
00036
00037        size_t Size() const;
00038        void Print(std::ostream &os) const;
00039
00040      private:
00041        static const uint8_t ItemType;
00042        static const uint8_t Reserved2;
00043        uint16_t ItemLength;
00044        std::string ImplementationVersionName;
00045      };
00046
00047    } // end namespace network
00048
00049  } // end namespace gdcm
00050
00051  #endif //GDCMMAXIMUMLENGTHSUB_H

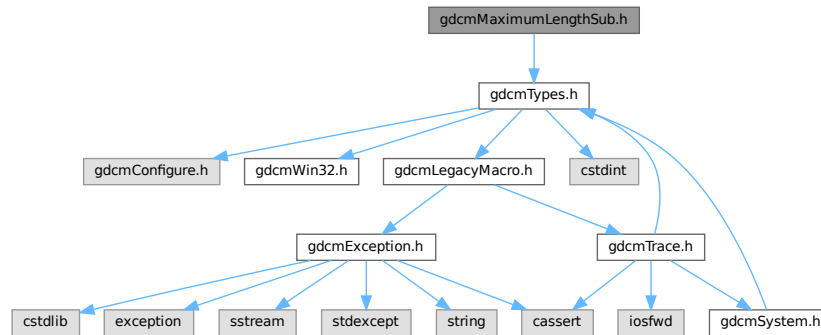
```



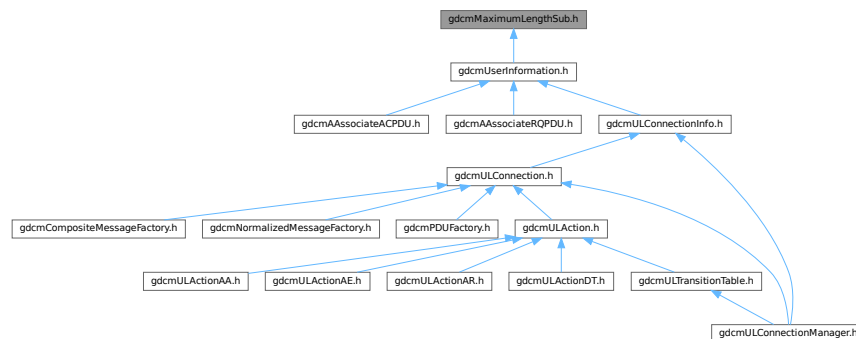
## 11.509 gdcmMaximumLengthSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmMaximumLengthSub.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class `gdcm::network::MaximumLengthSub`  
*MaximumLengthSub.*

### Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

## 11.510 gdcmMaximumLengthSub.h

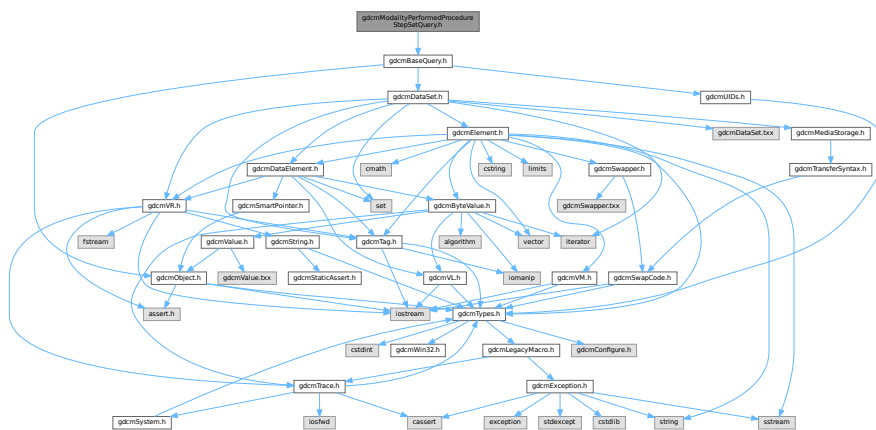
[Go to the documentation of this file.](#)

```
00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMMAXIMUMLENGTHSUB_H
00015 #define GDCMMAXIMUMLENGTHSUB_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021
00022   namespace network
00023   {
00024
00025     class MaximumLengthSub
00026     {
00027     public:
00028       MaximumLengthSub();
00029       std::istream &Read(std::istream &is);
00030       const std::ostream &Write(std::ostream &os) const;
00031
00032       size_t Size() const;
00033
00034       uint32_t GetMaximumLength() const { return MaximumLength; }
00035       void SetMaximumLength(uint32_t maximumlength);
00036
00037       void Print(std::ostream &os) const;
00038
00039     private:
00040       static const uint8_t ItemType;
00041       static const uint8_t Reserved2;
00042       uint16_t ItemLength;
00043       uint32_t MaximumLength;
00044     };
00045
00046   } // end namespace network
00047
00048 } // end namespace gdcm
00049
00050 #endif //GDCMMAXIMUMLENGTHSUB_H
```



## 11.513 gdcmModalityPerformedProcedureStepSetQuery.h File Reference

Include dependency graph for `gdcModalityPerformedProcedureStepSetQuery.h`:



- class `gdcmm::ModalityPerformedProcedureStepSetQuery`  
*ModalityPerformedProcedureStepSetQuery*.

- namespace **gdcm**

## 11.514 gdcmModalityPerformedProcedureStepSetQuery.h

[Go to the documentation of this file.](#)

```

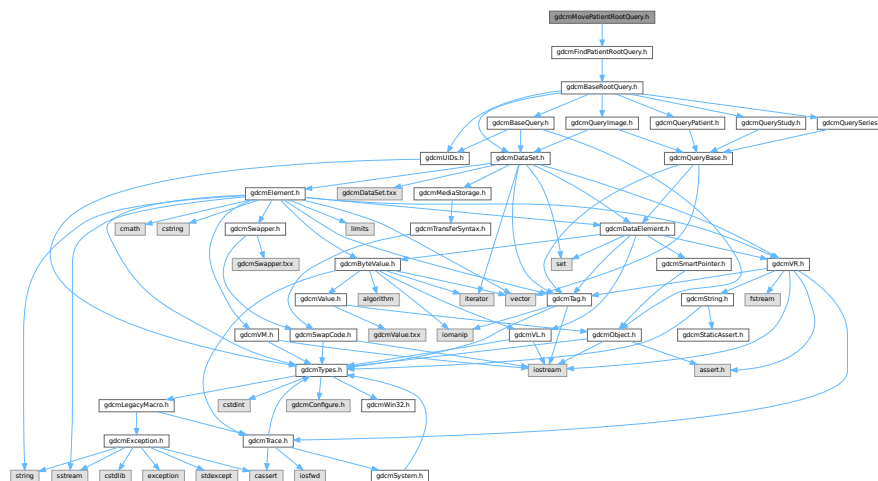
00001 /*
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 #ifndef GDCMMODALITYPERFORMEDPROCEDURESTEPSETQUERY_H
00015 #define GDCMMODALITYPERFORMEDPROCEDURESTEPSETQUERY_H
00016
00017 #include "gdcmBaseQuery.h"
00018
00019 namespace gdcm
00020 {
00021     class GDCM_EXPORT ModalityPerformedProcedureStepSetQuery : public BaseQuery{
00022     friend class QueryFactory;
00023     public:
00024         ModalityPerformedProcedureStepSetQuery( const std::string & iSopInstanceUID );
00025
00026         gdcm::DataSet GetRequiredDataSet() const;
00027         bool ValidateQuery(bool inStrict = true) const override;
00028         UIDs::TSName GetAbstractSyntaxUID() const override;
00029     };
00030
00031 } // end namespace gdcm
00032
00033 #endif // GDCMMODALITYPERFORMEDPROCEDURESTEPSETQUERY_H

```

## 11.515 gdcmmovepatientrootquery.h File Reference

```
#include "gdcmFindPatientRootQuery.h"
```

Include dependency graph for gdcmMovePatientRootQuery.h:



## Classes

- class `gdcm::MovePatientRootQuery`  
*MovePatientRootQuery.*

## Namespaces

- namespace `gdcm`

## 11.516 gdcmMovePatientRootQuery.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMMOVEPATIENTROOTQUERY_H
00015 #define GDCMMOVEPATIENTROOTQUERY_H
00016
00017 #include "gdcmFindPatientRootQuery.h"
00018
00019 namespace gdcm
00020 {
00021     class GDCM_EXPORT MovePatientRootQuery : public BaseRootQuery
00022     {
00023     public:
00024         MovePatientRootQuery();
00025
00026         void InitializeDataSet(const EQueryLevel& inQueryLevel) override;
00027
00028         std::vector<Tag> GetTagListByLevel(const EQueryLevel& inQueryLevel) override;
00029
00030         bool ValidateQuery(bool inStrict = true) const override;
00031
00032         UIDs::TSName GetAbstractSyntaxUID() const override;
00033     };
00034 } // end namespace gdcm
00035
00036 #endif // GDCMMOVEPATIENTROOTQUERY_H

```









## Classes

- class [gdcm::network::NCreateRQ](#)  
*NCreateRQ.*
- class [gdcm::network::NCreateRSP](#)  
*NCreateRSP this file defines the messages for the ncreate action.*

## Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.522 gdcmNCreateMessages.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2014 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMCNCREATEMESSAGES_H
00015 #define GDCMCNCREATEMESSAGES_H
00016
00017 #include "gdcmBaseNormalizedMessage.h"
00018
00019 namespace gdcm{
00020     namespace network{
00021
00022     class ULConnection;
00023
00024     class NCreateRQ : public BaseNormalizedMessage {
00025     public:
00026         std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
00027             const BaseQuery* inQuery) override;
00028     };
00029
00030     class NCreateRSP : public BaseNormalizedMessage {
00031     public:
00032         std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
00033     };
00034     }
00035 }
00036
00037 #endif // GDCMCNCREATEMESSAGES_H

```



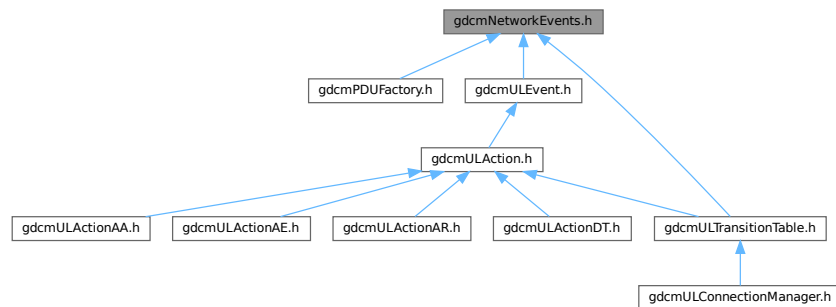
```

00018
00019 namespace gdcm{
00020   namespace network{
00021
00022   class ULConnection;
00023
00028   class NDeleteRQ : public BaseNormalizedMessage {
00029   public:
00030     std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
00031     const BaseQuery* inQuery) override;
00032   };
00033
00038   class NDeleteRSP : public BaseNormalizedMessage {
00039   public:
00040     std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
00041   };
00042   }
00043 }
00044 #endif // GDCMCNDELETEMESSAGES_H

```

## 11.525 gdcmNetworkEvents.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

### Enumerations

- enum `gdcm::network::EEventID` {  
`gdcm::network::eAASSOCIATERequestLocalUser = 0` ,  
`gdcm::network::eTransportConnConfirmLocal` ,  
`gdcm::network::eASSOCIATE_ACPDUreceived` ,  
`gdcm::network::eASSOCIATE_RJPDUreceived` ,  
`gdcm::network::eTransportConnIndicLocal` ,  
`gdcm::network::eAASSOCIATE_RQPDUreceived` ,  
`gdcm::network::eAASSOCIATEResponseAccept` ,  
`gdcm::network::eAASSOCIATEResponseReject` ,

```

gdcm::network::ePDATArequest ,
gdcm::network::ePDATATFPDU ,
gdcm::network::eARELEASERequest ,
gdcm::network::eARELEASE_RQPDUReceivedOpen ,
gdcm::network::eARELEASE_RPPDUReceived ,
gdcm::network::eARELEASEResponse ,
gdcm::network::eAABORTRequest ,
gdcm::network::eAABORTPDUReceivedOpen ,
gdcm::network::eTransportConnectionClosed ,
gdcm::network::eARTIMTimerExpired ,
gdcm::network::eUnrecognizedPDUReceived ,
gdcm::network::eEventDoesNotExist }

```

## Variables

- const int gdcm::network::cMaxEventID = eEventDoesNotExist

## 11.526 gdcmNetworkEvents.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  * http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 /*
00019 The NetworkEvents enumeration defines the inputs into the state of the network connection.
00020
00021 These inputs can come either from user input or input from other things on the socket,
00022 ie, responses from the peer or ARTIM timeouts.
00023
00024 Note that this enumeration is not 'power of two', like the states, because you can't have
00025 multiple simultaneous events. Multiple state outputs in transition tables, however, is possible.
00026
00027 */
00028 #ifndef GDCMNETWORKEVENTS_H
00029 #define GDCMNETWORKEVENTS_H
00030
00031 namespace gdcm {
00032 namespace network {
00033 typedef enum {
00034 eAASSOCIATERequestLocalUser = 0,
00035 eTransportConnConfirmLocal,
00036 eASSOCIATE_ACPDUreceived,
00037 eASSOCIATE_RJPDUreceived,
00038 eTransportConnIndicLocal,
00039 eAASSOCIATE_RQPDUreceived,
00040 eAASSOCIATEResponseAccept,
00041 eAASSOCIATEResponseReject,
00042 ePDATArequest,
00043 ePDATATFPDU,
00044 eARELEASERequest,
00045 eARELEASE_RQPDUReceivedOpen,
00046 eARELEASE_RPPDUReceived,

```

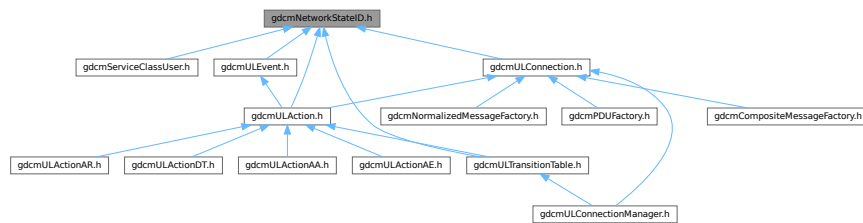
```

00047     eARELEASEResponse,
00048     eAABORTRequest,
00049     eAABORTPDUReceivedOpen,
00050     eTransportConnectionClosed,
00051     eARTIMTimerExpired,
00052     eUnrecognizedPDUReceived,
00053     eEventDoesNotExist
00054 } EEventID;
00055
00056 const int cMaxEventID = eEventDoesNotExist;
00057 }
00058 }
00059
00060 #endif //NETWORKEVENTS_H

```

## 11.527 gdcmlNetworkStateID.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

- namespace [gdcml](#)
- namespace [gdcml::network](#)

### Enumerations

- enum [gdcml::network::EStateID](#) {  
[gdcml::network::eStaDoesNotExist](#) = 0 ,  
[gdcml::network::eSta1Idle](#) = 1 ,  
[gdcml::network::eSta2Open](#) = 2 ,  
[gdcml::network::eSta3WaitLocalAssoc](#) = 4 ,  
[gdcml::network::eSta4LocalAssocDone](#) = 8 ,  
[gdcml::network::eSta5WaitRemoteAssoc](#) = 16 ,  
[gdcml::network::eSta6TransferReady](#) = 32 ,  
[gdcml::network::eSta7WaitRelease](#) = 64 ,  
[gdcml::network::eSta8WaitLocalRelease](#) = 128 ,  
[gdcml::network::eSta9ReleaseCollisionRqLocal](#) = 256 ,  
[gdcml::network::eSta10ReleaseCollisionAc](#) = 512 ,  
[gdcml::network::eSta11ReleaseCollisionRq](#) = 1024 ,  
[gdcml::network::eSta12ReleaseCollisionAcLocal](#) = 2048 ,  
[gdcml::network::eSta13AwaitingClose](#) = 4096 }

**Functions**

- int `gdc::network::GetStateIndex` (EStateID inState)

**Variables**

- const int `gdc::network::cMaxStateID` = 13

**11.528 gdcNetworkStateID.h**

[Go to the documentation of this file.](#)

```

00001 /*=====
00002  *
00003  *   Copyright NumFOCUS
00004  *
00005  *   Licensed under the Apache License, Version 2.0 (the "License");
00006  *   you may not use this file except in compliance with the License.
00007  *   You may obtain a copy of the License at
00008  *
00009  *       http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  *   Unless required by applicable law or agreed to in writing, software
00012  *   distributed under the License is distributed on an "AS IS" BASIS,
00013  *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  *   See the License for the specific language governing permissions and
00015  *   limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMNETWORKSTATEID_H
00019 #define GDCMNETWORKSTATEID_H
00020
00021 namespace gdc {
00022     namespace network {
00023
00024         enum EStateID {
00025             eStaDoesNotExist = 0,
00026             eStaIdle = 1,
00027             eSta2Open = 2,
00028             eSta3WaitLocalAssoc = 4,
00029             eSta4LocalAssocDone = 8,
00030             eSta5WaitRemoteAssoc = 16,
00031             eSta6TransferReady = 32,
00032             eSta7WaitRelease = 64,
00033             eSta8WaitLocalRelease = 128,
00034             eSta9ReleaseCollisionRqLocal = 256,
00035             eSta10ReleaseCollisionAc = 512,
00036             eSta11ReleaseCollisionRq = 1024,
00037             eSta12ReleaseCollisionAcLocal = 2048,
00038             eSta13AwaitingClose = 4096
00039         };
00040
00041         const int cMaxStateID = 13;
00042
00043         //the transition table is built on state indices
00044         //this function will produce the index from the power-of-two EStateID
00045         inline int GetStateIndex(EStateID inState){
00046             switch (inState){
00047                 case eStaDoesNotExist:
00048                     default:
00049                         return -1;
00050                 case eStaIdle:
00051                     return 0;
00052                 case eSta2Open:
00053                     return 1;
00054                 case eSta3WaitLocalAssoc:
00055                     return 2;
00056                 case eSta4LocalAssocDone:
00057                     return 3;
00058                 case eSta5WaitRemoteAssoc:
00059                     return 4;
00060             }
00061         }
00062     }
00063 }

```







## Classes

- class [gdcm::network::NGetRQ](#)  
*NGetRQ.*
- class [gdcm::network::NGetRSP](#)  
*NGetRSP this file defines the messages for the nget action.*

## Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.532 gdcmNGetMessages.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2014 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMCNGETMESSAGES_H
00015 #define GDCMCNGETMESSAGES_H
00016
00017 #include "gdcmBaseNormalizedMessage.h"
00018
00019 namespace gdcm{
00020     namespace network{
00021
00022     class ULConnection;
00023
00028     class NGetRQ : public BaseNormalizedMessage {
00029     public:
00030         std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
00031             const BaseQuery* inQuery) override;
00032     };
00033
00038     class NGetRSP : public BaseNormalizedMessage {
00039     public:
00040         std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
00041     };
00042     }
00043 }
00044 #endif // GDCMCNGETMESSAGES_H

```

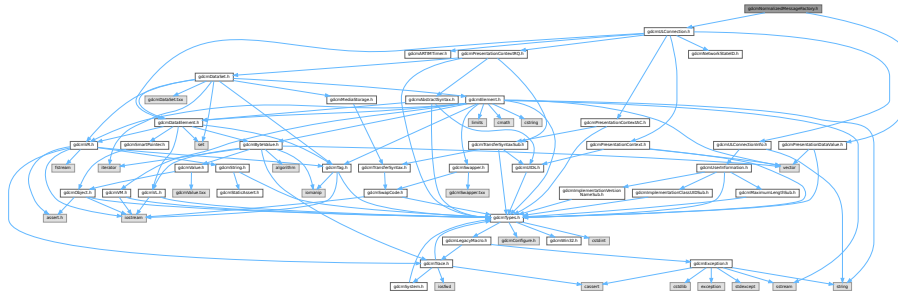
## 11.533 gdcmNormalizedMessageFactory.h File Reference

```

#include "gdcmPresentationDataValue.h"
#include "gdcmULConnection.h"

```

Include dependency graph for gdcmNormalizedMessageFactory.h:



## Classes

- class [gdcm::network::NormalizedMessageFactory](#)

## Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.534 gdcmNormalizedMessageFactory.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2014 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMNORMALIZEDMESSAGEFACTORY_H
00015 #define GDCMNORMALIZEDMESSAGEFACTORY_H
00016
00017 #include "gdcmPresentationDataValue.h"
00018 #include "gdcmULConnection.h"
00019
00020 namespace gdcm {
00021     class BaseQuery;
00022     class File;
00023     namespace network {
00024         class BasePDU;
00025
00026     class NormalizedMessageFactory
00027     {
00028     public:
00029         static std::vector<PresentationDataValue> ConstructNEventReport (const ULConnection& inConnection,
00030 const BaseQuery* inQuery);
00031         static std::vector<PresentationDataValue> ConstructNGet (const ULConnection& inConnection,
00032 const BaseQuery* inQuery);
00033         static std::vector<PresentationDataValue> ConstructNSet (const ULConnection& inConnection,
00034 const BaseQuery* inQuery);

```



## 11.536 gdcmNormalizedNetworkFunctions.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2014 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMNORMALIZEDNETWORKFUNCTIONS_H
00015  #define GDCMNORMALIZEDNETWORKFUNCTIONS_H
00016
00017  #include "gdcmDirectory.h"
00018  #include "gdcmBaseQuery.h" // EQueryLevel / EQueryType
00019
00020  #include <vector>
00021  #include <string>
00022
00023  namespace gdcm
00024  {
00046  class GDCM_EXPORT NormalizedNetworkFunctions
00047  {
00048  public:
00049      static BaseQuery* ConstructQuery( const std::string & sopInstanceUID,
00050                                       const DataSet& queryds, ENQueryType queryType = eCreateMMPS );
00051      static bool NEventReport( const char *remote, uint16_t portno,
00052                               const BaseQuery* query, std::vector<DataSet> &retDataSets,
00053                               const char *aetitle, const char *call );
00054      static bool NGet( const char *remote, uint16_t portno,
00055                       const BaseQuery* query, std::vector<DataSet> &retDataSets,
00056                       const char *aetitle, const char *call );
00057      static bool NSet( const char *remote, uint16_t portno,
00058                       const BaseQuery* query, std::vector<DataSet> &retDataSets,
00059                       const char *aetitle, const char *call );
00060      static bool NAction( const char *remote, uint16_t portno,
00061                          const BaseQuery* query, std::vector<DataSet> &retDataSets,
00062                          const char *aetitle, const char *call );
00063      static bool NCreate( const char *remote, uint16_t portno,
00064                          BaseQuery* query, std::vector<DataSet> &retDataSets,
00065                          const char *aetitle, const char *call );
00066      static bool NDelete( const char *remote, uint16_t portno,
00067                          const BaseQuery* query, std::vector<DataSet> &retDataSets,
00068                          const char *aetitle, const char *call );
00069  };
00070
00071  } // end namespace gdcm
00072
00073  #endif // GDCMCOMPOSITENETWORKFUNCTIONS_H

```



```

00018
00019 namespace gdcm{
00020     namespace network{
00021
00022     class ULConnection;
00023
00028     class NSetRQ : public BaseNormalizedMessage {
00029     public:
00030         std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
00031             const BaseQuery* inQuery) override;
00032     };
00033
00038     class NSetRSP : public BaseNormalizedMessage {
00039     public:
00040         std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
00041     };
00042     }
00043 }
00044 #endif // GDCMCNSETMESSAGES_H

```

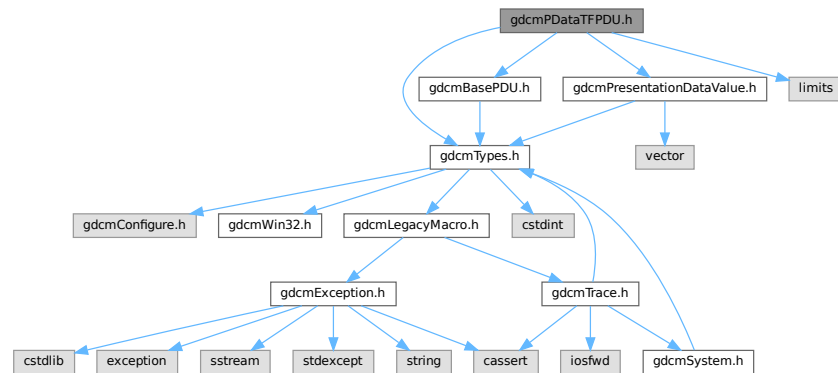
## 11.539 gdcmPDataTFPDU.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmPresentationDataValue.h"
#include "gdcmBasePDU.h"
#include <limits>

```

Include dependency graph for gdcmPDataTFPDU.h:



### Classes

- class [gdcm::network::PDataTFPDU](#)  
*PDataTFPDU*.

### Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.540 gdcmPDataTFPDU.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMPDATATFPDU_H
00015 #define GDCMPDATATFPDU_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmPresentationDataValue.h"
00019 #include "gdcmBasePDU.h"
00020 #include <limits>
00021
00022 namespace gdcm
00023 {
00024
00025     namespace network
00026     {
00027
00033         class GDCM_EXPORT PDataTFPDU : public BasePDU
00034         {
00035         public:
00036             PDataTFPDU();
00037             std::istream &Read(std::istream &is) override;
00038             const std::ostream &Write(std::ostream &os) const override;
00039
00041             size_t Size() const override;
00042
00043             void AddPresentationDataValue( PresentationDataValue const &pdv ) {
00044                 V.push_back( pdv );
00045                 assert(Size() < std::numeric_limits<uint32_t>::max());
00046                 ItemLength = (uint32_t)Size() - 6;
00047             }
00048
00049             typedef std::vector<PresentationDataValue>::size_type SizeType;
00050             PresentationDataValue const &GetPresentationDataValue(SizeType i) const {
00051                 assert( !V.empty() && i < V.size() );
00052                 return V[i];
00053             }
00054             SizeType GetNumberOfPresentationDataValues() const {
00055                 return V.size();
00056             }
00057
00058             void Print(std::ostream &os) const override;
00059             bool IsLastFragment() const override;
00060
00061         protected:
00062             std::istream &ReadInto(std::istream &is, std::ostream &os);
00063         private:
00064             static const uint8_t ItemType; // PDUType ?
00065             static const uint8_t Reserved2;
00066             uint32_t ItemLength; // PDU Length ?
00067             std::vector<PresentationDataValue> V;
00068         };
00069
00070     } // end namespace network
00071
00072 } // end namespace gdcm
00073
00074 #endif //GDCMPDATATFPDU_H

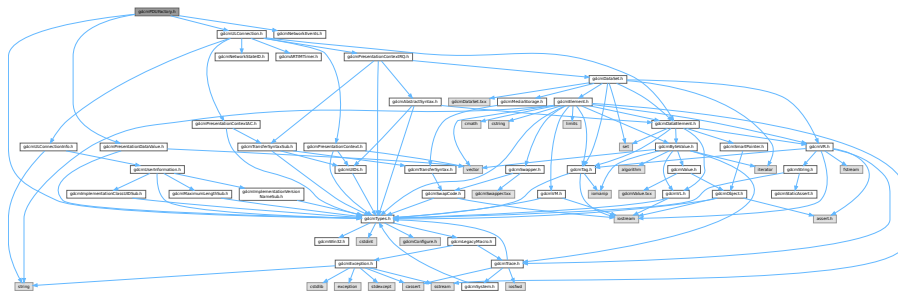
```



## 11.541 gdcmPDUFactory.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmNetworkEvents.h"
#include "gdcmULConnection.h"
#include "gdcmPresentationDataValue.h"
```

Include dependency graph for gdcmPDUFactory.h:



### Classes

- class [gdcm::network::PDUFactory](#)  
*PDUFactory* basically, given an initial byte, construct the.

### Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.542 gdcmPDUFactory.h

[Go to the documentation of this file.](#)

```
00001 /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  * http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMPDUFACTORY_H
00019 #define GDCMPDUFACTORY_H
00020
00021 #include "gdcmTypes.h"
00022 #include "gdcmNetworkEvents.h"
00023 #include "gdcmULConnection.h"
```

```

00024 #include "gdcmPresentationDataValue.h"
00025
00026 namespace gdcm{
00027     class BaseRootQuery;
00028     class BaseQuery;
00029     class File;
00030     namespace network{
00031         class BasePDU;
00032
00033     class PDUFactory {
00034     public:
00035         static BasePDU* ConstructPDU(uint8_t itemtype); //eventually needs to be smartpointer'd
00036         static EEventID DetermineEventByPDU(const BasePDU* inPDU);
00037         static BasePDU* ConstructReleasePDU();
00038         static BasePDU* ConstructAbortPDU();
00039
00040         //these are the composite PDU construction methods for the PDataPDUs.
00041         //basically, builds a pdatapdu, and then puts the appropriate information in
00042         //for the appropriate composite service (c-echo, c-find, c-store, c-get, c-move)
00043         //the connection is necessary to construct the stream of PDVs that will
00044         //be then placed into the vector of PDUs
00045         static std::vector<BasePDU*> CreateCEchoPDU(const ULConnection& inConnection);
00046         static std::vector<BasePDU*> CreateCStoreRQPDU(const ULConnection& inConnection, const File &file,
00047 bool writeDataSet = true );
00048         static std::vector<BasePDU*> CreateCStoreRSPDPDU(const DataSet *inDataSet, const BasePDU* inPC);
00049         static std::vector<BasePDU*> CreateCFindPDU(const ULConnection& inConnection, const BaseRootQuery*
00050 inRootQuery);
00051         static std::vector<BasePDU*> CreateCMovePDU(const ULConnection& inConnection, const BaseRootQuery*
00052 inRootQuery);
00053
00054         static std::vector<BasePDU*> CreateNEventReportPDU (const ULConnection& inConnection, const BaseQuery
00055 *inQuery);
00056         static std::vector<BasePDU*> CreateNGetPDU (const ULConnection& inConnection, const BaseQuery
00057 *inQuery);
00058         static std::vector<BasePDU*> CreateNSetPDU (const ULConnection& inConnection, const BaseQuery
00059 *inQuery);
00060         static std::vector<BasePDU*> CreateNActionPDU (const ULConnection& inConnection, const BaseQuery
00061 *inQuery);
00062         static std::vector<BasePDU*> CreateNCreatePDU (const ULConnection& inConnection, const BaseQuery
00063 *inQuery);
00064         static std::vector<BasePDU*> CreateNDeletePDU (const ULConnection& inConnection, const BaseQuery
00065 *inQuery);
00066
00067         //given data pdus, produce the presentation data values stored within.
00068         //all operations have these as the payload of the data sending operation
00069         //however, echo does not have a dataset in the pdv.
00070         static std::vector<PresentationDataValue> GetPDVs(const std::vector<BasePDU*> & inDataPDUs);
00071     };
00072 }
00073 #endif //GDCMPDUFACORY_H

```

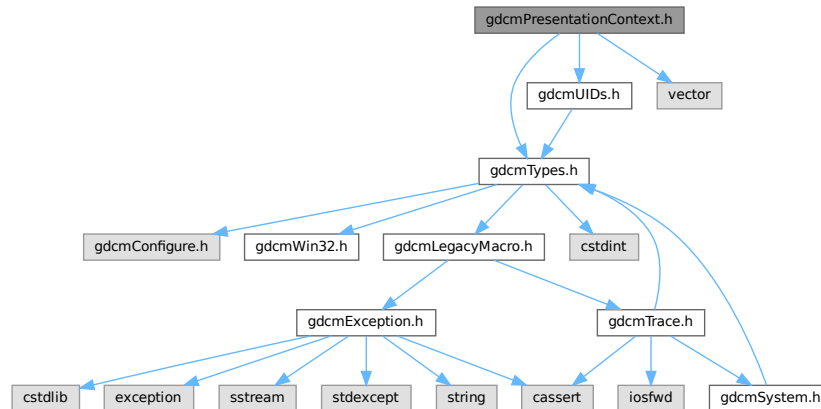
## 11.543 gdcmPresentationContext.h File Reference

```

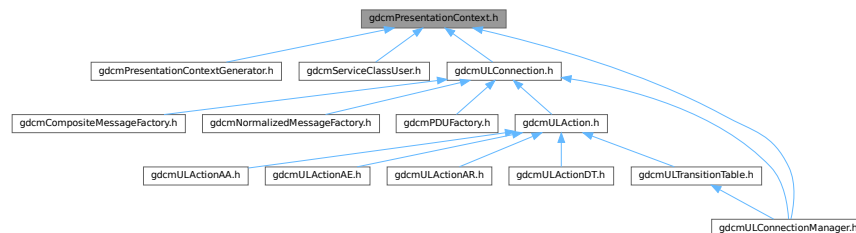
#include "gdcmTypes.h"
#include "gdcmUIDs.h"
#include <vector>

```

Include dependency graph for gdcmPresentationContext.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::PresentationContext](#)  
*PresentationContext.*

## Namespaces

- namespace [gdcm](#)

## 11.544 gdcmPresentationContext.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003 Program: GDCM (Grassroots DICOM). A DICOM library

```

```

00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMPRESENTATIONCONTEXT_H
00015 #define GDCMPRESENTATIONCONTEXT_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmUIDs.h"
00019
00020 #include <vector>
00021
00022 namespace gdcm
00023 {
00024
00029 class GDCM_EXPORT PresentationContext
00030 {
00031 public:
00032     PresentationContext();
00033
00037     PresentationContext( UIDs::TSName asname,
00038         UIDs::TSName tsname = UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM );
00039
00040     void SetAbstractSyntax( const char *absyn ) { AbstractSyntax = absyn; }
00041     const char *GetAbstractSyntax() const { return AbstractSyntax.c_str(); }
00042
00043     void AddTransferSyntax( const char *tsstr );
00044     typedef std::vector<std::string> TransferSyntaxArrayType;
00045     typedef TransferSyntaxArrayType::size_type SizeType;
00046     const char *GetTransferSyntax(SizeType i) const { return TransferSyntaxes[i].c_str(); }
00047     SizeType GetNumberOfTransferSyntaxes() const { return TransferSyntaxes.size(); }
00048
00049     void SetPresentationContextID( uint8_t id );
00050     uint8_t GetPresentationContextID() const;
00051
00052     void Print(std::ostream &os) const;
00053
00054     bool operator==(const PresentationContext & pc) const
00055     {
00056         assert( TransferSyntaxes.size() == 1 ); // TODO
00057         assert( pc.TransferSyntaxes.size() == 1 );
00058         return AbstractSyntax == pc.AbstractSyntax && TransferSyntaxes == pc.TransferSyntaxes;
00059     }
00060
00061 protected :
00062     std::string AbstractSyntax;
00063     std::vector<std::string> TransferSyntaxes;
00064     uint8_t /*PresentationContext*/ID;
00065 };
00066
00067 } // end namespace gdcm
00068
00069 #endif //GDCMPRESENTATIONCONTEXT_H

```

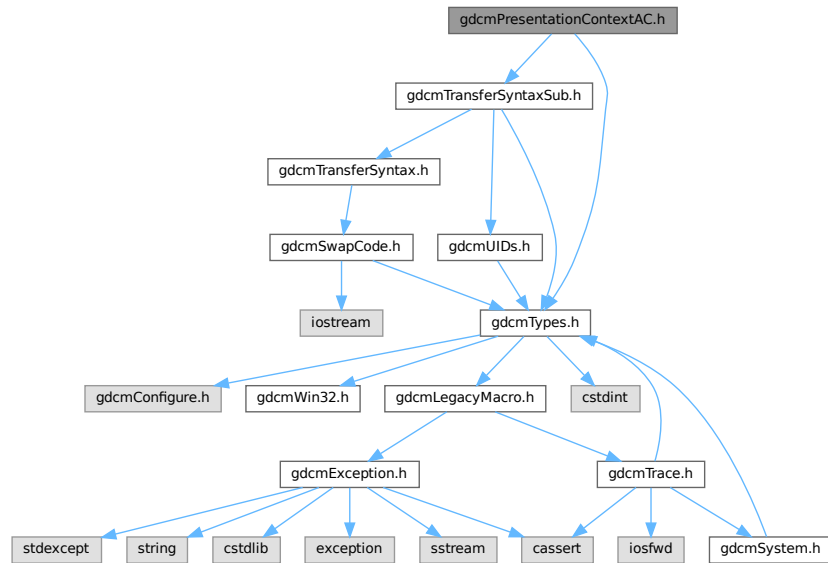
## 11.545 gdcmPresentationContextAC.h File Reference

```

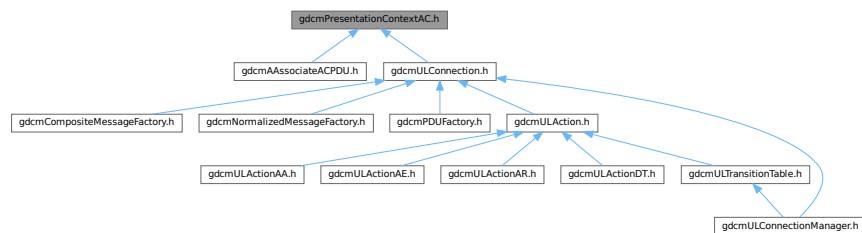
#include "gdcmTypes.h"
#include "gdcmTransferSyntaxSub.h"

```

Include dependency graph for gdcmPresentationContextAC.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::network::PresentationContextAC](#)  
*PresentationContextAC.*

## Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.546 gdcmPresentationContextAC.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMPRESENTATIONCONTEXTAC_H
00015  #define GDCMPRESENTATIONCONTEXTAC_H
00016
00017  #include "gdcmTypes.h"
00018  #include "gdcmTransferSyntaxSub.h"
00019
00020  namespace gdcm
00021  {
00022
00023  namespace network
00024  {
00025
00032  class PresentationContextAC
00033  {
00034  public:
00035      PresentationContextAC();
00036      std::istream &Read(std::istream &is);
00037      const std::ostream &Write(std::ostream &os) const;
00038
00039      size_t Size() const;
00040
00041      void SetTransferSyntax( TransferSyntaxSub const &ts );
00042      void SetPresentationContextID( uint8_t id );
00043
00044      void Print(std::ostream &os) const;
00045
00046      uint8_t GetPresentationContextID() const
00047      {
00048          return ID;
00049      }
00050      TransferSyntaxSub const & GetTransferSyntax() const { return SubItems; }
00051
00052      void SetReason( uint8_t r ) { Result = r; }
00053      uint8_t GetReason() const { return Result; }
00054
00055  private:
00056      static const uint8_t ItemType;
00057      static const uint8_t Reserved2;
00058      uint16_t ItemLength; // len of last transfer syntax
00059      uint8_t /*PresentationContext*/ID;
00060      static const uint8_t Reserved6;
00061      uint8_t /*Reason*/Result;
00062      static const uint8_t Reserved8;
00063      TransferSyntaxSub SubItems;
00064  };
00065
00066  } // end namespace network
00067
00068  } // end namespace gdcm
00069
00070  #endif //GDCMPRESENTATIONCONTEXTAC_H

```

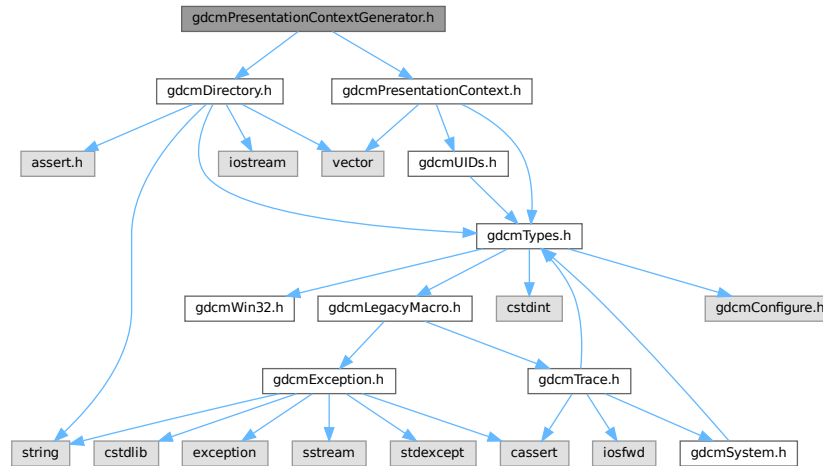
## 11.547 gdcmPresentationContextGenerator.h File Reference

```

#include "gdcmDirectory.h"
#include "gdcmPresentationContext.h"

```

Include dependency graph for gdcmPresentationContextGenerator.h:



## Classes

- class [gdcm::PresentationContextGenerator](#)  
*PresentationContextGenerator.*

## Namespaces

- namespace [gdcm](#)

## 11.548 gdcmPresentationContextGenerator.h

[Go to the documentation of this file.](#)

```

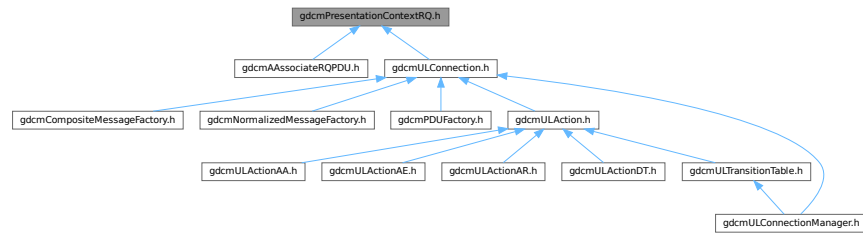
00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMPRESENTATIONCONTEXTGENERATOR_H
00015 #define GDCMPRESENTATIONCONTEXTGENERATOR_H
00016
00017 #include "gdcmDirectory.h"
00018 #include "gdcmPresentationContext.h"
00019
00020 namespace gdcm
00021 {
00022   class TransferSyntax;
00023   class File;

```

## 11.549 gdcmpresentationContextRQ.h File Reference



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcmm::network::PresentationContextRQ`  
*PresentationContextRQ.*

## Namespaces

- namespace `gdcmm`
- namespace `gdcmm::network`

## 11.550 gdcmpresentationcontextrq.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMPRESENTATIONCONTEXTTRQ_H
00015 #define GDCMPRESENTATIONCONTEXTTRQ_H
00016
00017 #include "gdcmmTypes.h"
00018 #include "gdcmmAbstractSyntax.h"
00019 #include "gdcmmTransferSyntaxSub.h"
00020 #include "gdcmmDataSet.h"
00021
00022 namespace gdcmm
00023 {
00024 class PresentationContext;
00025 namespace network
00026 {
00027
00034 class GDCM_EXPORT PresentationContextRQ
00035 {
00036 public:
00037     PresentationContextRQ();
00038
00042     PresentationContextRQ(UIDs::TSName asname, UIDs::TSName tname =
00043         UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM );

```

```

00044
00045     std::istream &Read(std::istream &is);
00046     const std::ostream &Write(std::ostream &os) const;
00047     size_t Size() const;
00048
00049     void SetAbstractSyntax( AbstractSyntax const & absyn );
00050     AbstractSyntax const &GetAbstractSyntax() const { return SubItems; }
00051     AbstractSyntax &GetAbstractSyntax() { return SubItems; }
00052
00053     void AddTransferSyntax( TransferSyntaxSub const &ts );
00054     typedef std::vector<TransferSyntaxSub>::size_type SizeType;
00055     TransferSyntaxSub const & GetTransferSyntax(SizeType i) const { return TransferSyntaxes[i]; }
00056     TransferSyntaxSub & GetTransferSyntax(SizeType i) { return TransferSyntaxes[i]; }
00057     std::vector<TransferSyntaxSub> const & GetTransferSyntaxes() const {return TransferSyntaxes; }
00058     SizeType GetNumberOfTransferSyntaxes() const { return TransferSyntaxes.size(); }
00059
00060     void SetPresentationContextID( uint8_t id );
00061     uint8_t GetPresentationContextID() const;
00062
00063     void Print(std::ostream &os) const;
00064
00065     bool operator==(const PresentationContextRQ & pc) const
00066     {
00067         assert( TransferSyntaxes.size() == 1 ); // TODO
00068         assert( pc.TransferSyntaxes.size() == 1 );
00069         return SubItems == pc.SubItems && TransferSyntaxes == pc.TransferSyntaxes;
00070     }
00071
00072     PresentationContextRQ(const PresentationContext & pc);
00073
00074 private:
00075     static const uint8_t ItemType;
00076     static const uint8_t Reserved2;
00077     uint16_t ItemLength; // len of last transfer syntax
00078     uint8_t /*PresentationContext*/ID;
00079     static const uint8_t Reserved6;
00080     static const uint8_t Reserved7;
00081     static const uint8_t Reserved8;
00082     /*
00083     This variable field shall contain the following sub-items: one Abstract
00084     Syntax and one or more Transfer Syntax(es). For a complete
00085     description of the use and encoding of these sub-items see Sections
00086     9.3.2.2.1 and 9.3.2.2.2.
00087     */
00088     AbstractSyntax SubItems;
00089     std::vector<TransferSyntaxSub> TransferSyntaxes;
00090 };
00091
00092 } // end namespace network
00093
00094 } // end namespace gdcm
00095
00096 #endif //GDCMPRESENTATIONCONTEXTREQ_H

```

## 11.551 gdcmPresentationDataValue.h File Reference

```

#include "gdcmTypes.h"
#include <vector>

```



```

00016
00017 #include "gdcmTypes.h"
00018
00019 #include <vector>
00020
00021 namespace gdcm
00022 {
00023   class DataSet;
00024   namespace network
00025   {
00026
00032     class GDCM_EXPORT PresentationDataValue
00033     {
00034     public:
00035       PresentationDataValue();
00036       std::istream &Read(std::istream &is);
00037       std::istream &ReadInto(std::istream &is, std::ostream &os);
00038
00039       const std::ostream &Write(std::ostream &os) const;
00040
00042       size_t Size() const;
00043
00046       void SetDataSet(const DataSet & ds);
00047       void SetBlob(const std::string & partialblob);
00048       const std::string &GetBlob() const;
00049
00050       uint8_t GetPresentationContextID() const { return PresentationContextID; }
00051       void SetPresentationContextID(uint8_t id) {
00052         assert( id );
00053         PresentationContextID = id;
00054       }
00055       uint8_t GetMessageHeader() const {
00056         assert( MessageHeader <= 0x3 );
00057         return MessageHeader;
00058       }
00059       // E.2 MESSAGE CONTROL HEADER ENCODING
00060       // Only the first two bits are considered
00061       void SetMessageHeader(uint8_t messageheader) {
00062         MessageHeader = messageheader;
00063         assert( MessageHeader <= 0x3 );
00064       }
00065       //flip the least significant bit of the message header to 1
00066       //if this is a command, else set it to 0.
00067       void SetCommand(bool inCommand);
00068       void SetLastFragment(bool inLast); //set to true if this is the last PDV of a set
00069
00070       bool GetIsCommand() const;
00071       bool GetIsLastFragment() const;
00072
00073       void Print(std::ostream &os) const;
00074
00075       //NOTE that the PDVs have to be given in the order in which they were received!
00076       //also note that a dataset may be across multiple PDVs
00077       static DataSet ConcatenatePDVBlobs(const std::vector<PresentationDataValue>& inPDVs);
00078
00079       static DataSet ConcatenatePDVBlobsAsExplicit(const std::vector<PresentationDataValue>& inPDVs);
00080
00081     private:
00082       uint32_t ItemLength;
00083       uint8_t PresentationContextID;
00084       uint8_t MessageHeader;
00085       std::string Blob;
00086     };
00087   };
00088 } // end namespace network
00089
00090 } // end namespace gdcm
00091
00092 #endif //GDCMPRESENTATIONDATAVALUE_H

```

## 11.553 gdcmQueryBase.h File Reference

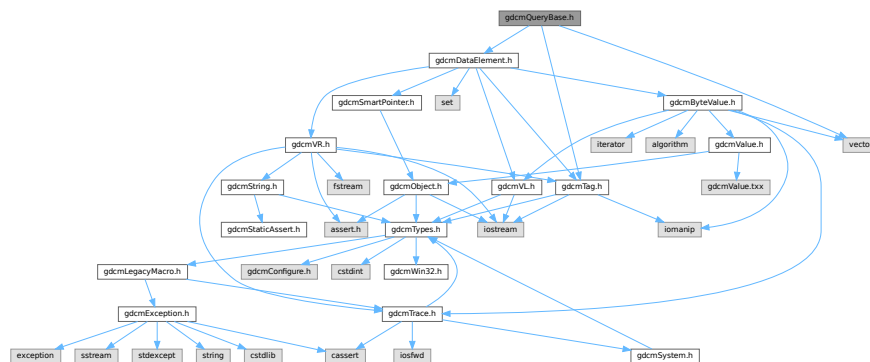
```

#include "gdcmTag.h"
#include "gdcmDataElement.h"

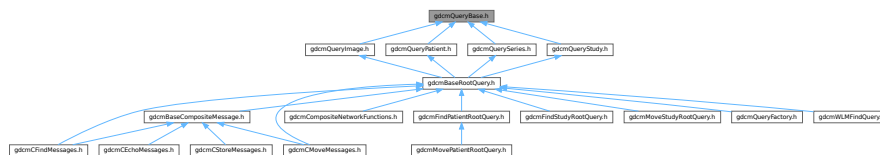
```

```
#include <vector>
```

Include dependency graph for gdcmQueryBase.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::QueryBase](#)  
*QueryBase.*

## Namespaces

- namespace [gdcm](#)

## Enumerations

- enum [gdcm::ERootType](#) {  
  [gdcm::ePatientRootType](#) ,  
  [gdcm::eStudyRootType](#) }

## 11.554 gdcmQueryBase.h

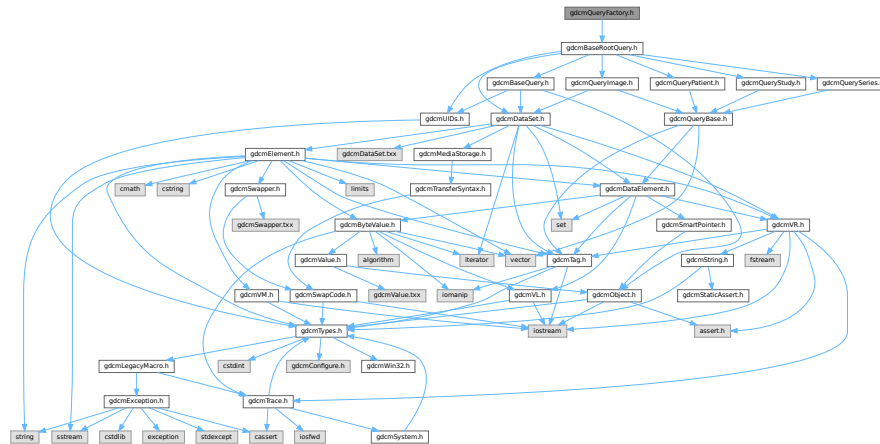
[Go to the documentation of this file.](#)

```

00001 /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  * http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMQUERYBASE_H
00019 #define GDCMQUERYBASE_H
00020
00021 #include "gdcmTag.h"
00022 #include "gdcmDataElement.h"
00023
00024 #include <vector>
00025
00026 namespace gdcm
00027 {
00028     enum ERootType
00029     {
00030         ePatientRootType,
00031         eStudyRootType
00032     };
00033
00060     class GDCM_EXPORT QueryBase
00061     {
00062     public:
00063         virtual ~QueryBase() = default;
00064
00065         virtual std::vector<Tag> GetRequiredTags(const ERootType& inRootType) const = 0;
00066         virtual std::vector<Tag> GetUniqueTags(const ERootType& inRootType) const = 0;
00067         virtual std::vector<Tag> GetOptionalTags(const ERootType& inRootType) const = 0;
00068         // C.4.1.2.1 Baseline Behavior of SCU
00069         // All C-FIND SCUs shall be capable of generating query requests which
00070         // meet the requirements of the Hierarchical Search.
00071         // The Identifier contained in a C-FIND request shall contain a single
00072         // value in the Unique Key Attribute for each level above the
00073         // Query/Retrieve level. No Required or Optional Keys shall be
00074         // specified which are associated with levels above the Query/Retrieve
00075         // level.
00077         virtual std::vector<Tag> GetHierarchicalSearchTags(const ERootType& inRootType) const = 0;
00078
00081         std::vector<Tag> GetAllTags(const ERootType& inRootType) const;
00082
00085         std::vector<Tag> GetAllRequiredTags(const ERootType& inRootType) const;
00086
00087         virtual const char * GetName() const = 0;
00088         virtual DataElement GetQueryLevel() const = 0;
00089     };
00090 }
00091
00092 #endif //GDCMQUERYBASE_H

```

```
#include "gdcmBaseRootQuery.h"
Include dependency graph for gdcmQueryFactory.h:
```



- class `gdcm::QueryFactory`  
*QueryFactory.h*.

- namespace **gdcm**

- enum gdcmm::ECharSet {  
gdcmm::eLatin1 = 0 ,  
gdcmm::eLatin2 ,  
gdcmm::eLatin3 ,  
gdcmm::eLatin4 ,  
gdcmm::eCyrillic ,  
gdcmm::eArabic ,  
gdcmm::eGreek ,  
gdcmm::eHebrew ,  
gdcmm::eLatin5 ,  
gdcmm::eJapanese ,  
gdcmm::eThai ,  
gdcmm::eJapaneseKanjiMultibyte ,  
gdcmm::eJapaneseSupplementaryKanjiMultibyte ,  
gdcmm::eKoreanHangulHanjaMultibyte ,  
gdcmm::eUTF8 ,  
gdcmm::eGB18030 }

## 11.556 gdcmQueryFactory.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  * http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMQUERYFACTORY_H
00019 #define GDCMQUERYFACTORY_H
00020
00021 #include "gdcmBaseRootQuery.h"
00022
00023 namespace gdcm{
00024     enum ECharSet {
00025         eLatin1 = 0,
00026         eLatin2,
00027         eLatin3,
00028         eLatin4,
00029         eCyrillic,
00030         eArabic,
00031         eGreek,
00032         eHebrew,
00033         eLatin5, // Latin Alphabet No. 5 (Turkish) Extended
00034         eJapanese, // JIS X 0201 (Shift JIS) Extended
00035         eThai, // TIS 620-2533 (Thai) Extended
00036         eJapaneseKanjiMultibyte, // JIS X 0208 (Kanji) Extended
00037         eJapaneseSupplementaryKanjiMultibyte, // JIS X 0212 (Kanji) Extended
00038         eKoreanHangulHanjaMultibyte, // KS X 1001 (Hangul and Hanja) Extended
00039         eUTF8,
00040         eGB18030 // Chinese (Simplified) Extended
00041     };
00042
00043     class GDCM_EXPORT QueryFactory
00044     {
00045     public:
00046         static BaseQuery* ProduceQuery( const std::string & sopInstanceUID, ENQueryType inQueryType );
00047         static BaseRootQuery* ProduceQuery(ERootType inRootType, EQueryType inQueryType,
00048             EQueryLevel inQueryLevel);
00049
00050         static DataElement ProduceCharacterSetDataElement(
00051             const std::vector<ECharSet>& inCharSetType);
00052
00053         static ECharSet GetCharacterFromCurrentLocale();
00054
00055         static void ListCharSets(std::ostream& os);
00056     };
00057 } // end namespace gdcm
00058
00059 #endif // GDCMQUERYFACTORY_H

```

## 11.557 gdcmQueryImage.h File Reference

```

#include "gdcmQueryBase.h"
#include "gdcmDataSet.h"

```





```

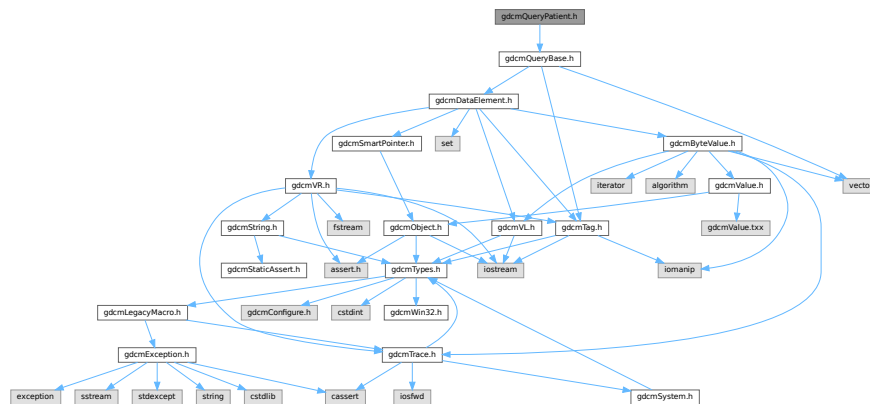
00017  *===== */
00018  #ifndef GDCMQUERYIMAGE_H
00019  #define GDCMQUERYIMAGE_H
00020
00021  #include "gdcmQueryBase.h"
00022  #include "gdcmDataSet.h"
00023
00024  namespace gdcm
00025  {
00030  class GDCM_EXPORT QueryImage : public QueryBase
00031  {
00032  public:
00033      std::vector<Tag> GetRequiredTags(const ERootType& inRootType) const override;
00034      std::vector<Tag> GetUniqueTags(const ERootType& inRootType) const override;
00035      std::vector<Tag> GetOptionalTags(const ERootType& inRootType) const override;
00036      std::vector<Tag> GetHierarchicalSearchTags(const ERootType& inRootType) const override;
00037
00038      const char * GetName() const override;
00039
00040      DataElement GetQueryLevel() const override;
00041  };
00042
00043  } // end namespace gdcm
00044
00045  #endif // GDCMQUERYIMAGE_H

```

## 11.559 gdcmQueryPatient.h File Reference

```
#include "gdcmQueryBase.h"
```

Include dependency graph for gdcmQueryPatient.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [gdcm::QueryPatient](#)  
*QueryPatient.*

## Namespaces

- namespace `gdcm`

## 11.560 gdcmQueryPatient.h

[Go to the documentation of this file.](#)

```

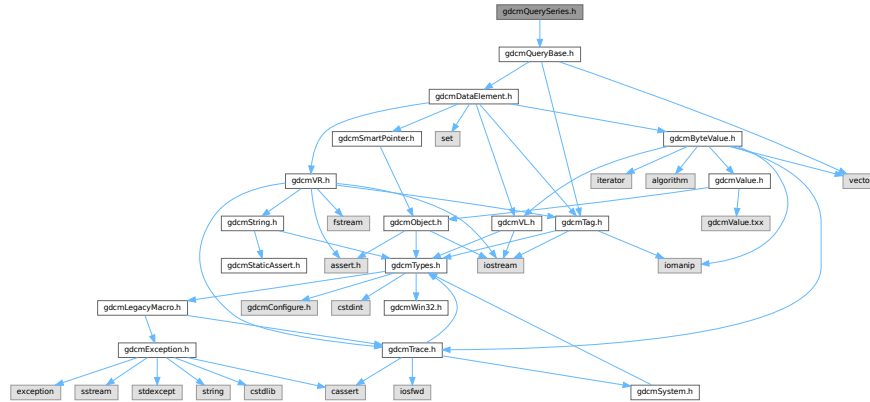
00001 /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  *      http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMQUERYPATIENT_H
00019 #define GDCMQUERYPATIENT_H
00020
00021 #include "gdcmQueryBase.h"
00022
00023 namespace gdcm
00024 {
00025     class GDCM_EXPORT QueryPatient : public QueryBase
00026     {
00027     public:
00028         std::vector<Tag> GetRequiredTags(const ERootType& inRootType) const override;
00029         std::vector<Tag> GetUniqueTags(const ERootType& inRootType) const override;
00030         std::vector<Tag> GetOptionalTags(const ERootType& inRootType) const override;
00031         std::vector<Tag> GetHierarchicalSearchTags(const ERootType& inRootType) const override;
00032
00033         const char * GetName() const override;
00034         DataElement GetQueryLevel() const override;
00035     };
00036
00037 } // end namespace gdcm
00038
00039 #endif //GDCMQUERYPATIENT_H

```

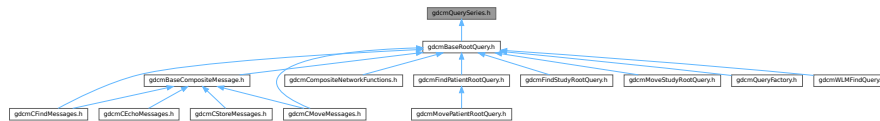
## 11.561 gdcmQuerySeries.h File Reference

```
#include "gdcmQueryBase.h"
```

Include dependency graph for gdcmQuerySeries.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [gdcm::QuerySeries](#)  
*QuerySeries.*

### Namespaces

- namespace [gdcm](#)

## 11.562 gdcmQuerySeries.h

[Go to the documentation of this file.](#)

```
00001 /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
```

```

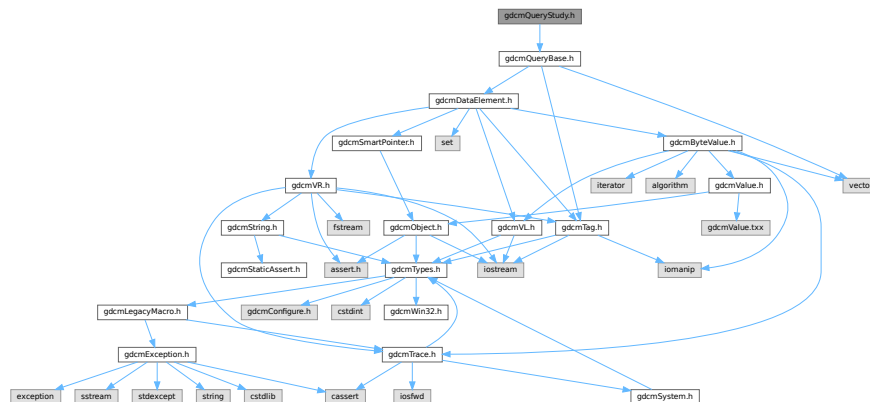
00008  *
00009  *      http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMQUERYSERIES_H
00019 #define GDCMQUERYSERIES_H
00020
00021 #include "gdcMQueryBase.h"
00022
00023 namespace gdcM
00024 {
00025     class GDCM_EXPORT QuerySeries : public QueryBase
00026     {
00027     public:
00028         std::vector<Tag> GetRequiredTags(const ERootType& inRootType) const override;
00029         std::vector<Tag> GetUniqueTags(const ERootType& inRootType) const override;
00030         std::vector<Tag> GetOptionalTags(const ERootType& inRootType) const override;
00031         std::vector<Tag> GetHierarchicalSearchTags(const ERootType& inRootType) const override;
00032
00033         const char * GetName() const override;
00034         DataElement GetQueryLevel() const override;
00035     };
00036 } // end namespace gdcM
00037
00038 #endif //GDCMQUERYSERIES_H

```

## 11.563 gdcMQueryStudy.h File Reference

```
#include "gdcMQueryBase.h"
```

Include dependency graph for gdcMQueryStudy.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::QueryStudy`  
*QueryStudy.h.*

## Namespaces

- namespace `gdcm`

## 11.564 gdcmQueryStudy.h

[Go to the documentation of this file.](#)

```

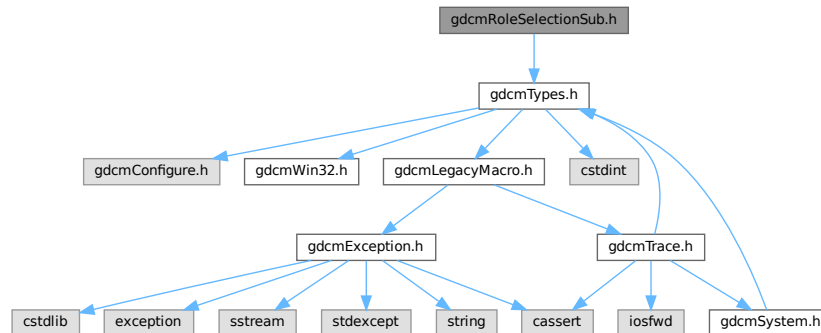
00001 /*=====
00002  *
00003  *   Copyright NumFOCUS
00004  *
00005  *   Licensed under the Apache License, Version 2.0 (the "License");
00006  *   you may not use this file except in compliance with the License.
00007  *   You may obtain a copy of the License at
00008  *
00009  *       http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  *   Unless required by applicable law or agreed to in writing, software
00012  *   distributed under the License is distributed on an "AS IS" BASIS,
00013  *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  *   See the License for the specific language governing permissions and
00015  *   limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMQUERYSTUDY_H
00019 #define GDCMQUERYSTUDY_H
00020
00021 #include "gdcmQueryBase.h"
00022
00023 namespace gdcm
00024 {
00025     class GDCM_EXPORT QueryStudy : public QueryBase
00026     {
00027     public:
00028         std::vector<Tag> GetRequiredTags(const ERootType& inRootType) const override;
00029         std::vector<Tag> GetUniqueTags(const ERootType& inRootType) const override;
00030         std::vector<Tag> GetOptionalTags(const ERootType& inRootType) const override;
00031         std::vector<Tag> GetHierarchicalSearchTags(const ERootType& inRootType) const override;
00032
00033         const char *GetName() const override;
00034         DataElement GetQueryLevel() const override;
00035     };
00036 } // end namespace gdcm
00037
00038 #endif //GDCMQUERYSTUDY_H

```

## 11.565 gdcmRoleSelectionSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmRoleSelectionSub.h:



### Classes

- class [gdcm::network::RoleSelectionSub](#)  
*RoleSelectionSub.*

### Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.566 gdcmRoleSelectionSub.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMROLESELECTIONSUB_H
00015  #define GDCMROLESELECTIONSUB_H
00016
00017  #include "gdcmTypes.h"
00018
00019  namespace gdcm
00020  {
00021

```

```

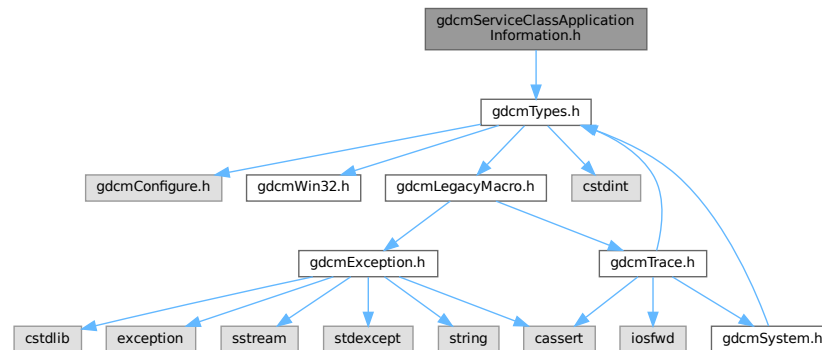
00022 namespace network
00023 {
00024
00031 class RoleSelectionSub
00032 {
00033 public:
00034     RoleSelectionSub();
00035     std::istream &Read(std::istream &is);
00036     const std::ostream &Write(std::ostream &os) const;
00037
00038     size_t Size() const;
00039     void Print(std::ostream &os) const;
00040
00041     void SetTuple(const char *uid, uint8_t scurole, uint8_t scprole);
00042
00043 private:
00044     static const uint8_t ItemType;
00045     static const uint8_t Reserved2;
00046     uint16_t ItemLength;
00047     uint16_t UIDLength;
00048     std::string /*SOP-class-uid*/ Name; // UID
00049     uint8_t SCURole;
00050     uint8_t SCPRole;
00051 };
00052
00053 } // end namespace network
00054
00055 } // end namespace gdcm
00056
00057 #endif // GDCMROLESELECTIONSUB_H

```

## 11.567 gdcmServiceClassApplicationInformation.h File Reference

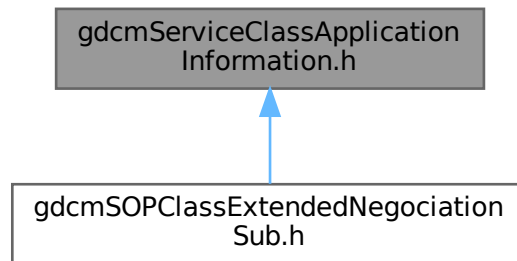
```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmServiceClassApplicationInformation.h:





This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::network::ServiceClassApplicationInformation](#)

## Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.568 gdcmServiceClassApplicationInformation.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMSERVICECLASSAPPLICATIONINFORMATION_H
00015 #define GDCMSERVICECLASSAPPLICATIONINFORMATION_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021
00022   namespace network
00023   {
00024
00030     class ServiceClassApplicationInformation
00031     {
00032     public:
00033       ServiceClassApplicationInformation();
  
```

## 11.569 gdcmServiceClassUser.h File Reference

The diagram illustrates the complex dependency structure of the glibmm library. It features a large number of header files, each represented by a box. The files are interconnected by blue arrows, indicating the direction of dependencies. The graph is organized into several layers, with some files at the top (like glibmmServiceClass.h) and others at the bottom (like glibmmSystem.h). The dependencies are highly interconnected, showing a dense web of relationships between the various components of the library.

- class `gdcm::ServiceClassUser`  
*ServiceClassUser.*

- namespace `gdcm`
- namespace `gdcm::network`

## 11.570 gdcmServiceClassUser.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMSERVICECLASSUSER_H
00015 #define GDCMSERVICECLASSUSER_H
00016
00017 #include "gdcmSubject.h"
00018
00019 #include "gdcmPresentationContext.h"
00020 #include "gdcmFile.h"
00021
00022 #include "gdcmNetworkStateID.h" // EStateID
00023
00024 namespace gdcm
00025 {
00026   class ServiceClassUserInternals;
00027   class BaseRootQuery;
00028   namespace network{
00029     class ULEvent;
00030     class ULConnection;
00031     class ULConnectionCallback;
00032   }
00033   class GDCM_EXPORT ServiceClassUser : public Subject
00034   {
00035   public:
00042     ServiceClassUser();
00043     ~ServiceClassUser() override;
00044     ServiceClassUser(const ServiceClassUser&) = delete;
00045     void operator=(const ServiceClassUser &) = delete;
00046
00047     void SetHostname( const char *hostname );
00048
00049     void SetPort( uint16_t port );
00050
00051     void SetPortSCP( uint16_t portscp );
00052
00053     void SetAETitle(const char *aetitle);
00054     const char *GetAETitle() const;
00055
00056     void SetCalledAETitle(const char *aetitle);
00057     const char *GetCalledAETitle() const;
00058
00059     void SetTimeout(double t);
00060     double GetTimeout() const;
00061
00062     bool InitializeConnection();
00063
00064     void SetPresentationContexts(std::vector<PresentationContext> const & pcs);
00065
00066     bool IsPresentationContextAccepted(const PresentationContext& pc) const;
00067
00068     bool StartAssociation();
00069
00070     bool StopAssociation();
00071
00072     bool SendEcho();
00073
00074     bool SendStore(const char *filename);
00075     bool SendStore(File const &file);
00076     bool SendStore(DataSet const &ds);
00077
00078     bool SendFind(const BaseRootQuery* query, std::vector<DataSet> &retDatasets);
00079
00080     bool SendMove(const BaseRootQuery* query, const char *outputdir);
00081     bool SendMove(const BaseRootQuery* query, std::vector<DataSet> &retDatasets);
00082     bool SendMove(const BaseRootQuery* query, std::vector<File> &retFile);

```

```

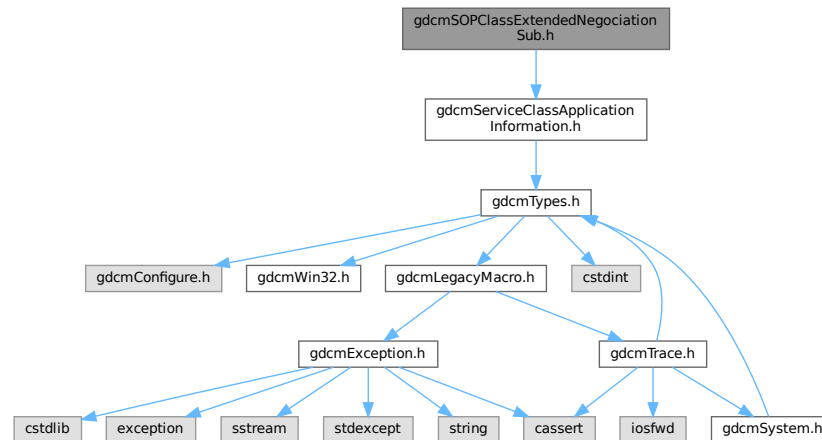
00105
00107     static SmartPointer<ServiceClassUser> New() { return new ServiceClassUser; }
00108
00109 private:
00110     network::EStateID RunEventLoop(network::ULEvent& inEvent,
00111         network::ULConnection* inWhichConnection,
00112         network::ULConnectionCallback* inCallback, const bool& startWaiting);
00113     network::EStateID RunMoveEventLoop(network::ULEvent& inEvent,
00114         network::ULConnectionCallback* inCallback);
00115
00116 private:
00117     ServiceClassUserInternals *Internals;
00118 };
00119
00120 } // end namespace gdcms
00121
00122 #endif // GDCMSERVICECLASSUSER_H

```

## 11.571 gdcmsOPClassExtendedNegociationSub.h File Reference

#include "gdcmsServiceClassApplicationInformation.h"

Include dependency graph for gdcmsOPClassExtendedNegociationSub.h:



### Classes

- class `gdcms::network::SOPClassExtendedNegociationSub`  
*SOPClassExtendedNegociationSub.*

### Namespaces

- namespace `gdcms`
- namespace `gdcms::network`

## 11.572 gdcmSOPClassExtendedNegociationSub.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMSOPCLASSEXTENDEDNEGOCIATIONSUB_H
00015 #define GDCMSOPCLASSEXTENDEDNEGOCIATIONSUB_H
00016
00017 #include "gdcmServiceClassApplicationInformation.h"
00018
00019 namespace gdcm
00020 {
00021 namespace network
00022 {
00023
00031 class SOPClassExtendedNegociationSub
00032 {
00033 public:
00034   SOPClassExtendedNegociationSub();
00035   std::istream &Read(std::istream &is);
00036   const std::ostream &Write(std::ostream &os) const;
00037
00038   size_t Size() const;
00039   void Print(std::ostream &os) const;
00040
00041   void SetTuple(const char *uid, uint8_t levelofsupport = 3,
00042                uint8_t levelofdigitalsig = 0,
00043                uint8_t elementcoercion = 2);
00044
00045 private:
00046   static const uint8_t ItemType;
00047   static const uint8_t Reserved2;
00048   uint16_t ItemLength;
00049   uint16_t UIDLength;
00050   std::string /*SOP-class-uid*/ Name; // UID
00051   ServiceClassApplicationInformation SCAI;
00052 };
00053
00054 } // end namespace network
00055
00056 } // end namespace gdcm
00057
00058 #endif // GDCMSOPCLASSEXTENDEDNEGOCIATIONSUB_H

```

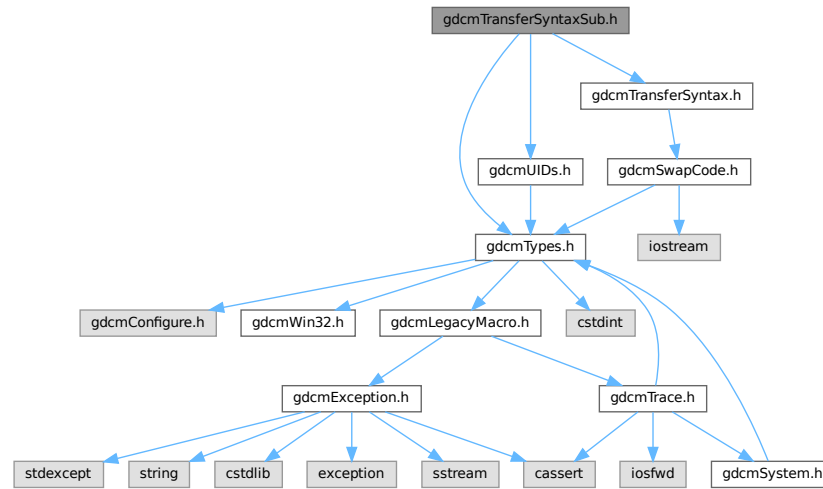
## 11.573 gdcmTransferSyntaxSub.h File Reference

```

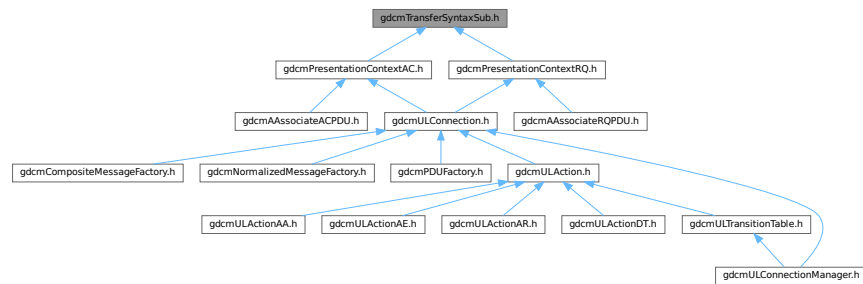
#include "gdcmTypes.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDs.h"

```

Include dependency graph for `gdcmTransferSyntaxSub.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::network::TransferSyntaxSub`  
*TransferSyntaxSub.*

## Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

## 11.574 gdcmTransferSyntaxSub.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 #ifndef GDCMTRANSFERSYNTAXSUB_H
00015 #define GDCMTRANSFERSYNTAXSUB_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmTransferSyntax.h"
00019 #include "gdcmUIDs.h"
00020
00021 namespace gdcm
00022 {
00023
00024   namespace network
00025   {
00026
00027     class TransferSyntaxSub
00028     {
00029     public:
00030       TransferSyntaxSub();
00031       void SetName( const char *name );
00032       const char *GetName() const { return Name.c_str(); }
00033
00034       // accept a UID::TSType also...
00035       void SetNameFromUID( UID::TSType tsname );
00036
00037       std::istream &Read(std::istream &is);
00038       const std::ostream &Write(std::ostream &os) const;
00039       size_t Size() const;
00040       void Print(std::ostream &os) const;
00041
00042       bool operator==(const TransferSyntaxSub & ts) const
00043       {
00044         return Name == ts.Name;
00045       }
00046
00047     private:
00048       void UpdateName( const char *name );
00049       static const uint8_t ItemType;
00050       static const uint8_t Reserved2;
00051       uint16_t ItemLength; // len of
00052       std::string /*TransferSyntaxSub*/ Name; // UID
00053     };
00054
00055   } // end namespace network
00056
00057 } // end namespace gdcm
00058
00059 #endif //GDCMTRANSFERSYNTAXSUB_H

```

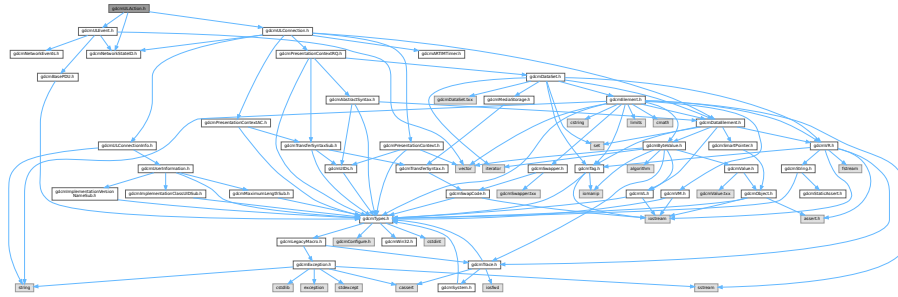
## 11.575 gdcmULAction.h File Reference

```

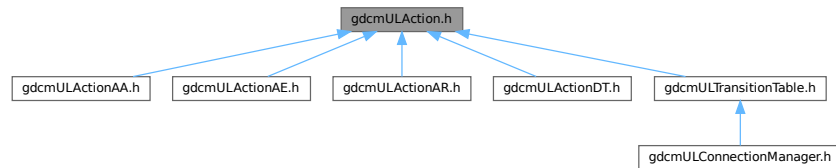
#include "gdcmNetworkStateID.h"
#include "gdcmULEvent.h"

```

```
#include "gdcmULConnection.h"
Include dependency graph for gdcmULAction.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::network::ULAction](#)  
*ULAction.*

## Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.576 gdcmULAction.h

[Go to the documentation of this file.](#)

```
00001 /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  * http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
```



```

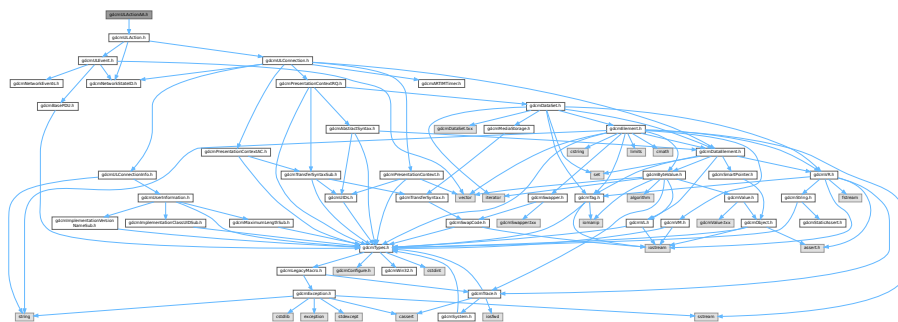
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMULACTION_H
00019 #define GDCMULACTION_H
00020
00021 #include "gdcmNetworkStateID.h"
00022 #include "gdcmULEvent.h"
00023 #include "gdcmULConnection.h"
00024
00025 namespace gdcm {
00026 class Subject;
00027     namespace network {
00028
00062 class ULAction {
00063
00064     protected:
00065
00066
00067     public:
00068         ULAction() = default;
00069         //make sure destructors are virtual to avoid memory leaks
00070         virtual ~ULAction() = default;
00071         //cannot copy a ULAction
00072         ULAction(const ULAction& inAction) = delete;
00073         void operator=(const ULAction&) = delete;
00074
00075         virtual EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00076             bool& outWaitingForEvent, EEventID& outRaisedEvent) = 0;
00077     };
00078 }
00079 }
00080
00081 #endif // GDCMULACTION_H

```

## 11.577 gdcmULActionAA.h File Reference

```
#include "gdcmULAction.h"
```

Include dependency graph for gdcmULActionAA.h:



### Classes

- class [gdcm::network::ULActionAA1](#)
- class [gdcm::network::ULActionAA2](#)
- class [gdcm::network::ULActionAA3](#)
- class [gdcm::network::ULActionAA4](#)

- class [gdcm::network::ULActionAA5](#)
- class [gdcm::network::ULActionAA6](#)
- class [gdcm::network::ULActionAA7](#)
- class [gdcm::network::ULActionAA8](#)

## Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.578 gdcmULActionAA.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002  *
00003  *   Copyright NumFOCUS
00004  *
00005  *   Licensed under the Apache License, Version 2.0 (the "License");
00006  *   you may not use this file except in compliance with the License.
00007  *   You may obtain a copy of the License at
00008  *
00009  *       http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  *   Unless required by applicable law or agreed to in writing, software
00012  *   distributed under the License is distributed on an "AS IS" BASIS,
00013  *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  *   See the License for the specific language governing permissions and
00015  *   limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMULACTIONAA_H
00019 #define GDCMULACTIONAA_H
00020
00021 #include "gdcmULAction.h"
00022
00031 namespace gdcm {
00032     namespace network {
00033
00034         //Send A-ABORT PDU (service-user source) and start (or restart if already started) ARTIM timer
00035         //Next State: eStal3AwaitingClose
00036         class ULActionAA1 : public ULAction {
00037         public:
00038             EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00039                 bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00040         };
00041
00042         //Stop ARTIM timer if running. Close transport connection.
00043         //Next State: eStalIdle
00044         class ULActionAA2 : public ULAction {
00045         public:
00046             EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00047                 bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00048         };
00049
00050         //If (service-user initiated abort)
00051         //- issue A-ABORT indication and close transport connection
00052         //otherwise (service-provider initiated abort):
00053         //- issue A-P-ABORT indication and close transport connection
00054         //Next State: eStalIdle
00055         class ULActionAA3 : public ULAction {
00056         public:
00057             EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00058                 bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00059         };
00060
00061         //Issue A-P-ABORT indication primitive
00062         //Next State: eStalIdle
00063         class ULActionAA4 : public ULAction {
00064         public:

```

```

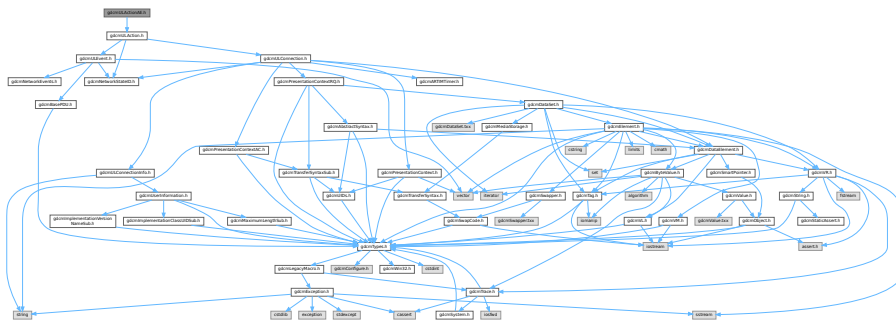
00065     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00066         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00067 };
00068
00069 //Stop ARTIM timer
00070 //Next State: eStalIdle
00071 class ULAActionAA5 : public ULAAction {
00072 public:
00073     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00074         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00075 };
00076
00077 //Ignore PDU
00078 //Next State: eStal3AwaitingClose
00079 class ULAActionAA6 : public ULAAction {
00080 public:
00081     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00082         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00083 };
00084
00085 //Send A-ABORT PDU
00086 //Next State: eStal3AwaitingClose
00087 class ULAActionAA7 : public ULAAction {
00088 public:
00089     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00090         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00091 };
00092
00093 //Send A-ABORT PDU (service-provider source), issue an A-P-ABORT indication, and start ARTIM timer
00094 //Next State: eStal3AwaitingClose
00095 class ULAActionAA8 : public ULAAction {
00096 public:
00097     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00098         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00099 };
00100 }
00101 }
00102
00103 #endif // GDCMULACTIONAA_H

```

## 11.579 gdcmlActionAE.h File Reference

#include "gdcmlAction.h"

Include dependency graph for gdcmlActionAE.h:



### Classes

- class [gdcml::network::ULAActionAE1](#)
- class [gdcml::network::ULAActionAE2](#)
- class [gdcml::network::ULAActionAE3](#)

- class [gdcm::network::ULActionAE4](#)
- class [gdcm::network::ULActionAE5](#)
- class [gdcm::network::ULActionAE6](#)
- class [gdcm::network::ULActionAE7](#)
- class [gdcm::network::ULActionAE8](#)

## Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.580 gdcmULActionAE.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  *      http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMULACTIONAE_H
00019 #define GDCMULACTIONAE_H
00020
00021 #include "gdcmULAction.h"
00022
00023 namespace gdcm {
00024     namespace network {
00025
00026         //Issue TRANSPORT CONNECT request primitive to local transport service.
00027         class ULActionAE1 : public ULAction {
00028         public:
00029             EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00030                 bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00031         };
00032
00033         //Send A-ASSOCIATE-RQ-PDU
00034         //Next State: eSta5WaitRemoteAssoc
00035         class ULActionAE2 : public ULAction {
00036         public:
00037             EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00038                 bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00039         };
00040
00041         //Issue A-ASSOCIATE confirmation (accept) primitive
00042         //Next State: eSta6TransferReady
00043         class ULActionAE3 : public ULAction {
00044         public:
00045             EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00046                 bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00047         };
00048
00049         //Issue A-ASSOCIATE confirmation (reject) primitive and close transport connection
00050         //Next State: eSta1Idle
00051         class ULActionAE4 : public ULAction {
00052         public:
00053             EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00054                 bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00055         };
00056
00057
00058
00059
00060
00061
00062
00063

```

```

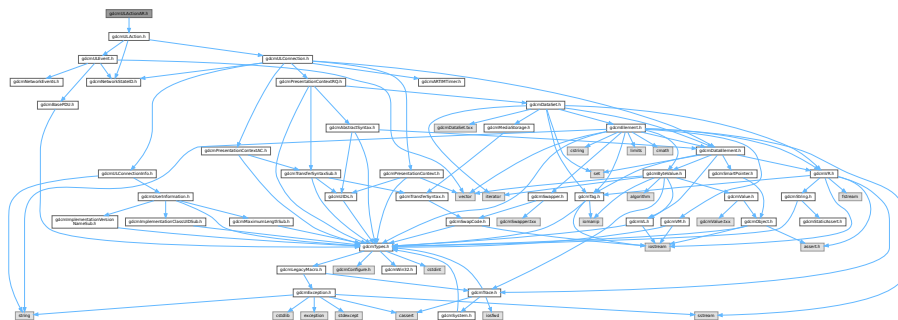
00064     };
00065
00066     //Issue Transport connection response primitive, start ARTIM timer
00067     //Next State: eSta2Open
00068     class ULActionAE5 : public ULAction {
00069     public:
00070         EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00071             bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00072     };
00073
00074     //Stop ARTIM timer and if A-ASSOCIATE-RQ acceptable by service-provider:
00075     //- issue A-ASSOCIATE indication primitive
00076     //Next state: eSta3WaitLocalAssoc
00077     //otherwise:
00078     //- issue A-ASSOCIATE-RJ-PDU and start ARTIM timer
00079     //Next state: eSta13AwaitingClose
00080     class ULActionAE6 : public ULAction {
00081     public:
00082         EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00083             bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00084     };
00085
00086     //Send A-ASSOCIATE-AC PDU
00087     //Next State: eSta6TransferReady
00088     class ULActionAE7 : public ULAction {
00089     public:
00090         EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00091             bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00092     };
00093
00094     //Send A-ASSOCIATE-RJ PDU and start ARTIM timer
00095     //Next State: eSta13AwaitingClose
00096     class ULActionAE8 : public ULAction {
00097     public:
00098         EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00099             bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00100     };
00101 }
00102 }
00103 #endif // GDCMULACTIONAE_H

```

## 11.581 gdcMULActionAR.h File Reference

#include "gdcMULAction.h"

Include dependency graph for gdcMULActionAR.h:



### Classes

- class [gdcM::network::ULActionAR1](#)
- class [gdcM::network::ULActionAR10](#)

- class [gdcm::network::ULActionAR2](#)
- class [gdcm::network::ULActionAR3](#)
- class [gdcm::network::ULActionAR4](#)
- class [gdcm::network::ULActionAR5](#)
- class [gdcm::network::ULActionAR6](#)
- class [gdcm::network::ULActionAR7](#)
- class [gdcm::network::ULActionAR8](#)
- class [gdcm::network::ULActionAR9](#)

## Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.582 gdcmULActionAR.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002  *
00003  *   Copyright NumFOCUS
00004  *
00005  *   Licensed under the Apache License, Version 2.0 (the "License");
00006  *   you may not use this file except in compliance with the License.
00007  *   You may obtain a copy of the License at
00008  *
00009  *       http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  *   Unless required by applicable law or agreed to in writing, software
00012  *   distributed under the License is distributed on an "AS IS" BASIS,
00013  *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  *   See the License for the specific language governing permissions and
00015  *   limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMULACTIONAR_H
00019 #define GDCMULACTIONAR_H
00020
00021 #include "gdcmULAction.h"
00022
00023 namespace gdcm {
00024     namespace network {
00025
00026         //Send A-RELEASE-RQ-PDU
00027         //Next State: eSta7WaitRelease
00028         class ULActionAR1 : public ULAction {
00029         public:
00030             EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00031                                     bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00032         };
00033
00034         //Issue A-RELEASE indication primitive
00035         //Next State: eSta8WaitLocalRelease
00036         class ULActionAR2 : public ULAction {
00037         public:
00038             EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00039                                     bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00040         };
00041
00042         //Issue A-RELEASE confirmation primitive, and close transport connection
00043         //Next State: eStaIdle
00044         class ULActionAR3 : public ULAction {
00045         public:
00046             EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00047                                     bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00048         };
00049
00050     }
00051 }

```

```

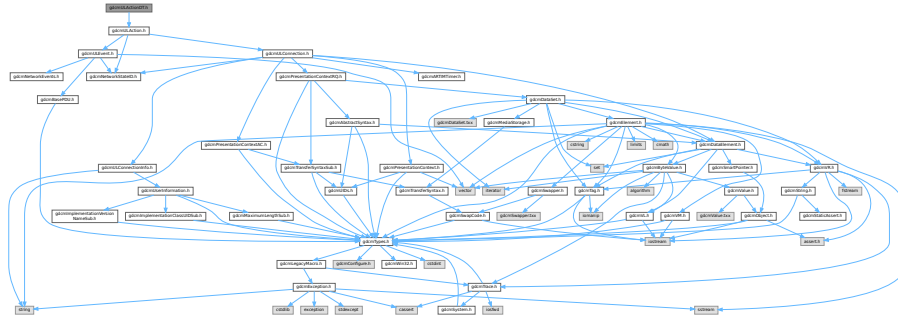
00058 //Issue A-RELEASE-RP PDU and start ARTIM timer
00059 //Next State: eSta13AwaitingClose
00060 class ULAActionAR4 : public ULAAction {
00061 public:
00062     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00063         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00064 };
00065
00066 //Stop ARTIM timer
00067 //Next State: eSta1Idle
00068 class ULAActionAR5 : public ULAAction {
00069 public:
00070     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00071         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00072 };
00073
00074 //Issue P-Data indication
00075 //Next State: eSta7WaitRelease
00076 class ULAActionAR6 : public ULAAction {
00077 public:
00078     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00079         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00080 };
00081
00082 //Issue P-DATA-TF PDU
00083 //Next State: eSta8WaitLocalRelease
00084 class ULAActionAR7 : public ULAAction {
00085 public:
00086     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00087         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00088 };
00089
00090 //Issue A-RELEASE indication (release collision):
00091 //- If association-requestor, next state is eSta9ReleaseCollisionRqLocal
00092 //- if not, next state is eSta10ReleaseCollisionAc
00093 class ULAActionAR8 : public ULAAction {
00094 public:
00095     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00096         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00097 };
00098
00099 //Send A-RELEASE-RP PDU
00100 //Next State: eSta11ReleaseCollisionRq
00101 class ULAActionAR9 : public ULAAction {
00102 public:
00103     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00104         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00105 };
00106
00107 //Issue A-RELEASE confirmation primitive
00108 //Next State: eSta12ReleaseCollisionAcLocal
00109 class ULAActionAR10 : public ULAAction {
00110 public:
00111     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00112         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00113 };
00114 }
00115 }
00116 #endif // GDCMULACTIONAR_H

```

## 11.583 gdcmULActionDT.h File Reference

```
#include "gdcmULAction.h"
```

Include dependency graph for gdcmULActionDT.h:



### Classes

- class [gdcm::network::ULActionDT1](#)
- class [gdcm::network::ULActionDT2](#)

### Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.584 gdcmULActionDT.h

[Go to the documentation of this file.](#)

```
00001 /*=====
00002  *
00003  *   Copyright NumFOCUS
00004  *
00005  *   Licensed under the Apache License, Version 2.0 (the "License");
00006  *   you may not use this file except in compliance with the License.
00007  *   You may obtain a copy of the License at
00008  *
00009  *       http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  *   Unless required by applicable law or agreed to in writing, software
00012  *   distributed under the License is distributed on an "AS IS" BASIS,
00013  *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  *   See the License for the specific language governing permissions and
00015  *   limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMULACTIONDT_H
00019 #define GDCMULACTIONDT_H
00020
00021 #include "gdcmULAction.h"
00022
00023 namespace gdcm {
00024     namespace network {
00025
00026         //Send P-DATA-TF PDU
```



## 11.585 gdcmlULBasicCallback.h File Reference

- class `gdcm::network::ULBasicCallback`  
*ULBasicCallback.*

- namespace `gdcm`
- namespace `gdcm::network`

## 11.586 gdcmULBasicCallback.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  * http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMULCONNECTIONBASICCALLBACK_H
00019 #define GDCMULCONNECTIONBASICCALLBACK_H
00020
00021 #include "gdcmULConnectionCallback.h"
00022 #include "gdcmDataSet.h"
00023 #include <vector>
00024
00025 namespace gdcm
00026 {
00027     namespace network
00028     {
00029         class GDCM_EXPORT ULBasicCallback : public ULConnectionCallback
00030         {
00031         {
00032             std::vector<DataSet> mDataSets;
00033             std::vector<DataSet> mResponses;
00034         public:
00035             ULBasicCallback() = default;
00036             ~ULBasicCallback() override = default; //empty, for later inheritance
00037
00038             void HandleDataSet(const DataSet& inDataSet) override;
00039             void HandleResponse(const DataSet& inDataSet) override;
00040
00041             std::vector<DataSet> const & GetDataSets() const;
00042             std::vector<DataSet> const & GetResponses() const;
00043         };
00044     } // end namespace network
00045 } // end namespace gdcm
00046
00047 #endif // GDCMULCONNECTIONBASICCALLBACK_H

```

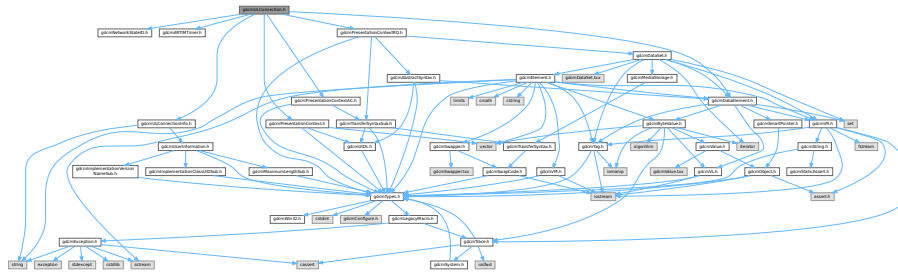
## 11.587 gdcmULConnection.h File Reference

```

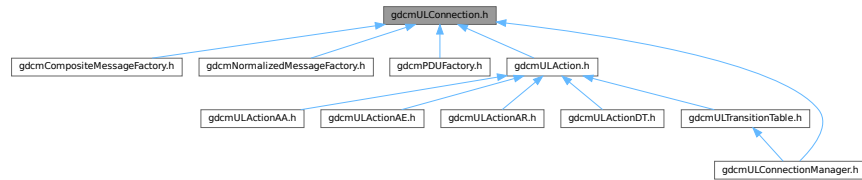
#include "gdcmNetworkStateID.h"
#include "gdcmARTIMTimer.h"
#include "gdcmULConnectionInfo.h"
#include "gdcmPresentationContextRQ.h"
#include "gdcmDataElement.h"
#include "gdcmPresentationContextAC.h"
#include "gdcmPresentationContext.h"

```

Include dependency graph for gdcmULConnection.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::network::ULConnection`  
*ULConnection.*

## Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

## 11.588 gdcmULConnection.h

[Go to the documentation of this file.](#)

```
00001 /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  * http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
```

```

00016  *
00017  *=====*/
00018  #ifndef GDCMULCONNECTION_H
00019  #define GDCMULCONNECTION_H
00020
00021  #include "gdcmNetworkStateID.h"
00022  #include "gdcmARTIMTimer.h"
00023  #include "gdcmULConnectionInfo.h"
00024  #include "gdcmPresentationContextRQ.h"
00025  #include "gdcmDataElement.h"
00026  #include "gdcmPresentationContextAC.h"
00027  #include "gdcmPresentationContext.h"
00028
00029  class iosocket;
00030  class echo;
00031  namespace gdcm{
00032      namespace network{
00033
00057  class GDCM_EXPORT ULConnection
00058  {
00059      ULConnectionInfo mInfo;
00060      //this is a dirty dirty hack
00061      //but to establish an outgoing connection (scu), we need the echo service
00062      //to establish incoming, we just need a port and localhost, so an iosocket works while an
00063      //echo would fail (probably because one already exists)
00064      echo* mEcho;
00065      iosocket* mSocket; //of the three protocols offered by socket++-- echo, smtp, and ftp--
00066      //echo most closely matches what the DICOM standard describes as a network connection
00067      ARTIMTimer mTimer;
00068
00069      EStateID mCurrentState;
00070
00071      std::vector<PresentationContextRQ> mPresentationContexts;
00072      //this is our list of presentation contexts of what we can send
00073      uint32_t mMaxPDUSize;
00074
00075      std::vector<PresentationContextAC> mAcceptedPresentationContexts; //these come back from the server
00076      //and tell us what can be sent over this connection
00077
00078      TransferSyntaxSub cstorets;
00079
00080      friend class ULActionAE6;
00081      void SetCStoreTransferSyntax( TransferSyntaxSub const & ts );
00082      friend class ULConnectionManager;
00083      TransferSyntaxSub const & GetCStoreTransferSyntax() const;
00084  public:
00085
00086      ULConnection(const ULConnectionInfo& inUserInformation);
00087      //destructors are virtual to prevent memory leaks by inherited classes
00088      virtual ~ULConnection();
00089
00090      EStateID GetState() const;
00091      void SetState(const EStateID& inState); //must be able to update state...
00092
00093      //echo* GetProtocol();
00094      std::iostream* GetProtocol();
00095      void StopProtocol();
00096
00097      ARTIMTimer& GetTimer();
00098
00099      const ULConnectionInfo &GetConnectionInfo() const;
00100
00101      //when the connection is first associated, the connection is told
00102      //the max packet/PDU size and the way in which to present data
00103      //(presentation contexts, etc). Store that here.
00104      void SetMaxPDUSize(uint32_t inSize);
00105      uint32_t GetMaxPDUSize() const;
00106
00107      const PresentationContextAC *GetPresentationContextACByID(uint8_t id) const;
00108      const PresentationContextRQ *GetPresentationContextRQByID(uint8_t id) const;
00109
00110      uint8_t GetPresentationContextIDFromPresentationContext(PresentationContextRQ const & pc) const;
00111
00112      std::vector<PresentationContextRQ> const & GetPresentationContexts() const;
00113      void SetPresentationContexts(const std::vector<PresentationContextRQ>& inContexts);
00114
00115      void SetPresentationContexts(const std::vector<PresentationContext>& inContexts);
00116
00117      //given a particular data element, presumably the SOP class,
00118      //find the presentation context for that SOP
00119      //NOT YET IMPLEMENTED
00120

```

```

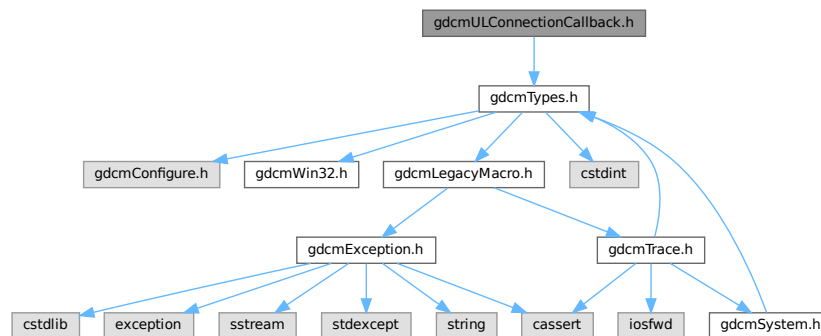
00121     PresentationContextRQ FindContext(const DataElement& de) const;
00122
00123     std::vector<PresentationContextAC> const & GetAcceptedPresentationContexts() const;
00124     std::vector<PresentationContextAC> & GetAcceptedPresentationContexts();
00125     void AddAcceptedPresentationContext(const PresentationContextAC& inPC);
00126
00128     bool InitializeConnection();
00129
00131     bool InitializeIncomingConnection();
00132
00133     ULConnection(const ULConnection&) = delete;
00134     void operator=(const ULConnection&) = delete;
00135 };
00136 }
00137 }
00138
00139 #endif // ULCONNECTION_H

```

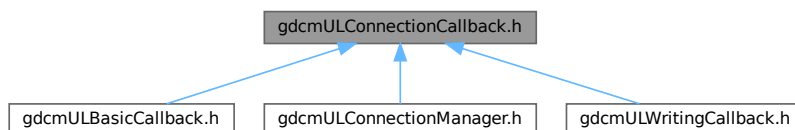
## 11.589 gdcmULConnectionCallback.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmULConnectionCallback.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::network::ULConnectionCallback`

## Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.590 gdcmULConnectionCallback.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  * http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMULCONNECTIONCALLBACK_H
00019 #define GDCMULCONNECTIONCALLBACK_H
00020
00021 #include "gdcmTypes.h" //to be able to export the class
00022
00023 namespace gdcm
00024 {
00025     class DataSet;
00026     namespace network
00027     {
00028         class GDCM_EXPORT ULConnectionCallback {
00029             bool mHandledDataSet;
00030         protected:
00031             bool mImplicit;
00032             //inherited callbacks MUST call this function for the cmove loop to work properly
00033             void DataSetHandled() { mHandledDataSet = true; }
00034         public:
00035             ULConnectionCallback():mHandledDataSet(false),mImplicit(true){}
00036             virtual ~ULConnectionCallback() = default; //placeholder for inherited objects
00037             virtual void HandleDataSet(const DataSet& inDataSet) = 0;
00038             virtual void HandleResponse(const DataSet& inDataSet) = 0;
00039
00040             bool DataSetHandles() const { return mHandledDataSet; }
00041             void ResetHandledDataSet() { mHandledDataSet = false; }
00042
00043             void SetImplicitFlag( const bool imp ) { mImplicit = imp; }
00044         };
00045     }
00046 }
00047 #endif //GDCMULCONNECTIONCALLBACK_H

```

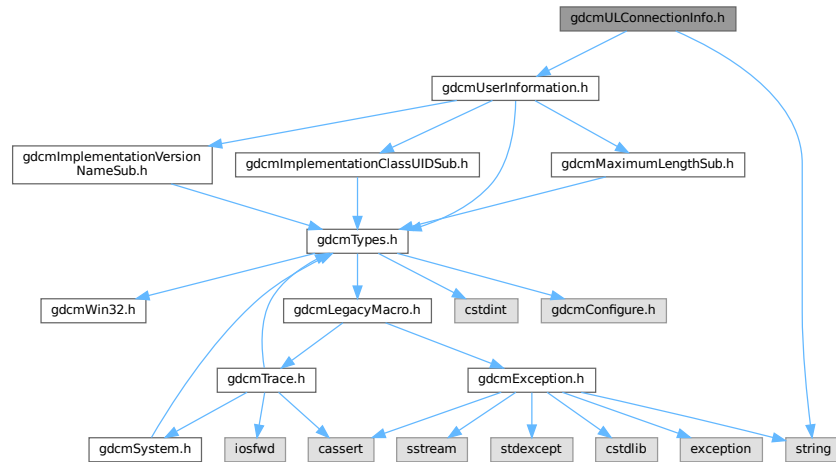
## 11.591 gdcmULConnectionInfo.h File Reference

```

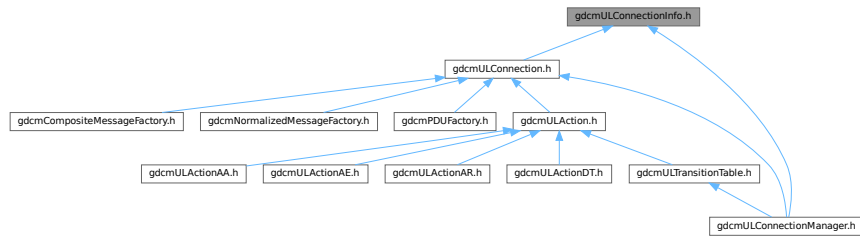
#include "gdcmUserInformation.h"
#include <string>

```

Include dependency graph for gdcmULConnectionInfo.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::network::ULConnectionInfo](#)  
*ULConnectionInfo.*

## Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.592 gdcmULConnectionInfo.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  * http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMULCONNECTIONINFO_H
00019 #define GDCMULCONNECTIONINFO_H
00020
00021 #include "gdcmUserInformation.h"
00022 #include <string>
00023
00024 namespace gdcm{
00025     namespace network {
00026         class ULConnectionInfo {
00027             UserInformation mUserInformation;
00028
00029             std::string mCalledAETitle;
00030             std::string mCallingAETitle;
00031
00032             unsigned long mCalledIPAddress;
00033             int mCalledIPPort;
00034             std::string mCalledComputerName; //either the IP or the name has to be filled in
00035
00036             unsigned long mMaxPDULength;
00037         public:
00038             ULConnectionInfo();
00039
00040             //it is possible to misinitialize this object, so
00041             //have it return false if something breaks (ie, given AEs are bigger than 16 characters,
00042             //no name or IP address).
00043             bool Initialize(UserInformation const &inUserInformation,
00044                 const char *inCalledAETitle, const char *inCallingAETitle,
00045                 unsigned long inCalledIPAddress, int inCalledIPPort,
00046                 std::string inCalledComputerName);
00047
00048             //UserInformation GetUserInformation() const;
00049             const char* GetCalledAETitle() const;
00050             const char* GetCallingAETitle() const;
00051
00052             unsigned long GetCalledIPAddress() const;
00053             int GetCalledIPPort() const;
00054             std::string GetCalledComputerName() const;
00055
00056             //CStore needs to know the max pdu length, so the value gets initialized
00057             //when a cstore connection is established (but not for the others).
00058             void SetMaxPDULength(unsigned long inMaxPDULength);
00059             unsigned long GetMaxPDULength() const;
00060         };
00061     }
00062 }
00063
00064 #endif //GDCMULCONNECTIONINFO_H

```

## 11.593 gdcmULConnectionManager.h File Reference

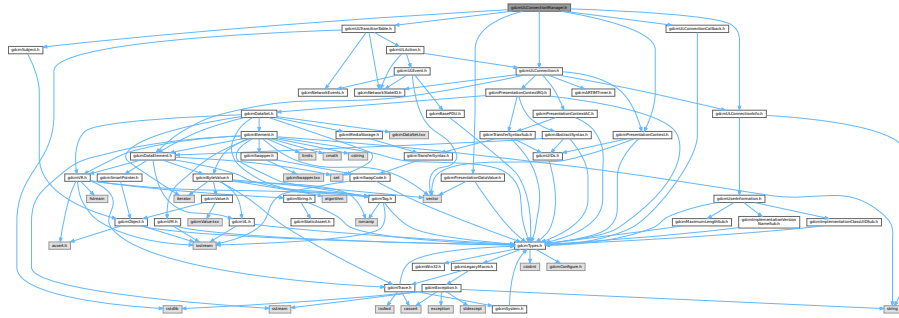
```

#include "gdcmULTransitionTable.h"
#include "gdcmULConnection.h"

```



```
#include "gdcmULConnectionInfo.h"
#include "gdcmPresentationDataValue.h"
#include "gdcmULConnectionCallback.h"
#include "gdcmSubject.h"
#include "gdcmPresentationContext.h"
Include dependency graph for gdcmULConnectionManager.h:
```



## Classes

- class [gdcm::network::ULConnectionManager](#)  
*ULConnectionManager.*

## Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.594 gdcmULConnectionManager.h

[Go to the documentation of this file.](#)

```
00001 /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  * http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMULCONNECTIONMANAGER_H
00019 #define GDCMULCONNECTIONMANAGER_H
00020
00021 #include "gdcmULTransitionTable.h"
00022 #include "gdcmULConnection.h"
00023 #include "gdcmULConnectionInfo.h"
00024 #include "gdcmPresentationDataValue.h"
```

```

00025 #include "gdcmlConnectionCallback.h"
00026 #include "gdcmlSubject.h"
00027 #include "gdcmlPresentationContext.h"
00028
00029 namespace gdcml {
00030     class File;
00031     class BaseRootQuery;
00032     class BaseQuery;
00033
00034     namespace network {
00035
00045     class GDCM_EXPORT ULConnectionManager : public Subject
00046     {
00047     protected:
00048         ULConnection* mConnection;
00049         ULConnection* mSecondaryConnection;
00050         ULTransitionTable mTransitions;
00051
00052         //no copying
00053         ULConnectionManager(const ULConnectionManager& inCM);
00054
00055         //event handler loop.
00056         //will just keep running until the current event is nonexistent.
00057         //at which point, it will return the current state of the connection
00058         //this starts by initiating an action, but can be put into a passive mode
00059         //for a cmove/cstore combination by setting startWaiting to true
00060         EStateID RunEventLoop(ULEvent& inEvent, ULConnection* inWhichConnection,
00061             ULConnectionCallback* inCallback, const bool& startWaiting);
00062
00063         //like the above, but will manage the event loop for a move event (which
00064         //is basically two simultaneous connections interwoven, one inbound and
00065         //the other outbound. Note, for instance, that cmoversp's can be sent back
00066         //during the other connection's operation.
00067         EStateID RunMoveEventLoop(ULEvent& inEvent, ULConnectionCallback* inCallback);
00068
00069     public:
00070         ULConnectionManager();
00071         ~ULConnectionManager() override;
00072
00073         // NOTE: (MM) The following two functions are difficult to use, therefore marking
00074         // them as internal for now.
00075
00076         // \internal
00077         bool EstablishConnection(const std::string& inAETitle,
00078             const std::string& inConnectAETitle,
00079             const std::string& inComputerName, long inIPAddress,
00080             uint16_t inConnectPort, double inTimeout,
00081             std::vector<PresentationContext> const & pcVector );
00082
00083         bool EstablishConnectionMove(const std::string& inAETitle,
00084             const std::string& inConnectAETitle,
00085             const std::string& inComputerName, long inIPAddress,
00086             uint16_t inConnectPort, double inTimeout,
00087             uint16_t inReturnPort,
00088             std::vector<PresentationContext> const & pcVector);
00089         // \endinternal
00090
00091         //bool ReestablishConnection(const EConnectionType& inConnectionType,
00092         //    const DataSet& inDS);
00093
00094         //allows for a connection to be broken, but waits for an acknowledgement
00095         //of the breaking for a certain amount of time. Returns true if the
00096         //other side acknowledges the break
00097         bool BreakConnection(const double& inTimeout);
00098
00099         //severs the connection, if it's open, without waiting for any kind of response.
00100         //typically done if the program is going down.
00101         void BreakConnectionNow();
00102
00103         //This function will send a given piece of data
00104         //across the network connection. It will return true if the
00105         //sending worked, false otherwise.
00106         //note that sending is asynchronous; as such, there's
00107         //also a 'receive' option, but that requires a callback function.
00108         //bool SendData();
00109
00110         //send the Data PDU associated with Echo (ie, a default DataPDU)
00111         //this lets the user confirm that the connection is alive.
00112         //the user should look to cout to see the response of the echo command
00113         //returns the PresentationDataValue that was returned by the remote

```

```

00124      //host. Note that the PDV can be uninitialized, which would indicate failure.
00125      //Echo does not use a callback for results.
00126      std::vector<PresentationDataValue> SendEcho();
00127
00128      // \internal
00129      // API will change...
00130      std::vector<DataSet> SendStore(const File &file, std::istream * pStream = nullptr, std::streampos
dataSetOffset = 0 );
00131      std::vector<DataSet> SendFind(const BaseRootQuery* inRootQuery);
00132      std::vector<DataSet> SendMove(const BaseRootQuery* inRootQuery);
00133
00134      std::vector<DataSet> SendNEventReport (const BaseQuery* inQuery);
00135      std::vector<DataSet> SendNGet      (const BaseQuery* inQuery);
00136      std::vector<DataSet> SendNSet      (const BaseQuery* inQuery);
00137      std::vector<DataSet> SendNAction   (const BaseQuery* inQuery);
00138      std::vector<DataSet> SendNCreate   (const BaseQuery* inQuery);
00139      std::vector<DataSet> SendNDelete   (const BaseQuery* inQuery);
00140      // \endinternal
00141
00143      void SendStore(const File & file, ULConnectionCallback* inCallback, std::istream * pStream = nullptr
, std::streampos dataSetOffset = 0 );
00144      void SendFind(const BaseRootQuery* inRootQuery, ULConnectionCallback* inCallback);
00146      bool SendMove(const BaseRootQuery* inRootQuery, ULConnectionCallback* inCallback);
00147
00148      void SendNEventReport (const BaseQuery* inQuery, ULConnectionCallback* inCallback);
00149      void SendNGet      (const BaseQuery* inQuery, ULConnectionCallback* inCallback);
00150      void SendNSet      (const BaseQuery* inQuery, ULConnectionCallback* inCallback);
00151      void SendNAction   (const BaseQuery* inQuery, ULConnectionCallback* inCallback);
00152      void SendNCreate   (const BaseQuery* inQuery, ULConnectionCallback* inCallback);
00153      void SendNDelete   (const BaseQuery* inQuery, ULConnectionCallback* inCallback);
00154
00155  };
00156  }
00157 }
00158
00159 #endif // GDCMULCONNECTIONMANAGER_H

```

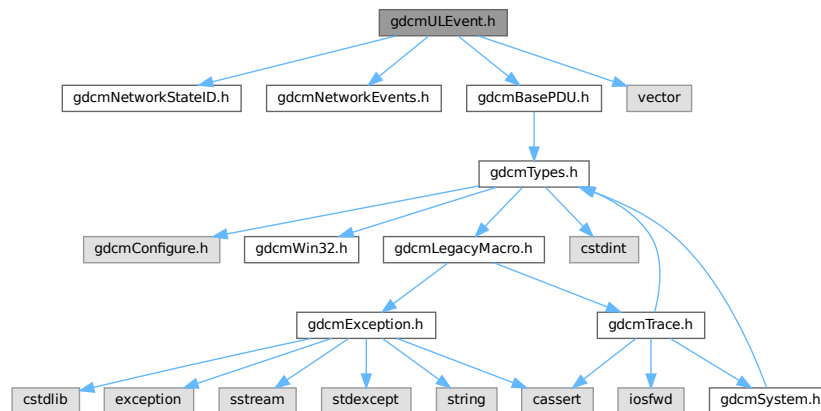
## 11.595 gdcmlEvent.h File Reference

```

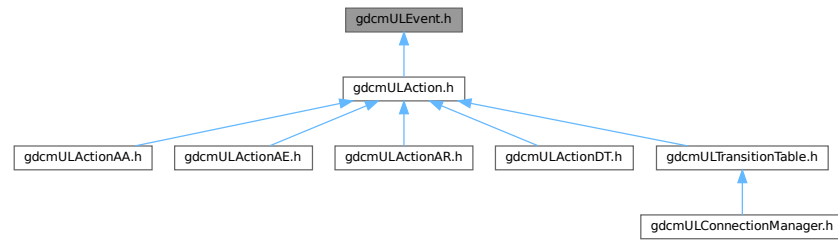
#include "gdcmlNetworkStateID.h"
#include "gdcmlNetworkEvents.h"
#include "gdcmlBasePDU.h"
#include <vector>

```

Include dependency graph for gdcmlEvent.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcml::network::UEvent`  
*UEvent.*

## Namespaces

- namespace `gdcml`
- namespace `gdcml::network`

## 11.596 gdcmlEvent.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002  *
00003  *   Copyright NumFOCUS
00004  *
00005  *   Licensed under the Apache License, Version 2.0 (the "License");
00006  *   you may not use this file except in compliance with the License.
00007  *   You may obtain a copy of the License at
00008  *
00009  *       http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  *   Unless required by applicable law or agreed to in writing, software
00012  *   distributed under the License is distributed on an "AS IS" BASIS,
00013  *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  *   See the License for the specific language governing permissions and
00015  *   limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMLEVENT_H
00019 #define GDCMLEVENT_H
00020
00021 #include "gdcmlNetworkStateID.h"
00022 #include "gdcmlNetworkEvents.h"
00023 #include "gdcmlBasePDU.h"
00024 #include <vector>
00025
00026 namespace gdcml {
00027     namespace network {
00028
00037     class UEvent {
00038     public:
00039         EEventID mEvent;
00040         std::vector<BasePDU*> mBasePDU;
00041         std::istream * m_pStream ;
00042     };
00043     }
00044 }
  
```

```

00041     std::streampos m_posDataSet ;
00042     void DeletePDUVector(){
00043         std::vector<BasePDU*>::iterator baseItror;
00044         for (baseItror = mBasePDU.begin(); baseItror < mBasePDU.end(); baseItror++){
00045             if (*baseItror != nullptr){
00046                 delete *baseItror;
00047                 *baseItror = nullptr;
00048             }
00049         }
00050     }
00051
00052     public:
00053     ULEvent(const EEventID& inEventID, std::vector<BasePDU*> inBasePDU, std::istream * iStream =
nullptr, std::streampos posDataSet = 0 ){
00054         mEvent = inEventID;
00055         mBasePDU = inBasePDU;
00056         m_pStream = iStream ;
00057         m_posDataSet = posDataSet ;
00058     }
00059     ULEvent(const EEventID& inEventID, BasePDU* inBasePDU, std::istream * iStream = nullptr,
std::streampos posDataSet = 0 ){
00060         mEvent = inEventID;
00061         mBasePDU.push_back(inBasePDU);
00062         m_pStream = iStream ;
00063         m_posDataSet = posDataSet ;
00064     }
00065     ~ULEvent(){
00066         DeletePDUVector();
00067     }
00068
00069     EEventID GetEvent() const { return mEvent; }
00070     std::vector<BasePDU*> const & GetPDUs() const { return mBasePDU; }
00071     std::istream * GetIStream() const { return m_pStream; }
00072     std::streampos GetDataSetPos() const { return m_posDataSet; }
00073
00074     void SetEvent(const EEventID& inEvent) { mEvent = inEvent; }
00075     void SetPDU(std::vector<BasePDU*> const & inPDU) {
00076         DeletePDUVector();
00077         mBasePDU = inPDU;
00078     }
00079 };
00080 }
00081 }
00082
00083 #endif //GDCMULEVENT_H

```

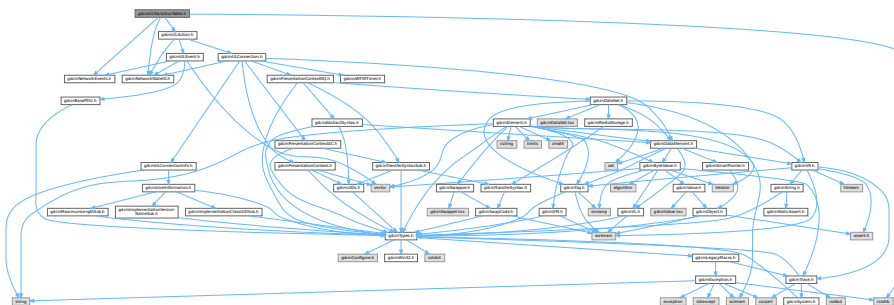
## 11.597 gdcmlTransitionTable.h File Reference

```

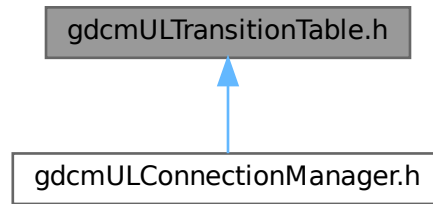
#include "gdcmlNetworkStateID.h"
#include "gdcmlNetworkEvents.h"
#include "gdcmlULAction.h"
#include <cstdlib>

```

Include dependency graph for gdcmlTransitionTable.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcml::network::TableRow](#)
- struct [gdcml::network::Transition](#)
- class [gdcml::network::ULTransitionTable](#)  
*ULTransitionTable* The transition table of all the ULEvents, new ULActions, and ULStates.

## Namespaces

- namespace [gdcml](#)
- namespace [gdcml::network](#)

## 11.598 gdcmlTransitionTable.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002  *
00003  *   Copyright NumFOCUS
00004  *
00005  *   Licensed under the Apache License, Version 2.0 (the "License");
00006  *   you may not use this file except in compliance with the License.
00007  *   You may obtain a copy of the License at
00008  *
00009  *       http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  *   Unless required by applicable law or agreed to in writing, software
00012  *   distributed under the License is distributed on an "AS IS" BASIS,
00013  *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  *   See the License for the specific language governing permissions and
00015  *   limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMULTRANSITIONTABLE_H
00019 #define GDCMULTRANSITIONTABLE_H
00020
00021 #include "gdcmlNetworkStateID.h"
00022 #include "gdcmlNetworkEvents.h"
00023 #include "gdcmlULAction.h"
00024
00025 #include <stdlib> // NULL
  
```

```

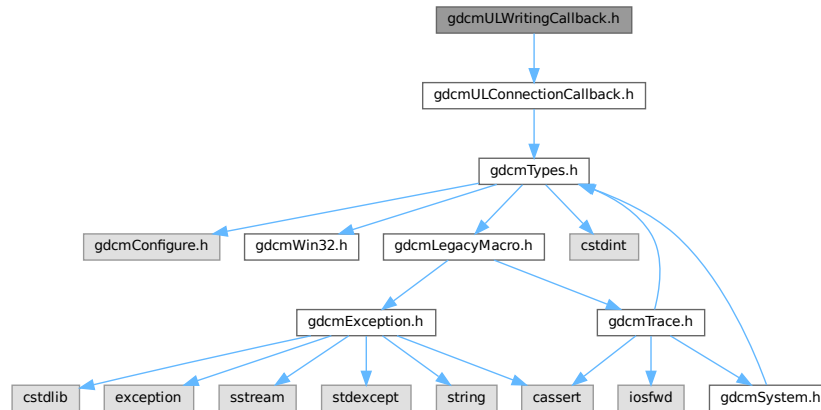
00026
00027 namespace gdcm {
00028   class Subject;
00029   namespace network{
00030     class ULConnection;
00031     class ULAction;
00032     class ULEvent;
00033
00034     //The transition dictates the action that should be taken from the start state to the end state
00035     struct Transition {
00036       int mEnd;
00037       ULAction* mAction;
00038       Transition(){
00039         mEnd = eStaDoesNotExist;
00040         mAction = nullptr;
00041       }
00042       ~Transition(){
00043         if (mAction != nullptr){
00044           delete mAction;
00045           mAction = nullptr;
00046         }
00047       }
00048       Transition(int inEndState, ULAction* inAction){
00049         mEnd = inEndState;
00050         mAction = inAction;
00051       }
00052       static Transition* MakeNew(int inEndState, ULAction* inAction){
00053         return new Transition(inEndState, inAction);
00054       }
00055     };
00056
00057     //used to define a row in table 9-10 of 3.8 2009
00058     //the transition table is events, then state,
00059     //then the transition itself (which has the event
00060     //and start state implied by their starting locations)
00061     //don't need to store the event; that's implicitly defined in the Table itself by location
00062     class TableRow{
00063     public:
00064       TableRow() {
00065         for(int stateIndex = 0; stateIndex < cMaxStateID; ++stateIndex)
00066         {
00067           transitions[stateIndex] = nullptr;
00068         }
00069       }
00070       ~TableRow() {
00071         for(int stateIndex = 0; stateIndex < cMaxStateID; ++stateIndex)
00072         {
00073           Transition *t = transitions[stateIndex];
00074           delete t;
00075         }
00076       }
00077       Transition *transitions[cMaxStateID];
00078
00079       //copy constructor for stl additions into the transition table below.
00080     };
00081
00082   class ULTransitionTable
00083   {
00084   private:
00085     TableRow mTable[cMaxEventID];
00086   public:
00087     ULTransitionTable();
00088
00089     void HandleEvent(Subject*s, ULEvent& inEvent, ULConnection& inConnection,
00090                     bool& outWaitingForEvent, EEventID& outRaisedEvent) const;
00091
00092     void PrintTable() const; //so that the table can be printed and verified against the DICOM standard
00093   };
00094 }
00095
00096 #endif // GDCMULTRANSITIONTABLE_H

```

## 11.599 gdcmULWritingCallback.h File Reference

```
#include "gdcmULConnectionCallback.h"
```

Include dependency graph for gdcmULWritingCallback.h:



### Classes

- class [gdcm::network::ULWritingCallback](#)

### Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

## 11.600 gdcmULWritingCallback.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002  *
00003  *   Copyright NumFOCUS
00004  *
00005  *   Licensed under the Apache License, Version 2.0 (the "License");
00006  *   you may not use this file except in compliance with the License.
00007  *   You may obtain a copy of the License at
00008  *
00009  *       http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  *   Unless required by applicable law or agreed to in writing, software
00012  *   distributed under the License is distributed on an "AS IS" BASIS,
00013  *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  *   See the License for the specific language governing permissions and
00015  *   limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMULCONNECTIONWRITINGCALLBACK_H
00019 #define GDCMULCONNECTIONWRITINGCALLBACK_H
00020

```



```

00021 #include "gdcmlULConnectionCallback.h"
00022
00023 namespace gdcml
00024 {
00025     class DataSet;
00026     namespace network
00027     {
00028         /* \brief ULWritingCallback
00029          * \details This is the most basic of callbacks for how the ULConnectionManager handles
00030          * incoming datasets. DataSets are immediately written to disk as soon as they
00031          * are received. NOTE that if the incoming connection is faster than the disk
00032          * writing speed, this callback could cause some pileups!
00033          */
00034         class GDCM_EXPORT ULWritingCallback : public ULConnectionCallback
00035         {
00036         public:
00037             std::string mDirectoryName;
00038             ULWritingCallback() = default;
00039             ~ULWritingCallback() override = default; //empty, for later inheritance
00040
00041             void SetDirectory(const std::string& inDirectoryName) { mDirectoryName = inDirectoryName; }
00042             void HandleDataSet(const DataSet& inDataSet) override;
00043             void HandleResponse(const DataSet& inDataSet) override;
00044         };
00045     } // end namespace network
00046 } // end namespace gdcml
00047
00048 #endif //GDCMLCONNECTIONWRITINGCALLBACK_H

```

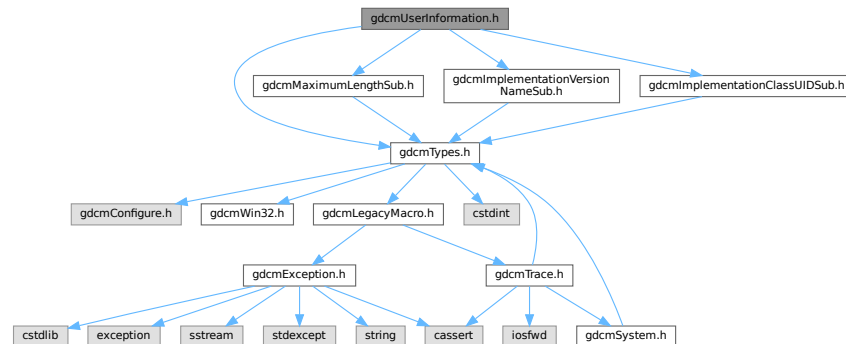
## 11.601 gdcmlUserInformation.h File Reference

```

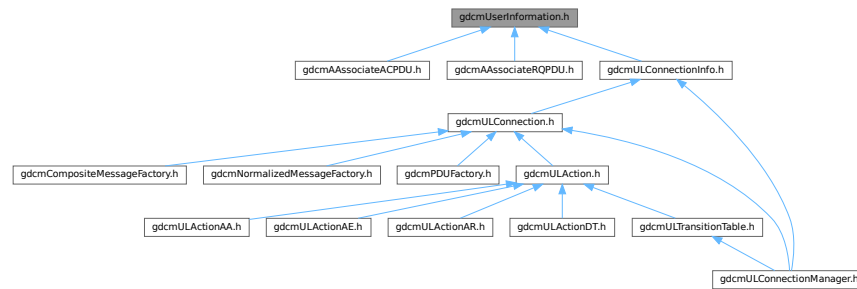
#include "gdcmlTypes.h"
#include "gdcmlMaximumLengthSub.h"
#include "gdcmlImplementationVersionNameSub.h"
#include "gdcmlImplementationClassUIDSub.h"

```

Include dependency graph for gdcmlUserInformation.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcml::network::UserInformation](#)  
*UserInformation.*

## Namespaces

- namespace [gdcml](#)
- namespace [gdcml::network](#)

## 11.602 gdcmlUserInformation.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMLUSERINFORMATION_H
00015  #define GDCMLUSERINFORMATION_H
00016
00017  #include "gdcmlTypes.h"
00018  #include "gdcmlMaximumLengthSub.h"
00019  #include "gdcmlImplementationVersionNameSub.h"
00020  #include "gdcmlImplementationClassUIDSub.h"
00021
00022  namespace gdcml
00023  {
00024
00025    namespace network
00026    {
00027
00028      class AsynchronousOperationsWindowSub;
00029      class RoleSelectionSub;
00030      struct RoleSelectionSubItems;
00031      class SOPClassExtendedNegociationSub;
00032    }
00033  }
  
```

## 11.603 gdcmlWLMFindQuery.h File Reference

## Classes

- class `gdcm::WLMFindQuery`  
*PatientRootQuery.*

## Namespaces

- namespace `gdcm`

## 11.604 gdcmWLMFindQuery.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMWLMFindQuery_H
00015 #define GDCMWLMFindQuery_H
00016
00017 #include "gdcmBaseRootQuery.h"
00018
00019 namespace gdcm
00020 {
00025     class GDCM_EXPORT WLMFindQuery : public BaseRootQuery
00026     {
00027     friend class QueryFactory;
00028     public:
00029         WLMFindQuery();
00030
00031         // no sense here
00032         void InitializeDataSet(const EQueryLevel& inQueryLevel) override;
00033         std::vector<Tag> GetTagListByLevel(const EQueryLevel& inQueryLevel) override;
00034         // validate query has required tag
00035         bool ValidateQuery(bool inStrict = true) const override;
00036
00037         UIDs::TSName GetAbstractSyntaxUID() const override;
00038     protected :
00039         DataSet GetValidDataSet() const;
00040     };
00041
00042 } // end namespace gdcm
00043
00044 #endif // GDCMWLMFindQuery_H

```

## 11.605 vtkGDCMImageReader.h File Reference

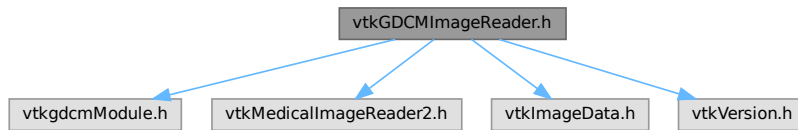
```

#include "vtkgdcmModule.h"
#include "vtkMedicalImageReader2.h"
#include "vtkImageData.h"

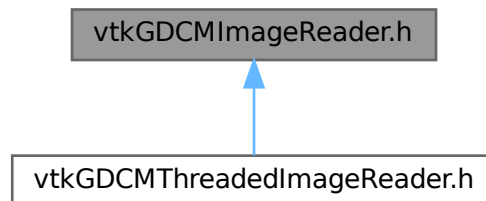
```

```
#include "vtkVersion.h"
```

Include dependency graph for vtkGDCMImageReader.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [vtkGDCMImageReader](#)

## Namespaces

- namespace [gdc](#)m

## Macros

- #define [VTK\\_CMYK](#) 8
- #define [VTK\\_INVERSE\\_LUMINANCE](#) 5
- #define [VTK\\_LOOKUP\\_TABLE](#) 6
- #define [VTK\\_YBR](#) 7

## 11.605.1 Macro Definition Documentation

### 11.605.1.1 VTK\_CMYK

```
#define VTK_CMYK 8
```

### 11.605.1.2 VTK\_INVERSE\_LUMINANCE

```
#define VTK_INVERSE_LUMINANCE 5
```

### 11.605.1.3 VTK\_LOOKUP\_TABLE

```
#define VTK_LOOKUP_TABLE 6
```

### 11.605.1.4 VTK\_YBR

```
#define VTK_YBR 7
```

## 11.606 vtkGDCMImageReader.h

[Go to the documentation of this file.](#)

```
00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 // .NAME vtkGDCMImageReader - read DICOM Image files (Pixel Data)
00015 // .SECTION Description
00016 // vtkGDCMImageReader is a source object that reads some DICOM files
00017 // this reader is single threaded.
00018 // .SECTION Implementation note: when FileLowerLeft is set to on the image is not flipped
00019 // upside down as VTK would expect, use this option only if you know what you are doing.
00020 // .SECTION Implementation note: when reading a series of 2D slices, user is
00021 // expected to provide an ordered list of filenames. No sorting will be applied afterward.
00022 // .SECTION Implementation note: Although 99% of the time the Zspacing as read
00023 // from a tag in a 2D DICOM file should be correct, there has been reports that this
00024 // value can be missing, or incorrect, in which case users are advised to override this
00025 // value using the return value from gdcm::IPPSorter::GetZSpacing() and set it via
00026 // vtkImageChangeInformation on the reader itself.
00027 // .SECTION TODO
00028 // This reader does not handle a series of 3D images, only a single 3D (multi frame) or a
00029 // list of 2D files are supported for now.
00030 // .SECTION TODO
00031 // Did not implement SetFilePattern / SetFilePrefix API, move it to protected section for now.
00032 // .SECTION BUG
00033 // Overlay are assumed to have the same extent as image. Right now if overlay origin is not
00034 // 0,0 the overlay will have an offset...
00035 // Only the very first overlay is loaded at the VTK level, for now (even if there are more than one in the
    file)
00036 // .SECTION DataOrigin
00037 // When the reader is instantiated with FileLowerLeftOn the DataOrigin and Image Position (Patient) are
00038 // identical. But when FileLowerLeft is Off, we have to reorder the Y-line of the image, and thus the
    DataOrigin
00039 // is then translated to the other side of the image.
00040 // .SECTION Spacing
00041 // When reading a 3D volume, the spacing along the Z dimension might be negative (so as to respect
    up-side-down)
00042 // as specified in the Image Orientation (Patient) tag. When Z-spacing is 0, this means the multi-frame
    object
00043 // contains image which do not represent uniform volume.
00044 // .SECTION Warning
00045 // When using vtkGDCMPolyDataReader in conjunction with vtkGDCMImageReader
```

```

00046 // it is *required* that FileLowerLeft is set to ON as coordinate system
00047 // would be inconsistent in between the two data structures.
00048 // .SECTION Color Space mapping:
00049 // * VTK_LUMINANCE      <-> MONOCHROME2
00050 // * VTK_LUMINANCE_ALPHA <-> Not supported
00051 // * VTK_RGB            <-> RGB
00052 // * VTK_RGBA           <-> ARGB (deprecated, DICOM 2008)
00053 // * VTK_INVERSE_LUMINANCE <-> MONOCHROME1
00054 // * VTK_LOOKUP_TABLE   <-> PALETTE COLOR
00055 // * VTK_YBR            <-> YBR_FULL
00056 //
00057 // For detailed information on color space transformation and true lossless transformation see:
00058 // http://gdcm.sourceforge.net/wiki/index.php/Color_Space_Transformations
00059
00060 // .SECTION See Also
00061 // vtkMedicalImageReader2 vtkMedicalImageProperties vtkGDCMPolyDataReader vtkGDCMImageWriter
00062 // vtkDICOMImageReader
00063
00064 #ifndef VTKGDCMIMAGEREADER_H
00065 #define VTKGDCMIMAGEREADER_H
00066
00067 #include "vtkgdcmModule.h"
00068 #include "vtkMedicalImageReader2.h"
00069 #include "vtkImageData.h"
00070 #include "vtkVersion.h"
00071
00072 #if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
00073 #else
00074 class vtkMedicalImageProperties;
00075 #endif
00076 #if (VTK_MAJOR_VERSION > 5) || ( VTK_MAJOR_VERSION == 5 && VTK_MINOR_VERSION > 0 )
00077 #else
00078 class vtkStringArray;
00079 #endif
00080 class vtkPolyData;
00081
00082 // vtkSystemIncludes.h defines:
00083 // #define VTK_LUMINANCE      1
00084 // #define VTK_LUMINANCE_ALPHA 2
00085 // #define VTK_RGB           3
00086 // #define VTK_RGBA          4
00087 #ifndef VTK_INVERSE_LUMINANCE
00088 #define VTK_INVERSE_LUMINANCE 5
00089 #endif
00090 #ifndef VTK_LOOKUP_TABLE
00091 #define VTK_LOOKUP_TABLE 6
00092 #endif
00093 #ifndef VTK_YBR
00094 #define VTK_YBR 7
00095 #endif
00096 #ifndef VTK_CMYK
00097 #define VTK_CMYK 8
00098 #endif
00099
00100 //BTX
00101 namespace gdcm { class ImageReader; }
00102 //ETX
00103 class vtkMatrix4x4;
00104 class VTKGDCM_EXPORT vtkGDCMImageReader : public vtkMedicalImageReader2
00105 {
00106 public:
00107     static vtkGDCMImageReader *New();
00108     vtkTypeMacro(vtkGDCMImageReader,vtkMedicalImageReader2);
00109     virtual void PrintSelf(ostream& os, vtkIndent indent);
00110
00111     // Description: is the given file name a DICOM file containing an image ?
00112     virtual int CanReadFile(const char* fname);
00113
00114     // Description:
00115     // Valid extensions
00116     virtual const char* GetFileExtensions()
00117     {
00118         // I would like to get rid of ACR/NEMA/IMA so only allow dcm extension for now
00119         return ".dcm .DCM";
00120     }
00121
00122     // Description:
00123     // A descriptive name for this format
00124     virtual const char* GetDescriptiveName()
00125     {
00126         return "DICOM";
00127     }
00128

```

```

00127     }
00128
00129     // Description:
00130     // Get the Image Position (Patient) as stored in the DICOM file
00131     // This is a read-only data member
00132     vtkGetObjectMacro(DirectionCosines, vtkMatrix4x4);
00133
00134     #if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
00135     #else
00136         // Description:
00137         // Get the medical image properties object
00138         vtkGetObjectMacro(MedicalImageProperties, vtkMedicalImageProperties);
00139     #endif
00140     virtual void SetMedicalImageProperties(vtkMedicalImageProperties *pd);
00141
00142     #if (VTK_MAJOR_VERSION > 5) || ( VTK_MAJOR_VERSION == 5 && VTK_MINOR_VERSION > 0 )
00143     #else
00144         virtual void SetFileNames(vtkStringArray*);
00145         vtkGetObjectMacro(FileNames, vtkStringArray);
00146     #endif
00147
00148     // Description:
00149     // Specifically request to load the overlay into the gdcm-VTK layer (gdcm always loads them when found).
00150     // If no overlay is found in the image, then the vtkImageData for the overlay will be empty.
00151     vtkGetMacro(LoadOverlays,int);
00152     vtkSetMacro(LoadOverlays,int);
00153     vtkBooleanMacro(LoadOverlays,int);
00154
00155     // Description:
00156     // Set/Get whether or not to load the Icon as vtkImageData (if found in the DICOM file)
00157     vtkGetMacro(LoadIconImage,int);
00158     vtkSetMacro(LoadIconImage,int);
00159     vtkBooleanMacro(LoadIconImage,int);
00160
00161     // Description:
00162     // Set/Get whether or not the image was compressed using a lossy compression algorithm
00163     vtkGetMacro(LossyFlag,int);
00164     vtkSetMacro(LossyFlag,int);
00165     vtkBooleanMacro(LossyFlag,int);
00166
00167     // Description:
00168     // Read only: number of overlays as found in this image (multiple overlays per slice is allowed)
00169     // Only valid when LoadOverlays is true
00170     vtkGetMacro(NumberOfOverlays,int);
00171
00172     // Description:
00173     // Read only: number of icon image (there can only be zero or one icon per file)
00174     // Only valid when LoadIconImage is true
00175     vtkGetMacro(NumberOfIconImages,int);
00176
00177     // Description:
00178     // Get Overlay/IconImage
00179     // Remember to ALWAYS use those methods in your code, as the internal number for the output port
00180     // is not guaranteed to remain the same, as features are added to the reader
00181     #if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
00182     //FIXME: Need to get rid of BTX/ETX if only the Python Wrapper of VTK 4.2 would let me
00183     //BTX
00184     vtkAlgorithmOutput* GetOverlayPort(int index);
00185     vtkAlgorithmOutput* GetIconImagePort();
00186     //ETX
00187     #endif
00188     vtkImageData* GetOverlay(int i);
00189     vtkImageData* GetIconImage();
00190
00191     // Description:
00192     // Load image with its associated Lookup Table
00193     vtkGetMacro(ApplyLookupTable,int);
00194     vtkSetMacro(ApplyLookupTable,int);
00195     vtkBooleanMacro(ApplyLookupTable,int);
00196
00197     // Description:
00198     // Load image as YBR
00199     vtkGetMacro(ApplyYBRToRGB,int)
00200     vtkSetMacro(ApplyYBRToRGB,int)
00201     vtkBooleanMacro(ApplyYBRToRGB,int);
00202
00203     // Description:
00204     // Return VTK_LUMINANCE, VTK_INVERSE_LUMINANCE, VTK_RGB, VTK_RGBA, VTK_LOOKUP_TABLE, VTK_YBR or VTK_CMYK
00205     // or 0 when ImageFormat is not handled.
00206     // Warning: For color image, PlanarConfiguration need to be taken into account.
00207     vtkGetMacro(ImageFormat,int);

```



```

00208
00209 // Description:
00210 // Return the Planar Configuration. This simply means that the internal DICOM image was stored
00211 // using a particular planar configuration (most of the time: 0)
00212 // For monochrome image, PlanarConfiguration is always 0
00213 vtkGetMacro(PlanarConfiguration,int);
00214
00215 // Description:
00216 // Return the 'raw' information stored in the DICOM file:
00217 // In case of a series of multiple files, only the first file is considered. The Image Orientation
(Patient)
00218 // is guaranteed to remain the same, and image Image Position (Patient) in other slice can be computed
00219 // using the ZSpacing (3rd dimension)
00220 // (0020,0032) DS [87.774866\~182.908510\168.629671] # 32, 3 ImagePositionPatient
00221 // (0020,0037) DS [0.001479\0.999989\~0.004376\~0.002039\~0.004372\~0.999988] # 58, 6
ImageOrientationPatient
00222 vtkGetVector3Macro(ImagePositionPatient,double);
00223 vtkGetVector6Macro(ImageOrientationPatient,double);
00224
00225 // Description:
00226 // Set/Get the first Curve Data:
00227 vtkGetObjectMacro(Curve,vtkPolyData);
00228 virtual void SetCurve(vtkPolyData *pd);
00229
00230 // Description:
00231 // \DEPRECATED:
00232 // Modality LUT
00233 // Value returned by GetShift/GetScale might be inaccurate since Shift/Scale could be
00234 // varying along the Series read. Therefore user are advices not to use those functions
00235 // anymore
00236 vtkGetMacro(Shift,double);
00237 vtkGetMacro(Scale,double);
00238
00239 protected:
00240 vtkGDCMImageReader();
00241 ~vtkGDCMImageReader();
00242
00243 vtkSetVector6Macro(ImageOrientationPatient,double);
00244
00245 //BTX
00246 void FillMedicalImageInformation(const gdcm::ImageReader &reader);
00247 //ETX
00248 int RequestInformationCompat();
00249 int RequestDataCompat();
00250
00251 #if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
00252 int ProcessRequest(vtkInformation* request,
00253                   vtkInformationVector** inputVector,
00254                   vtkInformationVector* outputVector);
00255 int RequestInformation(vtkInformation *request,
00256                       vtkInformationVector **inputVector,
00257                       vtkInformationVector *outputVector);
00258 int RequestData(vtkInformation *request,
00259                vtkInformationVector **inputVector,
00260                vtkInformationVector *outputVector);
00261 #else /*(VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )*/
00262 void ExecuteInformation();
00263 void ExecuteData(vtkDataObject *out);
00264 #endif /*(VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )*/
00265
00266 protected:
00267 #if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
00268 #else
00269 // Description:
00270 // Medical Image properties
00271 vtkMedicalImageProperties *MedicalImageProperties;
00272 #endif
00273 #if (VTK_MAJOR_VERSION > 5) || ( VTK_MAJOR_VERSION == 5 && VTK_MINOR_VERSION > 0 )
00274 #else
00275 vtkStringArray *FileNames;
00276 #endif
00277
00278 vtkMatrix4x4 *DirectionCosines;
00279 int LoadOverlays;
00280 int NumberOfOverlays;
00281 int LoadIconImage;
00282 int NumberOfIconImages;
00283 int IconImageDataExtent[6];
00284 double ImagePositionPatient[3];
00285 double ImageOrientationPatient[6];
00286 vtkPolyData *Curve;

```

```

00287
00288     int ImageFormat;
00289     // the following 3, should remain optional
00290     int ApplyInverseVideo;
00291     int ApplyLookupTable;
00292     int ApplyYBRToRGB;
00293     // I think that planar configuration need to always be applied as far as VTK is concerned
00294     int ApplyPlanarConfiguration;
00295     int ApplyShiftScale;
00296
00297     int LoadSingleFile(const char *filename, char *pointer, unsigned long &outlen);
00298
00299     double Shift;
00300     double Scale;
00301     int IconDataScalarType;
00302     int IconNumberOfScalarComponents;
00303     int PlanarConfiguration;
00304     int LossyFlag;
00305     int ForceRescale;
00306
00307 protected:
00308     // TODO / FIXME
00309     void SetFilePrefix(const char *) {}
00310     vtkGetStringMacro(FilePrefix);
00311     void SetFilePattern(const char *) {}
00312     vtkGetStringMacro(FilePattern);
00313
00314 private:
00315     vtkGDCMImageReader(const vtkGDCMImageReader&); // Not implemented.
00316     void operator=(const vtkGDCMImageReader&); // Not implemented.
00317 };
00318 #endif

```

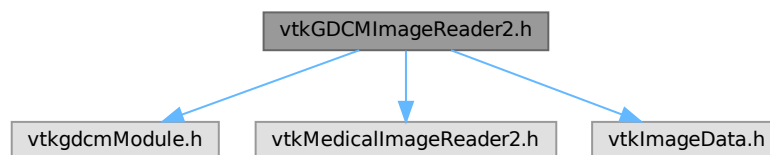
## 11.607 vtkGDCMImageReader2.h File Reference

```

#include "vtkgdcmModule.h"
#include "vtkMedicalImageReader2.h"
#include "vtkImageData.h"

```

Include dependency graph for vtkGDCMImageReader2.h:



### Classes

- class [vtkGDCMImageReader2](#)

### Namespaces

- namespace [gdcm](#)

**Macros**

- #define `VTK_CMYK` 8
- #define `VTK_INVERSE_LUMINANCE` 5
- #define `VTK_LOOKUP_TABLE` 6
- #define `VTK_YBR` 7

**11.607.1 Macro Definition Documentation****11.607.1.1 VTK\_CMYK**

```
#define VTK_CMYK 8
```

**11.607.1.2 VTK\_INVERSE\_LUMINANCE**

```
#define VTK_INVERSE_LUMINANCE 5
```

**11.607.1.3 VTK\_LOOKUP\_TABLE**

```
#define VTK_LOOKUP_TABLE 6
```

**11.607.1.4 VTK\_YBR**

```
#define VTK_YBR 7
```

**11.608 vtkGDCMImageReader2.h**

[Go to the documentation of this file.](#)

```
00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 // .NAME vtkGDCMImageReader2 - read DICOM Image files (Pixel Data)
00015 // .SECTION Description
00016 // vtkGDCMImageReader2 is a source object that reads some DICOM files
00017 // this reader is single threaded.
00018 // .SECTION Implementation note: when FileLowerLeft is set to on the image is not flipped
00019 // upside down as VTK would expect, use this option only if you know what you are doing.
00020 // .SECTION Implementation note: when reading a series of 2D slices, user is
00021 // expected to provide an ordered list of filenames. No sorting will be applied afterward.
00022 // .SECTION Implementation note: Although 99% of the time the Zspacing as read
00023 // from a tag in a 2D DICOM file should be correct, there has been reports that this
00024 // value can be missing, or incorrect, in which case users are advised to override this
```

```

00025 // value using the return value from gdcm::IPPSorter::GetZSpacing() and set it via
00026 // vtkImageChangeInformation on the reader itself.
00027 // .SECTION TODO
00028 // This reader does not handle a series of 3D images, only a single 3D (multi frame) or a
00029 // list of 2D files are supported for now.
00030 // .SECTION TODO
00031 // Did not implement SetFilePattern / SetFilePrefix API, move it to protected section for now.
00032 // .SECTION BUG
00033 // Overlay are assumed to have the same extent as image. Right now if overlay origin is not
00034 // 0,0 the overlay will have an offset...
00035 // Only the very first overlay is loaded at the VTK level, for now (even if there are more than one in the
    file)
00036 // .SECTION DataOrigin
00037 // When the reader is instantiated with FileLowerLeftOn the DataOrigin and Image Position (Patient) are
00038 // identical. But when FileLowerLeft is Off, we have to reorder the Y-line of the image, and thus the
    DataOrigin
00039 // is then translated to the other side of the image.
00040 // .SECTION Spacing
00041 // When reading a 3D volume, the spacing along the Z dimension might be negative (so as to respect
    up-side-down)
00042 // as specified in the Image Orientation (Patient) tag. When Z-spacing is 0, this means the multi-frame
    object
00043 // contains image which do not represent uniform volume.
00044 // .SECTION Warning
00045 // When using vtkGDCMPolyDataReader in conjunction with vtkGDCMImageReader2
00046 // it is *required* that FileLowerLeft is set to ON as coordinate system
00047 // would be inconsistent in between the two data structures.
00048 // .SECTION Color Space mapping:
00049 // * VTK_LUMINANCE <=> MONOCHROME2
00050 // * VTK_LUMINANCE_ALPHA <=> Not supported
00051 // * VTK_RGB <=> RGB
00052 // * VTK_RGBA <=> ARGB (deprecated, DICOM 2008)
00053 // * VTK_INVERSE_LUMINANCE <=> MONOCHROME1
00054 // * VTK_LOOKUP_TABLE <=> PALETTE COLOR
00055 // * VTK_YBR <=> YBR_FULL
00056 //
00057 // For detailed information on color space transformation and true lossless transformation see:
00058 // http://gdcm.sourceforge.net/wiki/index.php/Color\_Space\_Transformations
00059
00060 // .SECTION See Also
00061 // vtkMedicalImageReader2 vtkMedicalImageProperties vtkGDCMPolyDataReader vtkGDCMImageWriter
00062 // vtkDICOMImageReader
00063
00064 #ifndef VTKGDCMIMAGEREADER2_H
00065 #define VTKGDCMIMAGEREADER2_H
00066
00067 #include "vtkgdcmModule.h"
00068 #include "vtkMedicalImageReader2.h"
00069 #include "vtkImageData.h"
00070
00071 class vtkPolyData;
00072
00073 // vtkSystemIncludes.h defines:
00074 // #define VTK_LUMINANCE 1
00075 // #define VTK_LUMINANCE_ALPHA 2
00076 // #define VTK_RGB 3
00077 // #define VTK_RGBA 4
00078 #ifndef VTK_INVERSE_LUMINANCE
00079 #define VTK_INVERSE_LUMINANCE 5
00080 #endif
00081 #ifndef VTK_LOOKUP_TABLE
00082 #define VTK_LOOKUP_TABLE 6
00083 #endif
00084 #ifndef VTK_YBR
00085 #define VTK_YBR 7
00086 #endif
00087 #ifndef VTK_CMYK
00088 #define VTK_CMYK 8
00089 #endif
00090
00091 //BTX
00092 namespace gdcm { class ImageReader; }
00093 //ETX
00094 class vtkMatrix4x4;
00095 class VTKGDCM_EXPORT vtkGDCMImageReader2 : public vtkMedicalImageReader2
00096 {
00097 public:
00098     static vtkGDCMImageReader2 *New();
00099     vtkTypeMacro(vtkGDCMImageReader2,vtkMedicalImageReader2);
00100     virtual void PrintSelf(ostream& os, vtkIndent indent);
00101

```

```
00102 // Description: is the given file name a DICOM file containing an image ?
00103 virtual int CanReadFile(const char* fname);
00104
00105 // Description:
00106 // Valid extensions
00107 virtual const char* GetFileExtensions()
00108 {
00109     // I would like to get rid of ACR/NEMA/IMA so only allow dcm extension for now
00110     return ".dcm .DCM";
00111 }
00112
00113 // Description:
00114 // A descriptive name for this format
00115 virtual const char* GetDescriptiveName()
00116 {
00117     return "DICOM";
00118 }
00119
00120 // Description:
00121 // Get the Image Position (Patient) as stored in the DICOM file
00122 // This is a read-only data member
00123 vtkGetObjectMacro(DirectionCosines, vtkMatrix4x4);
00124
00125 virtual void SetMedicalImageProperties(vtkMedicalImageProperties *pd);
00126
00127 // Description:
00128 // Specifically request to load the overlay into the gdcm-VTK layer (gdcm always loads them when found).
00129 // If no overlay is found in the image, then the vtkImageData for the overlay will be empty.
00130 vtkGetMacro(LoadOverlays,int);
00131 vtkSetMacro(LoadOverlays,int);
00132 vtkBooleanMacro(LoadOverlays,int);
00133
00134 // Description:
00135 // Set/Get whether or not to load the Icon as vtkImageData (if found in the DICOM file)
00136 vtkGetMacro(LoadIconImage,int);
00137 vtkSetMacro(LoadIconImage,int);
00138 vtkBooleanMacro(LoadIconImage,int);
00139
00140 // Description:
00141 // Set/Get whether or not the image was compressed using a lossy compression algorithm
00142 vtkGetMacro(LossyFlag,int);
00143 vtkSetMacro(LossyFlag,int);
00144 vtkBooleanMacro(LossyFlag,int);
00145
00146 // Description:
00147 // Read only: number of overlays as found in this image (multiple overlays per slice is allowed)
00148 // Only valid when LoadOverlays is true
00149 vtkGetMacro(NumberOfOverlays,int);
00150
00151 // Description:
00152 // Read only: number of icon image (there can only be zero or one icon per file)
00153 // Only valid when LoadIconImage is true
00154 vtkGetMacro(NumberOfIconImages,int);
00155
00156 // Description:
00157 // Get Overlay/IconImage
00158 // Remember to ALWAYS use those methods in your code, as the internal number for the output port
00159 // is not guaranteed to remain the same, as features are added to the reader
00160 vtkAlgorithmOutput* GetOverlayPort(int index);
00161 vtkAlgorithmOutput* GetIconImagePort();
00162 vtkImageData* GetOverlay(int i);
00163 vtkImageData* GetIconImage();
00164
00165 // Description:
00166 // Load image with its associated Lookup Table
00167 vtkGetMacro(ApplyLookupTable,int);
00168 vtkSetMacro(ApplyLookupTable,int);
00169 vtkBooleanMacro(ApplyLookupTable,int);
00170
00171 // Description:
00172 // Load image as YBR
00173 vtkGetMacro(ApplyYBRToRGB,int);
00174 vtkSetMacro(ApplyYBRToRGB,int);
00175 vtkBooleanMacro(ApplyYBRToRGB,int);
00176
00177 // Description:
00178 // Return VTK_LUMINANCE, VTK_INVERSE_LUMINANCE, VTK_RGB, VTK_RGBA, VTK_LOOKUP_TABLE, VTK_YBR or VTK_CMYK
00179 // or 0 when ImageFormat is not handled.
00180 // Warning: For color image, PlanarConfiguration need to be taken into account.
00181 vtkGetMacro(ImageFormat,int);
00182
```

```

00183 // Description:
00184 // Return the Planar Configuration. This simply means that the internal DICOM image was stored
00185 // using a particular planar configuration (most of the time: 0)
00186 // For monochrome image, PlanarConfiguration is always 0
00187 vtkGetMacro(PlanarConfiguration,int);
00188
00189 // Description:
00190 // Return the 'raw' information stored in the DICOM file:
00191 // In case of a series of multiple files, only the first file is considered. The Image Orientation
(Patient)
00192 // is guaranteed to remain the same, and image Image Position (Patient) in other slice can be computed
00193 // using the ZSpacing (3rd dimension)
00194 // (0020,0032) DS [87.774866\~-182.908510\168.629671] # 32, 3 ImagePositionPatient
00195 // (0020,0037) DS [0.001479\0.999989\~-0.004376\~-0.002039\~-0.004372\~-0.999988] # 58, 6
ImageOrientationPatient
00196 vtkGetVector3Macro(ImagePositionPatient,double);
00197 vtkGetVector6Macro(ImageOrientationPatient,double);
00198
00199 // Description:
00200 // Set/Get the first Curve Data:
00201 vtkGetObjectMacro(Curve,vtkPolyData);
00202 virtual void SetCurve(vtkPolyData *pd);
00203
00204 // Description:
00205 // \DEPRECATED:
00206 // Modality LUT
00207 // Value returned by GetShift/GetScale might be inaccurate since Shift/Scale could be
00208 // varying along the Series read. Therefore user are advices not to use those functions
00209 // anymore
00210 vtkGetMacro(Shift,double);
00211 vtkGetMacro(Scale,double);
00212
00213 protected:
00214 vtkGDCMImageReader2();
00215 ~vtkGDCMImageReader2();
00216
00217 vtkSetVector6Macro(ImageOrientationPatient,double);
00218
00219 //BTX
00220 void FillMedicalImageInformation(const gdcm::ImageReader &reader);
00221 //ETX
00222 int RequestInformationCompat();
00223 int RequestDataCompat();
00224
00225 int ProcessRequest(vtkInformation* request,
00226                   vtkInformationVector** inputVector,
00227                   vtkInformationVector* outputVector);
00228 int RequestInformation(vtkInformation *request,
00229                       vtkInformationVector **inputVector,
00230                       vtkInformationVector *outputVector);
00231 int RequestData(vtkInformation *request,
00232                vtkInformationVector **inputVector,
00233                vtkInformationVector *outputVector);
00234
00235 protected:
00236 vtkMatrix4x4 *DirectionCosines;
00237 int LoadOverlays;
00238 int NumberOfOverlays;
00239 int LoadIconImage;
00240 int NumberOfIconImages;
00241 int IconImageDataExtent[6];
00242 double ImagePositionPatient[3];
00243 double ImageOrientationPatient[6];
00244 vtkPolyData *Curve;
00245
00246 int ImageFormat;
00247 // the following 3, should remain optional
00248 int ApplyInverseVideo;
00249 int ApplyLookupTable;
00250 int ApplyYBRToRGB;
00251 // I think that planar configuration need to always be applied as far as VTK is concerned
00252 int ApplyPlanarConfiguration;
00253 int ApplyShiftScale;
00254
00255 int LoadSingleFile(const char *filename, char *pointer, unsigned long &outlen);
00256
00257 double Shift;
00258 double Scale;
00259 int IconDataScalarType;
00260 int IconNumberOfScalarComponents;
00261 int PlanarConfiguration;

```

```

00262     int LossyFlag;
00263     int ForceRescale;
00264
00265 protected:
00266     // TODO / FIXME
00267     void SetFilePrefix(const char *) {}
00268     vtkGetStringMacro(FilePrefix);
00269     void SetFilePattern(const char *) {}
00270     vtkGetStringMacro(FilePattern);
00271
00272 private:
00273     vtkGDCMImageReader2(const vtkGDCMImageReader2&); // Not implemented.
00274     void operator=(const vtkGDCMImageReader2&); // Not implemented.
00275 };
00276 #endif

```

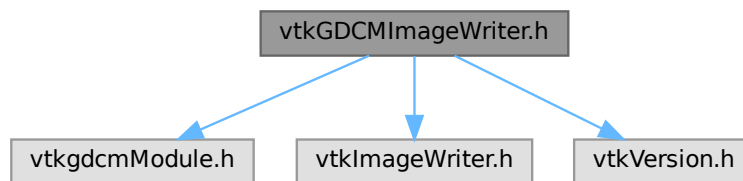
## 11.609 vtkGDCMImageWriter.h File Reference

```

#include "vtkgdcmModule.h"
#include "vtkImageWriter.h"
#include "vtkVersion.h"

```

Include dependency graph for vtkGDCMImageWriter.h:



### Classes

- class [vtkGDCMImageWriter](#)

## 11.610 vtkGDCMImageWriter.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 // .NAME vtkGDCMImageWriter - write DICOM files

```

```

00015 // .SECTION Description
00016 // vtkGDCMImageWriter is a sink object that write DICOM files
00017 // this writer is single threaded (see vtkGDCMThreadedImageReader2 for multi-thread)
00018 //
00019 // .SECTION Warning: vtkLookupTable from the vtkImageData object taken into account
00020 // only if ImageFormat is set to VTK_LOOKUP_TABLE
00021 //
00022 // .SECTION NOTE We are not using the usual API SetFilePrefix / SetFilePattern,
00023 // but instead a list of filenames: see SetFileNames and class gdcm::FilenameGenerator
00024 //
00025 // .SECTION Warning
00026 // You need to specify the correct ImageFormat (taken from the reader)
00027 // You need to explicitly specify the DirectionCosines (taken from the reader)
00028 // Since VTK 5.4 vtkMedicalImageProperties has its own DirectionCosine (no 's')
00029 // user need to make sure the vtkMatrix4x4 is compatible with the 6-vector DirectionCosine.
00030 //
00031 // .SECTION NOTE Shift/Scale are global to all DICOM frames (=files) written
00032 // as 2D slice, therefore the shift/scale operation might not be optimized for
00033 // all slices. This is not recommended for image with a large dynamic range.
00034 //
00035 // .SECTION See Also
00036 // vtkImageWriter vtkMedicalImageProperties vtkGDCMImageReader
00037
00038 #ifndef VTKGDCMIMAGEWRITER_H
00039 #define VTKGDCMIMAGEWRITER_H
00040
00041 #include "vtkgdcmModule.h"
00042 #include "vtkImageWriter.h"
00043 #include "vtkVersion.h"
00044
00045 class vtkLookupTable;
00046 class vtkMedicalImageProperties;
00047 class vtkMatrix4x4;
00048 class vtkStringArray;
00049 class VTKGDCM_EXPORT vtkGDCMImageWriter : public vtkImageWriter
00050 {
00051 public:
00052     static vtkGDCMImageWriter *New();
00053     vtkTypeMacro(vtkGDCMImageWriter,vtkImageWriter);
00054     virtual void PrintSelf(ostream& os, vtkIndent indent);
00055
00056     // Description:
00057     // Pass in the vtkmedicalimageproperties object for medical information
00058     // to be mapped to DICOM attributes.
00059     vtkGetObjectMacro(MedicalImageProperties, vtkMedicalImageProperties);
00060     virtual void SetMedicalImageProperties(vtkMedicalImageProperties*);
00061
00062     // Description:
00063     // Pass in the list of filename to be used to write out the DICOM file(s)
00064     virtual void SetFileNames(vtkStringArray*);
00065     vtkGetObjectMacro(FileNames, vtkStringArray);
00066
00067     // Description:
00068     // Set/Get whether or not the image was compressed using a lossy compression algorithm
00069     vtkGetMacro(LossyFlag,int);
00070     vtkSetMacro(LossyFlag,int);
00071     vtkBooleanMacro(LossyFlag,int);
00072
00073     // I need that...
00074     virtual void Write();
00075
00076     // Description:
00077     // Get the extension for this file format.
00078     virtual const char* GetFileExtensions() {
00079         return ".dcm .DCM"; }
00080
00081     // Description:
00082     // Get the name of this file format.
00083     virtual const char* GetDescriptiveName() {
00084         return "DICOM"; }
00085
00086     // Description:
00087     // You need to manually specify the direction the image is in to write a valid DICOM file
00088     // since vtkImageData do not contains one (eg. MR Image Storage, CT Image Storage...)
00089     virtual void SetDirectionCosines(vtkMatrix4x4 *matrix);
00090     vtkGetObjectMacro(DirectionCosines, vtkMatrix4x4);
00091     virtual void SetDirectionCosinesFromImageOrientationPatient(const double dircos[6]);
00092
00093     // Description:
00094     // Modality LUT
00095     vtkSetMacro(Shift, double);

```



```

00096     vtkGetMacro(Shift, double);
00097     vtkSetMacro(Scale, double);
00098     vtkGetMacro(Scale, double);
00099
00100     // Description:
00101     // See vtkGDCMImageReader for list of ImageFormat
00102     vtkGetMacro(ImageFormat,int);
00103     vtkSetMacro(ImageFormat,int);
00104
00105     // Description:
00106     // Set/Get whether the data comes from the file starting in the lower left
00107     // corner or upper left corner.
00108     vtkBooleanMacro(FileLowerLeft, int);
00109     vtkGetMacro(FileLowerLeft, int);
00110     vtkSetMacro(FileLowerLeft, int);
00111
00112     // Description:
00113     // For color image (more than a single comp) you can specify the planar configuration you prefer
00114     vtkSetMacro(PlanarConfiguration,int);
00115     vtkGetMacro(PlanarConfiguration,int);
00116
00117     // Description:
00118     // Set/Get specific StudyUID / SeriesUID
00119     vtkSetStringMacro(StudyUID);
00120     vtkGetStringMacro(StudyUID);
00121     vtkSetStringMacro(SeriesUID);
00122     vtkGetStringMacro(SeriesUID);
00123
00124 //BTX
00125     enum CompressionTypes {
00126         NO_COMPRESSION = 0,    // raw (default)
00127         JPEG_COMPRESSION,      // JPEG
00128         JPEG2000_COMPRESSION,  // J2K
00129         JPEGLS_COMPRESSION,    // JPEG-LS
00130         RLE_COMPRESSION        // RLE
00131     };
00132 //ETX
00133     // Set/Get the compression type
00134     vtkSetMacro(CompressionType, int);
00135     vtkGetMacro(CompressionType, int);
00136
00137     //void SetCompressionTypeFromString(const char *);
00138     //const char *GetCompressionTypeAsString();
00139
00140 protected:
00141     vtkGDCMImageWriter();
00142     ~vtkGDCMImageWriter();
00143
00144 #if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
00145     int FillInputPortInformation(int port, vtkInformation *info);
00146     int RequestInformation(
00147         vtkInformation *request,
00148         vtkInformationVector **inputVector,
00149         vtkInformationVector *outputVector);
00150     int RequestUpdateExtent(
00151         vtkInformation *request,
00152         vtkInformationVector **inputVector,
00153         vtkInformationVector *outputVector);
00154     int RequestData(
00155         vtkInformation *request,
00156         vtkInformationVector **inputVector,
00157         vtkInformationVector *outputVector);
00158 #else
00159     void WriteSlice(vtkImageData *data);
00160 #endif /*(VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )*/
00161     int WriteGDCMData(vtkImageData *data, int timeStep);
00162
00163 protected:
00164     virtual /*const*/ char *GetFileName();
00165
00166 private:
00167     vtkGDCMImageWriter(const vtkGDCMImageWriter&); // Not implemented.
00168     void operator=(const vtkGDCMImageWriter&); // Not implemented.
00169
00170     // VTK structs:
00171     //vtkLookupTable *LookupTable;
00172     vtkMedicalImageProperties *MedicalImageProperties;
00173     char *StudyUID;
00174     char *SeriesUID;
00175
00176     int DataUpdateExtent[6];

```

```

00177     int ImageFormat;
00178
00179     vtkStringArray *FileNames;
00180     vtkMatrix4x4 *DirectionCosines;
00181
00182     double Shift;
00183     double Scale;
00184     int FileLowerLeft;
00185     int PlanarConfiguration;
00186     int LossyFlag;
00187     int CompressionType;
00188 };
00189
00190 #endif

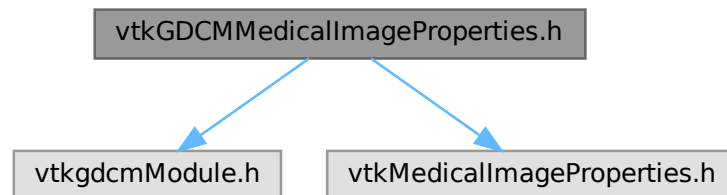
```

## 11.611 vtkGDCMMedicalImageProperties.h File Reference

```
#include "vtkgdcmModule.h"
```

```
#include "vtkMedicalImageProperties.h"
```

Include dependency graph for vtkGDCMMedicalImageProperties.h:



### Classes

- class [vtkGDCMMedicalImageProperties](#)

### Namespaces

- namespace [gdcm](#)

## 11.612 vtkGDCMMedicalImageProperties.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

```

```

00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 // .NAME vtkGDCMMedicalImageProperties - some medical image properties.
00015 // .SECTION Description
00016 // vtkGDCMMedicalImageProperties is a helper class that can be used by medical
00017 // image readers and applications to encapsulate medical image/acquisition
00018 // properties. Later on, this should probably be extended to add
00019 // any user-defined property.
00020 // .SECTION See Also
00021 // vtkMedicalImageReader2
00022
00023 #ifndef VTKGDCMMEDICALIMAGEPROPERTIES_H
00024 #define VTKGDCMMEDICALIMAGEPROPERTIES_H
00025
00026 #include "vtkgdcmmodule.h"
00027 #include "vtkMedicalImageProperties.h"
00028
00029 class vtkGDCMMedicalImagePropertiesInternals;
00030 //BTX
00031 namespace gdcmm { class File; }
00032 //ETX
00033
00034 class VTKGDCM_EXPORT vtkGDCMMedicalImageProperties : public vtkMedicalImageProperties
00035 {
00036 public:
00037     static vtkGDCMMedicalImageProperties *New();
00038     vtkTypeMacro(vtkGDCMMedicalImageProperties,vtkMedicalImageProperties);
00039     void PrintSelf(ostream& os, vtkIndent indent);
00040
00041     // Description:
00042     // Convenience method to reset all fields to an empty string/value
00043     virtual void Clear();
00044
00045     /*
00046     // Description:
00047     // Patient name
00048     // For ex: DICOM (0010,0010) = DOE,JOHN
00049     vtkSetStringMacro(PatientName);
00050     vtkGetStringMacro(PatientName);
00051
00052     // Description:
00053     // Patient ID
00054     // For ex: DICOM (0010,0020) = 1933197
00055     vtkSetStringMacro(PatientID);
00056     vtkGetStringMacro(PatientID);
00057
00058     // Description:
00059     // Patient age
00060     // Format: nnnD, nnW, nnnM or nnnY (eventually nnD, nnW, nnY)
00061     // with D (day), M (month), W (week), Y (year)
00062     // For ex: DICOM (0010,1010) = 031Y
00063     vtkSetStringMacro(PatientAge);
00064     vtkGetStringMacro(PatientAge);
00065
00066     // Description:
00067     // Take as input a string in VR=AS (DICOM PS3.5) and extract either
00068     // different fields namely: year month week day
00069     // Return 0 on error, 1 on success
00070     // One can test fields if they are different from -1 upon success
00071     static int GetAgeAsFields(const char *age, int &year, int &month, int &week, int &day);
00072
00073     // For Tcl:
00074     // From C++ use GetPatientAge + GetAgeAsField
00075     // Those function parse a DICOM string, and return the value of the number expressed
00076     // this is either expressed in year, month or days. Thus if a string is expressed in years
00077     // GetPatientAgeDay/GetPatientAgeWeek/GetPatientAgeMonth will return 0
00078     int GetPatientAgeYear();
00079     int GetPatientAgeMonth();
00080     int GetPatientAgeWeek();
00081     int GetPatientAgeDay();
00082
00083     // Description:
00084     // Patient sex
00085     // For ex: DICOM (0010,0040) = M
00086     vtkSetStringMacro(PatientSex);
00087     vtkGetStringMacro(PatientSex);
00088

```

```
00089 // Description:
00090 // Patient birth date
00091 // Format: yyyymmdd
00092 // For ex: DICOM (0010,0030) = 19680427
00093 vtkSetStringMacro(PatientBirthDate);
00094 vtkGetStringMacro(PatientBirthDate);
00095
00096 // For Tcl:
00097 // From C++ use GetPatientBirthDate + GetDateAsFields
00098 int GetPatientBirthDateYear();
00099 int GetPatientBirthDateMonth();
00100 int GetPatientBirthDateDay();
00101
00102 // Description:
00103 // Study Date
00104 // Format: yyyymmdd
00105 // For ex: DICOM (0008,0020) = 20030617
00106 vtkSetStringMacro(StudyDate);
00107 vtkGetStringMacro(StudyDate);
00108
00109 // Description:
00110 // Acquisition Date
00111 // Format: yyyymmdd
00112 // For ex: DICOM (0008,0022) = 20030617
00113 vtkSetStringMacro(AcquisitionDate);
00114 vtkGetStringMacro(AcquisitionDate);
00115
00116 // For Tcl:
00117 // From C++ use GetAcquisitionDate + GetDateAsFields
00118 int GetAcquisitionDateYear();
00119 int GetAcquisitionDateMonth();
00120 int GetAcquisitionDateDay();
00121
00122 // Description:
00123 // Study Time
00124 // Format: hhmmss.frac (any trailing component(s) can be omitted)
00125 // For ex: DICOM (0008,0030) = 162552.0705 or 230012, or 0012
00126 vtkSetStringMacro(StudyTime);
00127 vtkGetStringMacro(StudyTime);
00128
00129 // Description:
00130 // Acquisition time
00131 // Format: hhmmss.frac (any trailing component(s) can be omitted)
00132 // For ex: DICOM (0008,0032) = 162552.0705 or 230012, or 0012
00133 vtkSetStringMacro(AcquisitionTime);
00134 vtkGetStringMacro(AcquisitionTime);
00135
00136 // Description:
00137 // Image Date aka Content Date
00138 // Format: yyyymmdd
00139 // For ex: DICOM (0008,0023) = 20030617
00140 vtkSetStringMacro(ImageDate);
00141 vtkGetStringMacro(ImageDate);
00142
00143 // For Tcl:
00144 // From C++ use GetImageDate + GetDateAsFields
00145 int GetImageDateYear();
00146 int GetImageDateMonth();
00147 int GetImageDateDay();
00148
00149 // Description:
00150 // Take as input a string in ISO 8601 date (YYYY/MM/DD) and extract the
00151 // different fields namely: year month day
00152 // Return 0 on error, 1 on success
00153 static int GetDateAsFields(const char *date, int &year, int &month, int &day);
00154
00155 // Description:
00156 // Take as input a string in ISO 8601 date (YYYY/MM/DD) and construct a
00157 // locale date based on the different fields (see GetDateAsFields to extract
00158 // different fields)
00159 // Return 0 on error, 1 on success
00160 static int GetDateAsLocale(const char *date, char *locale);
00161
00162 // Description:
00163 // Image Time
00164 // Format: hhmmss.frac (any trailing component(s) can be omitted)
00165 // For ex: DICOM (0008,0033) = 162552.0705 or 230012, or 0012
00166 vtkSetStringMacro(ImageTime);
00167 vtkGetStringMacro(ImageTime);
00168
00169 // Description:
```

```
00170 // Image number
00171 // For ex: DICOM (0020,0013) = 1
00172 vtkSetStringMacro(ImageNumber);
00173 vtkGetStringMacro(ImageNumber);
00174
00175 // Description:
00176 // Series number
00177 // For ex: DICOM (0020,0011) = 902
00178 vtkSetStringMacro(SeriesNumber);
00179 vtkGetStringMacro(SeriesNumber);
00180
00181 // Description:
00182 // Series Description
00183 // User provided description of the Series
00184 // For ex: DICOM (0008,103e) = SCOUT
00185 vtkSetStringMacro(SeriesDescription);
00186 vtkGetStringMacro(SeriesDescription);
00187
00188 // Description:
00189 // Study ID
00190 // For ex: DICOM (0020,0010) = 37481
00191 vtkSetStringMacro(StudyID);
00192 vtkGetStringMacro(StudyID);
00193
00194 // Description:
00195 // Study description
00196 // For ex: DICOM (0008,1030) = BRAIN/C-SP/FACIAL
00197 vtkSetStringMacro(StudyDescription);
00198 vtkGetStringMacro(StudyDescription);
00199
00200 // Description:
00201 // Modality
00202 // For ex: DICOM (0008,0060)= CT
00203 vtkSetStringMacro(Modality);
00204 vtkGetStringMacro(Modality);
00205
00206 // Description:
00207 // Manufacturer
00208 // For ex: DICOM (0008,0070) = Siemens
00209 vtkSetStringMacro(Manufacturer);
00210 vtkGetStringMacro(Manufacturer);
00211
00212 // Description:
00213 // Manufacturer's Model Name
00214 // For ex: DICOM (0008,1090) = LightSpeed QX/i
00215 vtkSetStringMacro(ManufacturerModelName);
00216 vtkGetStringMacro(ManufacturerModelName);
00217
00218 // Description:
00219 // Station Name
00220 // For ex: DICOM (0008,1010) = LSPD_OC8
00221 vtkSetStringMacro(StationName);
00222 vtkGetStringMacro(StationName);
00223
00224 // Description:
00225 // Institution Name
00226 // For ex: DICOM (0008,0080) = FooCity Medical Center
00227 vtkSetStringMacro(InstitutionName);
00228 vtkGetStringMacro(InstitutionName);
00229
00230 // Description:
00231 // Convolution Kernel (or algorithm used to reconstruct the data)
00232 // For ex: DICOM (0018,1210) = Bone
00233 vtkSetStringMacro(ConvolutionKernel);
00234 vtkGetStringMacro(ConvolutionKernel);
00235
00236 // Description:
00237 // Slice Thickness (Nominal reconstructed slice thickness, in mm)
00238 // For ex: DICOM (0018,0050) = 0.273438
00239 vtkSetStringMacro(SliceThickness);
00240 vtkGetStringMacro(SliceThickness);
00241 virtual double GetSliceThicknessAsDouble();
00242
00243 // Description:
00244 // Peak kilo voltage output of the (x-ray) generator used
00245 // For ex: DICOM (0018,0060) = 120
00246 vtkSetStringMacro(KVP);
00247 vtkGetStringMacro(KVP);
00248
00249 // Description:
00250 // Gantry/Detector tilt (Nominal angle of tilt in degrees of the scanning
```

```

00251 // gantry.)
00252 // For ex: DICOM (0018,1120) = 15
00253 vtkSetStringMacro(GantryTilt);
00254 vtkGetStringMacro(GantryTilt);
00255 virtual double GetGantryTiltAsDouble();
00256
00257 // Description:
00258 // Echo Time
00259 // (Time in ms between the middle of the excitation pulse and the peak of
00260 // the echo produced)
00261 // For ex: DICOM (0018,0081) = 105
00262 vtkSetStringMacro(EchoTime);
00263 vtkGetStringMacro(EchoTime);
00264
00265 // Description:
00266 // Echo Train Length
00267 // (Number of lines in k-space acquired per excitation per image)
00268 // For ex: DICOM (0018,0091) = 35
00269 vtkSetStringMacro(EchoTrainLength);
00270 vtkGetStringMacro(EchoTrainLength);
00271
00272 // Description:
00273 // Repetition Time
00274 // The period of time in msec between the beginning of a pulse sequence and
00275 // the beginning of the succeeding (essentially identical) pulse sequence.
00276 // For ex: DICOM (0018,0080) = 2040
00277 vtkSetStringMacro(RepetitionTime);
00278 vtkGetStringMacro(RepetitionTime);
00279
00280 // Description:
00281 // Exposure time (time of x-ray exposure in msec)
00282 // For ex: DICOM (0018,1150) = 5
00283 vtkSetStringMacro(ExposureTime);
00284 vtkGetStringMacro(ExposureTime);
00285
00286 // Description:
00287 // X-ray tube current (in mA)
00288 // For ex: DICOM (0018,1151) = 400
00289 vtkSetStringMacro(XRayTubeCurrent);
00290 vtkGetStringMacro(XRayTubeCurrent);
00291
00292 // Description:
00293 // Exposure (The exposure expressed in mAs, for example calculated
00294 // from Exposure Time and X-ray Tube Current)
00295 // For ex: DICOM (0018,1152) = 114
00296 vtkSetStringMacro(Exposure);
00297 vtkGetStringMacro(Exposure);
00298
00299 // Interface to allow insertion of user define values, for instance in DICOM one would want to
00300 // store the Protocol Name (0018,1030), in this case one would do:
00301 // AddUserDefinedValue( "Protocol Name", "T1W/SE/1024" );
00302 void AddUserDefinedValue(const char *name, const char *value);
00303 // Get a particular user value
00304 const char *GetUserDefinedValue(const char *name);
00305 // Get the number of user defined values
00306 unsigned int GetNumberOfUserDefinedValues();
00307 // Get a name/value by index
00308 const char *GetUserDefinedNameByIndex(unsigned int idx);
00309 const char *GetUserDefinedValueByIndex(unsigned int idx);
00310
00311 // Description:
00312 // Copy the contents of p to this instance.
00313 virtual void DeepCopy(vtkGDCMMedicalImageProperties *p);
00314
00315 // Description:
00316 // Add/Remove/Query the window/level presets that may have been associated
00317 // to a medical image. Window is also known as 'width', level is also known
00318 // as 'center'. The same window/level pair can not be added twice.
00319 // As a convenience, a comment (aka Explanation) can be associated to a preset.
00320 // For ex: DICOM Window Center (0028,1050) = 00045\000470
00321 //           DICOM Window Width (0028,1051) = 0106\03412
00322 //           DICOM Window Center Width Explanation (0028,1055) = WINDOW1\WINDOW2
00323 virtual void AddWindowLevelPreset(double w, double l);
00324 virtual void RemoveWindowLevelPreset(double w, double l);
00325 virtual void RemoveAllWindowsLevelPresets();
00326 virtual int GetNumberOfWindowLevelPresets();
00327 virtual int HasWindowLevelPreset(double w, double l);
00328 virtual int GetNthWindowLevelPreset(int idx, double *w, double *l);
00329 virtual double* GetNthWindowLevelPreset(int idx);
00330 virtual void SetNthWindowLevelPresetComment(int idx, const char *comment);
00331 virtual const char* GetNthWindowLevelPresetComment(int idx);

```

```

00332
00333 // Description:
00334 // Mapping from a sliceidx within a volumeidx into a DICOM Instance UID
00335 // Some DICOM reader can populate this structure so that later on from a slice index
00336 // in a vtkImageData volume we can backtrack and find out which 2d slice it was coming from
00337 const char *GetInstanceUIDFromSliceID(int volumeidx, int sliceid);
00338 void SetInstanceUIDFromSliceID(int volumeidx, int sliceid, const char *uid);
00339
00340 // Description:
00341 // Provides the inverse mapping. Returns -1 if a slice for this uid is
00342 // not found.
00343 int GetSliceIDFromInstanceUID(int &volumeidx, const char *uid);
00344
00345 //BTX
00346 typedef enum {
00347     AXIAL = 0,
00348     CORONAL,
00349     SAGITTAL
00350 } OrientationType;
00351 //ETX
00352 int GetOrientationType(int volumeidx);
00353 void SetOrientationType(int volumeidx, int orientation);
00354 static const char *GetStringFromOrientationType(unsigned int type);
00355 */
00356 protected:
00357     vtkGDCMMedicalImageProperties();
00358     ~vtkGDCMMedicalImageProperties();
00359
00360 //BTX
00361 friend class vtkGDCMImageReader;
00362 friend class vtkGDCMImageReader2;
00363 friend class vtkGDCMImageWriter;
00364 void PushBackFile(gdcm::File const &f);
00365 gdcm::File const & GetFile(unsigned int t);
00366 //ETX
00367
00368 private:
00369     vtkGDCMMedicalImagePropertiesInternals *Internals;
00370
00371     vtkGDCMMedicalImageProperties(const vtkGDCMMedicalImageProperties&); // Not implemented.
00372     void operator=(const vtkGDCMMedicalImageProperties&); // Not implemented.
00373 };
00374
00375 #endif

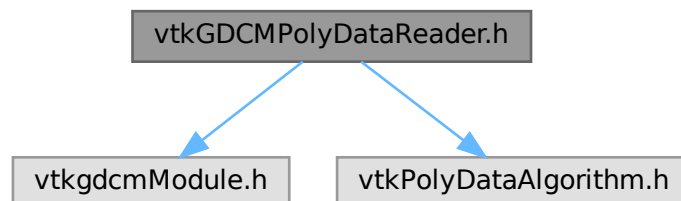
```

## 11.613 vtkGDCMPolyDataReader.h File Reference

```
#include "vtkgdcmModule.h"
```

```
#include "vtkPolyDataAlgorithm.h"
```

Include dependency graph for vtkGDCMPolyDataReader.h:



## Classes

- class [vtkGDCMPolyDataReader](#)

## Namespaces

- namespace [gdcmm](#)

# 11.614 vtkGDCMPolyDataReader.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 // .NAME vtkGDCMPolyDataReader - read DICOM PolyData files (Contour Data...)
00015 // .SECTION Description
00016 // For now only support RTSTRUCT (RT Structure Set Storage)
00017 // .SECTION TODO
00018 // Need to do the same job for DVH Sequence/DVH Data...
00019 // .SECTION Warning
00020 // When using vtkGDCMPolyDataReader in conjunction with vtkGDCMImageReader
00021 // it is *required* that FileLowerLeft is set to ON as coordinate system
00022 // would be inconsistent in between the two data structures.
00023 //
00024 // .SECTION See Also
00025 // vtkGDCMImageReader vtkGDCMPolyDataWriter vtkRTStructSetProperties
00026
00027
00028 #ifndef VTKGDCMPOLYDATAREADER_H
00029 #define VTKGDCMPOLYDATAREADER_H
00030
00031 #include "vtkgdcmModule.h"
00032 #include "vtkPolyDataAlgorithm.h"
00033
00034 class vtkMedicalImageProperties;
00035 class vtkRTStructSetProperties;
00036 //BTX
00037 namespace gdcmm { class Reader; }
00038 //ETX
00039 class VTKGDCM_EXPORT vtkGDCMPolyDataReader : public vtkPolyDataAlgorithm
00040 {
00041 public:
00042     static vtkGDCMPolyDataReader *New();
00043     vtkTypeMacro(vtkGDCMPolyDataReader, vtkPolyDataAlgorithm);
00044     virtual void PrintSelf(ostream& os, vtkIndent indent);
00045
00046     // Description:
00047     // Set/Get the filename of the file to be read
00048     vtkSetStringMacro(FileName);
00049     vtkGetStringMacro(FileName);
00050
00051     // Description:
00052     // Get the medical image properties object
00053     vtkGetObjectMacro(MedicalImageProperties, vtkMedicalImageProperties);
00054
00055     vtkGetObjectMacro(RTStructSetProperties, vtkRTStructSetProperties);
00056
00057 protected:
00058     vtkGDCMPolyDataReader();
00059     ~vtkGDCMPolyDataReader();

```



```

00060
00061     char *FileName;
00062     vtkMedicalImageProperties *MedicalImageProperties;
00063     vtkRTStructSetProperties *RTStructSetProperties;
00064     //BTX
00065     void FillMedicalImageInformation(const gdcm::Reader &reader);
00066     //ETX
00067
00068     int RequestData(vtkInformation *, vtkInformationVector **, vtkInformationVector *);
00069     int RequestInformation(
00070         vtkInformation *vtkNotUsed(request),
00071         vtkInformationVector **vtkNotUsed(inputVector),
00072         vtkInformationVector *outputVector);
00073     //BTX
00074     int RequestInformation_RTStructureSetStorage(gdcm::Reader const & reader);
00075     int RequestData_RTStructureSetStorage(gdcm::Reader const &reader, vtkInformationVector *outputVector);
00076     int RequestInformation_HemodynamicWaveformStorage(gdcm::Reader const & reader);
00077     int RequestData_HemodynamicWaveformStorage(gdcm::Reader const &reader, vtkInformationVector
00078         *outputVector);
00078     //ETX
00079
00080 private:
00081     vtkGDCMPolyDataReader(const vtkGDCMPolyDataReader&); // Not implemented.
00082     void operator=(const vtkGDCMPolyDataReader&); // Not implemented.
00083 };
00084
00085 #endif

```

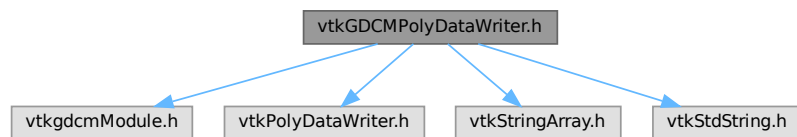
## 11.615 vtkGDCMPolyDataWriter.h File Reference

```

#include "vtkgdcmModule.h"
#include "vtkPolyDataWriter.h"
#include "vtkStringArray.h"
#include "vtkStdString.h"

```

Include dependency graph for vtkGDCMPolyDataWriter.h:



### Classes

- class [vtkGDCMPolyDataWriter](#)

### Namespaces

- namespace [gdcm](#)

## 11.616 vtkGDCMPolyDataWriter.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  // .NAME vtkGDCMPolyDataWriter - writer DICOM PolyData files (Contour Data...)
00015  // .SECTION Description
00016  // For now only support RTSTRUCT (RT Structure Set Storage)
00017  // .SECTION TODO
00018  // Need to do the same job for DVH Sequence/DVH Data...
00019  // .SECTION Warning
00020  //
00021  // .SECTION See Also
00022  // vtkGDCMImageReader vtkGDCMPolyDataReader vtkRTStructSetProperties
00023
00024
00025  #ifndef VTKGDCMPOLYDATAWRITER_H
00026  #define VTKGDCMPOLYDATAWRITER_H
00027
00028  #include "vtkgdcmModule.h"
00029  #include "vtkPolyDataWriter.h"
00030  #include "vtkStringArray.h"
00031  #include "vtkStdString.h"
00032
00033
00034  class vtkMedicalImageProperties;
00035  class vtkRTStructSetProperties;
00036  //BTX
00037  namespace gdcms { class File; }
00038  //ETX
00039  class VTKGDCM_EXPORT vtkGDCMPolyDataWriter : public vtkPolyDataWriter
00040  {
00041  public:
00042      static vtkGDCMPolyDataWriter *New();
00043      vtkTypeMacro(vtkGDCMPolyDataWriter, vtkPolyDataWriter);
00044      virtual void PrintSelf(ostream& os, vtkIndent indent);
00045
00046      // Description:
00047      // Set/Get the filename of the file to be read
00048      // vtkSetStringMacro(Filename);
00049      // vtkGetStringMacro(Filename);
00050
00051      // Description:
00052      // Get the medical image properties object
00053      // vtkGetObjectMacro(MedicalImageProperties, vtkMedicalImageProperties);
00054      virtual void SetMedicalImageProperties(vtkMedicalImageProperties *pd);
00055
00056      virtual void SetRTStructSetProperties(vtkRTStructSetProperties *pd);
00057
00058
00059      //this function will initialize the contained rtstructset with
00060      //the inputs of the writer and the various extra information
00061      //necessary for writing a complete rtstructset.
00062      //NOTE: inputs must be set BEFORE calling this function!
00063      //NOTE: the number of outputs for the appendpolydata MUST MATCH the ROI vectors!
00064      void InitializeRTStructSet(vtkStdString inDirectory,
00065                               vtkStdString inStructLabel, vtkStdString inStructName,
00066                               vtkStringArray* inROINames,
00067                               vtkStringArray* inROIAlgorithmName,
00068                               vtkStringArray* inROIType);
00069
00070      // make parent class public...
00071      void SetNumberOfInputPorts(int n);
00072
00073  protected:
00074      vtkGDCMPolyDataWriter();
00075      ~vtkGDCMPolyDataWriter();
00076

```

```

00077   vtkMedicalImageProperties *MedicalImageProperties;
00078   vtkRTStructSetProperties *RTStructSetProperties;
00079
00080   void WriteData();
00081   //BTX
00082   void WriteRTSTRUCTInfo(gdcm::File &file);
00083   void WriteRTSTRUCTData(gdcm::File &file, int num);
00084   //ETX
00085
00086 private:
00087   vtkGDCMPolyDataWriter(const vtkGDCMPolyDataWriter&); // Not implemented.
00088   void operator=(const vtkGDCMPolyDataWriter&); // Not implemented.
00089 };
00090
00091 #endif

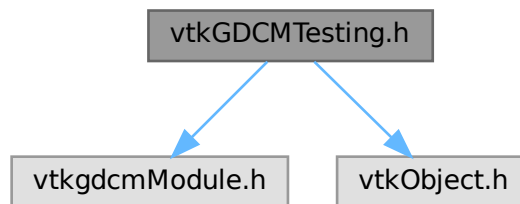
```

## 11.617 vtkGDCMTesting.h File Reference

```
#include "vtkgdcmModule.h"
```

```
#include "vtkObject.h"
```

Include dependency graph for vtkGDCMTesting.h:



### Classes

- class [vtkGDCMTesting](#)

## 11.618 vtkGDCMTesting.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/

```

```

00014 // .NAME vtkGDCMTesting - GDCM Testing
00015 // .SECTION Description
00016 // GDCM Testing
00017
00018 // .SECTION See Also
00019 // vtkTesting
00020
00021 #ifndef VTKGDCMTESTING_H
00022 #define VTKGDCMTESTING_H
00023
00024 #include "vtkgdcmModule.h"
00025 #include "vtkObject.h"
00026
00027 class VTKGDCM_EXPORT vtkGDCMTesting : public vtkObject
00028 {
00029 public:
00030     static vtkGDCMTesting *New();
00031     vtkTypeMacro(vtkGDCMTesting,vtkObject);
00032     void PrintSelf(ostream& os, vtkIndent indent);
00033
00034     static const char *GetVTKDataRoot();
00035     static const char *GetGDCMDataRoot();
00036
00037 //BTX
00038     typedef const char* const (*MD5MetaImagesType)[3];
00039     static const char * const * GetMD5MetaImage(unsigned int file);
00040 //ETX
00041     static unsigned int GetNumberOfMD5MetaImages();
00042
00043     static const char * GetMHDMD5FromFile(const char *filepath);
00044     static const char * GetRAWMD5FromFile(const char *filepath);
00045
00046 protected:
00047     vtkGDCMTesting();
00048     ~vtkGDCMTesting();
00049
00050 private:
00051     vtkGDCMTesting(const vtkGDCMTesting&); // Not implemented.
00052     void operator=(const vtkGDCMTesting&); // Not implemented.
00053 };
00054
00055 #endif

```

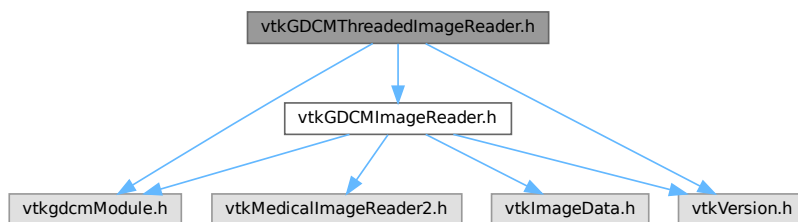
## 11.619 vtkGDCMThreadedImageReader.h File Reference

```

#include "vtkgdcmModule.h"
#include "vtkGDCMImageReader.h"
#include "vtkVersion.h"

```

Include dependency graph for vtkGDCMThreadedImageReader.h:



### Classes

- class [vtkGDCMThreadedImageReader](#)

## 11.620 vtkGDCMThreadedImageReader.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 // .NAME vtkGDCMThreadedImageReader - read DICOM files with multiple threads
00015 // .SECTION Description
00016 // vtkGDCMThreadedImageReader is a source object that reads some DICOM files
00017 // This reader is threaded. Meaning that on a multiple core CPU with N cpu, it will
00018 // read approx N times faster than when reading in a single thread.
00019 //
00020 // .SECTION Warning: Advanced users only. Do not use this class in the general case,
00021 // you have to understand how physically medium works first (sequential reading for
00022 // instance) before playing with this class
00023 //
00024 // .SECTION Implementation note: when FileLowerLeft is set to on the image is not flipped
00025 // upside down as VTK would expect, use this option only if you know what you are doing
00026 //
00027 // .SECTION FIXME: need to implement the other mode where FileLowerLeft is set to OFF
00028 //
00029 // .SECTION FIXME: you need to call SetFileName when reading a volume file (multiple slices DICOM)
00030 // since SetFileNames expect each single file to be single slice (see parent class)
00031 //
00032 // .SECTION BUG: you should really consider using vtkGDCMThreadedImageReader2 instead !
00033 //
00034 // .SECTION See Also
00035 // vtkMedicalImageReader2 vtkMedicalImageProperties vtkGDCMThreadedImageReader2
00036
00037 #ifndef VTKGDCMTHREADEDIMAGEREADER_H
00038 #define VTKGDCMTHREADEDIMAGEREADER_H
00039
00040 #include "vtkgdcmModule.h"
00041 #include "vtkGDCMImageReader.h"
00042 #include "vtkVersion.h"
00043
00044 class VTKGDCM_EXPORT vtkGDCMThreadedImageReader : public vtkGDCMImageReader
00045 {
00046 public:
00047     static vtkGDCMThreadedImageReader *New();
00048     vtkTypeMacro(vtkGDCMThreadedImageReader,vtkGDCMImageReader);
00049     virtual void PrintSelf(ostream& os, vtkIndent indent);
00050
00051     // Description:
00052     // Explicitly set the Rescale Intercept (0028,1052)
00053     vtkSetMacro(Shift,double);
00054
00055     // Description:
00056     // Explicitly get/set the Rescale Slope (0028,1053)
00057     vtkSetMacro(Scale,double);
00058
00059     // Description:
00060     // Determine whether or not reader should use value from Shift/Scale
00061     // Default is 1
00062     vtkSetMacro(UseShiftScale,int);
00063     vtkGetMacro(UseShiftScale,int);
00064     vtkBooleanMacro(UseShiftScale,int);
00065
00066     // Within this class this is allowed to set the Number of Overlays from outside
00067     //vtkSetMacro(NumberOfOverlays,int);
00068
00069 protected:
00070     vtkGDCMThreadedImageReader();
00071     ~vtkGDCMThreadedImageReader();
00072
00073 #if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
00074     int RequestInformation(vtkInformation *request,
00075                           vtkInformationVector **inputVector,
00076                           vtkInformationVector *outputVector);

```

```

00077   int RequestData(vtkInformation *request,
00078                   vtkInformationVector **inputVector,
00079                   vtkInformationVector *outputVector);
00080 #else /*(VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )*/
00081   void ExecuteInformation();
00082   void ExecuteData(vtkDataObject *out);
00083 #endif /*(VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )*/
00084
00085   void ReadFiles(unsigned int nfiles, const char *filenames[]);
00086   void RequestDataCompat();
00087
00088 private:
00089   vtkGDCMThreadedImageReader(const vtkGDCMThreadedImageReader&); // Not implemented.
00090   void operator=(const vtkGDCMThreadedImageReader&); // Not implemented.
00091
00092   int UseShiftScale;
00093 };
00094
00095 #endif

```

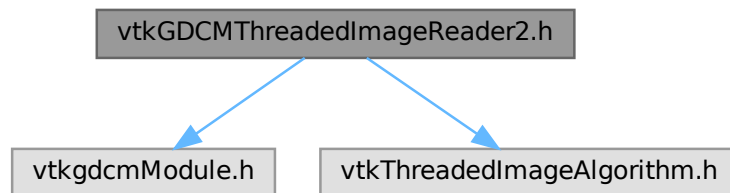
## 11.621 vtkGDCMThreadedImageReader2.h File Reference

```

#include "vtkgdcmModule.h"
#include "vtkThreadedImageAlgorithm.h"

```

Include dependency graph for vtkGDCMThreadedImageReader2.h:



### Classes

- class [vtkGDCMThreadedImageReader2](#)

## 11.622 vtkGDCMThreadedImageReader2.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR

```

```

00011     PURPOSE.  See the above copyright notice for more information.
00012
00013     =====*/
00014 // .NAME vtkGDCMThreadedImageReader2 - read DICOM files with multiple threads
00015 // .SECTION Description
00016 // vtkGDCMThreadedImageReader2 is a source object that reads some DICOM files
00017 // This reader is threaded. Meaning that on a multiple core CPU with N cpu, it will
00018 // read approx N times faster than when reading in a single thread assuming the IO is
00019 // not a bottleneck operation.
00020 // If looking for a single threaded class see: vtkGDCMImageReader
00021 //
00022 // .SECTION Warning: Advanced users only. Do not use this class in the general case,
00023 // you have to understand how physically medium works first (sequential reading for
00024 // instance) before playing with this class
00025 //
00026 // .SECTION Implementation note: when FileLowerLeft is set to on the image is not flipped
00027 // upside down as VTK would expect, use this option only if you know what you are doing
00028 //
00029 // .SECTION FIXME: need to implement the other mode where FileLowerLeft is set to OFF
00030 //
00031 // .SECTION FIXME: need to implement reading of series of 3D files
00032 //
00033 // .SECTION Implementation note: this class is meant to supersede vtkGDCMThreadedImageReader
00034 // because it had support for ProgressEvent support even from python layer. There is a
00035 // subtle trick down in the threading mechanism in VTK were the main thread (talking to the
00036 // python interpreter) is also part of the execution process (and the N-1 other thread
00037 // are just there to execute the remaining of ThreadedRequestData), this separation into
00038 // two types of thread is necessary to achieve a working implementation of UpdateProgress
00039 //
00040 // .SECTION See Also
00041 // vtkMedicalImageReader2 vtkMedicalImageProperties vtkGDCMImageReader
00042
00043 #ifndef VTKGDCMTHREADEDIMAGEREADER2_H
00044 #define VTKGDCMTHREADEDIMAGEREADER2_H
00045
00046 #include "vtkgdcModule.h"
00047 #include "vtkThreadedImageAlgorithm.h"
00048
00049 class vtkStringArray;
00050 class VTKGDCM_EXPORT vtkGDCMThreadedImageReader2 : public vtkThreadedImageAlgorithm
00051 {
00052 public:
00053     static vtkGDCMThreadedImageReader2 *New();
00054     vtkTypeMacro(vtkGDCMThreadedImageReader2,vtkThreadedImageAlgorithm);
00055     virtual void PrintSelf(ostream& os, vtkIndent indent);
00056
00057     vtkGetMacro(FileLowerLeft,int);
00058     vtkSetMacro(FileLowerLeft,int);
00059     vtkBooleanMacro(FileLowerLeft,int);
00060
00061     vtkGetMacro(NumberOfOverlays,int);
00062
00063     vtkSetMacro(DataScalarType,int);
00064     vtkGetMacro(DataScalarType,int);
00065
00066     vtkSetMacro(NumberOfScalarComponents,int);
00067     vtkGetMacro(NumberOfScalarComponents,int);
00068
00069     vtkGetMacro(LoadOverlays,int);
00070     vtkSetMacro(LoadOverlays,int);
00071     vtkBooleanMacro(LoadOverlays,int);
00072
00073     vtkSetVector6Macro(DataExtent,int);
00074     vtkGetVector6Macro(DataExtent,int);
00075
00076     vtkSetVector3Macro(DataOrigin,double);
00077     vtkGetVector3Macro(DataOrigin,double);
00078
00079     vtkSetVector3Macro(DataSpacing,double);
00080     vtkGetVector3Macro(DataSpacing,double);
00081
00082     //vtkGetStringMacro(FileName);
00083     //vtkSetStringMacro(FileName);
00084     virtual const char *GetFileName(int i = 0);
00085     virtual void SetFileName(const char *filename);
00086
00087     virtual void SetFileNames(vtkStringArray*);
00088     vtkGetObjectMacro(FileNames, vtkStringArray);
00089
00090     int SplitExtent(int splitExt[6], int startExt[6],
00091                     int num, int total);

```

```

00092
00093 // Description:
00094 // Explicitly set the Rescale Intercept (0028,1052)
00095 vtkSetMacro(Shift,double);
00096 vtkGetMacro(Shift,double);
00097
00098 // Description:
00099 // Explicitly get/set the Rescale Slope (0028,1053)
00100 vtkSetMacro(Scale,double);
00101 vtkGetMacro(Scale,double);
00102
00103 // Description:
00104 // Determine whether or not reader should use value from Shift/Scale
00105 // Default is 1
00106 vtkSetMacro(UseShiftScale,int);
00107 vtkGetMacro(UseShiftScale,int);
00108 vtkBooleanMacro(UseShiftScale,int);
00109
00110 protected:
00111   vtkGDCMThreadedImageReader2();
00112   ~vtkGDCMThreadedImageReader2();
00113
00114   int RequestInformation(vtkInformation *request,
00115                         vtkInformationVector **inputVector,
00116                         vtkInformationVector *outputVector);
00117
00118 protected:
00119   void ThreadedRequestData (
00120     vtkInformation * request,
00121     vtkInformationVector** inputVector,
00122     vtkInformationVector * outputVector,
00123     vtkImageData **inData,
00124     vtkImageData **outData,
00125     int outExt[6], int id);
00126
00127 private:
00128   int FileLowerLeft;
00129   char *FileName;
00130   vtkStringArray *FileNames;
00131   int LoadIconImage;
00132   int DataExtent[6];
00133   int LoadOverlays;
00134   int NumberOfOverlays;
00135   int DataScalarType;
00136
00137   int NumberOfScalarComponents;
00138   double DataSpacing[3];
00139   double DataOrigin[3];
00140   int IconImageDataExtent[6];
00141
00142   double Shift;
00143   double Scale;
00144   int UseShiftScale;
00145
00146 private:
00147   vtkGDCMThreadedImageReader2(const vtkGDCMThreadedImageReader2&); // Not implemented.
00148   void operator=(const vtkGDCMThreadedImageReader2&); // Not implemented.
00149 };
00150
00151 #endif

```

## 11.623 vtkImageColorViewer.h File Reference

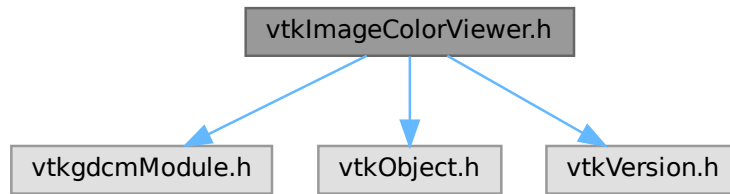
```

#include "vtkgdcmModule.h"
#include "vtkObject.h"
#include "vtkVersion.h"

```



Include dependency graph for vtkImageColorViewer.h:



## Classes

- class [vtkImageColorViewer](#)

## 11.624 vtkImageColorViewer.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 // .NAME vtkImageColorViewer - Display a 2D image.
00015 // .SECTION Description
00016 // vtkImageColorViewer is a convenience class for displaying a 2D image. It
00017 // packages up the functionality found in vtkRenderWindow, vtkRenderer,
00018 // vtkImageActor and vtkImageMapToWindowLevelColors into a single easy to use
00019 // class. This class also creates an image interactor style
00020 // (vtkInteractorStyleImage) that allows zooming and panning of images, and
00021 // supports interactive window/level operations on the image. Note that
00022 // vtkImageColorViewer is simply a wrapper around these classes.
00023 //
00024 // vtkImageColorViewer uses the 3D rendering and texture mapping engine
00025 // to draw an image on a plane. This allows for rapid rendering,
00026 // zooming, and panning. The image is placed in the 3D scene at a
00027 // depth based on the z-coordinate of the particular image slice. Each
00028 // call to SetSlice() changes the image data (slice) displayed AND
00029 // changes the depth of the displayed slice in the 3D scene. This can
00030 // be controlled by the AutoAdjustCameraClippingRange ivar of the
00031 // InteractorStyle member.
00032 //
00033 // It is possible to mix images and geometry, using the methods:
00034 //
00035 // viewer->SetInput( myImage );
00036 // viewer->GetRenderer()->AddActor( myActor );
00037 //
00038 // This can be used to annotate an image with a PolyData of "edges" or
00039 // or highlight sections of an image or display a 3D isosurface
00040 // with a slice from the volume, etc. Any portions of your geometry
00041 // that are in front of the displayed slice will be visible; any

```

```

00042 // portions of your geometry that are behind the displayed slice will
00043 // be obscured. A more general framework (with respect to viewing
00044 // direction) for achieving this effect is provided by the
00045 // vtkImagePlaneWidget .
00046 //
00047 // Note that pressing 'r' will reset the window/level and pressing
00048 // shift+'r' or control+'r' will reset the camera.
00049 //
00050 // .SECTION See Also
00051 // vtkRenderWindow vtkRenderer vtkImageActor vtkImageMapToWindowLevelColors
00052
00053 #ifndef VTKIMAGECOLORVIEWER_H
00054 #define VTKIMAGECOLORVIEWER_H
00055
00056 #include "vtkgdcmModule.h"
00057 #include "vtkObject.h"
00058 #include "vtkVersion.h"
00059
00060 class vtkAlgorithm;
00061 class vtkAlgorithmOutput;
00062 class vtkImageActor;
00063 class vtkImageData;
00064 class vtkImageMapToWindowLevelColors2;
00065 class vtkInformation;
00066 class vtkInteractorStyleImage;
00067 class vtkRenderWindow;
00068 class vtkRenderer;
00069 class vtkRenderWindowInteractor;
00070 class vtkPolyData;
00071
00072 class VTKGDCM_EXPORT vtkImageColorViewer : public vtkObject
00073 {
00074 public:
00075     static vtkImageColorViewer *New();
00076     vtkTypeMacro(vtkImageColorViewer,vtkObject);
00077     void PrintSelf(ostream& os, vtkIndent indent);
00078
00079     // Description:
00080     // Get the name of rendering window.
00081     virtual const char *GetWindowName();
00082
00083     // Description:
00084     // Render the resulting image.
00085     virtual void Render(void);
00086
00087     // Description:
00088     // Set/Get the input image to the viewer.
00089     #if (VTK_MAJOR_VERSION >= 6)
00090     virtual void SetInputData(vtkImageData *in);
00091     #else
00092     virtual void SetInput(vtkImageData *in);
00093     #endif
00094     virtual vtkImageData *GetInput();
00095     virtual void SetInputConnection(vtkAlgorithmOutput* input);
00096     virtual void AddInputConnection(vtkAlgorithmOutput* input);
00097     virtual void AddInput(vtkImageData * input);
00098     //virtual void AddInput(vtkPolyData * input);
00099
00100     double GetOverlayVisibility();
00101     void SetOverlayVisibility(double vis);
00102
00103     // Description:
00104     // Set/get the slice orientation
00105     //BTX
00106     enum
00107     {
00108         SLICE_ORIENTATION_YZ = 0,
00109         SLICE_ORIENTATION_XZ = 1,
00110         SLICE_ORIENTATION_XY = 2
00111     };
00112     //ETX
00113     vtkGetMacro(SliceOrientation, int);
00114     virtual void SetSliceOrientation(int orientation);
00115     virtual void SetSliceOrientationToXY()
00116     { this->SetSliceOrientation(vtkImageColorViewer::SLICE_ORIENTATION_XY); };
00117     virtual void SetSliceOrientationToYZ()
00118     { this->SetSliceOrientation(vtkImageColorViewer::SLICE_ORIENTATION_YZ); };
00119     virtual void SetSliceOrientationToXZ()
00120     { this->SetSliceOrientation(vtkImageColorViewer::SLICE_ORIENTATION_XZ); };
00121
00122     // Description:

```

```

00123 // Set/Get the current slice to display (depending on the orientation
00124 // this can be in X, Y or Z).
00125 vtkGetMacro(Slice, int);
00126 virtual void SetSlice(int s);
00127
00128 // Description:
00129 // Update the display extent manually so that the proper slice for the
00130 // given orientation is displayed. It will also try to set a
00131 // reasonable camera clipping range.
00132 // This method is called automatically when the Input is changed, but
00133 // most of the time the input of this class is likely to remain the same,
00134 // i.e. connected to the output of a filter, or an image reader. When the
00135 // input of this filter or reader itself is changed, an error message might
00136 // be displayed since the current display extent is probably outside
00137 // the new whole extent. Calling this method will ensure that the display
00138 // extent is reset properly.
00139 virtual void UpdateDisplayExtent();
00140
00141 // Description:
00142 // Return the minimum and maximum slice values (depending on the orientation
00143 // this can be in X, Y or Z).
00144 virtual int GetSliceMin();
00145 virtual int GetSliceMax();
00146 virtual void GetSliceRange(int range[2])
00147 { this->GetSliceRange(range[0], range[1]); }
00148 virtual void GetSliceRange(int &min, int &max);
00149 virtual int* GetSliceRange();
00150
00151 // Description:
00152 // Set window and level for mapping pixels to colors.
00153 virtual double GetColorWindow();
00154 virtual double GetColorLevel();
00155 virtual void SetColorWindow(double s);
00156 virtual void SetColorLevel(double s);
00157
00158 // Description:
00159 // These are here when using a Tk window.
00160 virtual void SetDisplayId(void *a);
00161 virtual void SetWindowId(void *a);
00162 virtual void SetParentId(void *a);
00163
00164 // Description:
00165 // Set/Get the position in screen coordinates of the rendering window.
00166 virtual int* GetPosition();
00167 virtual void SetPosition(int a, int b);
00168 virtual void SetPosition(int a[2]) { this->SetPosition(a[0], a[1]); }
00169
00170 // Description:
00171 // Set/Get the size of the window in screen coordinates in pixels.
00172 virtual int* GetSize();
00173 virtual void SetSize(int a, int b);
00174 virtual void SetSize(int a[2]) { this->SetSize(a[0], a[1]); }
00175
00176 // Description:
00177 // Get the internal render window, renderer, image actor, and
00178 // image map instances.
00179 vtkGetObjectMacro(RenderWindow, vtkRenderWindow);
00180 vtkGetObjectMacro(Renderer, vtkRenderer);
00181 vtkGetObjectMacro(ImageActor, vtkImageActor);
00182 vtkGetObjectMacro(WindowLevel, vtkImageMapToWindowLevelColors2);
00183 vtkGetObjectMacro(InteractorStyle, vtkInteractorStyleImage);
00184
00185 // Description:
00186 // Set your own renderwindow and renderer
00187 virtual void SetRenderWindow(vtkRenderWindow *arg);
00188 virtual void SetRenderer(vtkRenderer *arg);
00189
00190 // Description:
00191 // Attach an interactor for the internal render window.
00192 virtual void SetupInteractor(vtkRenderWindowInteractor*);
00193
00194 // Description:
00195 // Create a window in memory instead of on the screen. This may not
00196 // be supported for every type of window and on some windows you may
00197 // need to invoke this prior to the first render.
00198 virtual void SetOffScreenRendering(int);
00199 virtual int GetOffScreenRendering();
00200 vtkBooleanMacro(OffScreenRendering, int);
00201
00202 protected:
00203     vtkImageColorViewer();

```

```

00204 ~vtkImageColorViewer();
00205
00206 virtual void InstallPipeline();
00207 virtual void UnInstallPipeline();
00208
00209 vtkImageMapToWindowLevelColors2 *WindowLevel;
00210 vtkRenderWindow *RenderWindow;
00211 vtkRenderer *Renderer;
00212 vtkImageActor *ImageActor;
00213 vtkImageActor *OverlayImageActor;
00214 vtkRenderWindowInteractor *Interactor;
00215 vtkInteractorStyleImage *InteractorStyle;
00216
00217 int SliceOrientation;
00218 int FirstRender;
00219 int Slice;
00220
00221 virtual void UpdateOrientation();
00222
00223 #if (VTK_MAJOR_VERSION >= 6)
00224   vtkAlgorithm* GetInputAlgorithm();
00225   vtkInformation* GetInputInformation();
00226 #endif
00227
00228 friend class vtkImageColorViewerCallback;
00229
00230 private:
00231   vtkImageColorViewer(const vtkImageColorViewer&); // Not implemented.
00232   void operator=(const vtkImageColorViewer&); // Not implemented.
00233 };
00234
00235 #endif

```

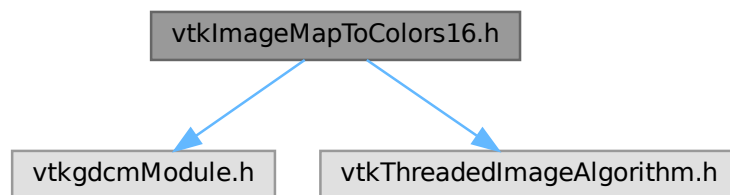
## 11.625 vtkImageMapToColors16.h File Reference

```

#include "vtkgdcmModule.h"
#include "vtkThreadedImageAlgorithm.h"

```

Include dependency graph for vtkImageMapToColors16.h:



### Classes

- class [vtkImageMapToColors16](#)

## 11.626 vtkImageMapToColors16.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 /*=====
00015
00016 Portions of this file are subject to the VTK Toolkit Version 3 copyright.
00017
00018 Program: Visualization Toolkit
00019 Module: $RCSfile: vtkImageMapToColors16.h,v $
00020
00021 Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
00022 All rights reserved.
00023 See Copyright.txt or http://www.kitware.com/Copyright.htm for details.
00024
00025 This software is distributed WITHOUT ANY WARRANTY; without even
00026 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00027 PURPOSE. See the above copyright notice for more information.
00028
00029 =====*/
00030 // .NAME vtkImageMapToColors16 - map the input image through a lookup table
00031 // .SECTION Description
00032 // The vtkImageMapToColors16 filter will take an input image of any valid
00033 // scalar type, and map the first component of the image through a
00034 // lookup table. The result is an image of type VTK_UNSIGNED_CHAR.
00035 // If the lookup table is not set, or is set to NULL, then the input
00036 // data will be passed through if it is already of type VTK_UNSIGNED_CHAR.
00037
00038 // .SECTION See Also
00039 // vtkLookupTable vtkScalarsToColors
00040
00041 #ifndef VTKIMAGEMAPTOCOLORS16_H
00042 #define VTKIMAGEMAPTOCOLORS16_H
00043
00044
00045 #include "vtkgdcmModule.h"
00046 #include "vtkThreadedImageAlgorithm.h"
00047
00048 class vtkScalarsToColors;
00049
00050 class VTKGDCM_EXPORT vtkImageMapToColors16 : public vtkThreadedImageAlgorithm
00051 {
00052 public:
00053     static vtkImageMapToColors16 *New();
00054     vtkTypeMacro(vtkImageMapToColors16,vtkThreadedImageAlgorithm);
00055     void PrintSelf(ostream& os, vtkIndent indent);
00056
00057     // Description:
00058     // Set the lookup table.
00059     virtual void SetLookupTable(vtkScalarsToColors*);
00060     vtkGetObjectMacro(LookupTable,vtkScalarsToColors);
00061
00062     // Description:
00063     // Set the output format, the default is RGBA.
00064     vtkSetMacro(OutputFormat,int);
00065     vtkGetMacro(OutputFormat,int);
00066     void SetOutputFormatToRGBA() { this->OutputFormat = VTK_RGBA; };
00067     void SetOutputFormatToRGB() { this->OutputFormat = VTK_RGB; };
00068     void SetOutputFormatToLuminanceAlpha() { this->OutputFormat = VTK_LUMINANCE_ALPHA; };
00069     void SetOutputFormatToLuminance() { this->OutputFormat = VTK_LUMINANCE; };
00070
00071     // Description:
00072     // Set the component to map for multi-component images (default: 0)
00073     vtkSetMacro(ActiveComponent,int);
00074     vtkGetMacro(ActiveComponent,int);
00075
00076     // Description:

```

```

00077 // Use the alpha component of the input when computing the alpha component
00078 // of the output (useful when converting monochrome+alpha data to RGBA)
00079 vtkSetMacro(PassAlphaToOutput,int);
00080 vtkBooleanMacro(PassAlphaToOutput,int);
00081 vtkGetMacro(PassAlphaToOutput,int);
00082
00083 // Description:
00084 // We need to check the modified time of the lookup table too.
00085 #ifdef VTK_HAS_MTIME_TYPE
00086     virtual vtkMTimeType GetMTime();
00087 #else
00088     virtual unsigned long GetMTime();
00089 #endif
00090
00091 protected:
00092     vtkImageMapToColors16();
00093     ~vtkImageMapToColors16();
00094
00095     virtual int RequestInformation (vtkInformation *, vtkInformationVector **, vtkInformationVector *);
00096
00097     void ThreadedRequestData(vtkInformation *request,
00098                             vtkInformationVector **inputVector,
00099                             vtkInformationVector *outputVector,
00100                             vtkImageData **inData, vtkImageData **outData,
00101                             int extent[6], int id);
00102
00103     virtual int RequestData(vtkInformation *request,
00104                             vtkInformationVector **inputVector,
00105                             vtkInformationVector *outputVector);
00106
00107     vtkScalarsToColors *LookupTable;
00108     int OutputFormat;
00109
00110     int ActiveComponent;
00111     int PassAlphaToOutput;
00112
00113     int DataWasPassed;
00114 private:
00115     vtkImageMapToColors16(const vtkImageMapToColors16&); // Not implemented.
00116     void operator=(const vtkImageMapToColors16&); // Not implemented.
00117 };
00118
00119 #endif

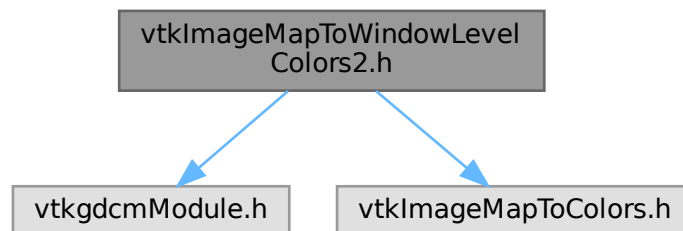
```

## 11.627 vtkImageMapToWindowLevelColors2.h File Reference

```
#include "vtkgdcmModule.h"
```

```
#include "vtkImageMapToColors.h"
```

Include dependency graph for vtkImageMapToWindowLevelColors2.h:



## Classes

- class [vtkImageMapToWindowLevelColors2](#)

## 11.628 vtkImageMapToWindowLevelColors2.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 /*=====
00015
00016   Portions of this file are subject to the VTK Toolkit Version 3 copyright.
00017
00018   Program: Visualization Toolkit
00019   Module:   $RCSfile: vtkImageMapToWindowLevelColors2.h,v $
00020
00021   Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
00022   All rights reserved.
00023   See Copyright.txt or http://www.kitware.com/Copyright.htm for details.
00024
00025   This software is distributed WITHOUT ANY WARRANTY; without even
00026   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00027   PURPOSE. See the above copyright notice for more information.
00028
00029   =====*/
00030 // .NAME vtkImageMapToWindowLevelColors2 - map the input image through a lookup table and window / level
00031 // it
00032 // .SECTION Description
00033 // The vtkImageMapToWindowLevelColors2 filter will take an input image of any
00034 // valid scalar type, and map the first component of the image through a
00035 // lookup table. This resulting color will be modulated with value obtained
00036 // by a window / level operation. The result is an image of type
00037 // VTK_UNSIGNED_CHAR. If the lookup table is not set, or is set to NULL, then
00038 // the input data will be passed through if it is already of type
00039 // UNSIGNED_CHAR.
00040 // .SECTION See Also
00041 // vtkLookupTable vtkScalarsToColors
00042
00043 #ifndef VTKIMAGEMAPTOWINDOWLEVELCOLORS2_H
00044 #define VTKIMAGEMAPTOWINDOWLEVELCOLORS2_H
00045
00046 #include "vtkgdcModule.h"
00047 #include "vtkImageMapToColors.h"
00048
00049 class VTKGDCM_EXPORT vtkImageMapToWindowLevelColors2 : public vtkImageMapToColors
00050 {
00051 public:
00052   static vtkImageMapToWindowLevelColors2 *New();
00053   vtkTypeMacro(vtkImageMapToWindowLevelColors2,vtkImageMapToColors);
00054   void PrintSelf(ostream& os, vtkIndent indent);
00055
00056   // Description:
00057   // Set / Get the Window to use -> modulation will be performed on the
00058   // color based on (S - (L - W/2))/W where S is the scalar value, L is
00059   // the level and W is the window.
00060   vtkSetMacro( Window, double );
00061   vtkGetMacro( Window, double );
00062
00063   // Description:
00064   // Set / Get the Level to use -> modulation will be performed on the
00065   // color based on (S - (L - W/2))/W where S is the scalar value, L is
00066   // the level and W is the window.

```

```

00067     vtkSetMacro( Level, double );
00068     vtkGetMacro( Level, double );
00069
00070 protected:
00071     vtkImageMapToWindowLevelColors2();
00072     ~vtkImageMapToWindowLevelColors2();
00073
00074     virtual int RequestInformation( vtkInformation *, vtkInformationVector **, vtkInformationVector * );
00075     void ThreadedRequestData( vtkInformation *request,
00076                             vtkInformationVector **inputVector,
00077                             vtkInformationVector *outputVector,
00078                             vtkImageData **inData, vtkImageData **outData,
00079                             int extent[6], int id );
00080     virtual int RequestData( vtkInformation *request,
00081                             vtkInformationVector **inputVector,
00082                             vtkInformationVector *outputVector );
00083
00084     double Window;
00085     double Level;
00086
00087 private:
00088     vtkImageMapToWindowLevelColors2(const vtkImageMapToWindowLevelColors2&); // Not implemented.
00089     void operator=(const vtkImageMapToWindowLevelColors2&); // Not implemented.
00090 };
00091
00092 #endif

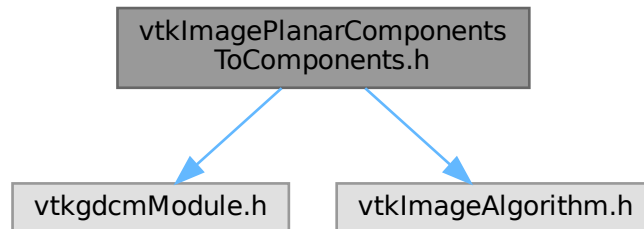
```

## 11.629 vtkImagePlanarComponentsToComponents.h File Reference

```
#include "vtkgdcmModule.h"
```

```
#include "vtkImageAlgorithm.h"
```

Include dependency graph for vtkImagePlanarComponentsToComponents.h:



### Classes

- class [vtkImagePlanarComponentsToComponents](#)

## 11.630 vtkImagePlanarComponentsToComponents.h

[Go to the documentation of this file.](#)



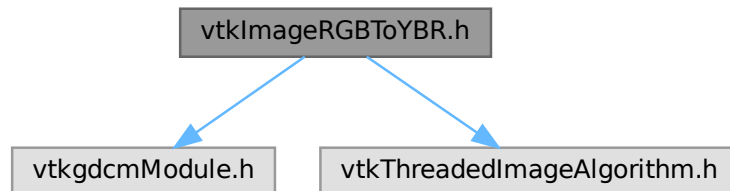
```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 /*=====
00015
00016   Portions of this file are subject to the VTK Toolkit Version 3 copyright.
00017
00018   Program: Visualization Toolkit
00019   Module:   $RCSfile: vtkImagePlanarComponentsToComponents.h,v $
00020
00021   Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
00022   All rights reserved.
00023   See Copyright.txt or http://www.kitware.com/Copyright.htm for details.
00024
00025   This software is distributed WITHOUT ANY WARRANTY; without even
00026   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00027   PURPOSE. See the above copyright notice for more information.
00028
00029 =====*/
00030 // .NAME vtkImagePlanarComponentsToComponents - Converts planar comp to pixel comp
00031 // .SECTION Description
00032
00033 // .SECTION See Also
00034 // TODO: Can I make this filter threaded ?
00035 // TODO: How do I handle the VTK-flipping (FileLowerLeft)?
00036
00037 #ifndef VTKIMAGEPLANARCOMPONENTSTOCOMPONENTS_H
00038 #define VTKIMAGEPLANARCOMPONENTSTOCOMPONENTS_H
00039
00040 #include "vtkgdcModule.h"
00041 #include "vtkImageAlgorithm.h"
00042
00043 // everything is now handled within the vtkGDCMImageReader as Planar Configuration can not
00044 // be externalized (conflict with file lower left)
00045
00046 #error do not use this class
00047
00048 //class VTKGDCM_EXPORT vtkImagePlanarComponentsToComponents : public vtkThreadedImageAlgorithm
00049 class VTKGDCM_EXPORT vtkImagePlanarComponentsToComponents : public vtkImageAlgorithm
00050 {
00051 public:
00052   static vtkImagePlanarComponentsToComponents *New();
00053   //vtkTypeMacro(vtkImagePlanarComponentsToComponents,vtkThreadedImageAlgorithm);
00054   vtkTypeMacro(vtkImagePlanarComponentsToComponents,vtkImageAlgorithm);
00055
00056   void PrintSelf(ostream& os, vtkIndent indent);
00057
00058 protected:
00059   vtkImagePlanarComponentsToComponents();
00060   ~vtkImagePlanarComponentsToComponents() {};
00061
00062   // void ThreadedExecute (vtkImageData *inData, vtkImageData *outData,
00063   // int ext[6], int id);
00064   // virtual int RequestInformation (vtkInformation *, vtkInformationVector**, vtkInformationVector *);
00065   virtual int RequestData(vtkInformation *, vtkInformationVector **, vtkInformationVector *);
00066
00067 private:
00068   vtkImagePlanarComponentsToComponents(const vtkImagePlanarComponentsToComponents&); // Not implemented.
00069   void operator=(const vtkImagePlanarComponentsToComponents&); // Not implemented.
00070 };
00071
00072 #endif

```

## 11.631 vtkImageRGBToYBR.h File Reference

```
#include "vtkgdcmModule.h"
#include "vtkThreadedImageAlgorithm.h"
Include dependency graph for vtkImageRGBToYBR.h:
```



### Classes

- class [vtkImageRGBToYBR](#)

## 11.632 vtkImageRGBToYBR.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 /*=====
00015
00016   Portions of this file are subject to the VTK Toolkit Version 3 copyright.
00017
00018   Program: Visualization Toolkit
00019   Module:   $RCSfile: vtkImageRGBToYBR.h,v $
00020
00021   Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
00022   All rights reserved.
00023   See Copyright.txt or http://www.kitware.com/Copyright.htm for details.
00024
00025   This software is distributed WITHOUT ANY WARRANTY; without even
00026   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00027   PURPOSE. See the above copyright notice for more information.
00028
00029  =====*/
00030 // .NAME vtkImageRGBToYBR - Converts YBR components to RGB.
00031 // .SECTION Description
00032 // For each pixel with hue, saturation and value components this filter
00033 // outputs the color coded as red, green, blue. Output type must be the same

```

```

00034 // as input type.
00035
00036 // .SECTION See Also
00037 // vtkImageRGBToHSV
00038
00039 #ifndef VTKIMAGERGBTOYBR_H
00040 #define VTKIMAGERGBTOYBR_H
00041
00042 #include "vtkgdcmModule.h"
00043 #include "vtkThreadedImageAlgorithm.h"
00044
00045 class VTKGDCM_EXPORT vtkImageRGBToYBR : public vtkThreadedImageAlgorithm
00046 {
00047 public:
00048     static vtkImageRGBToYBR *New();
00049     vtkTypeMacro(vtkImageRGBToYBR,vtkThreadedImageAlgorithm);
00050
00051     void PrintSelf(ostream& os, vtkIndent indent);
00052
00053 protected:
00054     vtkImageRGBToYBR();
00055     ~vtkImageRGBToYBR() {};
00056
00057     void ThreadedExecute (vtkImageData *inData, vtkImageData *outData,
00058                          int ext[6], int id);
00059 private:
00060     vtkImageRGBToYBR(const vtkImageRGBToYBR&); // Not implemented.
00061     void operator=(const vtkImageRGBToYBR&); // Not implemented.
00062 };
00063
00064 #endif

```

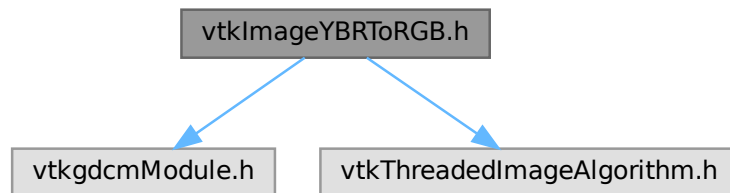
## 11.633 vtkImageYBRToRGB.h File Reference

```

#include "vtkgdcmModule.h"
#include "vtkThreadedImageAlgorithm.h"

```

Include dependency graph for vtkImageYBRToRGB.h:



### Classes

- class [vtkImageYBRToRGB](#)

## 11.634 vtkImageYBRToRGB.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 /*=====
00015
00016   Portions of this file are subject to the VTK Toolkit Version 3 copyright.
00017
00018   Program: Visualization Toolkit
00019   Module:   $RCSfile: vtkImageYBRToRGB.h,v $
00020
00021   Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
00022   All rights reserved.
00023   See Copyright.txt or http://www.kitware.com/Copyright.htm for details.
00024
00025   This software is distributed WITHOUT ANY WARRANTY; without even
00026   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00027   PURPOSE. See the above copyright notice for more information.
00028
00029   =====*/
00030 // .NAME vtkImageYBRToRGB - Converts YBR components to RGB.
00031 // .SECTION Description
00032 // For each pixel with hue, saturation and value components this filter
00033 // outputs the color coded as red, green, blue. Output type must be the same
00034 // as input type.
00035
00036 // .SECTION See Also
00037 // vtkImageRGBToHSV
00038
00039 #ifndef VTKIMAGEYBRTORGB_H
00040 #define VTKIMAGEYBRTORGB_H
00041
00042 #include "vtkgdcmModule.h"
00043 #include "vtkThreadedImageAlgorithm.h"
00044
00045 class VTKGDCM_EXPORT vtkImageYBRToRGB : public vtkThreadedImageAlgorithm
00046 {
00047 public:
00048   static vtkImageYBRToRGB *New();
00049   vtkTypeMacro(vtkImageYBRToRGB,vtkThreadedImageAlgorithm);
00050
00051   void PrintSelf(ostream& os, vtkIndent indent);
00052
00053 protected:
00054   vtkImageYBRToRGB();
00055   ~vtkImageYBRToRGB() {};
00056
00057   void ThreadedExecute (vtkImageData *inData, vtkImageData *outData,
00058                        int ext[6], int id);
00059 private:
00060   vtkImageYBRToRGB(const vtkImageYBRToRGB&); // Not implemented.
00061   void operator=(const vtkImageYBRToRGB&); // Not implemented.
00062 };
00063
00064 #endif

```

## 11.635 vtkLookupTable16.h File Reference

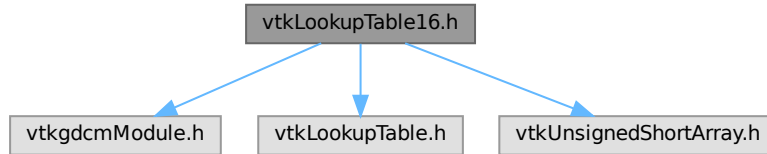
```

#include "vtkgdcmModule.h"
#include "vtkLookupTable.h"

```

```
#include "vtkUnsignedShortArray.h"
```

Include dependency graph for vtkLookupTable16.h:



## Classes

- class [vtkLookupTable16](#)

## 11.636 vtkLookupTable16.h

[Go to the documentation of this file.](#)

```

00001 /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 /*=====
00015
00016   Portions of this file are subject to the VTK Toolkit Version 3 copyright.
00017
00018   Program: Visualization Toolkit
00019   Module:   $RCSfile: vtkLookupTable16.h,v $
00020
00021   Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
00022   All rights reserved.
00023   See Copyright.txt or http://www.kitware.com/Copyright.htm for details.
00024
00025   This software is distributed WITHOUT ANY WARRANTY; without even
00026   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00027   PURPOSE. See the above copyright notice for more information.
00028
00029   =====*/
00030 // .NAME vtkLookupTable16 -
00031 // .SECTION Description
00032 //
00033 // .SECTION Caveats
00034 //
00035 // .SECTION See Also
00036 // vtkLookupTable
00037
00038 #ifndef VTKLOOKUPTABLE16_H
00039 #define VTKLOOKUPTABLE16_H
00040
00041 #include "vtkgdcmModule.h"
00042 #include "vtkLookupTable.h"
00043 #include "vtkUnsignedShortArray.h"

```

```

00044
00045 class VTKGDCM_EXPORT vtkLookupTable16 : public vtkLookupTable
00046 {
00047 public:
00048     static vtkLookupTable16 *New();
00049
00050     vtkTypeMacro(vtkLookupTable16,vtkLookupTable);
00051     void PrintSelf(ostream& os, vtkIndent indent);
00052
00053     void Build();
00054
00055     void SetNumberOfTableValues(vtkIdType number);
00056
00057     unsigned char *WritePointer(const vtkIdType id, const int number);
00058
00059     unsigned short *GetPointer(const vtkIdType id) {
00060         return this->Table16->GetPointer(4*id); };
00061
00062 protected:
00063     vtkLookupTable16(int size=256, int ext=256);
00064     ~vtkLookupTable16();
00065
00066     vtkUnsignedShortArray *Table16;
00067
00068     void MapScalarsThroughTable2(void *input,
00069                                 unsigned char *output,
00070                                 int inputDataType,
00071                                 int numberOfValues,
00072                                 int inputIncrement,
00073                                 int outputFormat);
00074
00075 private:
00076     vtkLookupTable16(const vtkLookupTable16&); // Not implemented.
00077     void operator=(const vtkLookupTable16&); // Not implemented.
00078 };
00079
00080 //-----
00081 inline unsigned char *vtkLookupTable16::WritePointer(const vtkIdType id,
00082                                                       const int number)
00083 {
00084     //this->InsertTime.Modified();
00085     return (unsigned char*)this->Table16->WritePointer(4*id,4*number);
00086 }
00087
00088 #endif

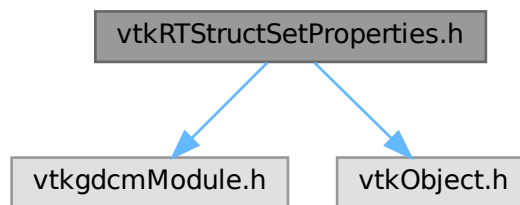
```

## 11.637 vtkRTStructSetProperties.h File Reference

```
#include "vtkgdcmModule.h"
```

```
#include "vtkObject.h"
```

Include dependency graph for vtkRTStructSetProperties.h:



**Classes**

- class [vtkRTStructSetProperties](#)

**11.638 vtkRTStructSetProperties.h**

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  // .NAME vtkRTStructSetProperties - some rtstruct properties.
00015  // .SECTION Description
00016  //
00017  // .SECTION See Also
00018  // vtkGDCMPolyDataReader vtkGDCMPolyDataWriter
00019
00020  #ifndef VTKRTSTRUCTSETPROPERTIES_H
00021  #define VTKRTSTRUCTSETPROPERTIES_H
00022
00023  #include "vtkgdcModule.h"
00024  #include "vtkObject.h"
00025
00026  class vtkRTStructSetPropertiesInternals;
00027
00028  class VTKGDCM_EXPORT vtkRTStructSetProperties : public vtkObject
00029  {
00030  public:
00031      static vtkRTStructSetProperties *New();
00032      vtkTypeMacro(vtkRTStructSetProperties,vtkObject);
00033      void PrintSelf(ostream& os, vtkIndent indent);
00034
00035      // Description:
00036      // Convenience method to reset all fields to an empty string/value
00037      virtual void Clear();
00038
00039      // Description:
00040      //
00041      vtkSetStringMacro(StructureSetLabel);
00042      vtkGetStringMacro(StructureSetLabel);
00043
00044      vtkSetStringMacro(StructureSetName);
00045      vtkGetStringMacro(StructureSetName);
00046
00047      vtkSetStringMacro(StructureSetDate);
00048      vtkGetStringMacro(StructureSetDate);
00049
00050      vtkSetStringMacro(StructureSetTime);
00051      vtkGetStringMacro(StructureSetTime);
00052
00053      vtkSetStringMacro(SOPInstanceUID);
00054      vtkGetStringMacro(SOPInstanceUID);
00055
00056      vtkSetStringMacro(StudyInstanceUID);
00057      vtkGetStringMacro(StudyInstanceUID);
00058
00059      vtkSetStringMacro(SeriesInstanceUID);
00060      vtkGetStringMacro(SeriesInstanceUID);
00061
00062      vtkSetStringMacro(ReferenceSeriesInstanceUID);
00063      vtkGetStringMacro(ReferenceSeriesInstanceUID);
00064
00065      vtkSetStringMacro(ReferenceFrameOfReferenceUID);
00066      vtkGetStringMacro(ReferenceFrameOfReferenceUID);
00067

```

```

00068 // Description:
00069 // Copy the contents of p to this instance.
00070 virtual void DeepCopy(vtkRTStructSetProperties *p);
00071
00072 void AddContourReferencedFrameOfReference( vtkIdType pdnum, const char *classuid , const char *
instanceuid );
00073 const char *GetContourReferencedFrameOfReferenceClassUID( vtkIdType pdnum, vtkIdType id );
00074 const char *GetContourReferencedFrameOfReferenceInstanceUID( vtkIdType pdnum, vtkIdType id );
00075 vtkIdType GetNumberOfContourReferencedFrameOfReferences();
00076 vtkIdType GetNumberOfContourReferencedFrameOfReferences(vtkIdType pdnum);
00077
00078 void AddReferencedFrameOfReference( const char *classuid , const char * instanceuid );
00079 const char *GetReferencedFrameOfReferenceClassUID( vtkIdType id );
00080 const char *GetReferencedFrameOfReferenceInstanceUID( vtkIdType id );
00081 vtkIdType GetNumberOfReferencedFrameOfReferences();
00082
00083 void AddStructureSetROI( int roinumber,
00084     const char* refframerefuid,
00085     const char* roiname,
00086     const char* ROIGenerationAlgorithm,
00087     const char* ROIDescription = 0
00088 );
00089 void AddStructureSetROIObservation( int refnumber,
00090     int observationnumber,
00091     const char *rtroiinterpretedtype,
00092     const char *roiinterpreter,
00093     const char *roiobservationlabel = 0
00094 );
00095
00096 vtkIdType GetNumberOfStructureSetROIs();
00097 int GetStructureSetObservationNumber(vtkIdType id);
00098 int GetStructureSetROINumber(vtkIdType id);
00099 const char *GetStructureSetROIRefFrameRefUID(vtkIdType);
00100 const char *GetStructureSetROIName(vtkIdType);
00101 const char *GetStructureSetROIGenerationAlgorithm(vtkIdType);
00102 const char *GetStructureSetROIDescription(vtkIdType id);
00103 const char *GetStructureSetRTROIInterpretedType(vtkIdType id);
00104 const char *GetStructureSetROIObservationLabel(vtkIdType id);
00105
00106 protected:
00107     vtkRTStructSetProperties();
00108     ~vtkRTStructSetProperties();
00109
00110     char *StructureSetLabel;
00111     char *StructureSetName;
00112     char *StructureSetDate;
00113     char *StructureSetTime;
00114
00115     char *SOPInstanceUID;
00116     char *StudyInstanceUID;
00117     char *SeriesInstanceUID;
00118
00119     char *ReferenceSeriesInstanceUID;
00120     char *ReferenceFrameOfReferenceUID;
00121
00122 // Description:
00123 // PIMPL Encapsulation for STL containers
00124 //BTX
00125     vtkRTStructSetPropertiesInternals *Internals;
00126 //ETX
00127
00128 private:
00129     vtkRTStructSetProperties(const vtkRTStructSetProperties&); // Not implemented.
00130     void operator=(const vtkRTStructSetProperties&); // Not implemented.
00131 };
00132
00133 #endif

```

## 11.639 gdcmPythonFilter.h File Reference

```

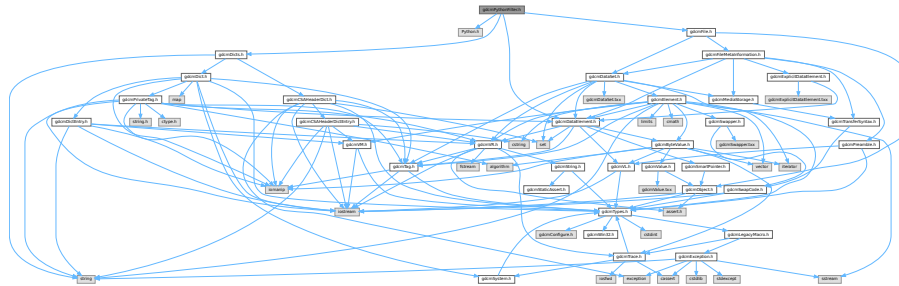
#include <Python.h>
#include "gdcmDataElement.h"
#include "gdcmDicts.h"

```



```
#include "gdcmFile.h"
```

Include dependency graph for gdcmPythonFilter.h:



## Classes

- class [gdcm::PythonFilter](#)

*PythonFilter PythonFilter* is the class that make *gdcm2.x* looks more like *gdcm1* and transform the binary blob contained in a *DataElement* into a string, typically this is a nice feature to have for wrapped language.

## Namespaces

- namespace [gdcm](#)

## 11.640 gdcmPythonFilter.h

[Go to the documentation of this file.](#)

```
00001 /*=====
00002
00003     Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005     Copyright (c) 2006-2011 Mathieu Malaterre
00006     All rights reserved.
00007     See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 #ifndef GDCMPYTHONFILTER_H
00015 #define GDCMPYTHONFILTER_H
00016
00017 #include <Python.h>
00018
00019 #include "gdcmDataElement.h"
00020 #include "gdcmDicts.h"
00021 #include "gdcmFile.h"
00022
00023 namespace gdcm
00024 {
00025
00031 class GDCM_EXPORT PythonFilter
00032 {
00033 public:
00034     PythonFilter();
00035     ~PythonFilter();
00036
00037     void UseDictAlways(bool ) {}
00038 }
```

```
00038
00039 // Allow user to pass in there own dicts
00040 void SetDicts(const Dicts &dicts);
00041
00042 // Convert to string the ByteValue contained in a DataElement
00043 PyObject *ToPyObject(const Tag& t) const;
00044
00045 void SetFile(const File& f);
00046 File &GetFile();
00047 const File &GetFile() const;
00048
00049 private:
00050     SmartPointer<File> F;
00051 };
00052
00053 } // end namespace gdcms
00054
00055 #endif //GDCMPYTHONFILTER_H
```

# Chapter 12

## Examples

### 12.1 TestByteSwap.cxx

This is a C++ example on how to use [gdcm::ByteSwap](#)

```
/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmTypes.h"
#include "gdcmSwapCode.h"
#include "gdcmByteSwap.h"

#include <cstring> // memcpy

int myfunc()
{
    char vl_str[4];
    const char raw[] = "\000\000\000\004";
    memcpy(vl_str, raw, 4);
    uint32_t vl;
    memcpy(&vl, vl_str, 4);
    gdcm::ByteSwap<uint32_t>::SwapRangeFromSwapCodeIntoSystem(&vl, gdcm::SwapCode::BigEndian, 1);
    if( vl != 0x00000004 )
    {
        std::cerr << std::hex << "vl: " << vl << std::endl;
        return 1;
    }

    gdcm::ByteSwap<uint32_t>::SwapFromSwapCodeIntoSystem(vl, gdcm::SwapCode::LittleEndian);
    if( vl != 0x00000004 )
    {
        std::cerr << std::hex << "vl: " << vl << std::endl;
        return 1;
    }

    gdcm::ByteSwap<uint32_t>::SwapFromSwapCodeIntoSystem(vl, gdcm::SwapCode::BigEndian);
    if( vl != 0x40000000 )
    {
        std::cerr << std::hex << "vl: " << vl << std::endl;
        return 1;
    }

    return 0;
}
```

```

}

int TestByteSwap(int , char *[])
{
    gdcm::SwapCode sc = gdcm::SwapCode::Unknown;
    if ( gdcm::ByteSwap<uint16_t>::SystemIsBigEndian() )
    {
        sc = gdcm::SwapCode::BigEndian;
    }
    else if ( gdcm::ByteSwap<uint16_t>::SystemIsLittleEndian() )
    {
        sc = gdcm::SwapCode::LittleEndian;
    }
    if( sc == gdcm::SwapCode::Unknown )
    {
        std::cerr << "unk" << std::endl;
        return 1;
    }

    //std::cout << "sc: " << sc << std::endl;

    uint16_t t = 0x1234;
    gdcm::ByteSwap<uint16_t>::SwapFromSwapCodeIntoSystem(t, sc);
    if( sc == gdcm::SwapCode::BigEndian )
    {
        if( t != 0x3412 )
        {
            std::cerr << std::hex << "t: " << t << std::endl;
            return 1;
        }
        // ok test pass rest value to old one
        t = 0x1234;
    }
    else if ( sc == gdcm::SwapCode::LittleEndian )
    {
        if( t != 0x1234 )
        {
            std::cerr << std::hex << "t: " << t << std::endl;
            return 1;
        }
    }
}

union { char n[2]; uint16_t tn; } ul6;
memcpy(ul6.n, &t, 2 );
gdcm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem(&ul6.tn, sc, 1);
uint16_t tn = ul6.tn;
if( sc == gdcm::SwapCode::BigEndian )
{
    if( tn != 0x3412 )
    {
        std::cerr << std::hex << "tn: " << tn << std::endl;
        return 1;
    }
    // ok test pass rest value to old one
    t = 0x1234;
}
else if ( sc == gdcm::SwapCode::LittleEndian )
{
    if( tn != 0x1234 )
    {
        std::cerr << std::hex << "tn: " << tn << std::endl;
        return 1;
    }
}
gdcm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem(&ul6.tn, gdcm::SwapCode::BigEndian, 1);
tn = ul6.tn;
if( sc == gdcm::SwapCode::LittleEndian )
{
    if( tn != 0x3412 )
    {
        std::cerr << std::hex << "tn: " << tn << std::endl;
        return 1;
    }
}
else if ( sc == gdcm::SwapCode::BigEndian )
{
    if( tn != 0x1234 )
    {
        std::cerr << std::hex << "tn: " << tn << std::endl;
        return 1;
    }
}

```

```

    }

    if( myfunc() )
    {
        return 1;
    }

    uint16_t array[] = { 0x1234 };
    gdcm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem(array,
        gdcm::SwapCode::BigEndian,1);
    if ( array[0] != 0x3412 )
    {
        std::cerr << std::hex << "array: " << array[0] << std::endl;
        return 1;
    }

    return 0;
}

```

## 12.2 PatchFile.cxx

This is a C++ example on how to use [gdcm::Attribute](#)

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * The image was a broken file where the Pixel Data element was 8 times too big
 * Apparently multiplying the BitsAllocated to 4 and multiplying the number of
 * frames by 2 would solve the problem
 *
 * This C++ code can be used to patch the header.
 */

#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmWriter.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        return 1;
    }
    const char *f = argv[1];
    const char *out = argv[2];
    gdcm::Reader r;
    r.SetFileName( f );
    if( !r.Read() )
    {
        return 1;
    }

    gdcm::File &file = r.GetFile();
    gdcm::DataSet& ds = file.GetDataSet();
    // (0028,0100) US 16 # 2, 1 BitsAllocated
    // (0028,0101) US 16 # 2, 1 BitsStored
    // (0028,0102) US 15 # 2, 1 HighBit
    //
    {
        gdcm::Attribute<0x28,0x100> at;
        at.SetFromDataElement( ds.GetDataElement( at.GetTag() ) );
        if( at.GetValue() != 8 )
        {

```

```

        return 1;
    }
    at.SetValue( 32 );
    ds.Replace( at.GetAsDataElement() );
}
{
    gdcm::Attribute<0x28,0x101> at;
    at.SetFromDataElement( ds.GetDataElement( at.GetTag() ) );
    if( at.GetValue() != 8 )
    {
        return 1;
    }
    at.SetValue( 32 );
    ds.Replace( at.GetAsDataElement() );
}
{
    gdcm::Attribute<0x28,0x102> at;
    at.SetFromDataElement( ds.GetDataElement( at.GetTag() ) );
    if( at.GetValue() != 7 )
    {
        return 1;
    }
    at.SetValue( 31 );
    ds.Replace( at.GetAsDataElement() );
}
// (0028,0008) IS [56] # 2, 1 NumberOfFrames

{
    gdcm::Attribute<0x28,0x8> at;
    at.SetFromDataElement( ds.GetDataElement( at.GetTag() ) );
    at.SetValue( at.GetValue() * 2 );
    ds.Replace( at.GetAsDataElement() );
}

gdcm::Writer w;
w.SetFile( file );
w.SetCheckFileMetaInformation( false );
w.SetFileName( out );
if( !w.Write() )
{
    return 1;
}

// Now let's see if we can read it as an image:
gdcm::ImageReader ir;
ir.SetFileName( out );
if(!ir.Read())
{
    return 1;
}
gdcm::Image &image = ir.GetImage();
unsigned long len = image.GetBufferLength();
const gdcm::ByteValue *bv = ir.GetFile().GetDataSet().GetDataElement( gdcm::Tag(0x7fe0,0x0010)
    ).GetByteValue();
if( !bv || len != bv->GetLength() )
{
    return 1;
}
std::cout << bv->GetLength() << " " << len << std::endl;

std::cout << "Success to rewrite image !" << std::endl;
image.Print( std::cout );
return 0;
}

```

## 12.3 SimplePrint.cs

This is a C# example on how to use gdcm::SWIGDataSet

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

```

```

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
    Convertor convertor = new Convertor();
    int a = convertor.Convert<int>( some_int_blob );
    double b = convertor.Convert<double>( some_double_blob );
*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/SimplePrint.exe gdcmData/012345.002.050.dcm
 */
using System;
using gdcm;

public class SimplePrint
{
    public static void RecurseDataSet(File f, DataSet ds, string indent)
    {
        CSharpDataSet cds = new CSharpDataSet(ds);
        while(!cds.IsAtEnd())
        {
            DataElement de = cds.GetCurrent();
            // Compute VR from the toplevel file, and the currently processed dataset:
            VR vr = DataSetHelper.ComputeVR(f, ds, de.GetTag() );

            if( vr.Compatible( new VR(VR.VRType.SQ) ) )
            {
                uint uvl = (uint)de.GetVL(); // Test cast is ok
                System.Console.WriteLine( indent + de.GetTag().toString() + ":" + uvl ); // why not ?
                //SequenceOfItems sq = de.GetSequenceOfItems();
                // GetValueAsSQ handle more cases than GetSequenceOfItems
                SmartPtrSQ sq = de.GetValueAsSQ();
                uint n = sq.GetNumberOfItems();
                for( uint i = 1; i <= n; i++) // item starts at 1, not 0
                {
                    Item item = sq.GetItem( i );
                    DataSet nested = item.GetNestedDataSet();
                    RecurseDataSet( f, nested, indent + "  " );
                }
            }
            else
            {
                System.Console.WriteLine( indent + de.toString() );
            }
            cds.Next();
        }
    }

    public static int Main(string[] args)
    {
        string filename = args[0];
        Reader reader = new Reader();
        reader.SetFileName( filename );
        bool ret = reader.Read();
        if( !ret )
        {
            return 1;
        }
        File f = reader.GetFile();
        DataSet ds = f.GetDataSet();

        RecurseDataSet( f, ds, "" );

        return 0;
    }
}

```

## 12.4 TestReader.cxx

```

/*=====

```

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre

All rights reserved.

See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even  
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR  
PURPOSE. See the above copyright notice for more information.

```

=====*/
#include "gdcmReader.h"
#include "gdcmFileMetaInformation.h"
#include "gdcmFile.h"
#include "gdcmTesting.h"
#include "gdcmMediaStorage.h"

int TestRead(const char* filename, bool verbose = false)
{
    if( verbose )
        std::cout << "TestRead: " << filename << std::endl;

    gdcm::Reader reader;
    reader.SetFileName( filename );
    if ( !reader.Read() )
    {
        std::cerr << "TestReadError: Failed to read: " << filename << std::endl;
        return 1;
    }

    //commenting out the fmi and ds to avoid warnings
    //const gdcm::FileMetaInformation &h = reader.GetFile().GetHeader();
    //std::cout << h << std::endl;

    //const gdcm::DataSet &ds = reader.GetFile().GetDataSet();
    //std::cout << ds << std::endl;

    const char *ref = gdcm::Testing::GetMediaStorageFromFile(filename);
    gdcm::MediaStorage ms;
    ms.SetFromFile( reader.GetFile() );
    if( !ref )
    {
        std::cerr << "TestReadError: Missing MediaStorage: " << filename << std::endl;
        std::cerr << "It should be: " << ms << std::endl;
        return 1;
    }

    if( ms.IsUndefined() && ref && *ref != 0 )
    {
        std::cerr << "TestReadError: MediaStorage: " << filename << std::endl;
        std::cerr << "It should be instead: " << ref << std::endl;
        return 1;
    }

    // Make sure it is the right one:

    if( ref && *ref != 0 && ms != gdcm::MediaStorage::GetMSType(ref) )
    {
        std::cerr << "Error: Found MediaStorage: " << ms << " for " << filename << std::endl;
        std::cerr << "It should be instead: " << ref << std::endl;
        return 1;
    }

    return 0;
}

int TestReader(int argc, char *argv[])
{
    if( argc == 2 )
    {
        const char *filename = argv[1];
        return TestRead(filename, true);
    }

    // else
    gdcm::Trace::DebugOff();
    gdcm::Trace::WarningOff();
    int r = 0, i = 0;
    const char *filename;
    const char * const *filenames = gdcm::Testing::GetFileNames();
    while( (filename = filenames[i]) )

```



```

    {
        r += TestRead( filename );
        ++i;
    }

    return r;
}

```

## 12.5 TestReader.py

This is a C++ example on how to use [gdcm::Reader](#)

```

00001
00014
00015 import os,sys
00016 import gdcm
00017
00018 def TestRead(filename, verbose = False):
00019     r = gdcm.Reader()
00020     r.SetFileName( filename )
00021     success = r.Read()
00022     #if verbose: print r.GetFile()
00023     if verbose: print (r.GetFile().GetDataSet())
00024     return success
00025
00026 if __name__ == "__main__":
00027     success = 0
00028     try:
00029         filename = os.sys.argv[1]
00030         success += TestRead( filename, True )
00031     except:
00032         # loop over all files:
00033         gdcm.Trace.DebugOff()
00034         gdcm.Trace.WarningOff()
00035         t = gdcm.Testing()
00036         nfiles = t.GetNumberOfFileNames()
00037         for i in range(0,nfiles):
00038             filename = t.GetFileName(i)
00039             success += TestRead( filename )
00040
00041
00042     # Test succeed ?
00043     sys.exit(success == 0)

```

## 12.6 DecompressJPEGFile.cs

This is a C# example on how to use [gdcm::SequenceOfFragments](#)

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/DecompressJPEGFile.exe somejpegfile.jpg
 */
using System;
using gdcm;

public class DecompressJPEGFile

```

```

{
    public static int Main(string[] args)
    {
        string file1 = args[0];
        System.IO.FileStream infile =
            new System.IO.FileStream(file1, System.IO.FileMode.Open, System.IO.FileAccess.Read);
        uint fsize = gdcm.PosixEmulation.FileSize(file1);

        byte[] jstream = new byte[fsize];
        infile.Read(jstream, 0, jstream.Length);

        Trace.DebugOn();
        Image image = new Image();
        image.SetNumberOfDimensions( 2 ); // important for now
        DataElement pixeldata = new DataElement( new gdcm.Tag(0x7fe0,0x0010) );

        // DO NOT set a ByteValue here, JPEG is a particular kind of encapsulated syntax
        // in which can one cannot use a simple byte array for storage. Instead, see
        // gdcm.SequenceOfFragments
        //pixeldata.SetByteValue( jstream, new gdcm.VL( (uint)jstream.Length ) );

        // Create a new SequenceOfFragments C++ object, store it as a SmartPointer :
        SmartPtrFrag sq = SequenceOfFragments.New();
        Fragment frag = new Fragment();
        frag.SetByteValue( jstream, new gdcm.VL( (uint)jstream.Length) );
        // Single file => single fragment
        sq.AddFragment( frag );
        // Pass by reference:
        pixeldata.SetValue( sq.__ref__() );

        // insert:
        image.SetDataElement( pixeldata );

        // JPEG use YBR to achieve better compression ratio by default (not RGB)
        // FIXME hardcoded:
        PhotometricInterpretation pi = new PhotometricInterpretation( PhotometricInterpretation.PIType.YBR_FULLL );
        image.SetPhotometricInterpretation( pi );
        // FIXME hardcoded:
        PixelFormat pixeltype = new PixelFormat(3,8,8,7);
        image.SetPixelFormat( pixeltype );

        // FIXME hardcoded:
        image.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.JPEGLosslessProcess14_1 ) );
        image.SetDimension(0, 692);
        image.SetDimension(1, 721);

        // Decompress !
        byte[] decompressedData = new byte[(int)image.GetBufferLength()];
        image.GetBuffer(decompressedData);

        // Write out the decompressed bytes
        System.Console.WriteLine(image.toString());
        using (System.IO.Stream stream =
            System.IO.File.Open(@"tmp/dd.raw",
                System.IO.FileMode.Create))
        {
            System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
            writer.Write(decompressedData);
        }

        return 0;
    }
}

```

## 12.7 ManipulateFile.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR

```

```

    PURPOSE.  See the above copyright notice for more information.
=====*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/ManipulateFile.exe gdcmData/012345.002.050.dcm out.dcm
 */
using System;
using gdcm;

public class ManipulateFile
{
    public static int Main(string[] args)
    {
        {
            string file1 = args[0];
            string file2 = args[1];
            Reader reader = new Reader();
            reader.SetFileName( file1 );
            bool ret = reader.Read();
            if( !ret )
            {
                return 1;
            }

            Anonymizer ano = new Anonymizer();
            ano.SetFile( reader.GetFile() );
            ano.RemovePrivateTags();
            ano.RemoveGroupLength();
            Tag t = new Tag(0x10,0x10);
            ano.Replace( t, "GDCM^Csharp^Test^Hello^World" );

            UIDGenerator g = new UIDGenerator();
            ano.Replace( new Tag(0x0008,0x0018), g.Generate() );
            ano.Replace( new Tag(0x0020,0x000d), g.Generate() );
            ano.Replace( new Tag(0x0020,0x000e), g.Generate() );
            ano.Replace( new Tag(0x0020,0x0052), g.Generate() );

            Writer writer = new Writer();
            writer.SetFileName( file2 );
            writer.SetFile( ano.GetFile() );
            ret = writer.Write();
            if( !ret )
            {
                return 1;
            }

            return 0;
        }
    }
}

```

## 12.8 ClinicalTrialIdentificationWorkflow.cs

This is a C# example on how to use Anonymizer

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

    This software is distributed WITHOUT ANY WARRANTY; without even
    the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
    PURPOSE.  See the above copyright notice for more information.
=====*/

/*
 * Typical usage on UNIX:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/ClinicalTrialIdentificationWorkflow.exe input_dir output_dir
 */
using System;

```

```

using gdc;

public class MyWatcher : SimpleSubjectWatcher
{
    public MyWatcher(Subject s):base(s,"Override String"){
        protected override void StartFilter() {
            System.Console.WriteLine( "This is my start" );
        }
        protected override void EndFilter(){
            System.Console.WriteLine( "This is my end" );
        }
        protected override void ShowProgress(Subject caller, Event evt){
            ProgressEvent pe = ProgressEvent.Cast(evt);
            System.Console.WriteLine( "This is my progress: " + pe.GetProgress() );
        }
        protected override void ShowIteration(){
            System.Console.WriteLine( "This is my iteration" );
        }
        protected override void ShowAnonymization(Subject caller, Event evt){
/*
 * A couple of explanation are necessary here to understand how SWIG work
 * http://www.swig.org/Doc1.3/Java.html#adding_downcasts
 *
 * System.Console.WriteLine( "This is my Anonymization. Type: " + evt.GetEventName() );
 * System.Type type = evt.GetType();
 * System.Console.WriteLine( "This is my Anonymization. System.Type: " + type.ToString() );
 * System.Console.WriteLine( "This is my Anonymization. CheckEvent: " + ae.CheckEvent( evt ) );
 * System.Console.WriteLine( "This is my Anonymization. Processing Tag #" + ae.GetTag().toString() );
 */
            AnonymizeEvent ae = AnonymizeEvent.Cast(evt);
            if( ae != null )
            {
                Tag t = ae.GetTag();
                System.Console.WriteLine( "This is my Anonymization. Processing Tag #" + t.toString() );
            }
            else
            {
                System.Console.WriteLine( "This is my Anonymization. Unhandled Event type: " + evt.GetEventName() );
            }
        }
        protected override void ShowAbort(){
            System.Console.WriteLine( "This is my abort" );
        }
    }
}

public class ClinicalTrialIdentificationWorkflow
{
    public static bool ProcessOneFile( gdc.Anonymizer ano , string filename, string outfilename )
    {
        Reader reader = new Reader();
        reader.SetFileName( filename );
        bool ret = reader.Read();
        if( !ret )
        {
            return false;
        }
        // Pass in the file:
        ano.SetFile( reader.GetFile() );

        // First step, let's protect all Patient information as per
        // PS 3.15 / E.1 / Basic Application Level Confidentiality Profile
        if( !ano.BasicApplicationLevelConfidentialityProfile() )
        {
            return false;
        }

        // Now let's pass in all Clinical Trial fields
        // PS 3.3 - 2008 / C.7.1.3 Clinical Trial Subject Module
        /*
Clinical Trial Sponsor Name (0012,0010) 1 The name of the clinical trial sponsor. See C.7.1.3.1.1.
Clinical Trial Protocol ID (0012,0020) 1 Identifier for the noted protocol. See C.7.1.3.1.2.
Clinical Trial Protocol Name (0012,0021) 2 The name of the clinical trial protocol. See C.7.1.3.1.3.
Clinical Trial Site ID (0012,0030) 2 The identifier of the site responsible for submitting clinical trial
data. See C.7.1.3.1.4.
Clinical Trial Site Name (0012,0031) 2 Name of the site responsible for submitting clinical trial data. See
C.7.1.3.1.5
Clinical Trial Subject ID (0012,0040) 1C The assigned identifier for the clinical trial subject. See
C.7.1.3.1.6. Shall be present if Clinical Trial Subject Reading ID (0012,0042) is absent. May be present
otherwise.
Clinical Trial Subject Reading ID (0012,0042) 1C Identifies the subject for blinded evaluations. Shall be
present if Clinical Trial Subject ID (0012,0040) is absent. May be present otherwise. See C.7.1.3.1.7.

```

```

    */
    ano.Replace( new gdcm.Tag(0x0012,0x0010), "MySponsorName");
    ano.Replace( new gdcm.Tag(0x0012,0x0020), "MyProtocolID");
    ano.Replace( new gdcm.Tag(0x0012,0x0021), "MyProtocolName");
    ano.Replace( new gdcm.Tag(0x0012,0x0030), "MySiteId");
    ano.Replace( new gdcm.Tag(0x0012,0x0031), "MySiteName");
    ano.Replace( new gdcm.Tag(0x0012,0x0040), "MySponsorId");
    ano.Replace( new gdcm.Tag(0x0012,0x0050), "MyTPId");
    ano.Replace( new gdcm.Tag(0x0012,0x0051), "MyTPDescription");

    // The following two are not required as they are guaranteed to be filled in by the
    // Basic Application Level Confidentiality Profile. Only override if you understand what
    // you are doing
    //ano.Replace( new gdcm.Tag(0x0012,0x0062), "YES");
    //ano.Replace( new gdcm.Tag(0x0012,0x0063), "My Super Duper Anonymization Overload");

    // We might be generating a subdirectory. Let's make sure the subdir exist:
    gdcm.FileMetaInformation fn = new gdcm.FileMetaInformation( outfilename );
    string subdir = fn.GetPath();
    if( !gdcm.PosixEmulation.MakeDirectory( subdir ) )
    {
        return false;
    }

    gdcm.FileMetaInformation fmi = ano.GetFile().GetHeader();
    // The following three lines make sure to regenerate any value:
    fmi.Remove( new gdcm.Tag(0x0002,0x0012) );
    fmi.Remove( new gdcm.Tag(0x0002,0x0013) );
    fmi.Remove( new gdcm.Tag(0x0002,0x0016) );

    Writer writer = new Writer();
    writer.SetFileName( outfilename );
    writer.SetFile( ano.GetFile() );
    ret = writer.Write();
    if( !ret )
    {
        return false;
    }

    return true;
}

public static int Main(string[] args)
{
    gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "My ClinicalTrial App" );

    // http://www.oid-info.com/get/1.3.6.1.4.17434
    string THERALYS_ORG_ROOT = "1.3.6.1.4.17434";
    gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
    System.Console.WriteLine( "Root dir is now: " + gdcm.UIDGenerator.GetRoot() );

    gdcm.Global global = gdcm.Global.GetInstance();
    if( !global.LoadResourcesFiles() )
    {
        System.Console.WriteLine( "Could not LoadResourcesFiles" );
        return 1;
    }

    if( args.Length != 2 )
    {
        System.Console.WriteLine( "Usage:" );
        System.Console.WriteLine( "ClinicalTrialIdentificationWorkflow input_dir output_dir" );
        return 1;
    }

    string dir1 = args[0];
    string dir2 = args[1];

    // Check input is valid:
    if( !gdcm.PosixEmulation.FileIsDirectory(dir1) )
    {
        System.Console.WriteLine( "Input directory: " + dir1 + " does not exist. Sorry" );
        return 1;
    }
    if( !gdcm.PosixEmulation.FileIsDirectory(dir2) )
    {
        System.Console.WriteLine( "Output directory: " + dir2 + " does not exist. Sorry" );
        return 1;
    }

    // Recursively search all file within this toplevel directory:
    Directory d = new Directory();

```

```

uint nfiles = d.Load( dir1, true );
if(nfiles == 0) return 1;

// Let's use the pre-shipped certificate of GDcm.
string certpath = gdcm.FileName.Join(gdcm.Testing.GetSourceDirectory(),
    "/Testing/Source/Data/certificate.pem" );
gdcm.CryptoFactory fact = gdcm.CryptoFactory.GetFactoryInstance();
gdcm.CryptographicMessageSyntax cms = fact.CreateCMSProvider();
if( !cms.ParseCertificateFile( certpath ) )
{
    System.Console.WriteLine( "PEM Certificate : " + certpath + " could not be read. Sorry" );
    return 1;
}

//Anonymizer ano = new Anonymizer();
// A reference to an actual C++ instance is required here:
SmartPtrAno sano = Anonymizer.New();
Anonymizer ano = sano.__ref__();

//SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(ano, "Anonymizer");
MyWatcher watcher = new MyWatcher(ano);

// Explicitly specify the Cryptographic Message Syntax to use:
ano.SetCryptographicMessageSyntax( cms );

// Process all filenames:
FilenameType filenames = d.GetFilenames();
for( uint i = 0; i < nfiles; ++i )
{
    string filename = filenames[ (int)i ];
    string outfilename = filename.Replace( dir1, dir2 );
    System.Console.WriteLine( "Filename: " + filename );
    System.Console.WriteLine( "Out Filename: " + outfilename );
    if( !ProcessOneFile( ano , filename, outfilename ) )
    {
        System.Console.WriteLine( "Could not process filename: " + filename );
        return 1;
    }
}

return 0;
}
}

```

## 12.9 GenerateDICOMDIR.cs

This is a C# example on how to use DICOMDIRGenerator

```

/*=====
Program: GDcm (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Simple C# example to show how to use DICOMDIRGenerator
 *
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/GenerateDICOMDIR.exe path output_filename
 */
using System;
using gdcm;

public class GenerateDICOMDIR
{
    public static int Main(string[] args)
    {

```

```

string directory = args[0];
string outfilename = args[1];

Directory d = new Directory();
uint nfiles = d.Load( directory, true );
if(nfiles == 0) return 1;
//System.Console.WriteLine( "Files:\n" + d.toString() );

// Implement fast path ?
// Scanner s = new Scanner();

string descriptor = "My_Descriptor";
FileNamesType filenames = d.GetFilesNames();

gdcm.DICOMDIRGenerator gen = new DICOMDIRGenerator();
gen.SetFilenames( filenames );
gen.SetDescriptor( descriptor );
if( !gen.Generate() )
{
    return 1;
}

gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "GenerateDICOMDIR" );
gdcm.Writer writer = new Writer();
writer.SetFile( gen.GetFile() );
writer.SetFileName( outfilename );
if( !writer.Write() )
{
    return 1;
}

return 0;
}

```

## 12.10 GenFakelImage.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
#include "gdcmImage.h"
#include "gdcmImageWriter.h"
#include "gdcmFileDerivation.h"
#include "gdcmUIDGenerator.h"
// #include "gdcmImageChangePhotometricInterpretation.h"

/*
 * This example shows two things:
 * 1. How to create an image ex-nihilo
 * 2. How to use the gdcm.FileDerivation filter. This filter is meant to create "DERIVED" image
 * object. FileDerivation has a simple API where you can reference *all* the input image that have been
 * used to generate the image. The API also allows user to specify the purpose of reference (see CID 7202,
 * PS 3.16 - 2008), and the image derivation type (CID 7203, PS 3.16 - 2008).
 */
int main(int, char *[])
{
    // Step 1: Fake Image
    gdcm::SmartPointer<gdcm::Image> im = new gdcm::Image;

    char * buffer = new char[ 256 * 256 * 3];
    char * p = buffer;
    int b = 128;
    //int ybr[3];
    int ybr2[3];
    //int rgb[3];

    for(int r = 0; r < 256; ++r)

```

```

for(int g = 0; g < 256; ++g)
    //for(int b = 0; b < 256; ++b)
    {
        //rgb[0] = r;
        //rgb[1] = g;
        //rgb[1] = 128;
        //rgb[2] = b;
        //ybr[0] = r;
        //ybr[1] = g;
        //ybr[1] = 128;
        //ybr[2] = b;

        ybr2[0] = r;
        ybr2[1] = g;
        ybr2[1] = 128;
        ybr2[2] = b;
        //gdcm::ImageChangePhotometricInterpretation::YBR2RGB(rgb, ybr);
        //gdcm::ImageChangePhotometricInterpretation::RGB2YBR(ybr2, rgb);
        *p++ = (char)ybr2[0];
        *p++ = (char)ybr2[1];
        *p++ = (char)ybr2[2];
    }

im->SetNumberOfDimensions( 2 );
im->SetDimension(0, 256 );
im->SetDimension(1, 256 );

im->GetPixelFormat().SetSamplesPerPixel(3);
//im->SetPhotometricInterpretation( gdcm::PhotometricInterpretation::RGB );
im->SetPhotometricInterpretation( gdcm::PhotometricInterpretation::YBR_FULL );

unsigned long l = im->GetBufferLength();
if( l != 256 * 256 * 3 )
{
    return 1;
}
gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
pixeldata.SetByteValue( buffer, (uint32_t)l );
delete[] buffer;
im->SetDataElement( pixeldata );

gdcm::UIDGenerator uid; // helper for uid generation

gdcm::SmartPointer<gdcm::File> file = new gdcm::File; // empty file

// Step 2: DERIVED object
gdcm::FileDerivation fd;
// For the purpose of this exercise we will pretend that this image is referencing
// two source image (we need to generate fake UID for that).
const char ReferencedSOPClassUID[] = "1.2.840.10008.5.1.4.1.1.7"; // Secondary Capture
fd.AddReference( ReferencedSOPClassUID, uid.Generate() );
fd.AddReference( ReferencedSOPClassUID, uid.Generate() );

// Again for the purpose of the exercise we will pretend that the image is a
// multiplanar reformat (MPR):
// CID 7202 Source Image Purposes of Reference
// { "DCM",121322,"Source image for image processing operation"},
fd.SetPurposeOfReferenceCodeSequenceCodeValue( 121322 );
// CID 7203 Image Derivation
// { "DCM",113072,"Multiplanar reformatting" },
fd.SetDerivationCodeSequenceCodeValue( 113072 );
fd.SetFile( *file );
// If all Code Value are ok the filter will execute properly
if( !fd.Derive() )
{
    std::cerr << "Sorry could not derive using input info" << std::endl;
    return 1;
}

// We pass both :
// 1. the fake generated image
// 2. the 'DERIVED' dataset object
// to the writer.
gdcm::ImageWriter w;
w.SetImage( *im );
w.SetFile( fd.GetFile() );

// Set the filename:
w.SetFileName( "ybr2.dcm" );
if( !w.Write() )
{

```



```

    return 1;
}

return 0;
}

```

## 12.11 ReformatFile.cs

This is a C++ example on how to use FileDerivation

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Simple C# example
 *
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/ReformatFile.exe input.dcm output.dcm
 */
using System;
using gdcm;

public class ReformatFile
{
    public static int Main(string[] args)
    {
        gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "My Reformat App" );

        // http://www.oid-info.com/get/1.3.6.1.4.17434
        string THERALYS_ORG_ROOT = "1.3.6.1.4.17434";
        gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
        System.Console.WriteLine( "Root dir is now: " + gdcm.UIDGenerator.GetRoot() );

        string filename = args[0];
        string outfilename = args[1];

        Reader reader = new Reader();
        reader.SetFileName( filename );
        if( !reader.Read() )
        {
            System.Console.WriteLine( "Could not read: " + filename );
            return 1;
        }

        UIDGenerator uid = new UIDGenerator(); // helper for uid generation
        FileDerivation fd = new FileDerivation();
        // For the purpose of this exercise we will pretend that this image is referencing
        // two source image (we need to generate fake UID for that).
        string ReferencedSOPClassUID = "1.2.840.10008.5.1.4.1.1.7"; // Secondary Capture
        fd.AddReference( ReferencedSOPClassUID, uid.Generate() );
        fd.AddReference( ReferencedSOPClassUID, uid.Generate() );

        // Again for the purpose of the exercise we will pretend that the image is a
        // multiplanar reformat (MPR):
        // CID 7202 Source Image Purposes of Reference
        // {"DCM",121322,"Source image for image processing operation"},
        fd.SetPurposeOfReferenceCodeSequenceCodeValue( 121322 );
        // CID 7203 Image Derivation
        // {"DCM",113072,"Multiplanar reformatting" },
        fd.SetDerivationCodeSequenceCodeValue( 113072 );
        fd.SetFile( reader.GetFile() );
        // If all Code Value are ok the filter will execute properly
        if( !fd.Derive() )
        {

```

```

        return 1;
    }

    gdcm.FileMetaInformation fmi = reader.GetFile().GetHeader();
    // The following three lines make sure to regenerate any value:
    fmi.Remove( new gdcm.Tag(0x0002,0x0012) );
    fmi.Remove( new gdcm.Tag(0x0002,0x0013) );
    fmi.Remove( new gdcm.Tag(0x0002,0x0016) );

    Writer writer = new Writer();
    writer.SetFileName( outfilename );
    writer.SetFile( fd.GetFile() );
    if( !writer.Write() )
    {
        System.Console.WriteLine( "Could not write: " + outfilename );
        return 1;
    }

    return 0;
}

```

## 12.12 DecompressImage.cs

This is a C# example on how to use Image

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/DecompressImage.exe gdcmData/012345.002.050.dcm decompress.dcm
 */
using System;
using gdcm;

public class DecompressImage
{
    public static int Main(string[] args)
    {
        string file1 = args[0];
        string file2 = args[1];
        ImageReader reader = new ImageReader();
        reader.SetFileName( file1 );
        bool ret = reader.Read();
        if( !ret )
        {
            return 1;
        }

        // check that one can access a Fragment from C#:
        var de = reader.GetFile().GetDataSet().GetDataElement( new Tag(0x7fe0, 0x0010) );
        var sq = de.GetSequenceOfFragments();
        sq.GetFragment( 0 );

        Image image = new Image();
        Image ir = reader.GetImage();

        image.SetNumberOfDimensions( ir.GetNumberOfDimensions() );

        //Just for fun:
        //int dircos = ir.GetDirectionCosines();
        //t = gdcm.Orientation.GetType(dircos);
    }
}

```

```

//int l = gdc.Orientation.GetLabel(t);
//System.Console.WriteLine( "Orientation label:" + l );

// Set the dimensions,
// 1. either one at a time
//image.SetDimension(0, ir.GetDimension(0) );
//image.SetDimension(1, ir.GetDimension(1) );

// 2. the array at once
uint[] dims = {0, 0};
// Just for fun let's invert the dimensions:
dims[0] = ir.GetDimension(1);
dims[1] = ir.GetDimension(0);
ir.SetDimensions( dims );

PixelFormat pixeltype = ir.GetPixelFormat();
image.SetPixelFormat( pixeltype );

PhotometricInterpretation pi = ir.GetPhotometricInterpretation();
image.SetPhotometricInterpretation( pi );

DataElement pixeldata = new DataElement( new Tag(0x7fe0,0x0010) );
byte[] str1 = new byte[ ir.GetBufferLength()];
ir.GetBuffer( str1 );
//System.Console.WriteLine( ir.GetBufferLength() );
pixeldata.SetByteValue( str1, new VL( (uint)str1.Length ) );
//image.SetDataElement( pixeldata );
ir.SetDataElement( pixeldata );

ImageWriter writer = new ImageWriter();
writer.SetFileName( file2 );
writer.SetFile( reader.GetFile() );
writer.SetImage( ir );
ret = writer.Write();
if( !ret )
{
    return 1;
}

return 0;
}
}

```

## 12.13 StandardizeFiles.cs

This is a C++ example on how to use ImageChangeTransferSyntax

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Simple C# example to show how one would 'Standardize' a DICOM File-Set
 *
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/StandardizeFiles.exe input_path output_path
 */
using System;
using gdcm;

public class StandardizeFiles
{
    public static bool ProcessOneFile( string filename, string outfilename )
    {

```

```

PixmapReader reader = new PixmapReader();
reader.SetFileName( filename );
if( !reader.Read() )
{
    System.Console.WriteLine( "Could not read: " + filename );
    return false;
}

ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
change.SetForce( false ); // do we really want to recompress when input is already compressed in same alg ?
change.SetCompressIconImage( false ); // Keep it simple
change.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.JPEG2000Lossless ) );
change.SetInput( reader.GetPixmap() );
if( !change.Change() )
{
    System.Console.WriteLine( "Could not change: " + filename );
    return false;
}

gdcm.FileMetaInformation fmi = reader.GetFile().GetHeader();
// The following three lines make sure to regenerate any value:
fmi.Remove( new gdcm.Tag(0x0002,0x0012) );
fmi.Remove( new gdcm.Tag(0x0002,0x0013) );
fmi.Remove( new gdcm.Tag(0x0002,0x0016) );

PixmapWriter writer = new PixmapWriter();
writer.SetFileName( outfilename );
writer.SetFile( reader.GetFile() );
gdcm.Pixmap pixout = ((PixmapToPixmapFilter)change).GetOutput();

writer.SetPixmap( pixout );
if( !writer.Write() )
{
    System.Console.WriteLine( "Could not write: " + outfilename );
    return false;
}

return true;
}

public static int Main(string[] args)
{
    gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "My Standardize App" );

    // http://www.oid-info.com/get/1.3.6.1.4.17434
    string THERALYS_ORG_ROOT = "1.3.6.1.4.17434";
    gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
    System.Console.WriteLine( "Root dir is now: " + gdcm.UIDGenerator.GetRoot() );

    string dir1 = args[0];
    string dir2 = args[1];

    // Check input is valid:
    if( !gdcm.PosixEmulation.FileIsDirectory(dir1) )
    {
        System.Console.WriteLine( "Input directory: " + dir1 + " does not exist. Sorry" );
        return 1;
    }
    if( !gdcm.PosixEmulation.FileIsDirectory(dir2) )
    {
        System.Console.WriteLine( "Output directory: " + dir2 + " does not exist. Sorry" );
        return 1;
    }

    Directory d = new Directory();
    uint nfiles = d.Load( dir1, true );
    if(nfiles == 0) return 1;

    // Process all filenames:
    FilenamesType filenames = d.GetFilenames();
    for( uint i = 0; i < nfiles; ++i )
    {
        string filename = filenames[ (int)i ];
        string outfilename = filename.Replace( dir1, dir2 );
        System.Console.WriteLine( "Filename: " + filename );
        System.Console.WriteLine( "Out Filename: " + outfilename );
        if( !ProcessOneFile( filename, outfilename ) )
        {
            System.Console.WriteLine( "Could not process filename: " + filename );
            //return 1;
        }
    }
}

```

```

    }

    return 0;
}

```

## 12.14 ScanDirectory.cs

This is a C# example on how to use Scanner

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Usage:
 * $ bin/ScanDirectory.exe /path/to/gdcmData/
 */
using System;
using gdcm;

// We will print each filename being processed
public class MyWatcher : SimpleSubjectWatcher
{
    public MyWatcher(Subject s):base(s,"Override String"){
    protected override void ShowFileName(Subject caller, Event evt){
        FileNameEvent fne = FileNameEvent.Cast(evt);
        if( fne != null )
        {
            string fn = fne.GetFileName();
            System.Console.WriteLine( "This is my Scanner. Processing FileName: " + fn );
        }
        else
        {
            System.Console.WriteLine( "This is my Anonymization. Unhandled Event type: " + evt.GetEventName() );
        }
    }
}

public class ScanDirectory
{
    public static int Main(string[] args)
    {
        string directory = args[0];
        Tag t = new Tag(0x8,0x80);

        Directory d = new Directory();
        uint nfiles = d.Load( directory );
        if(nfiles == 0) return 1;
        //System.Console.WriteLine( "Files:\n" + d.toString() );

        // Use a StrictScanner, need to use a reference to pass the C++ pointer to
        // MyWatcher implementation
        SmartPtrStrictScan sscan = StrictScanner.New();
        StrictScanner s = sscan.__ref__();
        MyWatcher watcher = new MyWatcher(s);

        s.AddTag( t );
        bool b = s.Scan( d.GetFileNames() );
        if(!b) return 1;

        for(int i = 0; i < (int)nfiles; ++i)
        {
            if( !s.IsKey( d.GetFileNames()[i] ) )
            {

```

```

        System.Console.WriteLine( "File is not DICOM or could not be read: " + d.GetFileNames()[i] );
    }
}

System.Console.WriteLine( "Scan:\n" + s.toString() );

System.Console.WriteLine( "success" );
return 0;
}
}

```

## 12.15 BasicAnonymizer.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/BasicAnonymizer.exe gdcmData/012345.002.050.dcm out.dcm
 */
using System;
using gdcm;

public class MyWatcher : SimpleSubjectWatcher
{
    public MyWatcher(Subject s):base(s,"Override String"){
    protected override void StartFilter() {
        System.Console.WriteLine( "This is my start" );
    }
    protected override void EndFilter(){
        System.Console.WriteLine( "This is my end" );
    }
    protected override void ShowProgress(Subject caller, Event evt){
        ProgressEvent pe = ProgressEvent.Cast(evt);
        System.Console.WriteLine( "This is my progress: " + pe.GetProgress() );
    }
    protected override void ShowIteration(){
        System.Console.WriteLine( "This is my iteration" );
    }
    protected override void ShowAnonymization(Subject caller, Event evt){
/*
 * A couple of explanation are necessary here to understand how SWIG work
 * http://www.swig.org/Doc1.3/Java.html#adding_downcasts
 *
 * System.Console.WriteLine( "This is my Anonymization. Type: " + evt.GetEventName() );
 * System.Type type = evt.GetType();
 * System.Console.WriteLine( "This is my Anonymization. System.Type: " + type.ToString() );
 * System.Console.WriteLine( "This is my Anonymization. CheckEvent: " + ae.CheckEvent( evt ) );
 * System.Console.WriteLine( "This is my Anonymization. Processing Tag #" + ae.GetTag().toString() );
 */
        AnonymizeEvent ae = AnonymizeEvent.Cast(evt);
        if( ae != null )
        {
            Tag t = ae.GetTag();
            System.Console.WriteLine( "This is my Anonymization. Processing Tag #" + t.toString() );
        }
        else
        {
            System.Console.WriteLine( "This is my Anonymization. Unhandled Event type: " + evt.GetEventName() );
        }
    }
    protected override void ShowAbort(){
        System.Console.WriteLine( "This is my abort" );
    }
}

```

```

}

public class BasicAnonymizer
{
    public static int Main(string[] args)
    {
        gdcm.Global global = gdcm.Global.GetInstance();
        if( !global.LoadResourcesFiles() )
        {
            System.Console.WriteLine( "Could not LoadResourcesFiles" );
            return 1;
        }

        string file1 = args[0];
        string file2 = args[1];
        Reader reader = new Reader();
        reader.SetFileName( file1 );
        bool ret = reader.Read();
        if( !ret )
        {
            return 1;
        }

        string certpath = gdcm.Filename.Join(gdcm.Testing.GetSourceDirectory(),
            "/Testing/Source/Data/certificate.pem" );
        gdcm.CryptoFactory fact = gdcm.CryptoFactory.GetFactoryInstance();
        gdcm.CryptographicMessageSyntax cms = fact.CreateCMSProvider();
        if( !cms.ParseCertificateFile( certpath ) )
        {
            return 1;
        }

        //Anonymizer ano = new Anonymizer();
        SmartPtrAno sano = Anonymizer.New();
        Anonymizer ano = sano.__ref__();

        //SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(ano, "Anonymizer");
        MyWatcher watcher = new MyWatcher(ano);

        ano.SetFile( reader.GetFile() );
        ano.SetCryptographicMessageSyntax( cms );
        if( !ano.BasicApplicationLevelConfidentialityProfile() )
        {
            return 1;
        }

        Writer writer = new Writer();
        writer.SetFileName( file2 );
        writer.SetFile( ano.GetFile() );
        ret = writer.Write();
        if( !ret )
        {
            return 1;
        }

        return 0;
    }
}

```

## 12.16 BasicImageAnonymizer.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
*/

```

```

using System;
using gdcm;

public class BasicImageAnonymizer
{
    public static int Main(string[] args)
    {
        string filename = args[0];

        // instantiate the reader:
        gdcm.ImageReader reader = new gdcm.ImageReader();
        reader.SetFileName( filename );

        if (!reader.Read()) return 1;

        Image ir = reader.GetImage();

        uint[] dims = {0, 0, 0};
        dims[0] = ir.GetDimension(0);
        dims[1] = ir.GetDimension(1);
        dims[2] = ir.GetDimension(2);
        System.Console.WriteLine( "Dim:" + dims[0] );
        System.Console.WriteLine( "Dim:" + dims[1] );
        System.Console.WriteLine( "Dim:" + dims[2] );

        // buffer to get the pixels
        byte[] buffer = new byte[ ir.GetBufferLength()];
        System.Console.WriteLine( "Dim:" + ir.GetBufferLength() );
        ir.GetBuffer( buffer );

        for (uint z = 0; z < dims[2]; z++)
        {
            for (uint y = 0; y < dims[1] / 2; y++) // only half Y
            {
                for (uint x = 0; x < dims[0] / 2; x++) // only half X
                {
                    buffer[ (z * dims[1] + y) * dims[0] + x ] = 0; // works when pixel type == UINT8
                }
            }
        }

        DataElement pixeldata = new DataElement( new Tag(0x7fe0,0x0010) );
        pixeldata.SetByteValue( buffer, new VL( (uint)buffer.Length ) );
        ir.SetDataElement( pixeldata );
        ir.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.ExplicitVRLittleEndian ) );

        ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
        change.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.JPEGLSLossless ) );
        change.SetInput( ir );
        if( !change.Change() )
        {
            System.Console.WriteLine( "Could not change: " + filename );
            return 1;
        }

        ImageWriter writer = new ImageWriter();
        writer.SetFileName( "out.dcm" );
        writer.SetFile( reader.GetFile() );
        writer.SetImage( change.GetOutput() );
        bool ret = writer.Write();
        if( !ret )
        {
            return 1;
        }

        return 0;
    }
}

```

## 12.17 Cleaner.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre

```



All rights reserved.  
See Copyright.txt or <http://gdcms.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even  
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR  
PURPOSE. See the above copyright notice for more information.

```
=====*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/Cleaner.exe gdcmData/012345.002.050.dcm out.dcm
 */
using System;
using gdcm;

public class MyWatcher : SimpleSubjectWatcher
{
    public MyWatcher(Subject s):base(s,"Override String"){
        protected override void StartFilter() {
            System.Console.WriteLine( "This is my start" );
        }
        protected override void EndFilter(){
            System.Console.WriteLine( "This is my end" );
        }
        protected override void ShowProgress(Subject caller, Event evt){
            ProgressEvent pe = ProgressEvent.Cast(evt);
            System.Console.WriteLine( "This is my progress: " + pe.GetProgress() );
        }
        protected override void ShowIteration(){
            System.Console.WriteLine( "This is my iteration" );
        }
        protected override void ShowAnonymization(Subject caller, Event evt){
/*
 * A couple of explanation are necessary here to understand how SWIG work
 * http://www.swig.org/Doc1.3/Java.html#adding\_downcasts
 *
 * System.Console.WriteLine( "This is my Anonymization. Type: " + evt.GetEventName() );
 * System.Type type = evt.GetType();
 * System.Console.WriteLine( "This is my Anonymization. System.Type: " + type.ToString() );
 * System.Console.WriteLine( "This is my Anonymization. CheckEvent: " + ae.CheckEvent( evt ) );
 * System.Console.WriteLine( "This is my Anonymization. Processing Tag #" + ae.GetTag().toString() );
 */
            AnonymizeEvent ae = AnonymizeEvent.Cast(evt);
            if( ae != null )
            {
                Tag t = ae.GetTag();
                System.Console.WriteLine( "This is my Anonymization. Processing Tag #" + t.toString() );
            }
            else
            {
                System.Console.WriteLine( "This is my Anonymization. Unhandled Event type: " + evt.GetEventName() );
            }
        }
        protected override void ShowAbort(){
            System.Console.WriteLine( "This is my abort" );
        }
    }
}

public class Cleaner
{
    public static int Main(string[] args)
    {
        {
            gdcm.Global global = gdcm.Global.GetInstance();
            if( !global.LoadResourcesFiles() )
            {
                System.Console.WriteLine( "Could not LoadResourcesFiles" );
                return 1;
            }

            string file1 = args[0];
            string file2 = args[1];
            Reader reader = new Reader();
            reader.SetFileName( file1 );
            bool ret = reader.Read();
            if( !ret )
            {
                return 1;
            }
        }
    }
}
```

```

    }

    SmartPtrCleaner scleaner = gdcm.Cleaner.New();
    gdcm.Cleaner cleaner = scleaner.__ref__();

    //SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(cleaner, "Anonymizer");
    MyWatcher watcher = new MyWatcher(cleaner);

    cleaner.SetFile( reader.GetFile() );
    cleaner.Empty( new gdcm.VR(gdcm.VR.VRType.PN) );
    gdcm.DPath dpath = new gdcm.DPath();
    dpath.ConstructFromString( "/0010,0010" );
    cleaner.Preserve( dpath );
    gdcm.Tag t1 = new gdcm.Tag(0x10, 0x30);
    cleaner.Empty( t1 );
    gdcm.PrivateTag pt0 = new gdcm.PrivateTag( new gdcm.Tag(0x29,0x60), "SIEMENS MEDCOM HEADER2" );
    cleaner.Remove( pt0 );
    gdcm.PrivateTag pt1 = new gdcm.PrivateTag( new gdcm.Tag(0x29,0x10), "SIEMENS CSA HEADER" );
    gdcm.PrivateTag pt2 = new gdcm.PrivateTag( new gdcm.Tag(0x29,0x20), "SIEMENS CSA HEADER" );
    cleaner.Scrub( pt1 );
    cleaner.Scrub( pt2 );
    if( !cleaner.Clean() )
    {
        return 1;
    }

    Writer writer = new Writer();
    writer.SetFileName( file2 );
    writer.SetFile( cleaner.GetFile() );
    ret = writer.Write();
    if( !ret )
    {
        return 1;
    }

    return 0;
}

```

## 12.18 CompressLossyJPEG.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Perso/gdcm/debug-gcc/bin
 * $ mono bin/CompressLossyJPEG.exe input.dcm output.dcm
 */

using System;
using gdcm;

public class CompressLossyJPEG
{
    public static int Main(string[] args)
    {
        {
            if( args.Length < 2 )
            {
                System.Console.WriteLine( " input.dcm output.dcm" );
                return 1;
            }
            string filename = args[0];
            string outfilename = args[1];

            ImageReader reader = new ImageReader();
            reader.SetFileName( filename );

```

```

if( !reader.Read() )
{
    System.Console.WriteLine( "Could not read: " + filename );
    return 1;
}

// The output of gdcm::Reader is a gdcm::File
File file = reader.GetFile();

// the dataset is the the set of element we are interested in:
DataSet ds = file.GetDataSet();

Image image = reader.GetImage();
//image.Print( cout );

ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
TransferSyntax targetts = new TransferSyntax( TransferSyntax.TType.JPEGBaselineProcess1 );
change.SetTransferSyntax( targetts );

// Setup our JPEGCodec, warning it should be compatible with JPEGBaselineProcess1
JPEGCodec jpegcodec = new JPEGCodec();
if( !jpegcodec.CanCode( targetts ) )
{
    System.Console.WriteLine( "Something went really wrong, JPEGCodec cannot handle JPEGBaselineProcess1" );
    return 1;
}
jpegcodec.SetLossless( false );
jpegcodec.SetQuality( 50 ); // poor quality !
change.SetUserCodec( jpegcodec ); // specify the codec to use to the ImageChangeTransferSyntax

change.SetInput( image );
bool b = change.Change();
if( !b )
{
    System.Console.WriteLine( "Could not change the Transfer Syntax" );
    return 1;
}

ImageWriter writer = new ImageWriter();
writer.SetImage( (gdcm.Image)change.GetOutput() );
writer.SetFile( reader.GetFile() );
writer.SetFileName( outfilename );
if( !writer.Write() )
{
    System.Console.WriteLine( "Could not write: " + outfilename );
    return 1;
}

return 0;
}
}

```

## 12.19 DecompressImageMultiframe.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
$ gdcminfo ~/Desktop/angiogram-06.dcm
MediaStorage is 1.2.840.10008.5.1.4.1.1.12.1 [X-Ray Angiographic Image Storage]
TransferSyntax is 1.2.840.10008.1.2.4.50 [JPEG Baseline (Process 1): Default Transfer Syntax for Lossy JPEG 8
    Bit Image Compression]
NumberOfDimensions: 3
Dimensions: (512,512,355)
Origin: (0,0,0)

```

```

Spacing: (1,1,40)
DirectionCosines: (1,0,0,0,1,0)
Rescale Intercept/Slope: (0,1)
SamplesPerPixel      :1
BitsAllocated        :8
BitsStored           :8
HighBit              :7
PixelRepresentation:0
ScalarType found     :UINT8
PhotometricInterpretation: MONOCHROME2
PlanarConfiguration: 0
TransferSyntax: 1.2.840.10008.1.2.4.50
Orientation Label: AXIAL
*/

/*
 * Description:
 *
 * Assume we have a file angiogram-06.dcm as described above.
 * the following program will decompress directly from the extracted jpeg stream.
 *
 * First step extract the jpeg stream (but not the Basic Offset Table):
 *
 * $ gdcmmraw -i angiogram-06.dcm -o /tmp/output/chris --split-frags --pattern %d.jpg
 *
 * Check that indeed there are 355 files, while there are 356 fragments in the original DICOM file, since
 * gdcmmraw always skip the first fragment (Basic Offset Table).
 *
 * Now from those individual jpeg stream, recreate a fake gdcm.DataElement...
 *
 * Usage:
 *
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono ./bin/DecompressImageMultiframe.exe /tmp/output
 */
using System;
using gdcm;

public class DecompressImageMultiframe
{
    public static int Main(string[] args)
    {
        string directory = args[0];
        gdcm.Directory dir = new gdcm.Directory();
        uint nfiles = dir.Load(directory);
        //System.Console.WriteLine(dir.toString());
        gdcm.FileNamesType filenames = dir.GetFilesNames();

        Image image = new Image();
        image.SetNumberOfDimensions( 3 ); // important for now
        DataElement pixeldata = new DataElement( new gdcm.Tag(0x7fe0,0x0010) );

        // Create a new SequenceOfFragments C++ object, store it as a SmartPointer :
        SmartPtrFrag sq = SequenceOfFragments.New();

        // Yeah, the files are not guaranteed to be in order, please adapt...
        for(uint i = 0; i < nfiles; ++i)
        {
            System.Console.WriteLine( filenames[(int)i] );
            string file = filenames[(int)i];
            System.IO.FileStream infile =
                new System.IO.FileStream(file, System.IO.FileMode.Open, System.IO.FileAccess.Read);
            uint fsize = gdcm.PosixEmulation.FileSize(file);

            byte[] jstream = new byte[fsize];
            infile.Read(jstream, 0 , jstream.Length);

            Fragment frag = new Fragment();
            frag.SetByteValue( jstream, new gdcm.VL( (uint)jstream.Length) );
            sq.AddFragment( frag );
        }

        // Pass by reference:
        pixeldata.SetValue( sq.__ref__() );

        // insert:
        image.SetDataElement( pixeldata );

        // JPEG use YBR to achieve better compression ratio by default (not RGB)
        // FIXME hardcoded:
        PhotometricInterpretation pi = new PhotometricInterpretation( PhotometricInterpretation.PIType.MONOCHROME2

```

```

    );
    image.SetPhotometricInterpretation( pi );
    // FIXME hardcoded:
    PixelFormat pixeltype = new PixelFormat(1,8,8,7);
    image.SetPixelFormat( pixeltype );

    // FIXME hardcoded:
    image.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.JPEGLosslessProcess14_1 ) );
    image.SetDimension(0, 512);
    image.SetDimension(1, 512);
    image.SetDimension(2, 355);

    // Decompress !
    byte[] decompressedData = new byte[(int)image.GetBufferLength()];
    image.GetBuffer(decompressedData);

    // Write out the decompressed bytes
    System.Console.WriteLine(image.ToString());
    using (System.IO.Stream stream =
        System.IO.File.Open(@"tmp/dd.raw",
            System.IO.FileMode.Create))
    {
        System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
        writer.Write(decompressedData);
    }

    return 0;
}

```

## 12.20 DumpCSA.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Usage:
 * $ bin/DumpCSA.exe input.dcm
 */
using System;
using gdcm;

public class DumpCSA
{
    public static int Main(string[] args)
    {
        string filename = args[0];

        gdcm.Reader reader = new gdcm.Reader();
        reader.SetFileName( filename );
        if (!reader.Read()) return 1;

        gdcm.File f = reader.GetFile();
        gdcm.DataSet ds = f.GetDataSet();

        string[] expectedSiemensTags = new string[] { "B_value", "AcquisitionMatrixText" };
        using (PrivateTag gtag = CSAHeader.GetCSAImageHeaderInfoTag())
        {
            if (ds.FindDataElement(gtag))
            {
                using (DataElement de = ds.GetDataElement(gtag))
                {
                    if (de != null && !de.IsEmpty())
                    {
                        using (CSAHeader csa = new CSAHeader())

```

```

        {
            if (csa.LoadFromDataElement(de))
            {
                foreach (string str in expectedSiemensTags)
                {
                    if (csa.FindCSAElementByName(str))
                    {
                        using (CSAElement elem = csa.GetCSAElementByName(str))
                        {
                            if (elem != null)
                            {
                                System.Console.WriteLine( elem.toString() );
                            }
                        }
                    }
                }
            }
        }
    }
}

return 0;
}
}

```

## 12.21 ExtractEncapsulatedFile.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * This example shows how one from C# context can extract a binary blob
 * and write out as a file.
 * This example is meant for pdf encapsulated file, but can be adapted for other type
 * of binary blob.
 *
 * DICOM file is:
 * ...
 * (0042,0010) ST (no value available) # 0, 0 DocumentTitle
 * (0042,0011) OB 25\50\44\46\2d\31\2e\32\20\0d\25\e2\e3\cf\d3\20\0d\31\30\20\30\20... # 40718, 1
 * EncapsulatedDocument
 * (0042,0012) LO [application/pdf] # 16, 1 MIMETimeTypeOfEncapsulatedDocument
 * ...
 *
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/ExtractEncapsulatedFile.exe some_pdf_encapsulated.dcm
 */
using System;
using gdcm;

public class ExtractEncapsulatedFile
{
    public static int Main(string[] args)
    {
        string file = args[0];
        Reader reader = new Reader();
        reader.SetFileName( file );
        bool ret = reader.Read();
        if( !ret )
        {
            return 1;
        }
    }
}

```

```

    }

    File f = reader.GetFile();
    DataSet ds = f.GetDataSet();
    Tag tencapsulated_stream = new Tag(0x0042, 0x0011); // Encapsulated Document
    if( !ds.FindDataElement( tencapsulated_stream ) )
    {
        return 1;
    }
    // else
    DataElement de = ds.GetDataElement( tencapsulated_stream );
    ByteValue bv = de.GetByteValue();
    uint len = bv.GetLength();
    byte[] encapsulated_stream = new byte[len];
    bv.GetBuffer( encapsulated_stream, len );

    // Write out the decompressed bytes
    //System.Console.WriteLine(image.toString());
    using (System.IO.Stream stream =
        System.IO.File.Open(@"tmp/dd.pdf",
            System.IO.FileMode.Create))
    {
        System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
        writer.Write( encapsulated_stream );
    }

    return 0;
}

```

## 12.22 ExtractImageRegion.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * This small code shows how to use the gdcm.ImageRegionReader API
 * In this example we are taking each frame by frame and dump them to
 * /tmp/frame.raw.
 *
 * Usage:
 * $ bin/ExtractImageRegion.exe input.dcm
 *
 * Example:
 * $ bin/ExtractImageRegion.exe gdcmData/012345.002.050.dcm
 * $ md5sum /tmp/frame.raw
 * d594a5e2fde12f32b6633ca859b4d4a6 /tmp/frame.raw
 * $ gdcminfo --md5sum gdcmData/012345.002.050.dcm
 * [...]
 * md5sum: d594a5e2fde12f32b6633ca859b4d4a6
 */
using System;
using gdcm;

public class ExtractImageRegion
{
    public static int Main(string[] args)
    {
        string filename = args[0];

        uint file_size = gdcm.PosixEmulation.FileSize(filename);

        // instantiate the reader:
        gdcm.ImageRegionReader reader = new gdcm.ImageRegionReader();
    }
}

```

```

reader.SetFileName( filename );

// pull DICOM info:
if (!reader.ReadInformation()) return 1;

// store current offset:
uint cur_pos = reader.GetStreamCurrentPosition();

uint remaining = file_size - cur_pos;

Console.WriteLine("Remaining bytes to read (Pixel Data): " + remaining.ToString() );

// Get file infos
gdcm.File f = reader.GetFile();

// get some info about image
UIntArrayType dims = ImageHelper.GetDimensionsValue(f);
PixelFormat pf = ImageHelper.GetPixelFormatValue(f);
int pixelsize = pf.GetPixelSize();
PhotometricInterpretation pi = ImageHelper.GetPhotometricInterpretationValue(f);
Console.WriteLine( pi.ToString() );

// buffer to get the pixels
byte[] buffer = new byte[ dims[0] * dims[1] * pixelsize ];

// define a simple box region.
BoxRegion box = new BoxRegion();
for (uint z = 0; z < dims[2]; z++)
{
    // Define that I want the image 0, full size (dimx x dimy pixels)
    // and do that for each z:
    box.SetDomain(0, dims[0] - 1, 0, dims[1] - 1, z, z);
    //System.Console.WriteLine( box.ToString() );
    reader.SetRegion( box );

    // reader will try to load the uncompressed image region into buffer.
    // the call returns an error when buffer.Length is too small. For instance
    // one can call:
    // uint buf_len = reader.ComputeBufferLength(); // take into account pixel size
    // to get the exact size of minimum buffer
    if (reader.ReadIntoBuffer(buffer, (uint)buffer.Length))
    {
        using (System.IO.Stream stream =
            System.IO.File.Open(@"tmp/frame.raw",
                System.IO.FileMode.Create))
        {
            System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
            writer.Write(buffer);
        }
    }
    else
    {
        throw new Exception("can't read pixels error");
    }
}

return 0;
}
}

```

## 12.23 ExtractImageRegionWithLUT.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*

```



```

* This small code shows how to use the gdcm.ImageRegionReader API
* In this example we are taking each frame by frame and dump them to
* /tmp/frame.raw.
* Furthermore we are applying the LUT on this image.
* Special care should be taken in case the image is not PALETTE COLOR
*
* Usage:
* $ bin/ExtractImageRegionWithLUT.exe input.dcm
*
* Example:
* $ bin/ExtractImageRegionWithLUT.exe gdcmData/rle16l00.dcm
* $ md5sum /tmp/frame_rgb.raw
* 73bf61325fdb6e2830244a2b7b0c4ae2 /tmp/frame_rgb.raw
* $ gdcming --depth 16 --spp 3 --size 600,430 /tmp/frame_rgb.raw rgb.dcm
* $ gdcmvviewer rgb.dcm
*/
using System;
using gdcm;

public class ExtractImageRegion
{
    public static int Main(string[] args)
    {
        string filename = args[0];

        // instantiate the reader:
        gdcm.ImageRegionReader reader = new gdcm.ImageRegionReader();
        reader.SetFileName( filename );

        // pull DICOM info:
        if (!reader.ReadInformation()) return 1;
        // Get file infos
        gdcm.File f = reader.GetFile();

        gdcm.LookupTable lut = reader.GetImage().GetLUT();

        // get some info about image
        UIntArrayType dims = ImageHelper.GetDimensionsValue(f);
        PixelFormat pf = ImageHelper.GetPixelFormatValue(f);
        int pixelsize = pf.GetPixelSize();

        // buffer to get the pixels
        byte[] buffer = new byte[ dims[0] * dims[1] * pixelsize ];

        // output buffer for the RGB decoded image:
        byte[] buffer2 = new byte[ dims[0] * dims[1] * pixelsize * 3 ];

        // define a simple box region.
        BoxRegion box = new BoxRegion();
        for (uint z = 0; z < dims[2]; z++)
        {
            // Define that I want the image 0, full size (dimx x dimy pixels)
            // and do that for each z:
            box.SetDomain(0, dims[0] - 1, 0, dims[1] - 1, z, z);
            //System.Console.WriteLine( box.toString() );
            reader.SetRegion( box );

            // reader will try to load the uncompressed image region into buffer.
            // the call returns an error when buffer.Length is too small. For instance
            // one can call:
            // uint buf_len = reader.ComputeBufferLength(); // take into account pixel size
            // to get the exact size of minimum buffer
            if (reader.ReadIntoBuffer(buffer, (uint)buffer.Length))
            {
                if (!lut.Decode( buffer2, (uint)buffer2.Length, buffer, (uint)buffer.Length ))
                {
                    throw new Exception("can't decode");
                }

                using (System.IO.Stream stream =
                    System.IO.File.Open(@"tmp/frame_rgb.raw",
                        System.IO.FileMode.Create))
                {
                    System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
                    writer.Write(buffer2);
                }
            }
            else
            {
                throw new Exception("can't read pixels error");
            }
        }
    }
}

```

```

    }
    return 0;
}
}

```

## 12.24 ExtractOneFrame.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/

/*
 * This small code shows how to use the gdcm.StreamImageReader API
 * to read a single (whole) frame at a time
 * The API allow extracting a smaller extent of the frame of course.
 * It will write out the extracted frame in /tmp/frame.raw
 *
 * Usage:
 * $ bin/ExtractOneFrame.exe input.dcm
 */
using System;
using gdcm;

public class ExtractOneFrame
{
    public static int Main(string[] args)
    {
        string filename = args[0];

        gdcm.StreamImageReader reader = new gdcm.StreamImageReader();

        reader.SetFileName( filename );

        if (!reader.ReadImageInformation()) return 1;
        // Get file infos
        gdcm.File f = reader.GetFile();

        // get some info about image
        UIntArrayType extent = ImageHelper.GetDimensionsValue(f);
        //System.Console.WriteLine( extent[0] );
        uint dimx = extent[0];
        //System.Console.WriteLine( extent[1] );
        uint dimy = extent[1];
        //System.Console.WriteLine( extent[2] );
        uint dimz = extent[2];
        PixelFormat pf = ImageHelper.GetPixelFormatValue( f);
        int pixelsize = pf.GetPixelSize();
        //System.Console.WriteLine( pixelsize );

        // buffer to get the pixels
        byte[] buffer = new byte[ dimx * dimy * pixelsize ];

        for (int i = 0; i < dimz; i++)
        {
            // Define that I want the image 0, full size (dimx x dimy pixels)
            reader.DefinePixelExtent(0, (ushort)dimx, 0, (ushort)dimy, (ushort)i, (ushort)(i+1));
            uint buf_len = reader.DefineProperBufferLength(); // take into account pixel size
            //System.Console.WriteLine( buf_len );
            if( buf_len > buffer.Length )
            {
                throw new Exception("buffer is too small for target");
            }

            if (reader.Read(buffer, (uint)buffer.Length))
            {
                using (System.IO.Stream stream =

```

```

        System.IO.File.Open(@"tmp/frame.raw",
            System.IO.FileMode.Create))
        {
            System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
            writer.Write(buffer);
        }
    }
    else
    {
        throw new Exception("can't read pixels error");
    }
}

return 0;
}
}

```

## 12.25 FileAnonymize.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Simple C# example
 *
 * Usage:
 * $ mono bin/FileAnonymize.exe input.dcm output.dcm
 */
using System;
using gdcml;

public class FileAnonymize
{
    public static int Main(string[] args)
    {
        string filename = args[0];
        string outfilename = args[1];

        gdcml.FileAnonymizer fa = new gdcml.FileAnonymizer();
        fa.SetInputFileName( filename );
        fa.SetOutputFileName( outfilename );

        // Empty Operations
        // It will create elements, since those tags are non-registered public elements (2011):
        fa.Empty( new Tag(0x0008,0x1313) );
        fa.Empty( new Tag(0x0008,0x1317) );
        // Remove Operations
        // The following Tag are actually carefully chosen, since they refer to SQ:
        fa.Remove( new Tag(0x0008,0x2112) );
        fa.Remove( new Tag(0x0008,0x9215) );
        // Replace Operations
        // do not call replace operation on SQ attribute !
        fa.Replace( new Tag(0x0018,0x5100), "MYVALUE " );
        fa.Replace( new Tag(0x0008,0x1160), "MYOTHERVAL" );

        if ( !fa.Write() )
        {
            System.Console.WriteLine( "Could not write" );
            return 1;
        }

        return 0;
    }
}

```

## 12.26 FileChangeTS.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Simple C# example
 *
 * Shows multiple steps:
 * Steps 1.
 * Create a fake (dummy) DICOM file, with size 512 x 512 x 2 We use a small
 * image to be able to create the volume in memory Of course you can use any
 * existing DICOM instead
 *
 * Step 2.
 * Hack the DICOM file to pretend the number of frames is 1000 (instead of 2)
 * At this point in time this makes the DICOM file invalid (truncated). But the
 * next step will fix this.
 *
 * Step 3.
 * Use C# to create a binary data which will represent our source object for
 * image.
 *
 * Step 4.
 * We use gdcm.FileStreamer to merge the template DICOM file from Step 2, with
 * the binary data from Step 3. We decide to read a scanline at a time, but
 * this can be read with any number of bytes. AppendToDataElement() will always
 * do the proper computation.
 *
 * Step 5.
 * We compress this gigantic file, into [JPEG Lossless, Non-Hierarchical,
 * First-Order Prediction (Process 14 [Selection Value 1])]
 *
 * Usage:
 * $ mono bin/FileChangeTS.exe small.dcm big.dcm raw.data merge.dcm jpeg.dcm
 */
using System;
using System.IO;
using gdcm;

public class FileChangeTS
{
    public static byte[] StrToByteArray(string str)
    {
        System.Text.ASCIIEncoding encoding=new System.Text.ASCIIEncoding();
        return encoding.GetBytes(str);
    }
    // Create a 256 x 256 Secondary Capture Image Storage
    static private void CreateSmallDICOM(string fileName)
    {
        using( var writer = new gdcm.PixmapWriter() )
        {
            gdcm.Pixmap img = writer.GetImage();
            img.SetNumberOfDimensions( 3 );
            img.SetDimension(0, 512 );
            img.SetDimension(1, 512 );
            img.SetDimension(2, 2 ); // fake a 3d volume
            PhotometricInterpretation pi = new PhotometricInterpretation( PhotometricInterpretation.PIType.MONOCHROME2
            );
            img.SetPhotometricInterpretation( pi );
            gdcm.DataElement pixeldata = new gdcm.DataElement( new gdcm.Tag(0x7fe0,0x0010) );
            byte[] buffer = new byte[ 512 * 512 * 2 ];
            pixeldata.SetByteValue( buffer, new gdcm.VL((uint)buffer.Length) );
            img.SetDataElement( pixeldata );

            gdcm.File file = writer.GetFile();
            gdcm.DataSet ds = file.GetDataSet();
            gdcm.DataElement ms = new gdcm.DataElement(new gdcm.Tag(0x0008,0x0016));
            string mediastorage = "1.2.840.10008.5.1.4.1.1.7.2"; // Multi-frame Grayscale Byte Secondary Capture Image

```

```

        Storage
        byte[] val = StrToByteArray(mediastorage);
        ms.SetByteValue( val, new gdcm.VL( (uint)val.Length) );
        ds.Insert( ms );

        writer.SetFileName( fileName );
        writer.Write();
    }
}

static private void CreateBigDICOM(string fileName, string outfilename)
{
    using( var ano = new gdcm.FileAnonymizer() )
    {
        // The following is somewhat dangerous, do not try at home:
        string nframes = "1000";
        ano.Replace( new gdcm.Tag(0x0028,0x0008), nframes );
        ano.SetInputFileName(fileName);
        ano.SetOutputFileName(outfilename);
        ano.Write(); // at this point the DICOM is invalid !
    }
}

static private void CreateDummyFile(string fileName, long length)
{
    using (var fileStream = new FileStream(fileName, FileMode.Create, FileAccess.Write, FileShare.None))
    {
        // Looks like C# always init to 0 (fallocate ?)
        // For the purpose of the test we could add some random noise
        fileStream.SetLength(length);
    }
}

static private void ReadBytesIntoArray( byte[] array, FileStream source )
{
    int numBytesToRead = array.Length;
    int numBytesRead = 0;
    while (numBytesToRead > 0)
    {
        // According to spec: Read() may return anything from 0 to numBytesToRead.
        int n = source.Read(array, numBytesRead, numBytesToRead);

        // Break when the end of the file is reached.
        if (n == 0)
            break;

        numBytesRead += n;
        numBytesToRead -= n;
    }
}

static private void AssembleDICOMAndRaw(string dicomfn, string rawdata, string outfn)
{
    using ( var fs = new gdcm.FileStreamer() )
    {
        fs.SetTemplateFileName(dicomfn);
        fs.SetOutputFileName(outfn);
        gdcm.Tag pixeldata = new gdcm.Tag(0x7fe0, 0x0010);
        // FileStreamer support automatic checking of pixel data length
        // based on DICOM attributes, only if we say so:
        fs.CheckDataElement( pixeldata );
        // Declare we are working on Pixel Data attribute:
        fs.StartDataElement( pixeldata );
        using (FileStream rawSource = new FileStream(rawdata,
            FileMode.Open, FileAccess.Read))
        {
            byte[] bytes = new byte[512];
            // Only read one scanline at a time
            // We could have been reading more at once, if this is more efficient,
            // AppendToDataElement will do the logic in all cases.
            for( int i = 0; i < 512 * 1000; ++i )
            {
                // Read the source file into a byte array.
                ReadBytesIntoArray( bytes, rawSource );
                fs.AppendToDataElement( pixeldata, bytes, (uint)bytes.Length );
            }
        }
        if( !fs.StopDataElement( pixeldata ) )
        {
            // Most likely an issue with Pixel Data Length computation:
            throw new Exception("StopDataElement failed");
        }
    }
}

static private void CompressIntoJPEG(string rawdicom, string jpegdicom)

```

```

{
using( var sfcts = FileChangeTransferSyntax.New() )
{
    // Need to retrieve the actual C++ reference, to pass to
    // SimpleSubjectWatcher:
    FileChangeTransferSyntax fcts = sfcts.__ref__();
    SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(fcts, "FileChangeTransferSyntax");
    gdcm.TransferSyntax ts = new TransferSyntax( TransferSyntax.TType.JPEGLosslessProcess14_1 );
    fcts.SetTransferSyntax( ts );
    fcts.SetInputFileName( rawdicom );
    fcts.SetOutputFileName( jpegdicom );
    fcts.Change();
}
}
public static int Main(string[] args)
{
    string filename = args[0];
    string outfilename = args[1];
    string rawfilename = args[2];
    string mergefn = args[3];
    string jpegfn = args[4];

    CreateSmallDICOM(filename);
    CreateBigDICOM(filename, outfilename);
    CreateDummyFile(rawfilename, 512 * 512 * 1000 );
    AssembleDICOMAndRaw(outfilename, rawfilename, mergefn);
    CompressIntoJPEG(mergefn, jpegfn);

    return 0;
}
}

```

## 12.27 FileChangeTSLossy.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Simple C# example
 *
 * Shows multiple steps:
 * Steps 1.
 * Create a fake (dummy) DICOM file, with size 512 x 512 x 2 We use a small
 * image to be able to create the volume in memory Of course you can use any
 * existing DICOM instead
 *
 * Step 2.
 * Hack the DICOM file to pretend the number of frames is 1000 (instead of 2)
 * At this point in time this makes the DICOM file invalid (truncated). But the
 * next step will fix this.
 *
 * Step 3.
 * Use C# to create a binary data which will represent our source object for
 * image.
 *
 * Step 4.
 * We use gdcm.FileStreamer to merge the template DICOM file from Step 2, with
 * the binary data from Step 3. We decide to read a scanline at a time, but
 * this can be read with any number of bytes. AppendToDataElement() will always
 * do the proper computation.
 *
 * Step 5.
 * We compress this gigantic file, into [JPEG Baseline (Process 1): Default Transfer Syntax for Lossy JPEG 8 Bit
 * Image Compression]
 *
 * Usage:

```

```

* $ bin/FileChangeTSLossy.exe small.dcm big.dcm raw.data merge.dcm jpeg.dcm
*/
using System;
using System.IO;
using gdcm;

public class FileChangeTS
{
    public static byte[] StrToByteArray(string str)
    {
        System.Text.ASCIIEncoding encoding=new System.Text.ASCIIEncoding();
        return encoding.GetBytes(str);
    }
    // Create a 256 x 256 Secondary Capture Image Storage
    static private void CreateSmallDICOM(string fileName)
    {
        using( var writer = new gdcm.PixmapWriter() )
        {
            gdcm.Pixmap img = writer.GetImage();
            img.SetNumberOfDimensions( 3 );
            img.SetDimension(0, 512 );
            img.SetDimension(1, 512 );
            img.SetDimension(2, 2 ); // fake a 3d volume
            PhotometricInterpretation pi = new PhotometricInterpretation( PhotometricInterpretation.PIType.MONOCHROME2
            );
            img.SetPhotometricInterpretation( pi );
            gdcm.DataElement pixeldata = new gdcm.DataElement( new gdcm.Tag(0x7fe0,0x0010) );
            byte[] buffer = new byte[ 512 * 512 * 2 ];
            pixeldata.SetByteValue( buffer, new gdcm.VL((uint)buffer.Length) );
            img.SetDataElement( pixeldata );

            gdcm.File file = writer.GetFile();
            gdcm.DataSet ds = file.GetDataSet();
            gdcm.DataElement ms = new gdcm.DataElement(new gdcm.Tag(0x0008,0x0016));
            string mediastorage = "1.2.840.10008.5.1.4.1.1.7.2"; // Multi-frame Grayscale Byte Secondary Capture Image
            Storage
            byte[] val = StrToByteArray(mediastorage);
            ms.SetByteValue( val, new gdcm.VL( (uint)val.Length) );
            ds.Insert( ms );

            writer.SetFileName( fileName );
            writer.Write();
        }
    }
    static private void CreateBigDICOM(string fileName, string outfilename)
    {
        using( var ano = new gdcm.FileAnonymizer() )
        {
            // The following is somewhat dangerous, do not try at home:
            string nframes = "1000";
            ano.Replace( new gdcm.Tag(0x0028,0x0008), nframes );
            ano.SetInputFileName(fileName);
            ano.SetOutputFileName(outfilename);
            ano.Write(); // at this point the DICOM is invalid !
        }
    }
    static private void CreateDummyFile(string fileName, long length)
    {
        using (var fileStream = new FileStream(fileName, FileMode.Create, FileAccess.Write, FileShare.None))
        {
            // Looks like C# always init to 0 (fallocate ?)
            // For the purpose of the test we could add some random noise
            fileStream.SetLength(length);
        }
    }
    static private void ReadBytesIntoArray( byte[] array, FileStream source )
    {
        int numBytesToRead = array.Length;
        int numBytesRead = 0;
        while (numBytesToRead > 0)
        {
            // According to spec: Read() may return anything from 0 to numBytesToRead.
            int n = source.Read(array, numBytesRead, numBytesToRead);

            // Break when the end of the file is reached.
            if (n == 0)
                break;

            numBytesRead += n;
            numBytesToRead -= n;
        }
    }
}

```

```

    }
    static private void AssembleDICOMAndRaw(string dicomfn, string rawdata, string outfn)
    {
        using ( var fs = new gdcm.FileStreamer() )
        {
            fs.SetTemplateFileName(dicomfn);
            fs.SetOutputFileName(outfn);
            gdcm.Tag pixeldata = new gdcm.Tag(0x7fe0, 0x0010);
            // FileStreamer support automatic checking of pixel data length
            // based on DICOM attributes, only if we say so:
            fs.CheckDataElement( pixeldata );
            // Declare we are working on Pixel Data attribute:
            fs.StartDataElement( pixeldata );
            using (FileStream rawSource = new FileStream(rawdata,
                FileMode.Open, FileAccess.Read))
            {
                byte[] bytes = new byte[512];
                // Only read one scanline at a time
                // We could have been reading more at once, if this is more efficient,
                // AppendToDataElement will do the logic in all cases.
                for( int i = 0; i < 512 * 1000; ++i )
                {
                    // Read the source file into a byte array.
                    ReadBytesIntoArray( bytes, rawSource );
                    fs.AppendToDataElement( pixeldata, bytes, (uint)bytes.Length );
                }
            }
            if( !fs.StopDataElement( pixeldata ) )
            {
                // Most likely an issue with Pixel Data Length computation:
                throw new Exception("StopDataElement failed");
            }
        }
    }
    static private void CompressIntoJPEG(string rawdicom, string jpegdicom)
    {
        using( var sfcts = FileChangeTransferSyntax.New() )
        {
            // Need to retrieve the actual C++ reference, to pass to
            // SimpleSubjectWatcher:
            FileChangeTransferSyntax fcts = sfcts.__ref__();
            SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(fcts, "FileChangeTransferSyntax");
            gdcm.TransferSyntax ts = new TransferSyntax( TransferSyntax.TType.JPEGBaselineProcess1 );
            fcts.SetTransferSyntax( ts );
            ImageCodec ic = fcts.GetCodec();
            JPEGCodec jpeg = JPEGCodec.Cast( ic );
            jpeg.SetLossless( false );
            jpeg.SetQuality( 50 ); // poor quality !

            fcts.SetInputFileName( rawdicom );
            fcts.SetOutputFileName( jpegdicom );
            fcts.Change();
        }
    }
    public static int Main(string[] args)
    {
        string filename = args[0];
        string outfilename = args[1];
        string rawfilename = args[2];
        string mergefn = args[3];
        string jpegfn = args[4];

        CreateSmallDICOM(filename);
        CreateBigDICOM(filename, outfilename);
        CreateDummyFile(rawfilename, 512 * 512 * 1000 );
        AssembleDICOMAndRaw(outfilename, rawfilename, mergefn);
        CompressIntoJPEG(mergefn, jpegfn);

        return 0;
    }
}

```

## 12.28 FileStreaming.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

```



```

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

    This software is distributed WITHOUT ANY WARRANTY; without even
    the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
    PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Simple C# example
 *
 * Usage:
 * $ mono bin/FileStreaming.exe gdcmData/CT_16b_signed-UsedBits13.dcm output.dcm
 *
 * The class will take care of group handling and will use the first available group:
 * (0009,0012) ?? (LO) [MYTEST] # 6,1 Private Creator
 */
using System;
using gdcm;

public class FileStreaming
{
    public static int Main(string[] args)
    {
        string filename = args[0];
        string outfilename = args[1];

        gdcm.PrivateTag pt = new gdcm.PrivateTag( new gdcm.Tag(0x9,0x10), "MYTEST" );

        gdcm.FileStreamer fs = new gdcm.FileStreamer();
        fs.SetTemplateFileName( filename );
        fs.SetOutputFileName( outfilename );

        byte[] buffer = new byte[ 8192 ];
        uint len = (uint)buffer.Length;

        // In this example, we want that each newly created Private Attribute
        // contains at most 1000 bytes of incoming dataset.
        // We are also calling the function twice to check that appending mode is
        // working from one call to the other. The last element will have a length
        // of (2 * 8192) % 1000 = 384
        if( !fs.StartGroupDataElement( pt, 1000, 1 )
            || !fs.AppendToGroupDataElement( pt, buffer, len )
            || !fs.AppendToGroupDataElement( pt, buffer, len )
            || !fs.StopGroupDataElement( pt ) )
        {
            System.Console.WriteLine( "Could not change private group" );
            return 1;
        }

        return 0;
    }
}

```

## 12.29 GetArray.cs

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

    This software is distributed WITHOUT ANY WARRANTY; without even
    the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
    PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/GetArray.exe gdcmData/012345.002.050.dcm

```

```

*/
using System;
using gdcm;

public class GetArray
{
    public static int Main(string[] args)
    {
        string file1 = args[0];
        ImageReader reader = new ImageReader();
        reader.SetFileName( file1 );
        bool ret = reader.Read();
        if( !ret )
        {
            return 1;
        }

        Image image = reader.GetImage();

        PixelFormat pixeltype = image.GetPixelFormat();

        if( image.GetNumberOfDimensions() != 2 )
        {
            // For the purpose of the test, exit early on
            return 1;
        }
        uint dimx = image.GetDimension(0);
        uint dimy = image.GetDimension(1);
        uint npixels = dimx * dimy;
        //LookupTable lut = image.GetLUT();
        //uint r1 = lut.GetLUTLength( LookupTable.LookupTableType.RED );
        //byte[] rbuf = new byte[ r1 ];
        //uint r12 = lut.GetLUT( LookupTable.LookupTableType.RED, rbuf );
        //assert r1 == r12;

        //byte[] str1 = new byte[ image.GetBufferLength()];
        //image.GetBuffer( str1 );
        if( pixeltype.GetScalarType() == PixelFormat.ScalarType.UINT8 )
        {
            System.Console.WriteLine( "Processing UINT8 image type" );
            byte[] str1 = new byte[ npixels ];
            image.GetArray( str1 );
        }
        else if( pixeltype.GetScalarType() == PixelFormat.ScalarType.INT16 )
        {
            System.Console.WriteLine( "Processing INT16 image type" );
            short[] str1 = new short[ npixels ];
            image.GetArray( str1 );
        }
        else if( pixeltype.GetScalarType() == PixelFormat.ScalarType.UINT16 )
        {
            System.Console.WriteLine( "Processing UINT16 image type" );
            ushort[] str1 = new ushort[ npixels ];
            image.GetArray( str1 );
        }
        else
        {
            //System.Console.WriteLine( "Default (unhandled pixel format): " + pixeltype.toString() );
            System.Console.WriteLine( "Default (unhandled pixel format): " + pixeltype.GetScalarTypeAsString() );
            // Get bytes
            byte[] str1 = new byte[ image.GetBufferLength()];
            image.GetBuffer( str1 );
        }

        return 0;
    }
}

```

## 12.30 MpegVideoInfo.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

```

```

    This software is distributed WITHOUT ANY WARRANTY; without even
    the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
    PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * This examples takes in a MPEG2 and write out a Video Endoscopic Imagae Storage
 * encoded using MPEG2 @ Main Profile
 * ref: http://chrisa.wordpress.com/2007/11/21/decoding-mpeg2-information/
 * See also:
 * http://dvd.sourceforge.net/dvdinfo/mpeghdrs.html#gop
 * http://cvs.linux.hr/cgi-bin/viewcvs.cgi/mpeg_mod/README.infrompeg?view=markup
 * http://www.guru-group.fi/~too/sw/m2vmp2cut/mpeg2info.c
 */

/*
 * Provides information about an MPEG2 file, including the duration, frame rate, aspect
 * ratio, and resolution. Good information about the MPEG2 file structure that helps
 * explain parts of the code can be found here:
 * http://dvd.sourceforge.net/dvdinfo/mpeghdrs.html#gop
 *
 * Copyright (c) 2007 Chris Anderson (chrisa@wordpress.com)
 *
 * This library is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 2 of the License, or (at your option) any later version.
 *
 * This library is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
 * Lesser General Public License for more details.
 */
using System;
using System.IO;
using gdc;

public class Mpeg2VideoInfo
{
    #region Member Variables
    private TimeSpan m_startTime = TimeSpan.Zero;
    private TimeSpan m_endTime = TimeSpan.Zero;
    private TimeSpan m_duration = TimeSpan.Zero;
    private eAspectRatios m_aspectRatio = eAspectRatios.Invalid;
    private eFrameRates m_frameRate = 0;
    private int m_pictureWidth = 0;
    private int m_pictureHeight = 0;
    #endregion

    #region Constants
    private const byte PADDING_PACKET = 0xBE;
    private const byte VIDEO_PACKET = 0xE0;
    private const byte AUDIO_PACKET = 0xC0;
    private const byte SYSTEM_PACKET = 0xBB;
    private const byte TIMESTAMP_PACKET = 0xB8;
    private const byte HEADER_PACKET = 0xB3;

    private const int BUFFER_SIZE = 8162; // 8K buffer

    private readonly static TimeSpan EMPTY_TIMESPAN = new TimeSpan(0, 0, -1);
    #endregion

    #region Enumerations
    public enum eFrameRates
    {
        Invalid,
        PulldownNTSC, // 24000d/1001d = 23.976 Hz
        Film, // 24 Hz
        PAL, // 25 Hz
        NTSC, // 30000d/1001d = 29.97 Hz
        DropFrameNTSC, // 30 Hz
        DoubleRatePAL, // 50 Hz
        DoubleRateNTSC, // 59.97 Hz
        DoubleRateDropFrameNTSC // 60 Hz
    }

    public enum eAspectRatios
    {
        Invalid,
        VGA, // 1/1
    }
    }

```

```

        StandardTV, // 4/3
        LargeTV,   // 16/9
        Cinema     // 2.21/1
    }
}
#endregion

#region Constructor
public Mpeg2VideoInfo(string file)
{
    ParseMpeg(file);
}
#endregion

#region Public Properties
public TimeSpan StartTime
{
    get { return m_startTime; }
}

public TimeSpan EndTime
{
    get { return m_endTime; }
}

public TimeSpan Duration
{
    get { return m_duration; }
}

public eAspectRatios AspectRatio
{
    get { return m_aspectRatio; }
}

public eFrameRates FrameRate
{
    get { return m_frameRate; }
}

public int PictureWidth
{
    get { return m_pictureWidth; }
}

public int PictureHeight
{
    get { return m_pictureHeight; }
}
#endregion

#region Private Functions
private void ParseMpeg(string file)
{
    FileStream fs = new FileStream(file, FileMode.Open, FileAccess.Read, FileShare.ReadWrite);
    BinaryReader br = new BinaryReader(fs);

    m_startTime = GetStartTimeStampInfo(br);
    m_endTime = GetEndTimeStampInfo(br);

    m_duration = m_endTime.Subtract(m_startTime);

    GetHeaderInfo(br);

    br.Close();
    fs.Close();
}

private TimeSpan GetStartTimeStampInfo(BinaryReader br)
{
    TimeSpan startTime = EMPTY_TIMESPAN;
    byte[] buffer = new byte[BUFFER_SIZE];

    br.BaseStream.Seek(0, SeekOrigin.Begin);

    while (startTime == EMPTY_TIMESPAN && br.BaseStream.Position < br.BaseStream.Length)
    {
        int readBytes = br.Read(buffer, 0, BUFFER_SIZE);

        for (int offset = 0; offset < readBytes - 8; offset++)
        {
            if (IsStreamMarker(ref buffer, offset, TIMESTAMP_PACKET))

```

```

        {
            offset += 4; // Move to the data position which follows the stream header
            uint timeStampEncoded = GetData(ref buffer, offset);
            startTime = DecodeTimeStamp(timeStampEncoded);

            if (startTime != EMPTY_TIMESPAN)
                break;
        }
    }

    return startTime;
}

private TimeSpan GetEndTimeStampInfo(BinaryReader br)
{
    TimeSpan endTime = EMPTY_TIMESPAN;
    byte[] buffer = new byte[BUFFER_SIZE];

    br.BaseStream.Seek(-BUFFER_SIZE, SeekOrigin.End);

    while (endTime == EMPTY_TIMESPAN && br.BaseStream.Position > BUFFER_SIZE)
    {
        int readBytes = br.Read(buffer, 0, BUFFER_SIZE);

        for (int offset = readBytes - 8; offset >= 0; offset--)
        {
            if (IsStreamMarker(ref buffer, offset, TIMESTAMP_PACKET))
            {
                offset += 4; // Move to the data position which follows the stream header
                uint timeStampEncoded = GetData(ref buffer, offset);
                endTime = DecodeTimeStamp(timeStampEncoded);

                if (endTime != EMPTY_TIMESPAN)
                    break;
            }
        }

        br.BaseStream.Seek(-BUFFER_SIZE * 2, SeekOrigin.Current);
    }

    return endTime;
}

private TimeSpan DecodeTimeStamp(uint timeStampEncoded)
{
    TimeSpan timeStamp = EMPTY_TIMESPAN;

    // Mask out the bits containing the property we are after, then
    // shift the data to the right to get its value
    int hour = (int)(timeStampEncoded & 0x7C000000) >> 26; // Bits 31 -> 27
    int minute = (int)(timeStampEncoded & 0x03F00000) >> 20; // Bits 26 -> 21
    int second = (int)(timeStampEncoded & 0x0007E000) >> 13; // Bits 19 -> 14
    int frame = (int)(timeStampEncoded & 0x00001F80) >> 7; // Bits 13 -> 8 - not used, but included for
    completeness

    timeStamp = new TimeSpan(hour, minute, second);
    return timeStamp;
}

private void GetHeaderInfo(BinaryReader br)
{
    byte[] buffer = new byte[BUFFER_SIZE];

    br.BaseStream.Seek(0, SeekOrigin.Begin);
    br.Read(buffer, 0, BUFFER_SIZE);

    for (int offset = 0; offset < buffer.Length - 4; offset++)
    {
        if (IsStreamMarker(ref buffer, offset, HEADER_PACKET))
        {
            offset += 4; // Move to the data position which follows the stream header
            uint headerData = GetData(ref buffer, offset);

            // Mask out the bits containing the property we are after, then
            // shift the data to the right to get its value
            m_pictureWidth = (int)(headerData & 0xFFF00000) >> 20;
            m_pictureHeight = (int)(headerData & 0x000FFF00) >> 8;

            uint aspectRatioIndex = (headerData & 0x000000F0) >> 4;

```

```

        uint fpsIndex = headerData & 0x0000000F;

        m_aspectRatio = (eAspectRatios)fpsIndex;
        m_frameRate = (eFrameRates)fpsIndex;

        break;
    }
}

private uint GetData(ref byte[] buffer, int offset)
{
    return (uint) ((buffer[offset] << 24) |
        (buffer[offset + 1] << 16) |
        (buffer[offset + 2] << 8) |
        (buffer[offset + 3]));
}

private bool IsStreamMarker(ref byte[] buffer, int offset, byte markerType)
{
    return (buffer[offset] == 0x00 &&
        buffer[offset + 1] == 0x00 &&
        buffer[offset + 2] == 0x01 &&
        buffer[offset + 3] == markerType);
}
#endregion
public static int Main(string[] args)
{
    string file1 = args[0];
    Mpeg2VideoInfo info = new Mpeg2VideoInfo(file1);
    System.Console.WriteLine( info.StartTime );
    System.Console.WriteLine( info.EndTime );
    System.Console.WriteLine( info.Duration );
    System.Console.WriteLine( info.AspectRatio );
    System.Console.WriteLine( info.FrameRate );
    System.Console.WriteLine( info.PictureWidth );
    System.Console.WriteLine( info.PictureHeight );

    ImageReader r = new ImageReader();
    //Image image = new Image();
    Image image = r.GetImage();
    image.SetNumberOfDimensions( 3 );
    DataElement pixeldata = new DataElement( new gdcm.Tag(0x7fe0,0x0010) );

    System.IO.FileStream infile =
        new System.IO.FileStream(file1, System.IO.FileMode.Open, System.IO.FileAccess.Read);
    uint fsize = gdcm.PosixEmulation.FileSize(file1);

    byte[] jstream = new byte[fsize];
    infile.Read(jstream, 0, jstream.Length);

    SmartPtrFrag sq = SequenceOfFragments.New();
    Fragment frag = new Fragment();
    frag.SetByteValue( jstream, new gdcm.VL( (uint)jstream.Length) );
    sq.AddFragment( frag );
    pixeldata.SetValue( sq.__ref__() );

    // insert:
    image.SetDataElement( pixeldata );

    PhotometricInterpretation pi = new PhotometricInterpretation(
        PhotometricInterpretation.PIType.YBR_PARTIAL_420 );
    image.SetPhotometricInterpretation( pi );
    // FIXME hardcoded:
    PixelFormat pixeltype = new PixelFormat(3,8,8,7);
    image.SetPixelFormat( pixeltype );

    // FIXME hardcoded:
    TransferSyntax ts = new TransferSyntax( TransferSyntax.TSType.MPEG2MainProfile);
    image.SetTransferSyntax( ts );

    image.SetDimension(0, (uint)info.PictureWidth);
    image.SetDimension(1, (uint)info.PictureHeight);
    image.SetDimension(2, 721);

    ImageWriter writer = new ImageWriter();
    gdcm.File file = writer.GetFile();
    file.GetHeader().SetDataSetTransferSyntax( ts );
    Anonymizer anon = new Anonymizer();

```

```

anon.SetFile( file );

MediaStorage ms = new MediaStorage( MediaStorage.MSType.VideoEndoscopicImageStorage);

UIDGenerator gen = new UIDGenerator();
anon.Replace( new Tag(0x0008,0x16), ms.GetString() );
anon.Replace( new Tag(0x0018,0x40), "25" );
anon.Replace( new Tag(0x0018,0x1063), "40.000000" );
anon.Replace( new Tag(0x0028,0x34), "4\\3" );
anon.Replace( new Tag(0x0028,0x2110), "01" );

writer.SetImage( image );
writer.SetFileName( "dummy.dcm" );
if( !writer.Write() )
{
    System.Console.WriteLine( "Could not write" );
    return 1;
}

return 0;
}

```

## 12.31 NewSequence.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/NewSequence.exe gdcmData/012345.002.050.dcm out.dcm
 */
using System;
//using gdcm;

public class NewSequence
{
    public static byte[] StrToByteArray(string str)
    {
        System.Text.ASCIIEncoding encoding=new System.Text.ASCIIEncoding();
        return encoding.GetBytes(str);
    }

    public static int Main(string[] argv)
    {
        string file1 = argv[0];
        string file2 = argv[1];

        gdcm.Reader r = new gdcm.Reader();
        r.SetFileName( file1 );
        if ( ! r.Read() )
        {
            return 1;
        }

        gdcm.File f = r.GetFile();
        gdcm.DataSet ds = f.GetDataSet();
        // tsis = gdcm.Tag(0x0008,0x2112) # SourceImageSequence

        // Create a dataelement
        gdcm.DataElement de = new gdcm.DataElement(new gdcm.Tag(0x0010, 0x2180));
        string occ = "Occupation";
        de.SetByteValue( StrToByteArray(occ), new gdcm.VL((uint)occ.Length));
        de.SetVR(new gdcm.VR(gdcm.VR.VRType.SH));

        // Create an item

```

```

gdcM.Item it = new gdcM.Item();
it.SetVLToUndefined(); // Needed to not popup error message
//it.InsertDataElement(de)
gdcM.DataSet nds = it.GetNestedDataSet();
nds.Insert(de);

// Create a Sequence
gdcM.SmartPtrSQ sq = gdcM.SequenceOfItems.New();
sq.SetLengthToUndefined();
sq.AddItem(it);

// Insert sequence into data set
gdcM.DataElement des = new gdcM.DataElement(new gdcM.Tag(0x0400, 0x0550));
des.SetVR(new gdcM.VR(gdcM.VR.VRType.SQ));
des.SetValue(sq.__ref__());
des.SetVLToUndefined();

ds.Insert(des);

gdcM.Writer w = new gdcM.Writer();
w.SetFile( f );
w.SetFileName( file2 );
if ( !w.Write() )
    return 1;

return 0;
}
}

```

## 12.32 RescaleImage.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcM.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcM/debug-gcc/bin
 * $ mono bin/DecompressImage.exe gdcMData/012345.002.050.dcm rescaled.dcm
 */
using System;
using gdcM;

public class DecompressImage
{
    public static int Main(string[] args)
    {
        string file1 = args[0];
        ImageReader reader = new ImageReader();
        reader.SetFileName( file1 );
        bool ret = reader.Read();
        if( !ret )
        {
            return 1;
        }

        Image image = reader.GetImage();
        PixelFormat pixeltype = image.GetPixelFormat();

        Rescaler r = new Rescaler();
        r.SetIntercept( 0 );
        r.SetSlope( 1.2 );
        r.SetPixelFormat( pixeltype );
        PixelFormat outputpt = new PixelFormat( r.ComputeInterceptSlopePixelType() );

        System.Console.WriteLine( "pixeltype" );
        System.Console.WriteLine( pixeltype.ToString() );
    }
}

```



```

System.Console.WriteLine( "outputpt" );
System.Console.WriteLine( outputpt.ToString() );

uint len = image.GetBufferLength();
short[] input = new short[ len / 2 ]; // sizeof(short) == 2
image.GetArray( input );

double[] output = new double[ len / 2 ];
r.Rescale( output, input, len );

// First Pixel is:
System.Console.WriteLine( "Input:" );
System.Console.WriteLine( input[0] );

System.Console.WriteLine( "Output:" );
System.Console.WriteLine( output[0] );

return 0;
}
}

```

## 12.33 SendFileSCU.cs

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Perso/gdcm-gcc/bin
 * $ mono bin/SendFileSCU.exe server port input.dcm
 */
using System;
using gdcm;

public class SendFileSCU
{
    public static int Main(string[] args)
    {
        {
            string server = args[0];
            ushort port = ushort.Parse(args[1]);
            string filename = args[2];

            bool b = CompositeNetworkFunctions.CEcho( server, port );
            if( !b ) return 1;

            FilenamesType files = new FilenamesType();
            files.Add( filename );
            b = CompositeNetworkFunctions.CStore( server, port, files );
            if( !b ) return 1;

            return 0;
        }
    }
}

```

## 12.34 SimplePrintPatientName.cs

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.

```

See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the above copyright notice for more information.

```

=====*/
/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Perso/gdcm/debug-gcc/bin
 * $ mono bin/SimplePrintPatientName.exe gdcmData/012345.002.050.dcm
 */
/*
This example was provided by Jonathan Morra /jonmorra gmail com/
on the gdcm mailing list (Fri, 28 May 2010)
*/
using System;
using gdcm;

namespace GDCMTest
{
    class SimplePrintPatientName
    {
        static int Main(string[] args)
        {
            if (args.Length != 1)
            {
                Console.WriteLine("This program prints the patient name of a dicom file with gdcm");
                Console.WriteLine("Usage: [input.dcm]");
                return 1;
            }

            gdcm.Reader reader = new gdcm.Reader();
            reader.SetFileName(args[0]);
            bool ret = reader.Read();
            //TagSetType tst = new TagSetType();
            //tst.Add( new Tag(0x7fe0,0x10) );
            //bool ret = reader.ReadUpToTag( new Tag(0x88,0x200), tst );
            if( !ret )
            {
                return 1;
            }

            gdcm.File file = reader.GetFile();

            gdcm.StringFilter filter = new gdcm.StringFilter();
            filter.SetFile(file);
            string value = filter.ToString(new gdcm.Tag(0x0010, 0x0010));

            Console.WriteLine("Patient Name: " + value);
            return 0;
        }
    }
}

```

## 12.35 SortImage2.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/SortImage.exe gdcmData/012345.002.050.dcm out.dcm
 */
using System;

```

```

using gdcm;

public class SortImage2
{
    bool mysort(DataSet ds1, DataSet ds2)
    {
        return false;
    }

    public static int Main(string[] args)
    {
        Sorter sorter = new Sorter();
        sorter.SetSortFunction( mysort );

        return 0;
    }
}

```

## 12.36 CStoreQtProgress.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * This small example show how one can use the virtual function
 * mechanism of the SimpleSubjectWatcher class to redirect progress
 * report to a custom Qt classes
 *
 * http://doc.qt.nokia.com/latest/qprogressdialog.html
 *
 * Usage:
 * CStoreQtProgress dicom.example.com 11112 gdcmData/MR_Spectroscopy_SIEMENS_OF.dcm
 *
 */

#include "gdcmServiceClassUser.h"
#include "gdcmSimpleSubjectWatcher.h"
#include "gdcmProgressEvent.h"
#include "gdcmDirectory.h"
#include "gdcmPresentationContextGenerator.h"

#include <QApplication>
#include <QProgressDialog>
#include <QVBoxLayout>

namespace gdcm {
/*
 * This class is a little more complicated than what this example demonstrate
 * This watcher is capable of handling nested progress. Since the Progress
 * grows from [0 to 1] on a per file basis and we only have one instance of a
 * watcher per association, we need some calculation to compute the global
 * (total) progress
 * In fact we simply divide the per-file progress by the number of files.
 *
 * This QtWatcher class will then update the progress bar according to the
 * progress.
 */
class MyQtWatcher : public SimpleSubjectWatcher
{
    size_t nfiles;
    double progress;
    size_t index;
    double refprogress;
    QWidget* win;
    QProgressDialog* qtprogress;
public:

```

```

MyQtWatcher(Subject * s, const char *comment = "", QWidget *w = NULL, QProgressDialog* p = NULL, size_t n =
1):
    SimpleSubjectWatcher(s,comment),nfiles(n),progress(0),index(0),refprogress(0),win(w),qtprogress(p){}
void ShowIteration()
{
    index++;
    assert( index <= nfiles );
    // update refprogress (we are moving to the next file)
    refprogress = progress;
}
void ShowProgress(Subject *, const Event &evt)
{
    // Retrieve the ProgressEvent:
    const ProgressEvent &pe = dynamic_cast<const ProgressEvent&>(evt);
    // compute global progress:
    progress = refprogress + (1. / (double)nfiles ) * pe.GetProgress();
    // Print Global and local progress to stdout:
    std::cout << "Global Progress: " << progress << " per file progress " << pe.GetProgress() << std::endl;
    //set progress value in the QtProgress bar
    int i = (int)(progress * 100 + 0.5); // round to next int
    qtprogress->setValue(i);
    win->show();
}
virtual void ShowDataSet(Subject *caller, const Event &evt)
{
    (void)caller;
    (void)evt;
}
};
} // end namespace gdcm

int main(int argc, char *argv[])
{
    if( argc < 4 )
    {
        std::cerr << argv[0] << " remote_server port filename" << std::endl;
        return 1;
    }
    QApplication a(argc, argv);

    std::ostringstream error_log;
    gdcm::Trace::SetErrorStream( error_log );

    const char *remote = argv[1];
    int portno = atoi(argv[2]);
    const char *filename = argv[3];

    QVBoxLayout* layout = new QVBoxLayout;
    QWidget* win = new QWidget;

    QProgressDialog* progress = new QProgressDialog("Sending data...", "Cancel", 0, 100);
    progress->setWindowModality(Qt::WindowModal);

    layout->addWidget(progress,Qt::AlignCenter);
    win->setLayout(layout);

    gdcm::SmartPointer<gdcm::ServiceClassUser> scup = new gdcm::ServiceClassUser;
    gdcm::ServiceClassUser &scu = *scup;
    //gdcm::SimpleSubjectWatcher w( &scu, "TestServiceClassUser" );
    // let's use a more complicated progress reported in this example
    gdcm::MyQtWatcher w( &scu, "QtWatcher", win, progress );

    scu.SetHostname( remote );
    scu.SetPort( (uint16_t)portno );
    scu.SetTimeout( 1000 );
    scu.SetCalledAETitle( "GDCM_STORE" );

    if( !scu.InitializeConnection() )
    {
        std::cerr << "Could not InitializeConnection" << std::endl;
        return 1;
    }

    gdcm::Directory::FileNamesType filenames;
    filenames.push_back( filename );

    // setup the PC(s) based on the filenames:
    gdcm::PresentationContextGenerator generator;
    if( !generator.GenerateFromFilenames(filenames) )
    {
        std::cerr << "Could not GenerateFromFilenames" << std::endl;
    }

```

```

    return 1;
}

// Setup PresentationContext(s)
scu.SetPresentationContexts( generator.GetPresentationContexts() );

// Start ASSOCIATION
if( !scu.StartAssociation() )
{
    std::cerr << "Could not Start" << std::endl;
    return 1;
}

// Send C-STORE
if( !scu.SendStore( filename ) )
{
    std::cerr << "Could not Store" << std::endl;
    std::cerr << "Error log is:" << std::endl;
    std::cerr << error_log.str() << std::endl;
    return 1;
}

// Stop ASSOCIATION
if( !scu.StopAssociation() )
{
    std::cerr << "Could not Stop" << std::endl;
    return 1;
}

win->show();

return a.exec();
}

```

## 12.37 ChangePrivateTags.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmPrivateTag.h"

int main(int argc, char* argv[] )
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " path/to/05148044-mr-siemens-avanto-syngo.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    gdcm::Reader reader;
    reader.SetFileName( filename );
    if ( ! reader.Read() )
    {
        return 1;
    }

    // (0029,0010) LO [SIEMENS CSA HEADER]          # 18,1 Private Creator
    // (0029,0011) LO [SIEMENS MEDCOM HEADER ]       # 22,1 Private Creator
    // (0029,0012) LO [SIEMENS MEDCOM HEADER2]       # 22,1 Private Creator
    // [...]
    // (0029,1018) CS [MR]                          # 2,1 CSA Series Header Type
    // (0029,1134) CS [DB TO DICOM ]                # 12,1 PMTF Information 4

```

```

// (0029,1260) LO [com ] # 4,1 Series Workflow Status

gdcmm::File &file = reader.GetFile();
gdcmm::DataSet &ds = file.GetDataSet();

// Declare private tag we need to find:
gdcmm::PrivateTag pt1( 0x29,0x18, "SIEMENS CSA HEADER" );
gdcmm::PrivateTag pt2( 0x29,0x34, "SIEMENS MEDCOM HEADER" );
gdcmm::PrivateTag pt3( 0x29,0x60, "SIEMENS MEDCOM HEADER2" );

const char str1[] = "GDCM was here 3!";
if( !ds.FindDataElement( pt1 ) ) return 1;
gdcmm::DataElement de1 = ds.GetDataElement( pt1 ); // Convert Private tag, into actual DataElement
std::cout << de1 << std::endl;
de1.SetByteValue( str1, (uint32_t)strlen(str1) );
ds.Replace( de1 );

const char str2[] = "GDCM was here 2!";
if( !ds.FindDataElement( pt2 ) ) return 1;
gdcmm::DataElement de2 = ds.GetDataElement( pt2 );
std::cout << de2 << std::endl;
de2.SetByteValue( str2, (uint32_t)strlen(str2) );
ds.Replace( de2 );

const char str3[] = "GDCM was here 3!";
if( !ds.FindDataElement( pt3 ) ) return 1;
gdcmm::DataElement de3 = ds.GetDataElement( pt3 );
std::cout << de3 << std::endl;
de3.SetByteValue( str3, (uint32_t)strlen(str3) );
ds.Replace( de3 );

gdcmm::Writer writer;
writer.SetFile( file );
writer.SetFileName( outfilename );
if ( !writer.Write() )
{
    return 1;
}

return 0;
}

```

## 12.38 ChangeSequenceUltrasound.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmmReader.h"
#include "gdcmmWriter.h"
#include "gdcmmSmartPointer.h"
#include "gdcmmDataSetHelper.h"

/*
./ChangeSequenceUltrasound gdcmmData/D_CLUNIE_CT1_J2KI.dcm myoutput.dcm

This is the exact C++ translation of the original python example: ManipulateSequence.py
*/

int main(int argc, char* argv[] )
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];

```

```

const char *outfilename = argv[2];

gdcmm::Reader reader;
reader.SetFileName( filename );
if ( ! reader.Read() )
{
    return 1;
}

gdcmm::File &file = reader.GetFile();
gdcmm::DataSet &ds = file.GetDataSet();
gdcmm::Tag tsis(0x0008,0x2112); // SourceImageSequence
if ( ds.FindDataElement( tsis ) )
{
    const gdcmm::DataElement &sis = ds.GetDataElement( tsis );
    gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqsis = sis.GetValueAsSQ();
    if ( sqsis && sqsis->GetNumberOfItems() )
    {
        gdcmm::Item &item1 = sqsis->GetItem(1);
        gdcmm::DataSet &nestedds = item1.GetNestedDataSet();
        gdcmm::Tag tprcs(0x0040,0xa170); // PurposeOfReferenceCodeSequence
        if( nestedds.FindDataElement( tprcs ) )
        {
            const gdcmm::DataElement &prcs = nestedds.GetDataElement( tprcs );
            gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqprcs = prcs.GetValueAsSQ();
            if ( sqprcs && sqprcs->GetNumberOfItems() )
            {
                gdcmm::Item &item2 = sqprcs->GetItem(1);
                gdcmm::DataSet &nestedds2 = item2.GetNestedDataSet();
                // (0008,0104) LO [Uncompressed predecessor] # 24, 1 CodeMeaning
                gdcmm::Tag tcm(0x0008,0x0104);
                if( nestedds2.FindDataElement( tcm ) )
                {
                    gdcmm::DataElement cm = nestedds2.GetDataElement( tcm );
                    std::string mystr = "GDCM was here";
                    cm.SetByteValue( mystr.c_str(), (uint32_t)mystr.size() );
                    nestedds2.Replace( cm );
                }
            }
        }
    }
}

gdcmm::Writer writer;
writer.SetFile( file );
writer.SetFileName( outfile );
if ( ! writer.Write() )
{
    return 1;
}

return 0;
}

```

## 12.39 CheckBigEndianBug.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * WARNING: This is a dev tool, do not use !
 *
 * Usage: after a gdcmmconv, you would like to know if the conversion process is acceptable
 * sometime a vbindiff is acceptable, sometime it is not. In the case of the famous Philips
 * Little/Big Endian Explicit Transfer Syntax it is not easy to compare two files. However
 * this only impact byte ordering, thus we can compute byte-independant information to still
 * compare the files.
 */

```

```

*/

#include "gdcmImageReader.h"
#include "gdcmImage.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"
#include "gdcmSystem.h"

#include <iostream>
#include <fstream>

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input1.dcm input2.dcm" << std::endl;
        return 1;
    }
    const char *filename1 = argv[1];
    const char *filename2 = argv[2];

    gdcm::ImageReader reader1;
    reader1.SetFileName( filename1 );
    if( !reader1.Read() )
    {
        std::cerr << "Could not read: " << filename1 << std::endl;
        return 1;
    }

    gdcm::ImageReader reader2;
    reader2.SetFileName( filename2 );
    if( !reader2.Read() )
    {
        std::cerr << "Could not read: " << filename2 << std::endl;
        return 1;
    }

    // TODO: need a DataSet== operator implementation

    std::cout << "Both files can be read and looks like DICOM" << std::endl;

    size_t s1 = gdcm::System::FileSize(filename1);
    size_t s2 = gdcm::System::FileSize(filename2);

    if( s1 != s2 )
    {
        std::cout << "Size mismatch: " << s1 << " != " << s2 << std::endl;
        return 1;
    }
    else
    {
        std::cout << "Size match: " << s1 << " = " << s2 << std::endl;
    }

    std::ifstream is1( filename1, std::ios::binary );
    char *buffer1 = new char[s1];
    is1.read(buffer1, s1);

    std::ifstream is2( filename2, std::ios::binary );
    char *buffer2 = new char[s2];
    is2.read(buffer2, s2);

    assert( s1 == s2 );
    if( memcmp(buffer1, buffer2, s1 ) == 0 )
    {
        std::cout << "memcmp succeed ! File are bit identical" << std::endl;
    }
    else
    {
        std::cout << "memcmp failed!" << std::endl;
    }

    // Hum...memcmp failed, for big endian/ little endian inversion the histogram of bytes
    // should still be the same. So let's compute it
    // buffer2[0] = 1; // let's make the test fail
    std::multiset<char> set1( buffer1, buffer1 + s1 );
    std::multiset<char> set2( buffer2, buffer2 + s2 );

    if( set1 == set2 )
    {

```



```

        std::cout << "set1 == set2. Byte histogram seems valid" << std::endl;
    }
    else
    {
        std::cout << "set1 != set2" << std::endl;
    }
    delete[] buffer1;
    delete[] buffer2;

    return 0;
}

```

## 12.40 ClinicalTrialAnnotate.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
/*
 * Dummy implementation of C.7.1.3 Clinical Trial Subject Module
 *
 * Usage:
 * ClinicalTrialAnnotate gdcmData/012345.002.050.dcm out.dcm
 */

#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmAnonymizer.h"

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Could not read: " << filename << std::endl;
        return 1;
    }

    // The output of gdcm::Reader is a gdcm::File
    //gdcm::File &file = reader.GetFile();

    // the dataset is the the set of element we are interested in:
    //gdcm::DataSet &ds = file.GetDataSet();

    gdcm::Anonymizer ano;
    ano.SetFile( reader.GetFile() );
    ano.RemoveGroupLength();
    ano.RemovePrivateTags();

    // PS 3.3 - 2008
    // C.7.1.3 Clinical Trial Subject Module
    // <entry group="0012" element="0010" vr="LO" vm="1" name="Clinical Trial Sponsor Name"/>
    ano.Replace( gdcm::Tag(0x12,0x10), "BigCompany name" );
    // <entry group="0012" element="0020" vr="LO" vm="1" name="Clinical Trial Protocol ID"/>
    ano.Replace( gdcm::Tag(0x12,0x20), "My Clinical Trial Protocol ID" );
    // <entry group="0012" element="0021" vr="LO" vm="1" name="Clinical Trial Protocol Name"/>
    ano.Replace( gdcm::Tag(0x12,0x21), "My Clinical Trial Protocol Name" );
    // <entry group="0012" element="0030" vr="LO" vm="1" name="Clinical Trial Site ID"/>

```

```

ano.Replace( gdcm::Tag(0x12,0x30), "My Clinical Trial Site ID" );
// <entry group="0012" element="0031" vr="LO" vm="1" name="Clinical Trial Site Name"/>
ano.Replace( gdcm::Tag(0x12,0x31), "My Clinical Trial Site Name" );
// <entry group="0012" element="0040" vr="LO" vm="1" name="Clinical Trial Subject ID"/>
ano.Replace( gdcm::Tag(0x12,0x40), "My Clinical Trial Subject ID" );
// <entry group="0012" element="0042" vr="LO" vm="1" name="Clinical Trial Subject Reading ID"/>
ano.Replace( gdcm::Tag(0x12,0x42), "My Clinical Trial Subject Reading ID" );

gdcm::Writer writer;
writer.SetFile( reader.GetFile() );
writer.SetFileName( outfilename );
if( !writer.Write() )
{
    return 1;
}

return 0;
}

```

## 12.41 CompressImage.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 *
 */

#include "gdcmImageReader.h"
#include "gdcmImage.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"
#include "gdcmImageWriter.h"
#include "gdcmImageChangeTransferSyntax.h"

#include <iostream>
#include <fstream>

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    gdcm::ImageReader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Could not read: " << filename << std::endl;
        return 1;
    }

    // The output of gdcm::Reader is a gdcm::File
    //gdcm::File &file = reader.GetFile();

    // the dataset is the the set of element we are interested in:
    //gdcm::DataSet &ds = file.GetDataSet();

    gdcm::Image &image = reader.GetImage();
    // image.SetSpacing(0, 0.1);
    // image.SetSpacing(1, 0.2);

```

```

image.Print( std::cout );

gdcm::ImageChangeTransferSyntax change;
change.SetTransferSyntax( gdcm::TransferSyntax::JPEG2000Lossless );
change.SetTransferSyntax( gdcm::TransferSyntax::JPEGLosslessProcess14_1 );
//change.SetTransferSyntax( gdcm::TransferSyntax::JPEGBaselineProcess1 );
//change.SetTransferSyntax( image.GetTransferSyntax() );
change.SetInput( image );
bool b = change.Change();
if( !b )
{
    std::cerr << "Could not change the Transfer Syntax" << std::endl;
    return 1;
}

//std::ofstream out( outfilename, std::ios::binary );
//image.GetBuffer2(out);
//out.close();
gdcm::ImageWriter writer;
writer.SetImage( change.GetOutput() );
writer.SetFile( reader.GetFile() );
writer.SetFileName( outfilename );
if( !writer.Write() )
{
    return 1;
}

return 0;
}

```

## 12.42 ConvertToQImage.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * This example shows how to setup the pipeline from a gdcm::ImageReader into a
 * Qt QImage data structure.
 * It only handles 2D image.
 *
 * Ref:
 * http://doc.trolltech.com/4.5/qimage.html
 *
 * Usage:
 * ConvertToQImage gdcmData/012345.002.050.dcm output.png
 *
 * Thanks:
 * Sylvain ADAM (sylvain51 hotmail com) for contributing this example
 */
#include "gdcmImageReader.h"
#include <QImage>
#include <QImageWriter>

bool ConvertToFormat_RGB888(gdcm::Image const & gimage, char *buffer, QImage* &imageQt)
{
    const unsigned int* dimension = gimage.GetDimensions();

    unsigned int dimX = dimension[0];
    unsigned int dimY = dimension[1];

    gimage.GetBuffer(buffer);

    // Let's start with the easy case:
    if( gimage.GetPhotometricInterpretation() == gdcm::PhotometricInterpretation::RGB )
    {
        if( gimage.GetPixelFormat() != gdcm::PixelFormat::UINT8 )

```

```

    {
        return false;
    }
    unsigned char *ubuffer = (unsigned char*)buffer;
    // QImage::Format_RGB888 13 The image is stored using a 24-bit RGB format (8-8-8).
    QImageQt = new QImage((unsigned char *)ubuffer, dimX, dimY, 3*dimX, QImage::Format_RGB888);
}
else if( gimage.GetPhotometricInterpretation() == gdcm::PhotometricInterpretation::MONOCHROME2 )
{
    if( gimage.GetPixelFormat() == gdcm::PixelFormat::UINT8 )
    {
        // We need to copy each individual 8bits into R / G and B:
        unsigned char *ubuffer = new unsigned char[dimX*dimY*3];
        unsigned char *pubuffer = ubuffer;
        for(unsigned int i = 0; i < dimX*dimY; i++)
        {
            *pubuffer++ = *buffer;
            *pubuffer++ = *buffer;
            *pubuffer++ = *buffer++;
        }

        QImageQt = new QImage(ubuffer, dimX, dimY, QImage::Format_RGB888);
    }
    else if( gimage.GetPixelFormat() == gdcm::PixelFormat::INT16 )
    {
        // We need to copy each individual 16bits into R / G and B (truncate value)
        short *buffer16 = (short*)buffer;
        unsigned char *ubuffer = new unsigned char[dimX*dimY*3];
        unsigned char *pubuffer = ubuffer;
        for(unsigned int i = 0; i < dimX*dimY; i++)
        {
            // Scalar Range of gdcmData/012345.002.050.dcm is [0,192], we could simply do:
            // *pubuffer++ = *buffer16;
            // *pubuffer++ = *buffer16;
            // *pubuffer++ = *buffer16;
            // instead do it right:
            *pubuffer++ = (unsigned char)std::min(255, (32768 + *buffer16) / 255);
            *pubuffer++ = (unsigned char)std::min(255, (32768 + *buffer16) / 255);
            *pubuffer++ = (unsigned char)std::min(255, (32768 + *buffer16) / 255);
            buffer16++;
        }

        QImageQt = new QImage(ubuffer, dimX, dimY, QImage::Format_RGB888);
    }
    else
    {
        std::cerr << "Pixel Format is: " << gimage.GetPixelFormat() << std::endl;
        return false;
    }
}
else
{
    std::cerr << "Unhandled PhotometricInterpretation: " << gimage.GetPhotometricInterpretation() << std::endl;
    return false;
}

return true;
}

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    gdcm::ImageReader ir;
    ir.SetFileName( filename );
    if(!ir.Read())
    {
        //Read failed
        return 1;
    }

    std::cout<<"Getting image from ImageReader..."<<std::endl;

    const gdcm::Image &gimage = ir.GetImage();
    std::vector<char> vbuffer;
    vbuffer.resize( gimage.GetBufferLength() );

```

```

char *buffer = &vbuffer[0];

QImage *imageQt = NULL;
if( !ConvertToFormat_RGB888( gimage, buffer, imageQt ) )
{
    return 1;
}

QImageWriter writer;
writer.setFormat("png");
writer.setFileName( outfilename );
if( !writer.write( *imageQt ) )
{
    return 1;
}

return 0;
}

```

## 12.43 CreateARGBImage.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/

/*
 * http://www.w3.org/Graphics/PNG/inline-alpha.html
 * alphatest.png: PNG image data, 380 x 287, 8-bit/color RGBA, non-interlaced
 *
 * $ convert alphatest.png alphatest.rgba
 */

#include "gdcmImageReader.h"
#include "gdcmSequenceOfFragments.h"
#include "gdcmSystem.h"
#include "gdcmImageWriter.h"

#include <iostream>
#include <fstream>

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.rgb output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    size_t len = gdcm::System::FileSize(filename);
    std::ifstream is(filename, std::ios::binary);

    char * buf = new char[len];
    is.read(buf, len);

    gdcm::ImageWriter writer;
    gdcm::Image &image = writer.GetImage();
    image.SetNumberOfDimensions( 2 );
    unsigned int dims[3] = {};
    dims[0] = 380;
    dims[1] = 287;
    image.SetDimensions( dims );
    gdcm::PixelFormat pf = gdcm::PixelFormat::UINT8;
    pf.SetSamplesPerPixel( 4 );
    image.SetPixelFormat( pf );
    gdcm::PhotometricInterpretation pi = gdcm::PhotometricInterpretation::ARGB;
    image.SetPhotometricInterpretation( pi );

```

```

image.SetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );

gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
pixeldata.SetByteValue( buf, (uint32_t)len );
image.SetDataElement( pixeldata );

writer.SetFileName( outfilename );
if( !writer.Write() )
{
    return 1;
}
delete[] buf;

return 0;
}

```

## 12.44 CreateCMYKImage.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/

/*
 * http://www.w3.org/Graphics/PNG/inline-alpha.html
 * alphatest.png: PNG image data, 380 x 287, 8-bit/color RGBA, non-interlaced
 *
 * $ convert alphatest.png alphatest.cmyk
 */

#include "gdcmImageReader.h"
#include "gdcmSequenceOfFragments.h"
#include "gdcmSystem.h"
#include "gdcmImageWriter.h"

#include <iostream>
#include <fstream>

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.cmyk output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    size_t len = gdcm::System::FileSize(filename);
    std::ifstream is(filename, std::ios::binary);

    char * buf = new char[len];
    is.read(buf, len);

    gdcm::ImageWriter writer;
    gdcm::Image &image = writer.GetImage();
    image.SetNumberOfDimensions( 2 );
    unsigned int dims[3] = {};
    dims[0] = 380;
    dims[1] = 287;
    image.SetDimensions( dims );
    gdcm::PixelFormat pf = gdcm::PixelFormat::UINT8;
    pf.SetSamplesPerPixel( 4 );
    image.SetPixelFormat( pf );
    gdcm::PhotometricInterpretation pi = gdcm::PhotometricInterpretation::CMYK;
    image.SetPhotometricInterpretation( pi );
    image.SetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );

    gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );

```

```

    pixeldata.SetByteValue( buf, (uint32_t)len );
    image.SetDataElement( pixeldata );

    writer.SetFileName( outfilename );
    if( !writer.Write() )
    {
        return 1;
    }
    delete[] buf;

    return 0;
}

```

## 12.45 CreateJPIPDataSet.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
/*
 * This example was created during the GSOC 2011 project for
 * JPIP
 */
#include "gdcmAnonymizer.h"
#include "gdcmWriter.h"
#include "gdcmUIDGenerator.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmSystem.h"
#include "gdcmAttribute.h"

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " output.dcm" << std::endl;
        return 1;
    }
    const char *outfilename = argv[1];

    gdcm::Writer w;
    gdcm::File &file = w.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();
    //w.SetCheckFileMetaInformation( true );
    w.SetFileName( outfilename );

    file.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::JPIPReferenced );

    gdcm::Anonymizer anon;
    anon.SetFile( file );

    gdcm::MediaStorage ms = gdcm::MediaStorage::SecondaryCaptureImageStorage;

    gdcm::UIDGenerator gen;
    anon.Replace( gdcm::Tag(0x0008,0x16), ms.GetString() );
    std::cout << ms.GetString() << std::endl;
    anon.Replace( gdcm::Tag(0x0008,0x18), gen.Generate() );
    //
    anon.Replace( gdcm::Tag(0x0010,0x10), "JPIP^EXAMPLE" );
    anon.Replace( gdcm::Tag(0x0010,0x20), "012345" );
    anon.Empty( gdcm::Tag(0x0010,0x30) );
    anon.Empty( gdcm::Tag(0x0010,0x40) );
    anon.Empty( gdcm::Tag(0x0008,0x20) );
    anon.Empty( gdcm::Tag(0x0008,0x30) );
    anon.Empty( gdcm::Tag(0x0008,0x90) );
    anon.Empty( gdcm::Tag(0x0020,0x10) );
    anon.Empty( gdcm::Tag(0x0020,0x11) );
    anon.Empty( gdcm::Tag(0x0008,0x50) );

```

```

anon.Empty( gdcmm::Tag(0x0020,0x0013) );
anon.Replace( gdcmm::Tag(0x0020,0xd), gen.Generate() );
anon.Replace( gdcmm::Tag(0x0020,0xe), gen.Generate() );
anon.Replace( gdcmm::Tag(0x0008,0x64), "WSD " );
anon.Replace( gdcmm::Tag(0x0008,0x60), "OT" );

gdcmm::Attribute<0x0028,0x7FE0> at;
at.SetValue( "http://dicom.example.com/jpipserver.cgi?target=img.jp2" );
ds.Insert( at.GetAsDataElement() );

// Need to retrieve the PixelFormat information from the given file

if (!w.Write() )
{
    std::cerr << "Could not write: " << outfilename << std::endl;
    return 1;
}

return 0;
}

```

## 12.46 DeriveSeries.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmmReader.h"
#include "gdcmmWriter.h"
#include "gdcmmAttribute.h"
#include "gdcmmFileDerivation.h"
#include "gdcmmUIDGenerator.h"

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        return 1;
    }
    const char * ref = argv[1];
    const char * in = argv[2];

    gdcmm::Reader r1;
    r1.SetFileName( ref );
    if( !r1.Read() ) return 1;

    gdcmm::Reader r2;
    r2.SetFileName( in );
    if( !r2.Read() ) return 1;

    // Fix Spatial info:
    gdcmm::DataSet & ds1 = r1.GetFile().GetDataSet();
    gdcmm::File & file2 = r2.GetFile();
    gdcmm::DataSet & ds2 = file2.GetDataSet();
    //gdcmm::Attribute<0x8,0x8> img_type = { "ORIGINAL", "PRIMARY" };
    ds2.Replace( ds1.GetDataElement( gdcmm::Tag(0x0008,0x0008) ) );
    ds2.Replace( ds1.GetDataElement( gdcmm::Tag(0x0020,0x0032) ) );
    ds2.Replace( ds1.GetDataElement( gdcmm::Tag(0x0020,0x0037) ) );
    ds2.Replace( ds1.GetDataElement( gdcmm::Tag(0x0018,0x0088) ) ); // Spacing between slices
    ds2.Replace( ds1.GetDataElement( gdcmm::Tag(0x0020,0x0013) ) ); // Instance Number
    ds2.Replace( ds1.GetDataElement( gdcmm::Tag(0x0018,0x5100) ) ); // Patient Position
    ds2.Replace( ds1.GetDataElement( gdcmm::Tag(0x0018,0x0050) ) ); // Slice Thickness
    ds2.Replace( ds1.GetDataElement( gdcmm::Tag(0x0008,0x0070) ) ); // Manufacturer
    ds2.Replace( ds1.GetDataElement( gdcmm::Tag(0x0018,0x0081) ) ); // Echo Time
    ds2.Replace( ds1.GetDataElement( gdcmm::Tag(0x0020,0x1041) ) ); // Slice Location

    gdcmm::Attribute<0x8,0x16> sopclassuid;

```



```

sopclassuid.SetFromDataSet( ds1 );
gdcm::Attribute<0x8,0x18> sopinstanceuid;
sopinstanceuid.SetFromDataSet( ds1 );

// Step 2: DERIVED object
gdcm::FileDerivation fd;
fd.AddReference( sopclassuid.GetValue(), sopinstanceuid.GetValue() );

// http://dicom.nema.org/MEDICAL/dicom/current/output/chtml/part16/chapter_D.html#DCM_121321
// CID 7202 "Source Image Purposes of Reference"
// DCM 121321 "Mask image for image processing operation"
fd.SetPurposeOfReferenceCodeSequenceCodeValue( 121321 );
// CID 7203 "Image Derivation"
// DCM 113047 "Pixel by pixel mask"
fd.SetDerivationCodeSequenceCodeValue( 113047 );
fd.SetFile( file2 );
// If all Code Value are ok the filter will execute properly
if( !fd.Derive() )
{
    std::cerr << "Sorry could not derive using input info" << std::endl;
    return 1;
}

gdcm::Writer w;
w.SetFile( r2.GetFile() );
w.SetFileName( "derived.dcm" );
if( !w.Write() )
{
    return 1;
}

return 0;
}

```

## 12.47 DiffFile.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input1.dcm input2.dcm" << std::endl;
        return 1;
    }
    const char *filename1 = argv[1];
    const char *filename2 = argv[2];

    gdcm::Reader reader1;
    reader1.SetFileName( filename1 );
    if( !reader1.Read() )
    {
        return 1;
    }

    gdcm::Reader reader2;
    reader2.SetFileName( filename2 );
    if( !reader2.Read() )
    {
        return 1;
    }

    const gdcm::File &file1 = reader1.GetFile();

```

```

const gdcm::File &file2 = reader2.GetFile();

const gdcm::DataSet &ds1 = file1.GetDataSet();
const gdcm::DataSet &ds2 = file2.GetDataSet();

gdcm::DataSet::ConstIterator it1 = ds1.Begin();
gdcm::DataSet::ConstIterator it2 = ds2.Begin();

const gdcm::DataElement &de1 = *it1;
const gdcm::DataElement &de2 = *it2;
if( de1 == de2 )
{
}
while( it1 != ds1.End() && it2 != ds2.End() && *it1 == *it2 )
{
  ++it1;
  ++it2;
}

if( it1 != ds1.End() || it2 != ds2.End() )
{
  std::cerr << "Problem with:" << std::endl;
  if( it1 != ds1.End() )
  {
    std::cerr << "ds1: " << *it1 << std::endl;
  }
  if( it2 != ds2.End() )
  {
    std::cerr << "ds2: " << *it2 << std::endl;
  }
  return 1;
}

return 0;
}

```

## 12.48 DiscriminateVolume.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
#include "gdcmScanner.h"
#include "gdcmTesting.h"
#include "gdcmIPPSorter.h"
#include "gdcmDirectionCosines.h"
#include <cmath>

/*
 * The following example is a basic sorted which should work in generic cases.
 * It sort files based on:
 * Study Instance UID
 *   Series Instance UID
 *     Frame of Reference UID
 *       Image Orientation (Patient)
 *         Image Position (Patient) (Sorting based on IPP + IOP)
 */

namespace gdcm {
  const Tag t1(0x0020,0x000d); // Study Instance UID
  const Tag t2(0x0020,0x000e); // Series Instance UID
  const Tag t3(0x0020,0x0052); // Frame of Reference UID
  const Tag t4(0x0020,0x0037); // Image Orientation (Patient)

  class DiscriminateVolume
  {
  private:
    std::vector< Directory::FilenameType > SortedFiles;

```

```

std::vector< Directory::FileNamesType > UnsortedFiles;

Directory::FileNamesType GetAllFileNamesFromTagToValue(
    Scanner const & s, Directory::FileNamesType const & filesubset, Tag const & t, const char *valueref)
{
    Directory::FileNamesType theReturn;
    if( valueref )
    {
        size_t len = strlen( valueref );
        Directory::FileNamesType::const_iterator file = filesubset.begin();
        for(; file != filesubset.end(); ++file)
        {
            const char *filename = file->c_str();
            const char * value = s.GetValue(filename, t);
            if( value && strncmp(value, valueref, len ) == 0 )
            {
                theReturn.push_back( filename );
            }
        }
    }
    return theReturn;
}

void ProcessAIOP(Scanner const & , Directory::FileNamesType const & subset, const char *iopval)
{
    std::cout << "IOP: " << iopval << std::endl;
    IPPSorter ipp;
    ipp.SetComputeZSpacing( true );
    ipp.SetZSpacingTolerance( 1e-3 ); // ??
    bool b = ipp.Sort( subset );
    if( !b )
    {
        // If you reach here this means you need one more parameter to discriminat this
        // series. Eg. T1 / T2 intertwined. Multiple Echo (0018,0081)
        std::cerr << "Failed to sort: " << subset.begin()->c_str() << std::endl;
        for(
            Directory::FileNamesType::const_iterator file = subset.begin();
            file != subset.end(); ++file)
        {
            std::cerr << *file << std::endl;
        }
        UnsortedFiles.push_back( subset );
        return ;
    }
    ipp.Print( std::cout );
    SortedFiles.push_back( ipp.GetFileNames() );
}

void ProcessAFrameOfRef(Scanner const & s, Directory::FileNamesType const & subset, const char * frameuid)
{
    // In this subset of files (belonging to same series), let's find those
    // belonging to the same Frame ref UID:
    Directory::FileNamesType files = GetAllFileNamesFromTagToValue(
        s, subset, t3, frameuid);

    std::set< std::string > iopset;

    for(
        Directory::FileNamesType::const_iterator file = files.begin();
        file != files.end(); ++file)
    {
        //std::cout << *file << std::endl;
        const char * value = s.GetValue(file->c_str(), gdcm::t4 );
        assert( value );
        iopset.insert( value );
    }
    size_t n = iopset.size();
    if ( n == 0 )
    {
        assert( files.empty() );
        return;
    }

    std::cout << "Frame of Ref: " << frameuid << std::endl;
    if ( n == 1 )
    {
        ProcessAIOP(s, files, iopset.begin()->c_str() );
    }
    else
    {
        const char *f = files.begin()->c_str();

```

```

std::cerr << "More than one IOP: " << f << std::endl;
// Make sure that there is actually 'n' different IOP
gdcmm::DirectionCosines ref;
gdcmm::DirectionCosines dc;
for(
    std::set< std::string >::const_iterator it = iopset.begin();
    it != iopset.end(); ++it )
{
    ref.SetFromString( it->c_str() );
    for(
        Directory::FilenameType::const_iterator file = files.begin();
        file != files.end(); ++file )
        {
            std::string value = s.GetValue(file->c_str(), gdcmm::t4 );
            if( value != it->c_str() )
                {
                    dc.SetFromString( value.c_str() );
                    const double crossdot = ref.CrossDot(dc);
                    const double eps = std::fabs( 1. - crossdot );
                    if( eps < 1e-6 )
                        {
                            std::cerr << "Problem with IOP discrimination: " << file->c_str()
                                << " " << it->c_str() << std::endl;
                            return;
                        }
                }
        }
    }
}
// If we reach here this means there is actually 'n' different IOP
for(
    std::set< std::string >::const_iterator it = iopset.begin();
    it != iopset.end(); ++it )
{
    const char *iopvalue = it->c_str();
    Directory::FilenameType iopfiles = GetAllFileNamesFromTagToValue(
        s, files, t4, iopvalue );
    ProcessAIOP(s, iopfiles, iopvalue );
}
}

void ProcessASeries(Scanner const & s, const char * seriesuid)
{
    std::cout << "Series: " << seriesuid << std::endl;
    // let's find all files belonging to this series:
    Directory::FilenameType seriesfiles = GetAllFileNamesFromTagToValue(
        s, s.GetFileNames(), t2, seriesuid);

    gdcmm::Scanner::ValuesType vt3 = s.GetValues(t3);
    for(
        gdcmm::Scanner::ValuesType::const_iterator it = vt3.begin()
        ; it != vt3.end(); ++it )
        {
            ProcessAFrameOfRef(s, seriesfiles, it->c_str());
        }
}

void ProcessAStudy(Scanner const & s, const char * studyuid)
{
    std::cout << "Study: " << studyuid << std::endl;
    gdcmm::Scanner::ValuesType vt2 = s.GetValues(t2);
    for(
        gdcmm::Scanner::ValuesType::const_iterator it = vt2.begin()
        ; it != vt2.end(); ++it )
        {
            ProcessASeries(s, it->c_str());
        }
}

public:

void Print( std::ostream & os )
{
    os << "Sorted Files: " << std::endl;
    for(
        std::vector< Directory::FilenameType >::const_iterator it = SortedFiles.begin();
        it != SortedFiles.end(); ++it )
        {
            os << "Group: " << std::endl;
            for(
                Directory::FilenameType::const_iterator file = it->begin();
                file != it->end(); ++file )

```

```

        {
            os << *file << std::endl;
        }
    }
    os << "Unsorted Files: " << std::endl;
    for(
        std::vector< Directory::FileNamesType >::const_iterator it = UnsortedFiles.begin();
        it != UnsortedFiles.end(); ++it )
    {
        os << "Group: " << std::endl;
        for(
            Directory::FileNamesType::const_iterator file = it->begin();
            file != it->end(); ++file)
        {
            os << *file << std::endl;
        }
    }
}

std::vector< Directory::FileNamesType > const & GetSortedFiles() const { return SortedFiles; }
std::vector< Directory::FileNamesType > const & GetUnsortedFiles() const { return UnsortedFiles; }

void ProcessIntoVolume( Scanner const & s )
{
    gdcm::Scanner::ValuesType vt1 = s.GetValues( gdcm::t1 );
    for(
        gdcm::Scanner::ValuesType::const_iterator it = vt1.begin();
        ; it != vt1.end(); ++it )
    {
        ProcessAStudy( s, it->c_str() );
    }
}

};

} // namespace gdcm

int main(int argc, char *argv[])
{
    std::string dirl;
    if( argc < 2 )
    {
        const char *extradataroot = nullptr;
#ifdef GDCM_BUILD_TESTING
        extradataroot = gdcm::Testing::GetDataExtraRoot();
#endif
        if( !extradataroot )
        {
            return 1;
        }
        dirl = extradataroot;
        dirl += "/gdcmSampleData/ForSeriesTesting/VariousIncidences/ST1";
    }
    else
    {
        dirl = argv[1];
    }

    gdcm::Directory d;
    d.Load( dirl, true ); // recursive !

    gdcm::Scanner s;
    s.AddTag( gdcm::t1 );
    s.AddTag( gdcm::t2 );
    s.AddTag( gdcm::t3 );
    s.AddTag( gdcm::t4 );
    bool b = s.Scan( d.GetFilesNames() );
    if( !b )
    {
        std::cerr << "Scanner failed" << std::endl;
        return 1;
    }

    gdcm::DiscriminateVolume dv;
    dv.ProcessIntoVolume( s );
    dv.Print( std::cout );

    return 0;
}

```

## 12.49 DumpADAC.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * the goal of this example is to mimic the behavior of disp_img_header
 * see http://www.gmccorp-usa.com/IM/NM/GC/ADAC/SV/adactechtips/Released_01Q3.pdf
 */
#include "gdcmlReader.h"
#include "gdcmlPrivateTag.h"
#include "gdcmlAttribute.h"
#include "gdcmlImageWriter.h"

#include <iostream>
#include <fstream>
#include <vector>

#include <string.h>
#include <assert.h>
#include <stdint.h>

struct dict
{
    uint16_t key;
    const char *name;
};

dict Array[] = {
    { 0x01, "Patient name" },
    { 0x02, "Patient ID" },
    { 0x03, "Patient sex" },
    { 0x04, "Patient age" },
    { 0x05, "Patient height" },
    { 0x06, "Patient weight" },
    { 0x07, "Exam date" },
    { 0x08, "Dose admin. time" },
    { 0x09, "Unique exam key" },
    { 0x0a, "Exam procedure" },
    { 0x0b, "Referring physician" },
    { 0x0c, "Attending physician" },
    { 0x0d, "Imaging modality" },
    { 0x0e, "Hospital ID" },
    { 0x0f, "Histogram crv file" },
    { 0x10, "Acq. start time" },
    { 0x11, "Object data type" },
    { 0x12, "Image viewid" },
    { 0x13, "Imaging device name" },
    { 0x14, "Device serial number" },
    { 0x15, "Collimator" },
    { 0x16, "Software version" },
    { 0x17, "Radiopharmaceutical #1" },
    { 0x18, "Energy window #1 center" },
    { 0x19, "Radiopharmaceutical #2" },
    { 0x1a, "Energy window #1 width" },
    { 0x1b, "Isotope imaging mode" },
    { 0x1c, "Energy window #2 center" },
    { 0x1d, "Energy window #2 width" },
    { 0x1e, "Energy window #3 center" },
    { 0x1f, "Energy window #3 width" },
    { 0x20, "Energy window #4 center" },
    { 0x21, "Energy window #4 width" },
    { 0x22, "??Energy window #5 center" },
    { 0x23, "??Energy window #5 width" },
    { 0x24, "Patient orientation" },
    { 0x25, "Spatial resolution" },
    { 0x26, "Slice thickness" },
    { 0x27, "Image X dimension" },
    { 0x28, "Image Y dimension" },
    { 0x29, "Image Z dimension" },
};

```

```

{ 0x2a, "Image pixel width" },
{ 0x2b, "Uniformity corr. file" },
{ 0x2c, "Acquisition zoom factor" },
{ 0x2d, "Total counts in set" },
{ 0x2e, "Time / frame" },
{ 0x2f, "Total acq. time" },
{ 0x30, "Maximum pixel value" },
{ 0x31, "Minimum pixel value" },
{ 0x32, "R-R interval time" },
{ 0x33, "Percent of cycle imaged" },
{ 0x34, "# of cycles accepted" },
{ 0x35, "# of cycles rejected" },
{ 0x36, "Approximate ED frame" },
{ 0x37, "Approximate ES frame" },
{ 0x38, "Approximate EF" },
{ 0x39, "Starting angle" },
{ 0x3a, "Degrees of rotation" },
{ 0x3b, "Direction of rotation" },
{ 0x3c, "Cont. or step/shoot" },
{ 0x3d, "Lim recon start frame" },
{ 0x3e, "Upper window grey shade" },
{ 0x3f, "Lower lvl grey shade" },
{ 0x40, "Associated color map" },
{ 0x41, "Custom color map file" },
{ 0x42, "Manipulated image" },
{ 0x43, "Axis of rotation corr." },
{ 0x44, "Reorientation azimuth" },
{ 0x45, "Reorientation elevation" },
{ 0x46, "Filter type" },
{ 0x47, "Filter order" },
{ 0x48, "Filter cutoff frequency" },
{ 0x49, "Reconstruction type" },
{ 0x4a, "Attenuation coefficient" },
{ 0x4b, "Associated parent file" },
{ 0x4c, "Unique patient key" },
{ 0x52, "Normalization crv file" },
{ 0x53, "Unique object key" },
{ 0x54, "This phase of VFR is" },
{ 0x55, "True color value" },
{ 0x56, "# of sets of x,y,z grps" },
{ 0x57, "Scale factor of set" },
{ 0x6d, "Date of birth" },
{ 0x6e, "Directional orientation" },
{ 0x6f, "Number of VFR studies" },
{ 0x70, "R-R low tolerance" },
{ 0x71, "R-R high tolerance" },
{ 0x72, "Prog specific results:" },

{ 0x99, nullptr }
};

void printname( int , int , uint16_t v )
{
    if( v == 0x1 )
    {
        std::cout << "DATABASE PARAMETERS" << std::endl;
        std::cout << "_____" << std::endl;
    }
    else if( v == 0x27 )
    {
        std::cout << "IMAGE PARAMETERS" << std::endl;
        std::cout << "_____" << std::endl;
    }
    else if( v == 0x13 )
    {
        std::cout << "EXTRA PARAMETERS" << std::endl;
        std::cout << "_____" << std::endl;
    }
    else if( v == 0x2e )
    {
        std::cout << "*** NOT CURRENTLY USED :" << std::endl;
    }
    static const unsigned int n = sizeof( Array ) / sizeof( *Array ) - 1;
    for( unsigned int i = 0; i < n; ++i )
    {
        if( v == Array[i].key )
        {
            std::cout << /*" << std::dec << len << "," << mult << " " << */ Array[i].name;
            std::cout << " : ";
            return;
        }
    }
}

```

```

    }
    std::cout << /*"\t# " << std::dec << len << ", " << mult << */ std::hex << v << "\t: ";
}

uint16_t readint16(std::istream &is )
{
    uint16_t val;
    is.read( (char*)&val, sizeof( val ));
    return (uint16_t)((val>>8) | (val<<8));
}

uint32_t readint32(std::istream &is )
{
    uint32_t val;
    is.read( (char*)&val, sizeof( val ));
    val= ((val<<8)&0xFF00FF00) | ((val>>8)&0x00FF00FF);
    return (val>>16) | (val<<16);
}

float readfloat32(std::istream &is )
{
    union { uint32_t val; float f;} dual;
    dual.val = readint32(is);
    return dual.f;
}

struct el
{
    uint16_t v1;
    uint16_t v2;
    uint16_t v3;
    void read( std::istream & is )
    {
        v1 = readint16(is);
        v2 = readint16(is);
        v3 = readint16(is);
    }
    void print( std::ostream & os )
    {
        os << std::hex << v1 << "\t" << v2 << "\t" << v3 << std::endl;
    }
};

std::vector<el> Vel;

void readelement( std::istream & is )
{
    el e;
    e.read( is );
    Vel.push_back( e );
}

void printascii( uint16_t tag, const char *buffer, size_t len )
{
    std::ostream & os = std::cout;
    if( tag == 0x72 )
    {
        os << "\n ";
        for(size_t i = 0; i < len; ++i)
        {
            const char &c = buffer[i];
            if( c == 0x0 ) os << "!";
            else if( c == 0x0f ) os << " ";
            else if( c == 0x17 ) os << ":";
            else if( c == 0x14 ) os << ":";
            else if( c == 0x10 ) os << ":";
            else if( c == 0x16 ) os << ":";
            else if( c == 0x08 ) os << ":";
            else if( c == 0x0b ) os << ":";
            else if( c == 0x0e ) os << ":";
            else if( c == 0x07 ) os << ":";
            else os << c;
        }
        os << " ";
    }
    else
    {
        (void)len;
        os << " " << buffer << " ";
    }
}

```



```

}

bool DumpADAC( std::istream & is )
{
    std::ostream &os = std::cout;

    char magic[6 + 1];
    magic[6] = 0;
    is.read( magic, 6);
    // std::cout << magic << " ";
    assert( strcmp( magic, "adac01" ) == 0 );
    int c = is.get();
    assert( c == 0 ); (void)c;
    c = is.get();
    assert( c == 'X' );

    uint16_t v;
    v = readint16(is);
    // std::cout << v << std::endl;
    assert( v == 512 ); (void)v; // ??

    int nel = 87;
    for( int i = 0; i <= nel; ++i )
    {
        readelement( is );
    }

    char buffer[512];
    for( int i = 0; i <= nel; ++i )
    {
        const el &e = Vel[i];
        int diff;
        if( i == nel )
        {
            diff = 2048 - e.v3;
            if( diff > 512 ) diff = 512;
        }
        else
        {
            const el &enext = Vel[i+1];
            diff = enext.v3 - e.v3;
        }
        is.seekg( e.v3, std::ios::beg );
        //std::cout << "(" << std::hex << std::setw( 2 ) << std::setfill( '0' ) << e.v1 << ")" " << std::hex << std::setw( 3 )
        // << std::setfill( '0' ) << e.v2 << " ";
        printname( diff, 0, e.v1 );
        int mult = 1;
        if( e.v2 == 0 )
        {
            is.read( buffer, diff);
            buffer[ diff ] = 0;
            printascii( e.v1, buffer, diff);
        }
        else if( e.v2 == 0x100 )
        {
            mult = diff / 2;
            assert( diff == 2 * mult );
            for( int ii = 0; ii < mult; ++ii )
            {
                if( ii ) os << "\\ ";
                uint16_t val = readint16(is);
                os << " " << std::dec << val << " ";
            }
        }
        else if( e.v2 == 0x200 )
        {
            assert( diff == 4 );
            uint32_t val = readint32(is);
            os << " " << std::dec << val << " ";
        }
        else if( e.v2 == 0x300 )
        {
            assert( diff == 4 );
            float val = readfloat32(is);
            os << " " << std::dec << val << " ";
        }
        else
        {
            assert( 0 );
        }
        os << std::endl;
    }
}

```

```

    }
    return true;
}

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }
    const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

    // (0019,1061) UN (OB) 61\64\61\63\30          # 2048,1 Ver200 ADAC Pegasys Headers
    const gdcm::PrivateTag tver200adacpegasysheaders(0x0019,0x61,"ADAC_IMG");
    if( !ds.FindDataElement( tver200adacpegasysheaders ) ) return 1;
    const gdcm::DataElement& ver200adacpegasysheaders = ds.GetDataElement( tver200adacpegasysheaders );
    if ( ver200adacpegasysheaders.IsEmpty() ) return 1;
    const gdcm::ByteValue * bv = ver200adacpegasysheaders.GetByteValue();

    // (0019,1021) US 1          # 2,1 Ver200 Number of ADAC Headers
    // TODO

    // (0019,1041) IS [2048\221184 ] # 12,1-n Ver200 ADAC Header/Image Size
    if( bv->GetLength() != 2048 ) return 1;

    gdcm::Element<gdcm::VR::IS,gdcm::VM::VM2> el;
    const gdcm::PrivateTag tver200adacheaderimagesize(0x0019,0x41,"ADAC_IMG");
    if( !ds.FindDataElement( tver200adacheaderimagesize ) ) return 1;
    const gdcm::DataElement& ver200adacheaderimagesize = ds.GetDataElement( tver200adacheaderimagesize );
    el.SetFromDataElement( ver200adacheaderimagesize );
    if( el.GetValue(0) != 2048 ) return 1;

    std::istringstream is;
    std::string dup( bv->GetPointer(), bv->GetLength() );
    is.str( dup );
    bool b = DumpADAC( is );
    if( !b ) return 1;

    return 0;
}

```

## 12.50 DumpExamCard.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*

Try to extract contents of Philips RAW storage class:

(0002,0002) UI [1.2.840.10008.5.1.4.1.1.66]          # 26,1 Media Storage SOP Class UID
(0002,0003) UI [1.3.46.670589.11.17240.5.23.4.1.3012.2010032409482568018] # 56,1 Media Storage SOP
Instance UID
(0002,0010) UI [1.2.840.10008.1.2.1]                # 20,1 Transfer Syntax UID
(0002,0012) UI [1.3.46.670589.11.0.0.51.4.4.1]       # 30,1 Implementation Class UID
(0002,0013) SH [MR DICOM 4.1]                       # 12,1 Implementation Version Name

* Everything done in this code is for the sole purpose of writing interoperable
* software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
* If you believe anything in this code violates any law or any of your rights,

```

```

* please contact us (gdcm-developers@lists.sourceforge.net) so that we can
* find a solution.
*
* Everything you do with this code is at your own risk, since decompression
* algorithm was not written from specification documents.
*
* Special thanks to:
* Triplett, William T for bringing to your attention on this ExamCard stuff
*/
#include "gdcmReader.h"
#include "gdcmDataSet.h"
#include "gdcmPrivateTag.h"
#include "gdcmBase64.h"

#include <iomanip>

static bool compfn(const char *s1, const char *s2)
{
    return strcmp(s1,s2) < 0 ? true : false;
}

static const char *PDFStrings[] = { // Keep me ordered please
    "COILSTATE", // series of string ?
    "HARDWARE_CONFIG", // series of number ?
    "PDF_CONTROL_GEN_PARS",
    "PDF_CONTROL_PREP_PARS",
    "PDF_CONTROL_RECON_PARS",
    "PDF_CONTROL_SCAN_PARS",
    "PDF_EXAM_PARS",
    "PDF_HARDWARE_PARS",
    "PDF_PREP_PARS",
    "PDF_PRESCAN_COIL_PARS",
    "PDF_SPT_PARS",
};

static bool isvalidpdfstring( const char *pdfstring )
{
    assert( pdfstring );
    static const size_t n = sizeof( PDFStrings ) / sizeof( *PDFStrings );
    static const char **begin = PDFStrings;
    static const char **end = begin + n;
    return std::binary_search(begin, end, pdfstring, compfn);
}

typedef enum
{
    param_float = 0,
    param_integer = 1, // 1 << 0
    param_string = 2, // 1 << 1
    param_3, // ??
    param_enum = 4 // 1 << 2
} param_type;

static const char *gettypenamefromtype( int i)
{
    const char *ret = nullptr;
    param_type e = (param_type)i;
    switch( e )
    {
        case param_float:
            ret = "float";
            break;
        case param_integer:
            ret = "int";
            break;
        case param_string:
            ret = "string";
            break;
        case param_3:
            ret = "??";
            break;
        case param_enum:
            ret = "enum";
            break;
    }
    assert( ret );
    return ret;
}

struct header
{

```

```

/*
 * TODO:
 * Looks as if we could read all int*, float* and string* at once...
 */
int32_t v1; // offset to int pointer array ?
uint16_t nints; // number of ints (max number?)
uint16_t v3; // always 0 ?
int32_t v4; // offset to float pointer array ?
uint32_t nfloats;
int32_t v6; // offset to string pointer array ?
uint32_t nstrings;
int32_t v8; // always 8 ??
uint32_t numparams;
uint32_t getnints() const { return nints; }
uint32_t getnfloats() const { return nfloats; }
uint32_t getnstrings() const { return nstrings; }
uint32_t getnparams() const { return numparams; }
void read( std::istream & is )
{
    is.read( (char*)&v1, sizeof(v1));
    if( v1 == 0x01 ) {
        // direct (FIXME how should we detect this, much like TIFF ???)
        nints = 0;
        v3 = 0;
        v4 = 0;
        nfloats = 0;
        v6 = 0;
        nstrings = 0;
        v8 = 0;
        numparams = 0;
        uint32_t bla;
        is.read( (char*)&bla, sizeof(bla) );
        assert( bla == 0x2 || bla == 0x3 );
        nstrings = 1;
        numparams = 1;
    } else {
        // indirect
        is.read( (char*)&nints, sizeof(nints));
        is.read( (char*)&v3, sizeof(v3));
        assert( v3 == 0 ); // looks like this is always 0
        is.read( (char*)&v4, sizeof(v4));
        is.read( (char*)&nfloats, sizeof(nfloats));
        is.read( (char*)&v6, sizeof(v6));
        is.read( (char*)&nstrings, sizeof(nstrings));
        is.read( (char*)&v8, sizeof(v8));
        assert( v8 == 8 );
        is.read( (char*)&numparams, sizeof(numparams));
    }
}

void print( std::ostream & os )
{
    os << v1 << ", ";
    os << nints << ", ";
    os << v3 << ", ";
    os << v4 << ", ";
    os << nfloats << ", ";
    os << v6 << ", ";
    os << nstrings << ", ";
    os << v8 << ", ";
    os << numparams << std::endl;
}

};

struct param
{
    char name[32+1];
    uint8_t boolean;
    int32_t type;
    uint32_t dim;
    union {
        uint32_t val;
        char * ptr; } v4;
    int32_t /*std::streamoff*/ offset;
    param_type gettype() const { return (param_type)type; }
    uint32_t getdim() const { return dim; }
    void read_direct_int( std::istream & is ) {
        uint32_t bla;
        int max = 9;
        std::vector<uint32_t> v;
        for( int i = 0; i < max; ++i ) {
            is.read( (char*)&bla, sizeof(bla) );

```

```

        v.push_back( bla );
    }
    is.read( (char*)&bla, sizeof(bla) );
    char name0[32];
    memset(name0,0,sizeof(name0));
    assert( bla < sizeof(name0) );
    is.read( name0, bla);
    size_t l = strlen(name0);
    assert( l == bla ); (void)l;
    char * ptr = strdup( name0 );
    v4.ptr = ptr;
    type = param_string;
    dim = 1;
    offset = 0; // important !
}
void read_direct_string( std::istream & is ) {
    uint32_t bla;
    is.read( (char*)&bla, sizeof(bla) );
    char name0[32];
    memset(name0,0,sizeof(name0));
    assert( bla < sizeof(name0) );
    is.read( name0, bla);
    size_t l = strlen(name0);
    assert( l == bla ); (void)l;
    memcpy( this->name, name0, bla );
    is.read( (char*)&bla, sizeof(bla) );
    assert( bla == 0x1 );
    is.read( (char*)&bla, sizeof(bla) );
    char value[32];
    memset(value,0,sizeof(value));
    assert( bla < sizeof(value) );
    is.read( value, bla);
    is.read( (char*)&bla, sizeof(bla) );
    assert( bla == 0 ); // trailing stuff ?
    is.read( (char*)&bla, sizeof(bla) );
    assert( bla == 0 ); // trailing stuff ?
    const uint32_t cur = (uint32_t)is.tellg();
    std::cerr << "offset:" << cur << std::endl;
    if( cur == 65 )
        is.read( (char*)&bla, 1 );
    else if( cur == 66 )
        is.read( (char*)&bla, 1 );
    else if( cur == 122 )
        is.read( (char*)&bla, 2 );
    else
        assert(0);
    type = param_string;
    dim = 1;
    // FIXME: store the value in v4 for now:
    char * ptr = strdup( value );
    v4.ptr = ptr;
    offset = 0; // important !
}
void read( std::istream & is )
{
    is.read( name, 32 + 1);
    // This is always the same issue the string can contains garbage from previous run,
    // we need to print only until the first \0 character:
    assert( strlen( name ) <= 32 );
    is.read( (char*)&boolean,1);
    assert( boolean == 0 || boolean == 1 || boolean == 0x69 ); // some kind of bool, or digital trash ?
    is.read( (char*)&type, sizeof( type ) );
    assert( gettypenamefromtype( type ) );
    is.read( (char*)&dim, sizeof( dim ) ); // number of elements
    is.read( (char*)&v4.val, sizeof( v4.val ) );
    //assert( v4.val == 0 ); // always 0 ? sometimes not...
    const uint32_t cur = (uint32_t)is.tellg();
    is.read( (char*)&offset, sizeof( offset ) );
    assert( offset != 0 );
    offset += cur;
}

void print( std::ostream & os ) const
{
    os << name << ",";
    os << (int)boolean << ",";
    os << type << ",";
    os << dim << ",";
    os << v4.val << ",";
    os << offset << std::endl;
}

```

```

void printvalue( std::ostream & os, std::istream & is ) const
{
    if( offset ) {
        is.seekg( offset );
        switch( type )
        {
            case param_float:
            {
                os.precision(2);
                os << std::fixed;
                for( uint32_t idx = 0; idx < dim; ++idx )
                {
                    if( idx ) os << ",";
                    float v;
                    is.read( (char*)&v, sizeof(v) );
                    os << v; // what if the string contains \0 ?
                }
            }
            break;
            case param_integer:
            {
                int32_t v;
                for( uint32_t idx = 0; idx < dim; ++idx )
                {
                    if( idx ) os << ",";
                    is.read( (char*)&v, sizeof(v) );
                    os << v;
                }
            }
            break;
            case param_string:
            {
                int size = 81;
                std::string v;
                v.resize( size );
                for( uint32_t idx = 0; idx < dim; ++idx )
                {
                    if( idx ) os << ",";
                    is.read( &v[0], size );
                    os << v.c_str();
                }
            }
            break;
            case param_enum:
            {
                int32_t v;
                for( uint32_t idx = 0; idx < dim; ++idx )
                {
                    if( idx ) os << ",";
                    is.read( (char*)&v, sizeof(v) );
                    os << v;
                }
            }
            break;
        }
    }
    else {
#ifdef 1
        // direct
        assert ( type == param_string );
        char * ptr = v4.ptr;
        //std::string v;
        //v.resize( dim );
        //is.read( &v[0], dim );
        os << ptr;
#endif
    }
}

void printxml( std::ostream & os, std::istream & is ) const
{
    // <Attribute Name="CGEN_force_par_mode" Type="enum">0</Attribute>
    os << " <Attribute";
    os << " Name=\"" << name << "\"";
    os << " Type=\"" << gettypenamefromtype(type) << "\"";
    if( dim != 1 )
    {
        os << " ArraySize=\"" << dim << "\"";
    }
    os << ">";
    printvalue( os, is );
    os << "</Attribute>\n";
}

```

```

    }
    void printcsv( std::ostream & os, std::istream & is ) const
    {
        os << std::setw(32) << std::left << name << ", ";
        os << std::setw(7) << std::right << gettypenameefromtype(type) << ", ";
        os << std::setw(4) << dim << ", ";
        os << " ";
        printvalue( os, is );
        os << ",\n";
    }
};

static bool ProcessNested( gdcm::DataSet & ds )
{
    /*
    TODO:
    Looks like the real length of the blob is stored here:
    (2005,1132) SQ                                     # u/1,1 ?
    (fffe,e000) na (Item with undefined length)
    (2005,0011) LO [Philips MR Imaging DD 002 ]         # 26,1 Private Creator
    (2005,1143) SL 3103                                 # 4,1 ?

    Wotsit ?
    (2005,1132) SQ                                     # u/1,1 ?
    (fffe,e000) na (Item with undefined length)
    (2005,0011) LO [Philips MR Imaging DD 002 ]         # 26,1 Private Creator
    (2005,1147) CS [Y ]                                # 2,1 ?
    */
    bool ret = false;

    // (2005,1137) PN (LO) [PDF_CONTROL_GEN_PARS]        # 20,1 Protocol Data Name
    const gdcm::PrivateTag pt0(0x2005,0x37,"Philips MR Imaging DD 002");
    if( !ds.FindDataElement( pt0 ) ) return false;
    const gdcm::DataElement &de0 = ds.GetDataElement( pt0 );
    if( de0.IsEmpty() ) return false;
    const gdcm::ByteValue * bv0 = de0.GetByteValue();
    std::string s0( bv0->GetPointer() , bv0->GetLength() );

    // (2005,1139) LO [IEEE_PDF]                        # 8,1 Protocol Data Type
    const gdcm::PrivateTag pt1(0x2005,0x39,"Philips MR Imaging DD 002");
    if( !ds.FindDataElement( pt1 ) ) return false;
    const gdcm::DataElement &de1 = ds.GetDataElement( pt1 );

    // (2005,1143) SL 53                                 # 4,1 Protocol Data Block Length (non-padded)
    const gdcm::PrivateTag pt2(0x2005,0x43,"Philips MR Imaging DD 002");
    if( !ds.FindDataElement( pt2 ) ) return false;
    const gdcm::DataElement &de2 = ds.GetDataElement( pt2 );

    // (2005,1147) CS [Y ]                              # 2,1 Protocol Data Boolean
    const gdcm::PrivateTag pt3(0x2005,0x47,"Philips MR Imaging DD 002");
    if( !ds.FindDataElement( pt3 ) ) return false;
    const gdcm::DataElement &de3 = ds.GetDataElement( pt3 );
    (void)de3;

    // (2005,1144) OW 00\00\00\00\05\00\00\00\35\2e\31\2e\37\00 # 54,1 Protocol Data Block
    const gdcm::PrivateTag pt(0x2005,0x44,"Philips MR Imaging DD 002");
    if( !ds.FindDataElement( pt ) ) return false;
    const gdcm::DataElement &de = ds.GetDataElement( pt );
    if( de.IsEmpty() ) return false;
    const gdcm::ByteValue * bv = de.GetByteValue();

    if( s0 == "ExamCardBlob" )
    {
        assert( de1.IsEmpty() );

        std::string fn = gdcm::LOComp::Trim( s0.c_str() ); // remove trailing space
        fn += ".xml";
        std::ofstream out( fn.c_str() );

        // remove trailing \0
        size_t len = strlen( bv->GetPointer() );
        out.write( bv->GetPointer() , len );
        out.close();

        // Extract binary64 thingy (this is a ugly hack, better use an XML parser)
        std::string dup( bv->GetPointer(), len );
        std::string::size_type pos1 = dup.find( "<ExamCardBlob>" );

```

```

std::string::size_type pos2 = dup.find( "</ExamCardBlob>" );

std::string b64( bv->GetPointer() + pos1 + 14, pos2 - (pos1 + 14) );

// ugly hack to remove \r\n from input base64:
std::string::iterator r_pos = std::remove(b64.begin(), b64.end(), '\r');
b64.erase(r_pos, b64.end());
std::string::iterator n_pos = std::remove(b64.begin(), b64.end(), '\n');
b64.erase(n_pos, b64.end());
#ifdef 0
std::ofstream out2( "debug" );
out2.write( b64.c_str(), b64.size() );
out2.close();
#endif

const size_t dlen = gdcm::Base64::GetDecodeLength(b64.c_str(), b64.size() );

std::string decoded;
decoded.resize( dlen );
gdcm::Base64::Decode( &decoded[0], decoded.size(), b64.c_str(), b64.size() );

std::ofstream f64( "soap.xml" );
f64.write( decoded.c_str(), decoded.size() );
f64.close();

ret = true;
}
else
{
    if( del.IsEmpty() ) return false;
    const gdcm::ByteValue * bv1 = del.GetByteValue();
    gdcm::Element<gdcm::VR::SL, gdcm::VM::VML> dlen = {{0L}};
    dlen.SetFromDataElement( de2 );
    std::string s1( bv1->GetPointer() , bv1->GetLength() );

    if( s1 == "IEEE_PDF" )
    {
        std::istream is;
        assert( bv->GetLength() == (size_t)dlen.GetValue() || bv->GetLength() == (size_t)(dlen.GetValue() + 1) );
        std::string dup( bv->GetPointer(), dlen.GetValue() /*bv->GetLength()*/ );
        is.str( dup );

        header h;
        h.read( is );
        //assert( is.peek() && is.eof() );
    }
#ifdef 1
    static int c = 0;
    std::string fn0 = gdcm::LOComp::Trim( s1.c_str() ); // remove trailing space
    std::stringstream ss;
    ss << fn0 << "_" << c++;
    if( h.v1 == 0x01 )
        ss << ".direct";
    else
        ss << ".indirect";
    std::cout << "fn0=" << ss.str() << " Len= " << bv->GetLength() << std::endl;
    std::ofstream out( ss.str().c_str() );
    out.write( bv->GetPointer(), bv->GetLength() );
    out.close();
#endif
#ifdef 1
    std::cout << dup.c_str() << std::endl;
    h.print( std::cout );
#endif

    std::vector< param > params;
    if( h.v1 == 0x01 ) {
        for( uint32_t i = 0; i < 1 /* h.getnparams()*/; ++i ) {
            param p;
            if( s0 == "HARDWARE_CONFIG" )
            {
                p.read_direct_int( is );
            }
            else if( s0 == "COILSTATE" )
            {
                p.read_direct_string( is );
            }
            else
            {
                assert(0);
            }
        }
    }
}

```



```

        params.push_back( p );
    }
} else {
    assert( is.tellg() == std::streampos(0x20) );
    is.seekg( 0x20 );

    param p;
    for( uint32_t i = 0; i < h.getnparams(); ++i )
    {
        p.read( is );
        //p.print( std::cout );
        params.push_back( p );
    }
}

std::string fn = gdc::LOComp::Trim( s0.c_str() ); // remove trailing space
bool b1 = isvalidpdfstring( fn.c_str() );
assert( b1 ); (void)b1;
fn += ".csv";
//fn += ".xml";
std::ofstream csv( fn.c_str() );

// let's do some bookkeeping:
uint32_t nfloats = 0;
uint32_t nints = 0;
uint32_t nstrings = 0;
for( std::vector<param>::const_iterator it = params.begin();
    it != params.end(); ++it )
{
    param_type type = it->gettype();
    switch( type )
    {
        case param_float:
            nfloats += it->getdim();
            break;
        case param_integer:
            nints += it->getdim();
            break;
        case param_string:
            nstrings += it->getdim();
            break;
        default:
            ;
    }
}

#if 0
std::cout << "Stats:" << std::endl;
std::cout << "nfloats:" << nfloats << std::endl;
std::cout << "nints:" << nints << std::endl;
std::cout << "nstrings:" << nstrings << std::endl;
#endif

#endif
assert( h.getnints() >= nints );
assert( h.getnfloats() >= nfloats );
assert( h.getnstrings() >= nstrings );

for( uint32_t i = 0; i < h.getnparams(); ++i )
{
    params[i].printcsv( csv, is );
    //params[i].printxml( csv, is );
}
csv.close();
ret = true;
}
else if( s1 == "ASCII " )
{
    #if 0
    std::cerr << "ASCII is not handled" << std::endl;
    std::string fn = gdc::LOComp::Trim( s0.c_str() ); // remove trailing space
    fn += ".asc";
    std::ofstream out( fn.c_str() );
    out.write( bv->GetPointer() , bv->GetLength() );
    out.close();
    #endif

    std::string fn = gdc::LOComp::Trim( s0.c_str() ); // remove trailing space
    fn += ".sin";
    std::ofstream sin( fn.c_str() );

    const char *beg = bv->GetPointer();
    const char *end = beg + bv->GetLength();
    assert( *beg == 0 );
    const char *p = beg + 1; // skip first \0

```

```

size_t prev = 0;
for( ; p != end; ++p )
{
    if( *p == 0 )
    {
        const char *s = beg + prev + 1;
        if( *s )
        {
            sin << s << std::endl;
        }
        else
        {
            sin << std::endl;
        }
        prev = p - beg;
    }
}
sin.close();

ret = true;
}
else if( sl == "BINARY" )
{
    std::cerr << "BINARY is not handled" << std::endl;
    std::string fn = gdcm::LOComp::Trim( s0.c_str() ); // remove trailing space
    fn += ".bin";
    std::ofstream out( fn.c_str() );
    //out.write( bv->GetPointer() + 512, bv->GetLength() - 512);
    out.write( bv->GetPointer() , bv->GetLength() );
    out.close();

#ifdef 0
    int array[ 128 ];
    memcpy( array, bv->GetPointer(), 512 );
    for( int i = 0; i < 14; ++i )
    {
        std::cout << array[i] << std::endl;
    }
#endif

    ret = true;
}
// else -> ret == false
assert( ret );

return ret;
}

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }
    const gdcm::DataSet& ds = reader.GetFile().GetDataSet();
    /*
(2005,1132) SQ                                     # u/1,1 ?
(fffe,e000) na (Item with undefined length)
(2005,0011) LO [Philips MR Imaging DD 002 ]          # 26,1 Private Creator
(2005,1137) PN (LO) [PDF_CONTROL_GEN_PARS]          # 20,1 ?
(2005,1138) PN (LO) (no value)                      # 0,1 ?
(2005,1139) PN (LO) [IEEE_PDF]                     # 8,1 ?
(2005,1140) PN (LO) (no value)                      # 0,1 ?
(2005,1141) PN (LO) (no value)                      # 0,1 ?
(2005,1143) SL 3103                                 # 4,1 ?
(2005,1144) OW
66050000003b010000004a0a0000000e000000007a0a00009501000000080000001b000000004347454e5f757365725
# 3104,1 ?
(2005,1147) CS [Y ]                                # 2,1 ?
(fffe,e00d)
*/
    const gdcm::PrivateTag pt(0x2005,0x32,"Philips MR Imaging DD 002");
    if( !ds.FindDataElement( pt ) ) return 1;
    const gdcm::DataElement &de = ds.GetDataElement( pt );

```

```

if( de.IsEmpty() ) return 1;

gdcm::SequenceOfItems *sqi = de.GetValueAsSQ();
if ( !sqi ) return 1;
gdcm::SequenceOfItems::SizeType s = sqi->GetNumberOfItems();
for( gdcm::SequenceOfItems::SizeType i = 1; i <= s; ++i )
{
    gdcm::Item &item = sqi->GetItem(i);

    gdcm::DataSet &nestedds = item.GetNestedDataSet();

    if( !ProcessNested( nestedds ) ) {
        std::cerr << "Error processing Item #" << i << std::endl;
    }
}

return 0;
}

```

## 12.51 DumpGEMSMovieGroup.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmImage.h"
#include "gdcmImageWriter.h"
#include "gdcmDataElement.h"
#include "gdcmPrivateTag.h"
#include "gdcmUIDGenerator.h"

#include <iostream>
#include <string>

#include <map>

bool PrintNameValueMapping( gdcm::SequenceOfItems *sqi_values,
gdcm::SequenceOfItems *sqi_names, std::string const & indent )
{
    using namespace gdcm;
    // prepare names mapping:
    typedef VRToType<VR::UL::Type> UL;
    std::map< UL, std::string > names;
    assert( sqi_names );
    assert( sqi_values );
    SequenceOfItems::SizeType s = sqi_names->GetNumberOfItems();
    PrivateTag tindex(0x7fe1,0x71,"GEMS_Ultrasound_MovieGroup_001");
    PrivateTag tname (0x7fe1,0x72,"GEMS_Ultrasound_MovieGroup_001");
    // First sequence contains all possible names (this is a dict)
    for( SequenceOfItems::SizeType i = 1; i <= s; ++i )
    {
        const Item & item = sqi_names->GetItem( i );
        const DataSet & ds = item.GetNestedDataSet();
        if( !ds.FindDataElement( tindex )
            || !ds.FindDataElement( tname ) )
        {
            assert( 0 );
            return false;
        }
        const DataElement & index = ds.GetDataElement( tindex );
        const DataElement & name = ds.GetDataElement( tname );
        if( index.IsEmpty() || name.IsEmpty() )
        {
            assert( 0 );
            return false;
        }
        gdcm::Element<VR::UL, VM::VM1> ell;
    }
}

```

```

    el1.SetFromDataElement( index );

    gdcm::Element<VR::LO, VM::VM1> el2;
    el2.SetFromDataElement( name );
//    std::cout << el1.GetValue() << " " << el2.GetValue() << std::endl;
    names.insert( std::make_pair( el1.GetValue(), el2.GetValue() ) );
}

SequenceOfItems::SizeType s2 = sqi_values->GetNumberOfItems();
assert( s2 <= s );
PrivateTag tindex2(0x7fe1,0x48,"GEMS_Ultrasound_MovieGroup_001");
for( SequenceOfItems::SizeType i = 1; i <= s2; ++i )
{
    const Item & item = sqi_values->GetItem( i );
    const DataSet & ds = item.GetNestedDataSet();
    if( !ds.FindDataElement( tindex2 ) )
    {
        assert( 0 );
        return false;
    }
    const DataElement & index2 = ds.GetDataElement( tindex2 );
    if( index2.IsEmpty() )
    {
        assert( 0 );
        return false;
    }
    gdcm::Element<VR::FD, VM::VM1_2> el1;
    el1.SetFromDataElement( index2 );

    UL copy = (UL)el1.GetValue();
#if 1
    std::cout << indent;
    std::cout << "( " << names[ copy ];
#endif
    // (7fe1,1052) FD 1560 # 8,1 ?
    // (7fe1,1057) LT [MscSkelSup] # 10,1 ?
    //PrivateTag tvalue(0x7fe1,0x52,"GEMS_Ultrasound_MovieGroup_001");
    PrivateTag tvalueint(0x7fe1,0x49,"GEMS_Ultrasound_MovieGroup_001"); // UL
    PrivateTag tvaluefloat1(0x7fe1,0x51,"GEMS_Ultrasound_MovieGroup_001"); // FL
    PrivateTag tvaluefloat(0x7fe1,0x52,"GEMS_Ultrasound_MovieGroup_001"); // FD
    PrivateTag tvalueul(0x7fe1,0x53,"GEMS_Ultrasound_MovieGroup_001"); // UL
    PrivateTag tvaluesl(0x7fe1,0x54,"GEMS_Ultrasound_MovieGroup_001"); // SL
    PrivateTag tvalueob(0x7fe1,0x55,"GEMS_Ultrasound_MovieGroup_001"); // OB
    PrivateTag tvaluetext(0x7fe1,0x57,"GEMS_Ultrasound_MovieGroup_001"); // LT
    PrivateTag tvaluefd(0x7fe1,0x77,"GEMS_Ultrasound_MovieGroup_001"); // FD / 1-N
    PrivateTag tvaluesl3(0x7fe1,0x79,"GEMS_Ultrasound_MovieGroup_001"); // SL / 1-N
    PrivateTag tvaluesl2(0x7fe1,0x86,"GEMS_Ultrasound_MovieGroup_001"); // SL ??
    PrivateTag tvaluefdl(0x7fe1,0x87,"GEMS_Ultrasound_MovieGroup_001"); // FD / 1-N
    PrivateTag tvaluefloat2(0x7fe1,0x88,"GEMS_Ultrasound_MovieGroup_001"); // FD ??
#if 1
    std::cout << " ) = ";
#endif
    if( ds.FindDataElement( tvalueint ) )
    {
        const DataElement & value = ds.GetDataElement( tvalueint );
        gdcm::Element<VR::UL,VM::VM1> el2;
        el2.SetFromDataElement( value );
        std::cout << el2.GetValue() << std::endl;
    }
    else if( ds.FindDataElement( tvaluefloat1 ) )
    {
        const DataElement & value = ds.GetDataElement( tvaluefloat1 );
        gdcm::Element<VR::FL,VM::VM1> el2;
        el2.SetFromDataElement( value );
        std::cout << el2.GetValue() << std::endl;
    }
    else if( ds.FindDataElement( tvaluefloat ) )
    {
        const DataElement & value = ds.GetDataElement( tvaluefloat );
        gdcm::Element<VR::FD,VM::VM1> el2;
        el2.SetFromDataElement( value );
        std::cout << el2.GetValue() << std::endl;
    }
    else if( ds.FindDataElement( tvaluesl ) )
    {
        const DataElement & value = ds.GetDataElement( tvaluesl );
        gdcm::Element<VR::SL,VM::VM1> el2;
        el2.SetFromDataElement( value );
        std::cout << el2.GetValue() << std::endl;
    }
    else if( ds.FindDataElement( tvalueul ) )

```

```

    {
        const DataElement & value = ds.GetDataElement( tvalueul );
        gdcmm::Element<VR::UL,VM::VM1_n> el2;
        el2.SetFromDataElement( value );
        assert( el2.GetLength() == 1 );
        std::cout << el2.GetValue() << std::endl;
    }
    else if( ds.FindDataElement( tvalueob ) )
    {
        const DataElement & value = ds.GetDataElement( tvalueob );
        gdcmm::Element<VR::SL,VM::VM1> el2;
        el2.SetFromDataElement( value );
        std::cout << el2.GetValue() << std::endl;
        std::cout << value << std::endl;
    }
    else if( ds.FindDataElement( tvaluetext ) )
    {
        const DataElement & value = ds.GetDataElement( tvaluetext );
        gdcmm::Element<VR::LT,VM::VM1> el2;
        el2.SetFromDataElement( value );
        std::cout << el2.GetValue() << std::endl;
    }
    else if( ds.FindDataElement( tvaluesl2 ) )
    {
        const DataElement & value = ds.GetDataElement( tvaluesl2 );
        gdcmm::Element<VR::SL,VM::VM1_n> el2;
        el2.SetFromDataElement( value );
        el2.Print( std::cout );
        assert( el2.GetLength() == 4 );
        std::cout << std::endl;
    }
    else if( ds.FindDataElement( tvaluesl3 ) )
    {
        const DataElement & value = ds.GetDataElement( tvaluesl3 );
        gdcmm::Element<VR::SL,VM::VM1_n> el2;
        el2.SetFromDataElement( value );
        el2.Print( std::cout );
        // assert( el2.GetLength() == 4 );
        std::cout << std::endl;
    }
    else if( ds.FindDataElement( tvaluefd ) )
    {
        const DataElement & value = ds.GetDataElement( tvaluefd );
        gdcmm::Element<VR::FD,VM::VM1_n> el2;
        el2.SetFromDataElement( value );
        el2.Print( std::cout );
        // assert( el2.GetLength() == 4 || el2.GetLength() == 3 || el2.GetLength() == 8 );
        std::cout << std::endl;
    }
    else if( ds.FindDataElement( tvaluefloat2 ) )
    {
        const DataElement & value = ds.GetDataElement( tvaluefloat2 );
        gdcmm::Element<VR::FD,VM::VM1_n> el2;
        el2.SetFromDataElement( value );
        el2.Print( std::cout );
        assert( el2.GetLength() == 2 );
        std::cout << std::endl;
    }
    else if( ds.FindDataElement( tvaluefd1 ) )
    {
        const DataElement & value = ds.GetDataElement( tvaluefd1 );
        gdcmm::Element<VR::FD,VM::VM1_n> el2;
        el2.SetFromDataElement( value );
        el2.Print( std::cout );
        assert( el2.GetLength() == 4 );
        std::cout << std::endl;
    }
    else
    {
        std::cout << "(no value)" << std::endl;
        // std::cout << ds << std::endl;
        assert( ds.Size() == 2 );
    }
}
return true;
}

bool PrintNameValueMapping2( gdcmm::PrivateTag const & privtag, const gdcmm::DataSet & ds,
    gdcmm::SequenceOfItems *sqi_names, std::string const & indent )
{
    if( !ds.FindDataElement( privtag ) ) return false;

```

```

const gdcm::DataElement& seq_values = ds.GetDataElement( privtag );
gdcm::SmartPointer<gdcm::SequenceOfItems> sqi = seq_values.GetValueAsSQ();

return PrintNameValueMapping( sqi, sqi_names, indent);
}

bool PrintNameValueMapping3( gdcm::PrivateTag const & privtag1, gdcm::PrivateTag const & privtag2, const
    gdcm::DataSet & ds,
    gdcm::SequenceOfItems *sqi_names, std::string const & indent )
{
    if( !ds.FindDataElement( privtag1 ) )
    {
        assert( 0 );
        return false;
    }
    const gdcm::DataElement& values10name = ds.GetDataElement( privtag1 );
    gdcm::Element<gdcm::VR::LO, gdcm::VM::VM1> el;
    el.SetFromDataElement( values10name );
    std::cout << std::endl;
    std::cout << " <" << el.GetValue().c_str() << ">" << std::endl;

    return PrintNameValueMapping2( privtag2, ds, sqi_names, indent);
}

bool print73( gdcm::DataSet const & ds10, gdcm::SequenceOfItems *sqi_dict, std::string const & indent )
{
    const gdcm::PrivateTag tseq_values73(0x7fe1, 0x73, "GEMS_Ultrasound_MovieGroup_001");
    if( !ds10.FindDataElement( tseq_values73 ) )
    {
        std::cout << indent << "No group 73" << std::endl;
        return false;
    }
    const gdcm::DataElement& seq_values73 = ds10.GetDataElement( tseq_values73 );
    gdcm::SmartPointer<gdcm::SequenceOfItems> sqi_values73 = seq_values73.GetValueAsSQ();

    size_t ni3 = sqi_values73->GetNumberOfItems();
    for( size_t i3 = 1; i3 <= ni3; ++i3 )
    {
        gdcm::Item &item_73 = sqi_values73->GetItem(i3);
        gdcm::DataSet &ds73 = item_73.GetNestedDataSet();
        assert( ds73.Size() == 3 );

        const gdcm::PrivateTag tseq_values74name(0x7fe1, 0x74, "GEMS_Ultrasound_MovieGroup_001");
        const gdcm::PrivateTag tseq_values75(0x7fe1, 0x75, "GEMS_Ultrasound_MovieGroup_001");
        PrintNameValueMapping3( tseq_values74name, tseq_values75, ds73, sqi_dict, indent);
        std::cout << std::endl;
    }
    return true;
}

bool print36( gdcm::DataSet const & ds10, gdcm::SequenceOfItems *sqi_dict, std::string const & indent )
{
    (void)sqi_dict;
    const gdcm::PrivateTag tseq_values36(0x7fe1, 0x36, "GEMS_Ultrasound_MovieGroup_001");
    if( !ds10.FindDataElement( tseq_values36 ) )
    {
        std::cout << indent << "No group 36" << std::endl;
        return false;
    }
    const gdcm::DataElement& seq_values36 = ds10.GetDataElement( tseq_values36 );
    gdcm::SmartPointer<gdcm::SequenceOfItems> sqi_values36 = seq_values36.GetValueAsSQ();

    size_t ni3 = sqi_values36->GetNumberOfItems();
    assert( ni3 >= 1 );
    for( size_t i3 = 1; i3 <= ni3; ++i3 )
    {
        gdcm::Item &item_36 = sqi_values36->GetItem(i3);
        gdcm::DataSet &ds36 = item_36.GetNestedDataSet();
        assert( ds36.Size() == 4 );

        // (7fe1,1037) UL 47 # 4,1 US MovieGroup Number of Frames
        // (7fe1,1043) OB 40\00\1c\c4\67\2f\0b\11\40 # 376,1 ?
        // (7fe1,1060) OB 4e\4e\49\4f\4e\47\46\43\2a # 4562714,1 US MovieGroup Image Data
        //
        const gdcm::PrivateTag timagedata(0x7fe1, 0x60, "GEMS_Ultrasound_MovieGroup_001");
        assert( ds36.FindDataElement( timagedata ) );
        gdcm::DataElement const & imagedata = ds36.GetDataElement( timagedata );

        const gdcm::ByteValue * bv = imagedata.GetByteValue();
        assert( bv );
        static int c = 0;

```

```

        std::stringstream ss;
        ss << "/tmp/debug";
        ss << c++;
        std::ofstream os( ss.str().c_str(), std::ios::binary );
        os.write( bv->GetPointer(), bv->GetLength() );
        os.close();

        //const gdcm::PrivateTag tseq_values85(0x7fe1,0x85,"GEMS_Ultrasound_MovieGroup_001");
        //PrintNameValueMapping3( tseq_values84name, tseq_values85, ds83, sqi_dict, indent);
        //std::cout << std::endl;
    }
    return true;
}

bool print83( gdcm::DataSet const & ds10, gdcm::SequenceOfItems *sqi_dict, std::string const & indent )
{
    const gdcm::PrivateTag tseq_values83(0x7fe1,0x83,"GEMS_Ultrasound_MovieGroup_001");
    if( !ds10.FindDataElement( tseq_values83 ) )
    {
        std::cout << indent << "No group 83" << std::endl;
        return false;
    }
    const gdcm::DataElement& seq_values83 = ds10.GetDataElement( tseq_values83 );
    gdcm::SmartPointer<gdcm::SequenceOfItems> sqi_values83 = seq_values83.GetValueAsSQ();

    size_t ni3 = sqi_values83->GetNumberOfItems();
    for( size_t i3 = 1; i3 <= ni3; ++i3 )
    {
        gdcm::Item &item_83 = sqi_values83->GetItem(i3);
        gdcm::DataSet &ds83 = item_83.GetNestedDataSet();
        assert( ds83.Size() == 3 );

        const gdcm::PrivateTag tseq_values84name(0x7fe1,0x84,"GEMS_Ultrasound_MovieGroup_001");
        const gdcm::PrivateTag tseq_values85(0x7fe1,0x85,"GEMS_Ultrasound_MovieGroup_001");
        PrintNameValueMapping3( tseq_values84name, tseq_values85, ds83, sqi_dict, indent);
        std::cout << std::endl;
    }
    return true;
}

bool PrintNameValueMapping4( gdcm::PrivateTag const & privtag0, const gdcm::DataSet & subds, gdcm::PrivateTag
    const & privtag1, gdcm::PrivateTag const & privtag2,
    gdcm::SequenceOfItems *sqi_dict, std::string const & indent )
{
    (void)indent;
    if( !subds.FindDataElement( privtag0 ) )
    {
        assert( 0 );
        return false;
    }
    const gdcm::DataElement& seq_values10 = subds.GetDataElement( privtag0 );
    gdcm::SmartPointer<gdcm::SequenceOfItems> sqi_values10 = seq_values10.GetValueAsSQ();

    size_t nil = sqi_values10->GetNumberOfItems();
    // assert( nil == 1 );
    for( size_t i1 = 1; i1 <= nil; ++i1 )
    {
        gdcm::Item &item_10 = sqi_values10->GetItem(i1);
        gdcm::DataSet &ds10 = item_10.GetNestedDataSet();
        assert( ds10.Size() == 2 + 3 );
        // (7fe1,0010)
        // (7fe1,1012)
        // (7fe1,1018)
        // (7fe1,1020)
        // (7fe1,1083)

        PrintNameValueMapping3( privtag1, privtag2, ds10, sqi_dict, " " );
        std::cout << std::endl;

        const gdcm::PrivateTag tseq_values20(0x7fe1,0x20,"GEMS_Ultrasound_MovieGroup_001");
        if( !ds10.FindDataElement( tseq_values20 ) )
        {
            assert( 0 );
            return false;
        }
        const gdcm::DataElement& seq_values20 = ds10.GetDataElement( tseq_values20 );
        gdcm::SmartPointer<gdcm::SequenceOfItems> sqi_values20 = seq_values20.GetValueAsSQ();

        size_t ni2 = sqi_values20->GetNumberOfItems();
        //assert( ni == 1 );
        for( size_t i2 = 1; i2 <= ni2; ++i2 )
        {

```

```

    gdcmm::Item &item_20 = sqi_values20->GetItem(i2);
    gdcmm::DataSet &ds20 = item_20.GetNestedDataSet();
    size_t count = ds20.Size(); (void)count;
    assert( ds20.Size() == 2 + 3 || ds20.Size() == 2 + 2 );
    // (7fe1,0010)
    // (7fe1,1024)
    // (7fe1,1026)
    // (7fe1,1036)
    // (7fe1,103a)
    // (7fe1,1083) (*)

    const gdcmm::PrivateTag tseq_values20name(0x7fe1,0x24,"GEMS_Ultrasound_MovieGroup_001");
    const gdcmm::PrivateTag tseq_values26(0x7fe1,0x26,"GEMS_Ultrasound_MovieGroup_001");
    PrintNameValueMapping3( tseq_values20name, tseq_values26, ds20, sqi_dict, "  ");
    std::cout << std::endl;

    print36(ds20, sqi_dict, "  ");
    print83(ds20, sqi_dict, "  ");
}

print83(ds10, sqi_dict, "  ");
}
return true;
}

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    using namespace gdcmm;
    const char *filename = argv[1];
    gdcmm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() ) return 1;

    gdcmm::File &file = reader.GetFile();
    gdcmm::DataSet &ds = file.GetDataSet();
    const PrivateTag tseq(0x7fe1,0x1,"GEMS_Ultrasound_MovieGroup_001");

    if( !ds.FindDataElement( tseq ) ) return 1;
    const DataElement& seq = ds.GetDataElement( tseq );

    SmartPointer<SequenceOfItems> sqi = seq.GetValueAsSQ();
    assert( sqi->GetNumberOfItems() == 1 );

    Item &item = sqi->GetItem(1);
    DataSet &subds = item.GetNestedDataSet();

    const PrivateTag tseq_dict(0x7fe1,0x70,"GEMS_Ultrasound_MovieGroup_001");
    if( !subds.FindDataElement( tseq_dict ) ) return 1;
    const DataElement& seq_dict = subds.GetDataElement( tseq_dict );
    SmartPointer<SequenceOfItems> sqi_dict = seq_dict.GetValueAsSQ();

    const PrivateTag tseq_values8(0x7fe1,0x8,"GEMS_Ultrasound_MovieGroup_001");
    if( !subds.FindDataElement( tseq_values8 ) ) return 1;
    const DataElement& seq_values8 = subds.GetDataElement( tseq_values8 );
    SmartPointer<SequenceOfItems> sqi_values8 = seq_values8.GetValueAsSQ();

    const PrivateTag tseq_values8name(0x7fe1,0x2,"GEMS_Ultrasound_MovieGroup_001");
    if( !subds.FindDataElement( tseq_values8name ) ) return 1;
    const DataElement& values8name = subds.GetDataElement( tseq_values8name );
    {
        Element<VR::LO,VM::VM1> el;
        el.SetFromDataElement( values8name );
        std::cout << el.GetValue() << std::endl;
    }

    size_t count = subds.Size(); (void)count;
    assert( subds.Size() == 3 + 2 + 1 || subds.Size() == 3 + 2 + 2 );

    // (7fe1,0010) # 30,1 Private Creator
    // (7fe1,1002) # 8,1 US MovieGroup Value 0008 Name
    // (7fe1,1003) # 4,1 ?
    // (7fe1,1008) # 8140,1 US MovieGroup Value 0008 Sequence
    // (7fe1,1010) # 1372196,1 ?
    // (7fe1,1070) # 33684,1 US MovieGroup Dict
    // (7fe1,1073) (*)
    PrintNameValueMapping( sqi_values8, sqi_dict, "  ");

    const PrivateTag tseq_values10(0x7fe1,0x10,"GEMS_Ultrasound_MovieGroup_001");
    const PrivateTag tseq_values10name(0x7fe1,0x12,"GEMS_Ultrasound_MovieGroup_001");
    const PrivateTag tseq_values18(0x7fe1,0x18,"GEMS_Ultrasound_MovieGroup_001");

```



```

PrintNameValueMapping4( tseq_values10, subds, tseq_values10name, tseq_values18, sqi_dict, " ");

print73( subds, sqi_dict, " " );

#if 0
gdcmm::DataSet::ConstIterator it = subds.Begin();
for( ; it != subds.End(); ++it )
{
    const gdcmm::DataElement &de = *it;
    std::cout << de.GetTag() << std::endl;
}
#endif

return 0;
}

```

## 12.52 DumpImageHeaderInfo.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * Dump TOSHIBA MDW HEADER / Image Header Info
 */
#include "gdcmmReader.h"
#include "gdcmmPrivateTag.h"
#include "gdcmmAttribute.h"
#include "gdcmmImageWriter.h"

#include <iostream>
#include <fstream>
#include <vector>

#include <string.h>
#include <assert.h>
#include <stdint.h>

struct element
{
    std::istream & read( std::istream & is );
};

std::istream & element::read( std::istream & is )
{
    static const uint32_t ref = 0xe000ffff;
    std::ostream &os = std::cout;
    if( is.eof() )
    {
        return is;
    }
    uint32_t magic;
    if( !is.read( (char*)&magic, sizeof(magic) ) )
    {
        return is;
    }
    //os << magic << std::endl;
    assert( magic == ref ); (void)ref;

    uint32_t l;
    is.read( (char*)&l, sizeof(l) );
    //os << l << std::endl;

    char str[17];
    str[16] = 0;
    is.read( str, 16 );
    os << str << " (" << l << ")" << std::endl;
    std::vector<char> bytes;

```



```

reader.SetFileName( filename );
if( !reader.Read() )
{
    std::cerr << "Failed to read: " << filename << std::endl;
    return 1;
}
const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

const gdcm::PrivateTag timageheaderinfo(0x0029,0x10,"TOSHIBA MDW HEADER");
if( !ds.FindDataElement( timageheaderinfo ) ) return 1;
const gdcm::DataElement& imageheaderinfo = ds.GetDataElement( timageheaderinfo );
if ( imageheaderinfo.IsEmpty() ) return 1;
const gdcm::ByteValue * bv = imageheaderinfo.GetByteValue();

std::istringstream is;
std::string dup( bv->GetPointer(), bv->GetLength() );
is.str( dup );
bool b = DumpImageHeaderInfo( is, bv->GetLength() );
if( !b ) return 1;

#if 0
const float d1 = 0.00416666668839752674; // 89 88 88 3B // 0x44c
//const float d1 = 0.053231674455417881;
const float d2 = 0.10828025639057159; // 0A C2 DD 3D // 0x1ac
//const float d1 = 0.17869562069272813;
//const unsigned int d2 = 4294967280;
const float d3 = 0.10828025639057159; // 0A C2 DD 3D // 0x15c
const int32_t d4 = 134;
const uint32_t d5 = 1153476;
std::ofstream t("/tmp/debug", std::ios::binary );
//t.write( (char*)&d0, sizeof( d0 ) );
t.write( (char*)&d1, sizeof( d1 ) );
t.write( (char*)&d2, sizeof( d2 ) );
t.write( (char*)&d3, sizeof( d3 ) );
t.write( (char*)&d4, sizeof( d4 ) );
t.write( (char*)&d5, sizeof( d5 ) );
t.close();
#endif

return 0;
}

```

## 12.53 DumpPhilipsECHO.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmDeflateStream.h"
#include "gdcm_zlib.h"

/*
 * This example extract the ZLIB compressed US image from a Philips private tag
 *
 * Everything done in this code is for the sole purpose of writing interoperable
 * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
 * If you believe anything in this code violates any law or any of your rights,
 * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
 * find a solution.
 *
 * Everything you do with this code is at your own risk, since decompression
 * algorithm was not written from specification documents.
 *
 * Usage:
 *
 * $ DumpPhilipsECHO private_us.dcm raw_us_img.raw
 * $ gdcming --sop-class-uid 1.2.840.10008.5.1.4.1.1.3.1 --size 608,427,88 raw_us_img.raw raw_us_img.dcm
 */

```

```

*/
// header:
struct hframe
{
    uint32_t val0; // 800 increment ?
    uint16_t val1[2];
    uint16_t val2[2];
    uint32_t imgsize;

    bool operator==(const hframe &h) const
    {
        return val0 == h.val0 &&
            val1[0] == h.val1[0] &&
            val1[1] == h.val1[1] &&
            val2[0] == h.val2[0] &&
            val2[1] == h.val2[1] &&
            imgsize == h.imgsize;
    }
};

static bool ProcessDeflate( const char *outfilename, const int nslices, const
    int buf_size, const char *buf, const std::streampos len,
    const char *crdbuf, const size_t crclen )
{
    std::vector< hframe > crchheaders;
    crchheaders.reserve( nslices );
    {
        std::istream is;
        is.str( std::string( crdbuf, crclen ) );
        hframe header;
        for( int r = 0; r < nslices; ++r )
        {
            is.read( (char*)&header, sizeof( header ) );
#ifdef 0
            std::cout << header.val0
                << " " << header.val1[0]
                << " " << header.val1[1]
                << " " << header.val2[0]
                << " " << header.val2[1]
                << " " << header.imgsize << std::endl;
#endif
            crchheaders.push_back( header );
        }

        std::istream is;
        is.str( std::string( buf, (size_t)len ) );

        std::streamoff totalsize;
        is.read( (char*)&totalsize, sizeof( totalsize ) );
        assert( totalsize == len );

        uint32_t nframes;
        is.read( (char*)&nframes, sizeof( nframes ) );
        assert( nframes == (uint32_t)nslices );

        std::vector< std::streamoff > offsets;
        offsets.reserve( nframes );
        for( uint32_t frame = 0; frame < nframes; ++frame )
        {
            uint32_t offset;
            is.read( (char*)&offset, sizeof( offset ) );
            offsets.push_back( offset );
        }

        std::vector<char> outbuf;

        const int size[2] = { 608, 427 }; // FIXME: where does it comes from ?
        std::stringstream ss;
        ss << outfile;
        ss << '\n';
        //ss << crchheaders[0].imgsize; // FIXME: Assume all header are identical !
        ss << size[0];
        ss << '\n';
        ss << size[1];
        ss << '\n';
        ss << nframes;
        ss << ".raw";

```

```

std::ofstream os( ss.str().c_str(), std::ios::binary );

assert( buf_size >= size[0] * size[1] );
outbuf.resize( buf_size );

hframe header;
//uint32_t prev = 0;
for( unsigned int r = 0; r < nframes; ++r )
{
    is.read( (char*)&header, sizeof( header ) );

    assert( header == crchheaders[r] );
    assert( header.val1[0] == 2000 );
    assert( header.val1[1] == 3 );
    assert( header.val2[0] == 1 );
    assert( header.val2[1] == 1280 );

    uLongf destLen = buf_size; // >= 608,427
    Bytef *dest = (Bytef*)outbuf.data();
    assert( is.tellg() == offsets[r] + 16 );
    const Bytef *source = (const Bytef*)buf + offsets[r] + 16;
    uLong sourceLen;
    if( r + 1 == nframes )
        sourceLen = (uLong)totalsize - (uLong)offsets[r] - 16;
    else
        sourceLen = (uLong)offsets[r+1] - (uLong)offsets[r] - 16;
    // FIXME: in-memory decompression:
    int ret = uncompress( dest, &destLen, source, sourceLen );
    assert( ret == Z_OK ); (void)ret;
    assert( destLen >= (uLongf)size[0] * size[1] ); // 16bytes padding ?
    assert( header.imgsize == (uint32_t)size[0] * size[1] );
    //os.write( &outbuf[0], outbuf.size() );
    os.write( outbuf.data(), size[0] * size[1] );

    // skip data:
    is.seekg( sourceLen, std::ios::cur );
}
os.close();
assert( is.tellg() == totalsize );

return true;
}

static bool ProcessNone( const char *outfilename, const int nslices, const
int buf_size, const char *buf, const std::streampos len,
const char *crdbuf, const size_t crclen )
{
    std::vector< hframe > crchheaders;
    crchheaders.reserve( nslices );
    {
        std::istringstream is;
        is.str( std::string( crdbuf, crclen ) );
        hframe header;
        for( int r = 0; r < nslices; ++r )
        {
            is.read( (char*)&header, sizeof( header ) );
            #if 0
                std::cout << header.val0
                    << " " << header.val1[0]
                    << " " << header.val1[1]
                    << " " << header.val2[0]
                    << " " << header.val2[1]
                    << " " << header.imgsize << std::endl;
            #endif
            crchheaders.push_back( header );
        }
    }

    std::istringstream is;
    is.str( std::string( buf, (size_t)len ) );

    std::streampos totalsize;
    is.read( (char*)&totalsize, sizeof( totalsize ) );
    assert( totalsize == len );

    uint32_t nframes;
    is.read( (char*)&nframes, sizeof( nframes ) );
    assert( nframes == (uint32_t)nslices );

    std::vector< uint32_t > offsets;
    offsets.reserve( nframes );

```

```

for( uint32_t frame = 0; frame < nframes ; ++frame )
{
    uint32_t offset;
    is.read( (char*)&offset, sizeof( offset ));
    offsets.push_back( offset );
    //std::cout << offset << std::endl;
}

std::vector<char> outbuf;
// No idea how to present the data, I'll just append everything, and present it as 2D
std::stringstream ss;
ss << outfilename;
ss << '_';
ss << crchheaders[0].imgsize; // FIXME: Assume all header are identical !
ss << '_';
ss << nframes;
ss << ".raw";
std::ofstream os( ss.str().c_str(), std::ios::binary );
outbuf.resize( buf_size ); // overallocated + 16
char *buffer = outbuf.data();

hframe header;
for( unsigned int r = 0; r < nframes; ++r )
{
    is.read( (char*)&header, sizeof( header ));
    #if 0
        std::cout << header.val0
            << " " << header.val1[0]
            << " " << header.val1[1]
            << " " << header.val2[0]
            << " " << header.val2[1]
            << " " << header.imgsize << std::endl;
    #endif
    assert( header == crchheaders[r] );

    is.read( buffer, buf_size - 16 );
    os.write( buffer, header.imgsize );
}
assert( is.tellg() == totalsize );
os.close();

return true;
}

#ifdef NDEBUG
static const char * const UDM_USD_DATATYPE_STRINGS[] = {
    "UDM_USD_DATATYPE_DIN_2D_ECHO",
    "UDM_USD_DATATYPE_DIN_2D_ECHO_CONTRAST",
    "UDM_USD_DATATYPE_DIN_DOPPLER_CW",
    "UDM_USD_DATATYPE_DIN_DOPPLER_PW",
    "UDM_USD_DATATYPE_DIN_DOPPLER_PW_TDI",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_FLOW",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_PMI",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_CPA",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_TDI",
    "UDM_USD_DATATYPE_DIN_MMODE_ECHO",
    "UDM_USD_DATATYPE_DIN_MMODE_COLOR",
    "UDM_USD_DATATYPE_DIN_MMODE_COLOR_TDI",
    "UDM_USD_DATATYPE_DIN_PARAM_BLOCK",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_VELOCITY",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_POWER",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_VARIANCE",
    "UDM_USD_DATATYPE_DIN_DOPPLER_AUDIO",
    "UDM_USD_DATATYPE_DIN_DOPPLER_HIGHQ",
    "UDM_USD_DATATYPE_DIN_PHYSIO",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_STRAIN",
    "UDM_USD_DATATYPE_DIN_COMPOSITE_RGB",
    "UDM_USD_DATATYPE_DIN_XFOV_REALTIME_GRAPHICS",
    "UDM_USD_DATATYPE_DIN_XFOV_MOSAIC",
    "UDM_USD_DATATYPE_DIN_COMPOSITE_R",
    "UDM_USD_DATATYPE_DIN_COMPOSITE_G",
    "UDM_USD_DATATYPE_DIN_COMPOSITE_B",
    "UDM_USD_DATATYPE_DIN_MMODE_COLOR_VELOCITY",
    "UDM_USD_DATATYPE_DIN_MMODE_COLOR_POWER",
    "UDM_USD_DATATYPE_DIN_MMODE_COLOR_VARIANCE",
    "UDM_USD_DATATYPE_DIN_2D_ELASTO",
};

static inline bool is_valid( const char * datatype_str )
{
    static const int n = sizeof( UDM_USD_DATATYPE_STRINGS ) / sizeof( *UDM_USD_DATATYPE_STRINGS );

```

```

bool found = false;
if( datatype_str )
{
    for( int i = 0; !found && i < n; ++i )
    {
        found = strcmp( datatype_str, UDM_USD_DATATYPE_STRINGS[i] ) == 0;
    }
}
return found;
}
#endif

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    using namespace gdcm;
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() ) return 1;

    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds1 = file.GetDataSet();

    const PrivateTag tseq1(0x200d,0x3cf8,"Philips US Imaging DD 045");
    if( !ds1.FindDataElement( tseq1 ) ) return 1;
    const DataElement& seq1 = ds1.GetDataElement( tseq1 );

    SmartPointer<SequenceOfItems> sq1 = seq1.GetValueAsSQ();
    assert( sq1->GetNumberOfItems() >= 1 );

    const size_t nitems = sq1->GetNumberOfItems();
    for( size_t item = 1; item < nitems; ++item )
    {
        Item &item1 = sq1->GetItem(item);
        DataSet &ds2 = item1.GetNestedDataSet();

        // (200d,300d) LO 28 UDM_USD_DATATYPE_DIN_2D_ECHO
        const PrivateTag tdatatype(0x200d,0x300d,"Philips US Imaging DD 033");
        if( !ds2.FindDataElement( tdatatype ) ) return 1;
        const DataElement& datatype = ds2.GetDataElement( tdatatype );
        const ByteValue *bvdatatype = datatype.GetByteValue();
        if( !bvdatatype ) return 1;

        const PrivateTag tseq2(0x200d,0x3cf1,"Philips US Imaging DD 045");
        if( !ds2.FindDataElement( tseq2 ) ) return 1;
        const DataElement& seq2 = ds2.GetDataElement( tseq2 );

        SmartPointer<SequenceOfItems> sq2 = seq2.GetValueAsSQ();
        assert( sq2->GetNumberOfItems() >= 1 );

        // FIXME: what if not in first Item ?
        assert( sq2->GetNumberOfItems() == 1 );
        Item &item2 = sq2->GetItem(1);
        DataSet &ds3 = item2.GetNestedDataSet();

        const PrivateTag tzlib(0x200d,0x3cfa,"Philips US Imaging DD 045");
        if( !ds3.FindDataElement( tzlib ) ) return 1;
        const DataElement& zlib = ds3.GetDataElement( tzlib );

        const ByteValue *bv = zlib.GetByteValue();
        if( !bv ) return 1;
        if( bv->GetLength() != 4 ) return 1;

        // (200d,3010) IS 2 88
        const PrivateTag tnslices(0x200d,0x3010,"Philips US Imaging DD 033");
        if( !ds3.FindDataElement( tnslices ) ) return 1;
        const DataElement& nslices = ds3.GetDataElement( tnslices );
        Element<VR::IS,VM::VM1> elnslices;
        elnslices.SetFromDataElement( nslices );
        const int nslicesref = elnslices.GetValue();
        assert( nslicesref >= 0 );
        // (200d,3011) IS 6 259648
        const PrivateTag tzalloc(0x200d,0x3011,"Philips US Imaging DD 033");
        if( !ds3.FindDataElement( tzalloc ) ) return 1;
        const DataElement& zalloc = ds3.GetDataElement( tzalloc );
        Element<VR::IS,VM::VM1> elzalloc;
        elzalloc.SetFromDataElement( zalloc );
        const int zallocref = elzalloc.GetValue();
        assert( zallocref >= 0 );
        // (200d,3021) IS 2 0

```

```

const PrivateTag tzero(0x200d,0x3021,"Philips US Imaging DD 033");
if( !ds3.FindDataElement( tzero ) ) return 1;
const DataElement& zero = ds3.GetDataElement( tzero );
Element<VR::IS,VM::VM1> elzero;
elzero.SetFromDataElement( zero );
const int zerocref = elzero.GetValue();
assert( zerocref == 0 ); (void)zerocref;

// (200d,3cf3) OB
const PrivateTag tdeflate(0x200d,0x3cf3,"Philips US Imaging DD 045");
if( !ds3.FindDataElement( tdeflate ) ) return 1;
const DataElement& deflate = ds3.GetDataElement( tdeflate );
const ByteValue *bv2 = deflate.GetByteValue();

// (200d,3cfb) OB
const PrivateTag tcrc(0x200d,0x3cfb,"Philips US Imaging DD 045");
if( !ds3.FindDataElement( tcrc ) ) return 1;
const DataElement& crc = ds3.GetDataElement( tcrc );
const ByteValue *bv3 = crc.GetByteValue();

std::string outfile = std::string( bvdatatype->GetPointer(), bvdatatype->GetLength() );
outfile = LOComp::Trim( outfile.c_str() );
const char *outfilename = outfile.c_str();
assert( is_valid(outfilename) );
if( bv2 )
{
    assert( bv3 );
    assert( zallocref > 0 );
    assert( nslicesref > 0 );
    std::cout << ds2 << std::endl;

    if( strcmp(bv->GetPointer(), "ZLib", 4) == 0 )
    {
        if( !ProcessDeflate( outfile, nslicesref, zallocref, bv2->GetPointer(),
            std::streampos(bv2->GetLength()), bv3->GetPointer(), bv3->GetLength() ) )
        {
            return 1;
        }
    }
    else if( strcmp(bv->GetPointer(), "None", 4) == 0 )
    {
        if( !ProcessNone( outfile, nslicesref, zallocref, bv2->GetPointer(),
            std::streampos(bv2->GetLength()), bv3->GetPointer(), bv3->GetLength() ) )
        {
            return 1;
        }
    }
    else
    {
        std::string str( bv->GetPointer(), bv->GetLength() );
        std::cerr << "Unhandled: " << str << std::endl;
        return 1;
    }
}
}

return 0;
}

```

## 12.54 DumpSiemensBase64.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * https://groups.google.com/forum/#!msg/comp.protocols.dicom/2kZ21LP8EcM/WzjFrtjnAgAJ
 */

```



```

#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmPrinter.h"
#include "gdcmDictPrinter.h"
#include "gdcmCSAHeader.h"
#include "gdcmBase64.h"
#include "gdcmExplicitDataElement.h"
#include "gdcmSwapper.h"

#include <iostream>
#include <fstream>
#include <vector>

#include <assert.h>

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }
    const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

    gdcm::CSAHeader csa;
    const gdcm::PrivateTag &t1 = csa.GetCSAImageHeaderInfoTag();
    if( !ds.FindDataElement( t1 ) ) return 1;
    csa.LoadFromDataElement( ds.GetDataElement( t1 ) );

    //const char name[] = "MRDiffusion";
    const char name[] = "MR_ASL";
    if( csa.FindCSAElementByName( name ) )
    {
        const gdcm::CSAElement &el = csa.GetCSAElementByName( name );
        const gdcm::ByteValue* bv = el.GetByteValue();
        std::string str( bv->GetPointer(), bv->GetLength() );
        str.erase( std::remove( str.begin(), str.end(), '\n' ), str.end() );
        size_t dl = gdcm::Base64::GetDecodeLength( str.c_str(), str.size() );
        std::vector<char> buf;
        buf.resize( dl );
        size_t dl2 = gdcm::Base64::Decode( buf.data(), buf.size(), str.c_str(), str.size() );
        (void)dl2;
        std::stringstream ss;
        ss.str( std::string( buf.data(), buf.size() ) );
        gdcm::File file;
        gdcm::DataSet &ds2 = file.GetDataSet();
        gdcm::DataElement xde;
        try
        {
            while( xde.Read<gdcm::ExplicitDataElement, gdcm::SwapperNoOp>( ss ) )
            {
                ds2.Insert( xde );
            }
            assert( ss.eof() );
        }
        catch( std::exception & )
        {
            return 1;
        }
        gdcm::Printer p;
        p.SetFile( file );
        p.Print( std::cout );
    }

    return 0;
}

```

## 12.55 DumpToSQLITE3.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

```

Copyright (c) 2006-2011 Mathieu Malaterre  
 All rights reserved.  
 See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even  
 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR  
 PURPOSE. See the above copyright notice for more information.

```

=====*/
/*
 * Ref:
 * http://massmail.spl.harvard.edu/public-archives/slicer-devel/2010/004408.html
 *
 * Implementation details:
 * http://www.sqlite.org/c3ref/bind_blob.html
 * http://www.adp-gmbh.ch/sqlite/bind_insert.html
 */
#include "gdcmScanner.h"
#include "gdcmDirectory.h"
#include "gdcmTag.h"
#include "gdcmTrace.h"

#include "sqlite3.h"

#include <stdio.h>
#include <time.h>

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        return 1;
    }
    time_t time_start = time(nullptr);

    gdcm::Trace::SetDebug( false );
    gdcm::Trace::SetWarning( false );
    const char *inputdirectory = argv[1];

    gdcm::Directory d;
    unsigned int nfiles = d.Load( inputdirectory, true);

    gdcm::Scanner s;
    using gdcm::Tag;
    s.AddTag( Tag(0x20,0xd) ); // Study Instance UID
    s.AddTag( Tag(0x20,0xe) ); // Series Instance UID

    bool b0 = s.Scan( d.GetFilesNames() );
    if( !b0 ) return 1;
    time_t time_scanner = time(nullptr);

    std::cout << "Finished loading data from : " << nfiles << " files" << std::endl;
    // MappingType const &mappings = s.GetMappings();

    sqlite3* db;
    sqlite3_open("./dicom.db", &db);

    if(db == nullptr)
    {
        std::cerr << "Could not open database." << std::endl;
        return 1;
    }

    const char sql_stmt[] = "create table browser (seriesuid, studyuid)";
    int ret;

    char *errmsg;
    ret = sqlite3_exec(db, sql_stmt, nullptr, nullptr, &errmsg);

    if(ret != SQLITE_OK)
    {
        printf("Error in statement: %s [%s].\n", sql_stmt, errmsg);
        return 1;
    }
    using gdcm::Directory;
    using gdcm::Scanner;
    const Directory::FileNamesType& files = d.GetFilesNames();

```

```

Directory::FileNamesType::const_iterator file = files.begin();

sqlite3_stmt *stmt;
if ( sqlite3_prepare(
    db,
    "insert into browser values (?,?)", // stmt
    -1, // If than zero, then stmt is read up to the first nul terminator
    &stmt,
    nullptr // Pointer to unused portion of stmt
)
    != SQLITE_OK)
{
    printf("\nCould not prepare statement.");
    return 1;
}
//printf("\nThe statement has %d wildcards\n", sqlite3_bind_parameter_count(stmt));
for(; file != files.end(); ++file)
{
    const char *filename = file->c_str();
    bool b = s.IsKey(filename);
    if( b )
    {
        const Scanner::TagToValue &mapping = s.GetMapping(filename);
        Scanner::TagToValue::const_iterator it = mapping.begin();

        sqlite3_reset(stmt);

        for( int index = 1; it != mapping.end(); ++it, ++index)
        {
            //const Tag & tag = it->first;
            const char *value = it->second;

            if (sqlite3_bind_text (
                stmt,
                index, // Index of wildcard
                value,
                (int)strlen(value), // length of text
                SQLITE_STATIC // SQLite assumes that the information is in static
            )
                != SQLITE_OK)
            {
                printf("\nCould not bind int.\n");
                return 1;
            }
        }
        if (sqlite3_step(stmt) != SQLITE_DONE)
        {
            printf("\nCould not step (execute) stmt.\n");
            return 1;
        }
    }
}

sqlite3_close(db);

time_t time_sqlite = time(nullptr);

std::cout << "Time to scan DICOM files: " << (time_scanner - time_start) << std::endl;
std::cout << "Time to build SQLITE3: " << (time_sqlite - time_scanner) << std::endl;

return 0;
}

```

## 12.56 DumpToshibaDTI.cxx

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

```

```

=====*/
/*
 * https://groups.google.com/d/msg/comp.protocols.dicom/7IaIkT0ZG5U/k7LPu81VvAMJ
 */
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmPrinter.h"
#include "gdcmDictPrinter.h"

#include <iostream>
#include <fstream>
#include <vector>

#include <assert.h>

static bool DumpToshibaDTI( const char * input, size_t len )
{
    static int i = 0;
    ++i;
    if( len % 2 ) return false;

    std::vector<char> copy( input, input + len );
    std::reverse( copy.begin(), copy.end() );

    if 0
    {
        std::ostringstream f;
        f << "debug" << i;
        std::ofstream of( f.str().c_str(), std::ios::binary );
        of.write( &copy[0], copy.size() );
        of.close();
    }
    else
    {
        std::istringstream is;
        std::string dup( copy.data(), copy.size() );
        is.str( dup );

        gdcm::File file;
        gdcm::FileMetaInformation & fmi = file.GetHeader();
        fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );
        gdcm::DataSet & ds = file.GetDataSet();
        ds.Read<gdcm::ExplicitDataElement, gdcm::SwapperNoOp>( is );

        //gdcm::DictPrinter p;
        gdcm::Printer p;
        p.SetFile( file );
        p.SetColor( true );
        p.Print( std::cout );
    }
    #endif

    return true;
}

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }
    const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

    // (0029,0010) ?? (LO) [PMTF INFORMATION DATA ]           # 22,1 Private Creator
    // (0029,1001) ?? (SQ) (Sequence with undefined length)    # u/1,1 ?

    const gdcm::PrivateTag tpmtf(0x0029,0x1,"PMTF INFORMATION DATA");
    if( !ds.FindDataElement( tpmtf ) ) return 1;
    const gdcm::DataElement& pmtf = ds.GetDataElement( tpmtf );
    if ( pmtf.IsEmpty() ) return 1;
    gdcm::SmartPointer<gdcm::SequenceOfItems> seq = pmtf.GetValueAsSQ();
    if ( !seq || !seq->GetNumberOfItems() ) return 1;

    size_t n = seq->GetNumberOfItems();
    for( size_t i = 1; i <= n; ++i )
    {
        gdcm::Item &item = seq->GetItem(i);
        gdcm::DataSet &subds = item.GetNestedDataSet();
        // (0029,0010) ?? (LO) [PMTF INFORMATION DATA ]           # 22,1 Private Creator
    }
}

```

```

// (0029,1090) ?? (OB) 00\05\00\13\00\12\00\22\ # 202,1 ?
const gdcm::PrivateTag tseq(0x0029,0x90,"PMTF INFORMATION DATA");

if( subds.FindDataElement( tseq ) )
{
    const gdcm::DataElement & de = subds.GetDataElement( tseq );
    const gdcm::ByteValue * bv = de.GetByteValue();
    if( !bv ) return 1;

    bool b = DumpToshibaDTI( bv->GetPointer(), bv->GetLength() );
    if( !b ) return 1;
}

}

return 0;
}

```

## 12.57 DumpToshibaDTI2.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * https://gazelle.ihe.net/EVSCClient/dicomResult.seam;jsessionid=x+Rf9Zs+ip49P+jC3L8SLZb8?&oid=1.3.6.1.4.1.12559.11.1.2.1.4.162
 */
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmPrinter.h"
#include "gdcmDictPrinter.h"

#include <iostream>
#include <fstream>
#include <vector>

#include <assert.h>

static bool DumpToshibaDTI2( const char * input, size_t len )
{
    static int i = 0;
    ++i;
    if( len % 2 ) return false;

    std::vector<char> copy( input, input + len );
    std::reverse( copy.begin(), copy.end() );

    if 0
    {
        std::ostringstream f;
        f << "debug" << i;
        std::ofstream of( f.str().c_str(), std::ios::binary );
        of.write( &copy[0], copy.size() );
        of.close();
    }
    else
    {
        std::istringstream is;
        std::string dup( copy.data(), copy.size() );
        is.str( dup );

        gdcm::File file;
        gdcm::FileMetaInformation & fmi = file.GetHeader();
        fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );
        gdcm::DataSet & ds = file.GetDataSet();
        ds.Read<gdcm::ExplicitDataElement, gdcm::SwapperNoOp>( is );

        //gdcm::DictPrinter p;
        gdcm::Printer p;
    }
}

```

```

    p.SetFile( file );
    p.SetColor( true );
    p.Print( std::cout );
#endif

    return true;
}

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }
    const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

    /*
    (0029,1001) SQ (Sequence with explicit length #=6)          # 18746, 1 Unknown Tag & Data
      (fffe,e000) na (Item with explicit length #=2)           # 206, 1 Item
        (0029,0010) LO [TOSHIBA_MEC_MR3]                       # 16, 1 PrivateCreator
        (0029,1090) OB 00\07\00\06\00\05\00\04\00\03\00\02\00\0c\00\01\00\00\00\00\00\12... # 170, 1 Unknown Tag &
        Data
      (fffe,e00d) na (ItemDelimitationItem for re-encoding)    # 0, 0 ItemDelimitationItem
      (fffe,e000) na (Item with explicit length #=2)           # 866, 1 Item
        (0029,0010) LO [TOSHIBA_MEC_MR3]                       # 16, 1 PrivateCreator
        (0029,1090) OB 45\4e\49\50\53\4c\20\52\41\5c\45\4e\49\50\53\4c\54\5c\52\45\53\55... # 830, 1 Unknown Tag &
        Data
    [...]
    (0029,1002) SQ (Sequence with explicit length #=1)        # 120, 1 Unknown Tag & Data
      (fffe,e000) na (Item with explicit length #=2)           # 112, 1 Item
        (0029,0010) LO [TOSHIBA_MEC_MR3]                       # 16, 1 PrivateCreator
        (0029,1090) OB 00\10\00\02\53\55\10\80\70\0d\30\31\5e\33\52\4d\5f\43\45\4d\5f\41... # 76, 1 Unknown Tag &
        Data
      (fffe,e00d) na (ItemDelimitationItem for re-encoding)    # 0, 0 ItemDelimitationItem
    */

    const gdcm::PrivateTag tmecmr3(0x0029,0x1,"TOSHIBA_MEC_MR3");
    if( !ds.FindDataElement( tmecmr3 ) ) return 1;
    const gdcm::DataElement& mecmr3 = ds.GetDataElement( tmecmr3 );
    if ( mecmr3.IsEmpty() ) return 1;
    gdcm::SmartPointer<gdcm::SequenceOfItems> seq = mecmr3.GetValueAsSQ();
    if ( !seq || !seq->GetNumberOfItems() ) return 1;

    size_t n = seq->GetNumberOfItems();
    for( size_t i = 1; i <= n; ++i )
    {
        gdcm::Item &item = seq->GetItem(i);
        gdcm::DataSet &subds = item.GetNestedDataSet();
        const gdcm::PrivateTag tseq(0x0029,0x90,"TOSHIBA_MEC_MR3");

        if( subds.FindDataElement( tseq ) )
        {
            const gdcm::DataElement &de = subds.GetDataElement( tseq );
            const gdcm::ByteValue *bv = de.GetByteValue();
            if( !bv ) return 1;

            bool b = DumpToshibaDTI2( bv->GetPointer(), bv->GetLength() );
            if( !b ) return 1;
        }
    }

    return 0;
}

```

## 12.58 DumpVisusChange.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre

```

All rights reserved.  
See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even  
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR  
PURPOSE. See the above copyright notice for more information.

```

=====*/
#include "gdcmReader.h"
#include "gdcmDirectory.h"
#include "gdcmStringFilter.h"

#include <vector>
#include <algorithm>

/*
*/
static bool process( std::vector<gdcm::DataElement> & ms, const char * filename)
{
    using namespace gdcm;
    Tag pd(0x7fe0,0x0000);
    std::set<gdcm::Tag> skiptags;
    skiptags.insert( pd );

    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.ReadUpToTag( pd, skiptags ) )
    {
        std::cerr << "Failure to read: " << filename << std::endl;
        return false;
    }

    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds1 = file.GetDataSet();

    const gdcm::PrivateTag tseq1(0x5533,0x33,"Visus Change");
    if( !ds1.FindDataElement( tseq1 ) ) return true;
    const gdcm::DataElement& seq1 = ds1.GetDataElement( tseq1 );

    SmartPointer<SequenceOfItems> sq1 = seq1.GetValueAsSQ();

    const size_t nitems = sq1->GetNumberOfItems();
    for( size_t item = 1; item < nitems; ++item )
    {
        Item &item1 = sq1->GetItem(item);
        DataSet &ds2 = item1.GetNestedDataSet();
        for(DataSet::ConstIterator it = ds2.Begin(); it != ds2.End(); ++it )
        {
            DataElement const &de = *it;
            // cannot simply use std::set here, see there is a discrepancy in between
            // operator== and operator<.
            // So only use operator== here:
            std::vector<DataElement>::iterator vit = std::find(ms.begin(), ms.end(), de);
            if( vit == ms.end() )
                ms.push_back(de);
        }
    }
    return true;
}

int main(int argc, char *argv[])
{
    bool usefastpath = true;

    if( argc < 2 ) return 1;
    using namespace gdcm;
    const char *filename = argv[1];
    gdcm::Directory::FileNamesType filenames;
    if( !gdcm::System::FileExists(filename) )
    {
        std::cerr << "Could not find file: " << filename << std::endl;
        return 1;
    }

    gdcm::Directory dir;
    if( gdcm::System::FileIsDirectory(filename) )
    {
        unsigned int nfiles = dir.Load(filename, false);
        if( nfiles == 0 )
        {
            std::cerr << "Could not find files: " << filename << std::endl;
        }
    }
}

```

```

        return 1;
    }
    filenames = dir.GetFilesNames();
}
else
{
    filenames.push_back( filename );
}
gdcm::StringFilter sf;

Tag pd(0x7fe0,0x0000);
std::set<gdcm::Tag> skiptags;
skiptags.insert( pd );

gdcm::Reader reader;
reader.SetFileName( filenames[0].c_str() );
if( !reader.ReadUpToTag( pd, skiptags ) )
{
    std::cerr << "Could not read file: " << filename << std::endl;
    return 1;
}
gdcm::File &file = reader.GetFile();
sf.SetFile(file);

if( usefastpath ) {
    // Heuristic, assume if private tag cannot be found in first file, skip the directory
    gdcm::DataSet &ds1 = file.GetDataSet();

    const gdcm::PrivateTag tseq1(0x5533,0x33,"Visus Change");
    if( !ds1.FindDataElement( tseq1 ) ){
        std::cerr << "Could not find private tag in first file skipping whole directory: " << filename << std::endl;
        return 0;
    }
}

std::vector<DataElement> ms;
for(gdcm::Directory::FilenamesType::const_iterator cit = filenames.begin(); cit != filenames.end(); ++cit )
{
    if( !process(ms, cit->c_str()) ) {
        return 1;
    }
}

if( !ms.empty() ) {
    std::sort(ms.begin(), ms.end());
    std::cout << filename << ",\n";
    for(std::vector<DataElement>::const_iterator it = ms.begin(); it != ms.end(); ++it )
    {
        DataElement const & de = *it;
        std::string const & s = sf.ToString( de );
        std::cout << de.GetTag() << " " << s << std::endl;
    }
    std::cout << "\n" << std::endl;
}

return 0;
}

```

## 12.59 DuplicatePCDE.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"

```



```
#include "gdcmFile.h"
#include "gdcmTag.h"
/*
  Usage:
  DuplicatePCDE gdcmData/D_CLUNIE_CT1_J2KI.dcm out.dcm

aka:
medical.nema.org/medical/dicom/DataSets/WG04/IMAGES/J2KI/CT1_J2KI

See:
gdcmConformanceTests/CT1_J2KI_DuplicatePCDE.dcm

Original thread can be found at:

http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/82f28c4db28963af
```

Question:

1.  
There is no restriction for a specific Private Creator Data Element (PCDE) to be unique within the same group, right ?  
Decoders of Private Data would have to handle the case where a PCDE would be repeated and should NOT stop on the first instance of a particular PCDE, right ?

Eg. when searching for the tag associated with (0x0029,0x0010,"SIEMENS CSA HEADER") in the following (pseudo) dataset:

```
(0029,0010) LO [SIEMENS CSA HEADER] # 18, 1
PrivateCreator
(0029,0011) LO [SIEMENS MEDCOM HEADER] # 22, 1
PrivateCreator
(0029,0012) LO [SIEMENS MEDCOM HEADER2] # 22, 1
PrivateCreator
(0029,0013) LO [SIEMENS CSA HEADER] # 18, 1
PrivateCreator
(0029,1008) CS [IMAGE NUM 4] # 12, 1
CSAImageHeaderType
(0029,1009) LO [20050723] # 8, 1
CSAImageHeaderVersion
(0029,1010) OB 53\56\31\30\04\03\02\01\38\00\00\00\4d
\00\00\00\45\63\68\6f\4c\69... # 6788, 1 CSAImageHeaderInfo
(0029,1018) CS [MR] # 2, 1
CSASeriesHeaderType
(0029,1019) LO [20050723] # 8, 1
CSASeriesHeaderVersion
(0029,1020) OB 53\56\31\30\04\03\02\01\2c\00\00\00\4d
\00\00\00\55\73\65\64\50\61... # 51520, 1 CSASeriesHeaderInfo
(0029,1131) LO [4.0.163088300] # 14, 1
PMTFInformation1
(0029,1132) UL 32768 # 4, 1
PMTFInformation2
(0029,1133) UL 0 # 4, 1
PMTFInformation3
(0029,1134) CS [DB TO DICOM] # 12, 1
PMTFInformation4
(0029,1260) ?? 63\6f\6d\20 # 4, 1
Unknown Tag & Data
(0029,1310) OB 53\56\31\30\04\03\02\01\38\00\00\00\4d
\00\00\00\45\63\68\6f\4c\69... # 6788, 1 CSAImageHeaderInfo
```

one should return two instances, correct ?

Answer:

I would say that this is covered in principle by the PS 3.5 7.1 "The Data Elements ... shall occur at most once in a Data Set" rule, since the data element is defined by the tuple (private creator,gggg,ee) where xxee is the element number and xx is arbitrary and has no inherent meaning and does not serve to disambiguate the data element.

E.g.:

```
(0019,0030) Private Creator ID = "Smith"
...
(0019,0032) Private Creator ID = "Smith"
...
(0019,3015) Fractal Index = "32"
...
(0019,3215) Fractal Index = "32"
```

would be illegal because even though they are assigned different (completely arbitrary) blocks, with the same group, element number and private creator, (0019,3015) and (0019,3215) are the "same" data element.

```

*/

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();

    // Let's get all private element from group 0x9:
    /*
(0009,0010) LO [GEMS_IDEN_01]                # 12,1 Private Creator
(0009,1001) LO [GE_GENESIS_FF ]              # 14,1 Full fidelity
(0009,1002) SH [CT01]                        # 4,1 Suite id
(0009,1004) SH [HiSpeed CT/i]                # 12,1 Product id
(0009,1027) SL 862399669                     # 4,1 Image actual date
(0009,1030) SH (no value)                    # 0,1 Service id
(0009,1031) SH (no value)                    # 0,1 Mobile location number
(0009,10e6) SH [05]                          # 2,1 Genesis Version - now
(0009,10e7) UL 973283917                     # 4,1 Exam Record checksum
(0009,10e9) SL 862399669                     # 4,1 Actual series data time stamp
*/
    gdcm::Tag start(0x0009,0x0);
    // Create a temporary duplicate dataset, since we cannot insert data element as we go over them (std::set
    // would reorganize itself as we go over it ...)
    gdcm::DataSet dup;
    gdcm::Tag new_private(0x0009,0x0);
    while (start.GetGroup() == 0x9 )
    {
        const gdcm::DataElement& de = ds.FindNextDataElement(start);
        const gdcm::Tag &t = de.GetTag();
        if( t.IsPrivateCreator() )
        {
            std::cout << t << std::endl;
            // Ok let's duplicate into the next available attribute:
            gdcm::DataElement duplicate = de;
            duplicate.GetTag().SetElement( (uint16_t)(t.GetElement() + 1) );
            dup.Insert( duplicate );
            new_private = duplicate.GetTag();
        }
        else if( t.IsPrivate() && !t.IsPrivateCreator() )
        {
            //std::cout << de << std::endl;
            std::string owner = ds.GetPrivateCreator( de.GetTag() );
            //std::cout << owner << std::endl;
            gdcm::DataElement duplicate = de;
            duplicate.GetTag().SetPrivateCreator( new_private );
            if( const gdcm::ByteValue *bv = duplicate.GetByteValue() )
            {
                // Warning: when doing : duplicate = de, only the pointer to the ByteValue is passed
                // (to avoid large memory duplicate). We need to explicitly duplicate the bytevalue ourselves:
                gdcm::ByteValue *dupbv = new gdcm::ByteValue( bv->GetPointer(),
                    bv->GetLength() );
                // Let's recognize the duplicated ASCII-type elements:
                if( duplicate.GetVR() & gdcm::VR::VRASCII )
                {
                    dupbv->Fill( 'X' );
                    duplicate.SetValue( *dupbv );
                }
                dup.Insert( duplicate );
            }
            start = t;
            // move to next possible 'public' element
            start.SetElement( (uint16_t)(start.GetElement() + 1) );
        }
    }
}

```

```

    }

    gdcmm::DataSet::ConstIterator it = dup.Begin();
    for( ; it != dup.End(); ++it )
    {
        ds.Insert( *it );
    }

    gdcmm::Writer w;
    w.SetFile( file );
    w.SetFileName( outfilename );
    if (!w.Write() )
    {
        return 1;
    }

    return 0;
}

```

## 12.60 ELSCINT1WaveToText.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"

/*
 * This example shows how to read a Wave Information tag from ELSCINT1
 * The wave information is stored in Tag (01e1,18,ELSCINT1) hidden in a
 * Secondary Capture Image Storage (usually a 'N' Symbol is shown)
 *
 * Everything done in this code is for the sole purpose of writing interoperable
 * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
 * If you believe anything in this code violates any law or any of your rights,
 * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
 * find a solution.
 *
 * Everything you do with this code is at your own risk, since decompression
 * algorithm was not written from specification documents.
 *
 * Special thanks to:
 * Gauthier Bouilhol
 */

template <typename T>
bool dumpargs(std::ostream & os, T c1, T c2, T c3, T c4, T c5, T c6, T c7, T c8)
{
    static const char sep = '\\t';
    os << c1 << sep << c2 << sep << c3 << sep << c4 << sep << c5 << sep << c6 << sep << c7 << sep << c8;
    os << std::endl;
    return true;
}

bool wave2stream( std::ostream &text_file, const char *in, size_t len )
{
    const short * buffer = (const short*)in;
    size_t length = len / sizeof( short );
    text_file << "COMPLETE_WAVE" << '\\t' << "MASK" << '\\t' << "AQUISITION_PROFIL" << '\\t' << "END-INHALE" << '\\t' <<
        "END-EXHALE" << '\\t' << "AQUISITION_WAVE" << '\\t' << "WAVE_STATISTICS" << '\\t' << "MASK" << std::endl;
    for (size_t i=0;i<length-76;i+=2)
    {
        if ( i < 74 )
        {
            if (buffer[i+75] == 0)
                text_file << buffer[i+74] << '\\t' << buffer[i+75] << '\\t' << 0 << '\\t' << " " <<
                '\\t' << " " << '\\t' << " " << '\\t' << buffer[i] << '\\t' << buffer[i+1] <<
                std::endl;

```

```

        if (buffer[i+75] == 16384)
            text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0
            '\t' << " " << '\t' << " " << '\t' << buffer[i] << '\t' << buffer[i+1] <<
            std::endl;
        if (buffer[i+75] == 256)
            text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0
            '\t' << buffer[i+74] << '\t' << " " << '\t' << buffer[i] << '\t' << buffer[i+1] <<
            std::endl;
        if (buffer[i+75] == -32768)
            text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1
            '\t' << " " << '\t' << buffer[i+74] << '\t' << buffer[i] << '\t' << buffer[i+1] <<
            std::endl;
        if (buffer[i+75] == -16384)
            text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1
            '\t' << " " << '\t' << buffer[i+74] << '\t' << buffer[i] << '\t' << buffer[i+1] <<
            std::endl;
        if (buffer[i+75] == -32512)
            text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1
            '\t' << buffer[i+74] << '\t' << buffer[i+74] << '\t' << buffer[i] << '\t' << buffer[i+1] <<
            std::endl;
    }
    else
    {
        if (buffer[i+75] == 0)
            text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0
            '\t' << " " << '\t' << " " << '\t' << " " << '\t' << " " <<
            std::endl;
        if (buffer[i+75] == 16384)
            text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0
            '\t' << " " << '\t' << " " << '\t' << " " << '\t' << " " <<
            std::endl;
        if (buffer[i+75] == 256)
            text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0
            '\t' << buffer[i+74] << '\t' << " " << '\t' << " " << '\t' << " " <<
            std::endl;
        if (buffer[i+75] == -32768)
            text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1
            '\t' << " " << '\t' << buffer[i+74] << '\t' << " " << '\t' << " " <<
            std::endl;
        if (buffer[i+75] == -16384)
            text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1
            '\t' << " " << '\t' << buffer[i+74] << '\t' << " " << '\t' << " " <<
            std::endl;
        if (buffer[i+75] == -32512)
            text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1
            '\t' << buffer[i+74] << '\t' << buffer[i+74] << '\t' << " " << '\t' << " " <<
            std::endl;
    }
}

return true;
}

int main(int argc, char *argv [])
{
    if( argc < 3 ) return 1;
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }
    const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

    const gdcm::PrivateTag twave(0x01e1,0x18,"ELSCINT1");
    if( !ds.FindDataElement( twave ) ) return 1;
    const gdcm::DataElement& wave = ds.GetDataElement( twave );
    if ( wave.IsEmpty() ) return 1;
    const gdcm::ByteValue *bv = wave.GetByteValue();
    assert( bv );

    std::ofstream os( outfile, std::ios::binary );
    // Dump that to a CSV file:
    wave2stream( os, bv->GetPointer(), bv->GetLength() );
    os.close();

    return 0;
}

```

## 12.61 EmptyMask.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
#include "gdcmEmptyMaskGenerator.h"

#include <string>
#include <cstring>

int main( int argc, char *argv[] )
{
    std::string inputdir;
    std::string outputdir;
    bool input_sopclassuid = true;
    bool grayscale_secondary_sopclassuid = false;
    if( argc < 3 ) return 1;
    inputdir = argv[1];
    outputdir = argv[2];
    // input_sopclassuid -> Use original SOP Class UID from input DICOM (Default).
    // grayscale_secondary_sopclassuid -> Use Grayscale Secondary Image Storage SOP Class UID.
    if( argc >= 3 )
    {
        input_sopclassuid = false;
        if( strcmp("input_sopclassuid", argv[3]) == 0 )
            input_sopclassuid = true;
        else if (strcmp("grayscale_secondary_sopclassuid", argv[3]) == 0 ) {
            grayscale_secondary_sopclassuid = true;
        }
    }

    //
    gdcm::EmptyMaskGenerator emg;
    if( input_sopclassuid )
        emg.SetSOPClassUIDMode( gdcm::EmptyMaskGenerator::UseOriginalSOPClassUID );
    else if( grayscale_secondary_sopclassuid )
        emg.SetSOPClassUIDMode( gdcm::EmptyMaskGenerator::UseGrayscaleSecondaryImageStorage );
    emg.SetInputDirectory( inputdir.c_str() );
    emg.SetOutputDirectory( outputdir.c_str() );
    if( !emg.Execute() )
    {
        return 1;
    }

    return 0;
}

```

## 12.62 EncapsulateFileInRawData.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
#include "gdcmAnonymizer.h"
#include "gdcmWriter.h"
#include "gdcmUIDGenerator.h"
#include "gdcmFile.h"
#include "gdcmTag.h"

```

```

#include "gdcmSystem.h"

#include "magic.h" // libmagic, API to file command line tool

/*
 * Let say you want to encapsulate a file type that is not defined in DICOM (exe, zip, png)
 * PNG is a bad example, unless it contains transparency (which has been deprecated).
 * It will take care of dispatching each chunk to an appropriate data item (pretty much like
 * WaveformData)
 *
 * Usage:
 * ./EncapsulateFileInRawData large_input_file.exe large_input_file.dcm
 */

// TODO:
// $ file -bi /tmp/gdcm-2.1.0.pdf
int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " inputfile output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    if( !gdcm::System::FileExists( filename ) ) return 1;

    size_t s = gdcm::System::FileSize(filename);
    if( !s ) return 1;

    magic_t cookie = magic_open(MAGIC_NONE);
    const char * file_type = magic_file(cookie, filename);
    if( !file_type ) return 1;
    magic_close(cookie);

    gdcm::Writer w;
    gdcm::File &file = w.GetFile();
    //gdcm::DataSet &ds = file.GetDataSet();
    //w.SetCheckFileMetaInformation( true );
    w.SetFileName( outfile );

    file.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::ImplicitVRLittleEndian );

    gdcm::Anonymizer anon;
    anon.SetFile( file );

    gdcm::MediaStorage ms = gdcm::MediaStorage::RawDataStorage;

    gdcm::UIDGenerator gen;
    anon.Replace( gdcm::Tag(0x0008,0x16), ms.GetString() );
    std::cout << ms.GetString() << std::endl;
    anon.Replace( gdcm::Tag(0x0008,0x18), gen.Generate() );

    if ( !w.Write() )
    {
        std::cerr << "Could not write: " << outfile << std::endl;
        return 1;
    }

    return 0;
}

```

## 12.63 ExtractEncryptedContent.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====

```

```

=====*/
#include "gdcMReader.h"

#include <fstream>

/*

openssl smime -encrypt -binary -aes256 -in outputfile.dcm -inform DER -out outputfile.der -outform DER
../trunk/Testing/Source/Data/certificate.pem

openssl smime -decrypt -binary -in out.der -inform DER -out outputfile.dcm -outform DER -inkey
../trunk/Testing/Source/Data/privatekey.pem ../trunk/Testing/Source/Data/certificate.pem

*/

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.der" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    gdcM::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcM::File &file = reader.GetFile();
    gdcM::DataSet &ds = file.GetDataSet();

    const gdcM::DataElement &EncryptedAttributesSequence = ds.GetDataElement( gdcM::Tag( 0x0400,0x0500 ) );

    gdcM::SequenceOfItems *sqi = EncryptedAttributesSequence.GetValueAsSQ();

    if ( !sqi || sqi->GetNumberOfItems() != 1 ) return 1;

    gdcM::Item &item = sqi->GetItem(1);

    gdcM::DataSet &nestedds = item.GetNestedDataSet();

    if( ! nestedds.FindDataElement( gdcM::Tag( 0x0400,0x0520 ) ) ) return 1;

    const gdcM::DataElement &EncryptedContent = nestedds.GetDataElement( gdcM::Tag( 0x0400,0x0520 ) );

    const gdcM::ByteValue *bv = EncryptedContent.GetByteValue();

    std::ofstream of( outfile, std::ios::binary );
    of.write( bv->GetPointer(), bv->GetLength() );
    of.close();

    return 0;
}

```

## 12.64 ExtractIconFromFile.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcM.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
* This example shows how to either retrieve an Icon if present somewhere
* in the file, or else generate one.

```

```

*/
#include "gdcmImageReader.h"
#include "gdcmPNMCodec.h"
#include "gdcmIconImageFilter.h"
#include "gdcmIconImageGenerator.h"

bool WriteIconAsPNM(const char* filename, const gdcm::IconImage& icon)
{
    gdcm::PNMCodec pnm;
    pnm.SetDimensions( icon.GetDimensions() );
    pnm.SetPixelFormat( icon.GetPixelFormat() );
    pnm.SetPhotometricInterpretation( icon.GetPhotometricInterpretation() );
    pnm.SetLUT( icon.GetLUT() );
    const gdcm::DataElement& in = icon.GetDataElement();
    bool b = pnm.Write( filename, in );
    assert( b );
    return b;
}

int main(int argc, char *argv [])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcm::ImageReader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read (or not image): " << filename << std::endl;
        return 1;
    }

    gdcm::IconImageFilter iif;
    iif.SetFile( reader.GetFile() );
    bool b = iif.Extract();

    if( b )
    {
        const gdcm::IconImage &icon = iif.GetIconImage(0);
        icon.Print( std::cout );

        if( !icon.GetTransferSyntax().IsEncapsulated() )
        {
            // Let's write out this icon as PNM file
            WriteIconAsPNM("icon.ppm", icon);
        }
        else if( icon.GetTransferSyntax() == gdcm::TransferSyntax::JPEGBaselineProcess1
            || icon.GetTransferSyntax() == gdcm::TransferSyntax::JPEGExtendedProcess2_4
        )
        {
            const gdcm::DataElement& in = icon.GetDataElement();
            const gdcm::ByteValue *bv = in.GetByteValue();
            assert( bv );
            std::ofstream out( "icon.jpg", std::ios::binary );
            out.write( bv->GetPointer(), bv->GetLength() );
            out.close();
        }
    }
    else
    {
        assert( iif.GetNumberOfIconImages() == 0 );
        std::cerr << "No Icon Found anywhere in file" << std::endl;

        const gdcm::Image &img = reader.GetImage();
        gdcm::IconImageGenerator iig;
        iig.AutoPixelMinMax(true);
        iig.SetPixmap( img );
        const unsigned int idims[2] = { 64, 64 };
        iig.SetOutputDimensions( idims );
        //iig.SetPixelMinMax(60, 868);
        if( !iig.Generate() ) return 1;
        const gdcm::IconImage &icon = iig.GetIconImage();
        WriteIconAsPNM("icon.ppm", icon);
    }

    return 0;
}

```



## 12.65 Extracting\_All\_Resolution.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
// This work was realised during the GSOC 2011 by Manoj Alwani

#include <fstream>
#include <stdint.h>
#include <string.h>
#include <assert.h>
#include <gdcm_j2k.h>
#include <gdcm_jp2.h>
#include <iostream>
#include <cstring>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmSystem.h"
#include <fstream>

#include "gdcm_openjpeg.h"
#include "gdcmMediaStorage.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmAttribute.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDGenerator.h"
#include "gdcmAnonymizer.h"
#include "gdcmStreamImageWriter.h"
#include "gdcmImageHelper.h"
#include "gdcmTrace.h"

void error_callback(const char *msg, void *) {
    (void)msg;
}
void warning_callback(const char *msg, void *) {
    (void)msg;
}
void info_callback(const char *msg, void *) {
    (void)msg;
}

bool Write_Resolution(gdcm::StreamImageWriter & theStreamWriter, const char *filename, int res, std::ostream&
    of, int flag, gdcm::SequenceOfItems *sq, int No_Of_Resolutions)
{
    std::ifstream is;
    is.open( filename, std::ios::binary );
    opj_dparameters_t parameters; /* decompression parameters */
    opj_event_mgr_t event_mgr; /* event manager */
    opj_dinfo_t* dinfo; /* handle to a decompressor */
    opj_cio_t *cio;
    opj_image_t *image = NULL;
    // FIXME: Do some stupid work:
    is.seekg( 0, std::ios::end);
    std::streampos buf_size = is.tellg();
    char *dummy_buffer = new char[(unsigned int)buf_size];
    is.seekg(0, std::ios::beg);
    is.read( dummy_buffer, buf_size);
    unsigned char *src = (unsigned char*)dummy_buffer;
    uint32_t file_length = (uint32_t)buf_size; // 32bits truncation should be ok since DICOM cannot have larger

```

```

        than 2Gb image

/* configure the event callbacks (not required) */
memset(&event_mgr, 0, sizeof(opj_event_mgr_t));
event_mgr.error_handler = error_callback;
event_mgr.warning_handler = warning_callback;
event_mgr.info_handler = info_callback;

/* set decoding parameters to default values */
opj_set_default_decoder_parameters(&parameters);

// default blindly copied
parameters.cp_layer=0;
parameters.cp_reduce= res;
// parameters.decode_format=-1;
// parameters.cod_format=-1;

const char jp2magic[] = "\x00\x00\x00\x0C\x6A\x50\x20\x20\x0D\x0A\x87\x0A";
if( memcmp( src, jp2magic, sizeof(jp2magic) ) == 0 )
{
    /* JPEG-2000 compressed image data ... sigh */
    // gdcmData/ELSCINT1_JP2vsJ2K.dcm
    // gdcmData/MAROTECH_CT_JP2Lossy.dcm
    //gdcmWarningMacro( "J2K start like JPEG-2000 compressed image data instead of codestream" );
    parameters.decode_format = 1; //JP2_CFMT;
    //assert(parameters.decode_format == JP2_CFMT);
}
else
{
    /* JPEG-2000 codestream */
    //parameters.decode_format = J2K_CFMT;
    //assert(parameters.decode_format == J2K_CFMT);
    assert( 0 );
}
parameters.cod_format = 11; // PGX_DFMT;
//assert(parameters.cod_format == PGX_DFMT);

/* get a decoder handle */
dinfo = opj_create_decompress(CODEC_JP2);

/* catch events using our callbacks and give a local context */
opj_set_event_mgr((opj_common_ptr)dinfo, &event_mgr, NULL);

/* setup the decoder decoding parameters using user parameters */
opj_setup_decoder(dinfo, &parameters);

/* open a byte stream */
cio = opj_cio_open((opj_common_ptr)dinfo, src, file_length);

/* decode the stream and fill the image structure */
image = opj_decode(dinfo, cio);
if(!image) {
    opj_destroy_decompress(dinfo);
    opj_cio_close(cio);
    //gdcmErrorMacro( "opj_decode failed" );
    return 1;
}

    opj_cp_t * cp = ((opj_jp2_t*)dinfo->jp2_handle)->j2k->cp;
    opj_tcp_t *tcp = &cp->tccps[0];
    opj_tccp_t *tccp = &tcp->tccps[0];
    /*      std::cout << "\n No of Cols In Image" << image->x1;
    std::cout << "\n No of Rows In Image" << image->y1;
    std::cout << "\n No of Components in Image" << image->numcomps;
    std::cout << "\n No of Resolutions" << tccp->numresolutions << "\n";
*/

    opj_j2k_t* j2k = NULL;
    opj_jp2_t* jp2 = NULL;
    jp2 = (opj_jp2_t*)dinfo->jp2_handle;
    int reversible = jp2->j2k->cp->tccps->tccps->qmfbid;
    //std::cout << reversible;
    int compno = 0;
    opj_image_comp_t *comp = &image->comps[compno];
    int Dimensions[2];
    Dimensions[0]= comp->w;
    Dimensions[1]= comp->h;
    opj_cio_close(cio);
    unsigned long len = Dimensions[0]*Dimensions[1] * image->numcomps;
    //std::cout << "\nTest" <<image->comps[0].factor;
    char *raw = new char[len];

```

```

for (unsigned int compno = 0; compno < (unsigned int)image->numcomps; compno++)
{
    obj_image_comp_t *comp = &image->comps[compno];

    int w = image->comps[compno].w;
    int h = image->comps[compno].h;
    uint8_t *data8 = (uint8_t*)raw + compno;
    for (int i = 0; i < w * h ; i++)
    {
        int v = image->comps[compno].data[i];
        *data8 = (uint8_t)v;
        data8 += image->numcomps;
    }
}

gdcm::Writer w;
gdcm::File &file = w.GetFile();
gdcm::DataSet &ds = file.GetDataSet();

file.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );

gdcm::UIDGenerator uid;
gdcm::DataElement de( gdcm::Tag(0x8,0x18) ); // SOP Instance UID
de.SetVR( gdcm::VR::UI );
const char *u = uid.Generate();
de.SetByteValue( u, strlen(u) );
ds.Insert( de );

gdcm::DataElement del( gdcm::Tag(0x8,0x16) );
del.SetVR( gdcm::VR::UI );
gdcm::MediaStorage ms( gdcm::MediaStorage::CTImageStorage );
del.SetByteValue( ms.GetString(), strlen(ms.GetString()) );
ds.Insert( del );

const char mystr[] = "MONOCHROME2 ";
gdcm::DataElement de2( gdcm::Tag(0x28,0x04) );
//de.SetTag(gdcm::Tag(0x28,0x04));
de2.SetVR( gdcm::VR::CS );
de2.SetByteValue(mystr, strlen(mystr));
ds.Insert( de2 );

gdcm::Attribute<0x0028,0x0010> row = {image->comps[0].w};
//row.SetValue(512);
ds.Insert( row.GetAsDataElement() );
// w.SetCheckFileMetaInformation( true );
gdcm::Attribute<0x0028,0x0011> col = {image->comps[0].h};
ds.Insert( col.GetAsDataElement() );
gdcm::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
ds.Insert( Number_Of_Frames.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0100> at = {8};
ds.Insert( at.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0002> at1 = {image->numcomps};
ds.Insert( at1.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0101> at2 = {8};
ds.Insert( at2.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0102> at3 = {7};
ds.Insert( at3.GetAsDataElement() );

if (flag == 1)
{
    for (int i=0; i < No_Of_Resolutions; i++)
    {
        int a = 1;
        int b = 1;

        while(a!=(No_Of_Resolutions)-i))
        {
            b = b*2;
            a = a+1;
        }
        uint16_t row = (image->y1)/b;
        uint16_t col = (image->x1)/b;
        //std::cout << row;
    }
}

```

```

    gdcmm::Element<gdcmm::VR::IS,gdcmm::VM::VM1> el2;
    el2.SetValue(i+1);
    gdcmm::DataElement rfn = el2.GetAsDataElement(); //ulr --> upper left row
    rfn.SetTag( gdcmm::Tag(0x0008,0x1160) );

    gdcmm::Element<gdcmm::VR::US,gdcmm::VM::VM2> el;
    el.SetValue(1,0);
    el.SetValue(1,1);
    gdcmm::DataElement ulr = el.GetAsDataElement(); //ulr --> upper left col/row
    ulr.SetTag( gdcmm::Tag(0x0048,0x0201) );

    gdcmm::Element<gdcmm::VR::US,gdcmm::VM::VM2> ell;
    ell.SetValue(col,0);
    ell.SetValue(row,1);
    gdcmm::DataElement brr = ell.GetAsDataElement();
    brr.SetTag( gdcmm::Tag(0x0048,0x0202) ); //brr --> bottom right col/row
    gdcmm::Item it;
    gdcmm::DataSet &nds = it.GetNestedDataSet();
    nds.Insert( rfn );
    nds.Insert( ulr );
    nds.Insert( brr );

    sq->AddItem(it);
}

gdcmm::Writer wl;
gdcmm::File &file1 = wl.GetFile();
gdcmm::DataSet &ds1 = file1.GetDataSet();
file1.GetHeader().SetDataSetTransferSyntax( gdcmm::TransferSyntax::ExplicitVRLittleEndian );

gdcmm::UIDGenerator uid1;
gdcmm::DataElement dea( gdcmm::Tag(0x8,0x18) ); // SOP Instance UID
dea.SetVR( gdcmm::VR::UI );
const char *ul = uid1.Generate();
dea.SetByteValue( ul, strlen(ul) );
ds1.Insert( dea );

gdcmm::DataElement deb( gdcmm::Tag(0x8,0x16) );
deb.SetVR( gdcmm::VR::UI );
gdcmm::MediaStorage ms1( gdcmm::MediaStorage::VLWholeSlideMicroscopyImageStorage );
deb.SetByteValue( ms1.GetString(), strlen(ms1.GetString()) );
ds1.Insert( deb );

const char mystr1[] = "MONOCHROME2 ";
gdcmm::DataElement dec( gdcmm::Tag(0x28,0x04) );
//de.SetTag(gdcmm::Tag(0x28,0x04));
dec.SetVR( gdcmm::VR::CS );
dec.SetByteValue(mystr, strlen(mystr1));
ds1.Insert( dec );

gdcmm::Attribute<0x0028,0x0010> row1 = {image->y1};
//row.SetValue(512);
ds1.Insert( row1.GetAsDataElement() );
// w.SetCheckFileMetaInformation( true );
gdcmm::Attribute<0x0028,0x0011> col1 = {image->x1};
ds1.Insert( col1.GetAsDataElement() );
gdcmm::Attribute<0x0028,0x0008> Number_Of_Frames1 = {tccp->numresolutions};
ds1.Insert( Number_Of_Frames1.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0100> ata = {8};
ds1.Insert( ata.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0002> atb = {image->numcomps};
ds1.Insert( atb.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0101> atc = {8};
ds1.Insert( atc.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0102> atd = {7};
ds1.Insert( atd.GetAsDataElement() );

theStreamWriter.SetFile(file1);

gdcmm::DataElement des( gdcmm::Tag(0x0048,0x0200) );
des.SetVR(gdcmm::VR::SQ);
//des.SetVR(gdcmm::VM::VM1);
des.SetValue(*sq);

```

```

des.SetVLToUndefined();

dsl.Insert(des);

if (!theStreamWriter.WriteImageInformation()){
    std::cerr << "unable to write image information" << std::endl;
    return 1; //the CanWrite function should prevent getting here, else,
    //that's a test failure
}

}

theStreamWriter.SetFile(file);

if (!theStreamWriter.CanWriteFile()){
    delete [] raw;
    std::cout << "Not able to write";
    return 0; //this means that the file was unwritable, period.
    //very similar to a ReadImageInformation failure
}
else
    std::cout << "\nable to read";

// Important to write here
std::vector<unsigned int> extent = gdcm::ImageHelper::GetDimensionsValue(file);

unsigned short xmax = extent[0];
unsigned short ymax = extent[1];
unsigned short theChunkSize = 4;
unsigned short ychunk = extent[1]/theChunkSize; //go in chunk sizes of theChunkSize
unsigned short zmax = extent[2];
std::cout << "\n" << xmax << " \n" << ymax << "\n" << zmax << "\n" << "image->numcomps" << "\n";

if (xmax == 0 || ymax == 0)
{
    std::cerr << "Image has no size, unable to write zero-sized image." << std::endl;
    return 0;
}

int z, y, nexty;
unsigned long prevLen = 0; //when going through the char buffer, make sure to grab
//the bytes sequentially. So, store how far you got in the buffer with each iteration.
for (z = 0; z < zmax; ++z){
    for (y = 0; y < ymax; y += ychunk){
        nexty = y + ychunk;
        if (nexty > ymax) nexty = ymax;
        theStreamWriter.DefinePixelExtent(0, xmax, y, nexty, z, z+1);
        unsigned long len = theStreamWriter.DefineProperBufferLength();
        std::cout << "\n" << len;
        char* finalBuffer = new char[len];
        memcpy(finalBuffer, &(raw[prevLen]), len);
        std::cout << "\n able to write";
        if (!theStreamWriter.Write(finalBuffer, len)){
            std::cerr << "writing failure:" << "output.dcm" << " at y = " << y << " and z = " << z << std::endl;
            delete [] raw;
            delete [] finalBuffer;
            return 1;
        }
        delete [] finalBuffer;
        prevLen += len;
    }
}
delete raw;

delete[] src; //FIXME

if(dinfo) {
    opj_destroy_decompress(dinfo);
}

opj_image_destroy(image);

return true;
}

bool Different_Resolution( gdcm::StreamImageWriter & theStreamWriter, const char *filename, int res,
    std::ostream& of)

```

```

{
    //std::vector<std::string>::const_iterator it = filenames.begin();
    bool b = true;
    int flag = 1;

    gdcmm::SmartPointer<gdcmm::SequenceOfItems> sq = new gdcmm::SequenceOfItems();
    sq->SetLengthToUndefined();

    for(int i = res-1 ; i>=0; --i)
    {
        b = b && Write_Resolution( theStreamWriter, filename, i, of ,flag,sq,res);
        // b = b && Get_Resolution( theStreamWriter, filename, i, of ,0);
        flag = 0;
    }
    //b = b && Get_Lowest_Resolution( writer, sq, filename, res-1 );
    //b = b && PopulateSingeFile( writer, sq, jpeg, filename2 );
    //image.SetDimension(2, res )
    return b;
}

int main(int argc, char *argv[])
{
    if( argc < 4 )
    {
        std::cerr << argv[0] << " input.jp2 output.dcm No. Of Resolutions " << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    char *resolutions = argv[3];
    int res = int((*resolutions)-48);
    //std::cout << "\nres"<< res;
    gdcmm::StreamImageWriter theStreamWriter;

    std::ofstream of;
    of.open( outfile, std::ios::out | std::ios::binary );
    theStreamWriter.SetStream(of);

    if( !Different_Resolution( theStreamWriter, filename,res,of ) ) return 1;

    uint16_t firstTag1 = 0xfffe;
    uint16_t secondTag1 = 0xe0dd;
    uint32_t thirdTag1 = 0x00000000;
    //uint16_t fourthTag1 = 0xffff;
    const int theBufferSize1 = 2*sizeof(uint16_t)+sizeof(uint32_t);
    char* tmpBuffer2 = new char[theBufferSize1];
    memcpy(&(tmpBuffer2[0]), &firstTag1, sizeof(uint16_t));
    memcpy(&(tmpBuffer2[sizeof(uint16_t)]), &secondTag1, sizeof(uint16_t));
    memcpy(&(tmpBuffer2[2*sizeof(uint16_t)]), &thirdTag1, sizeof(uint32_t));
    //memcpy(&(tmpBuffer2[3*sizeof(uint16_t)]), &fourthTag1, sizeof(uint16_t));
    assert( of && !of.eof() && of.good() );
    of.write(tmpBuffer2, theBufferSize1);
    of.flush();
    assert( of );

    return 0;
}

```

## 12.66 Fake\_Image\_Using\_Stream\_Image\_Writer.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

```

```

=====*/
// This work was realised during the GSOC 2011 by Manoj Alwani

#include "gdcmReader.h"
#include "gdcmMediaStorage.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmAttribute.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDGenerator.h"
#include "gdcmAnonymizer.h"
#include "gdcmStreamImageWriter.h"
#include "gdcmImageHelper.h"
#include "gdcmTrace.h"

int main(int, char *[])
{
    char * buffer = new char[ 256 * 256 *3 ];
    // *p = (uint8_t*)buffer;
    char * p = buffer;

    gdcm::Trace::DebugOn();
    gdcm::Trace::WarningOn();

    for(int row = 0; row < 256; ++row)
    {
        for(int col = 0; col < 256; ++col)
            //for(int b = 0; b < 256; ++b)
            {
                *p++ = 255;
                *p++ = 0;
                *p++ = 0;
            }

        gdcm::Writer w;
        gdcm::File &file = w.GetFile();
        gdcm::DataSet &ds = file.GetDataSet();

        file.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );

        gdcm::UIDGenerator uid;
        gdcm::DataElement de( gdcm::Tag(0x8,0x18) ); // SOP Instance UID
        de.SetVR( gdcm::VR::UI );
        const char *u = uid.Generate();
        de.SetByteValue( u, strlen(u) );
        ds.Insert( de );

        gdcm::DataElement del( gdcm::Tag(0x8,0x16) );
        del.SetVR( gdcm::VR::UI );
        gdcm::MediaStorage ms( gdcm::MediaStorage::VLWholeSlideMicroscopyImageStorage );
        del.SetByteValue( ms.GetString(), strlen(ms.GetString()) );
        ds.Insert( del );

        const char mystr[] = "RGB";
        gdcm::DataElement de2( gdcm::Tag(0x28,0x04) );
        //de.SetTag( gdcm::Tag(0x28,0x04) );
        de2.SetVR( gdcm::VR::CS );
        de2.SetByteValue(mystr, strlen(mystr));
        ds.Insert( de2 );

        gdcm::Attribute<0x0028,0x0010> row = {256};
        //row.SetValue(512);
        ds.Insert( row.GetAsDataElement() );
        // w.SetCheckFileMetaInformation( true );
        gdcm::Attribute<0x0028,0x0011> col = {256};
        ds.Insert( col.GetAsDataElement() );

        gdcm::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
        ds.Insert( Number_Of_Frames.GetAsDataElement() );

        gdcm::Attribute<0x0028,0x0100> at = {8};
        ds.Insert( at.GetAsDataElement() );

        gdcm::Attribute<0x0028,0x0002> at1 = {3}; //bits per pixel

```

```

    ds.Insert( at1.GetAsDataElement() );

    gdcm::Attribute<0x0028,0x0101> at2 = {8};
    ds.Insert( at2.GetAsDataElement() );

    gdcm::Attribute<0x0028,0x0102> at3 = {7};
    ds.Insert( at3.GetAsDataElement() );

    gdcm::Attribute<0x0028,0x0006> at4 = {0};
    ds.Insert( at4.GetAsDataElement() );

    gdcm::Attribute<0x0028,0x0103> at5 = {0};
    ds.Insert( at5.GetAsDataElement() );

    //de.SetTag(gdcm::Tag(0x7fe0,0x0010));
    //ds.Insert(de);

    gdcm::StreamImageWriter theStreamWriter;
    gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new gdcm::SequenceOfItems();
    sq->SetLengthToUndefined();

    uint16_t row1 = 256;
    uint16_t col1 = 256;
    //std::cout << row;

    gdcm::Element<gdcm::VR::IS, gdcm::VM::VM1> el2;
    el2.SetValue(1);
    gdcm::DataElement rfn = el2.GetAsDataElement(); //rfn ---> reference frame number
    rfn.SetTag( gdcm::Tag(0x0008,0x1160) );

    gdcm::Element<gdcm::VR::US, gdcm::VM::VM2> el;
    el.SetValue(1,0);
    el.SetValue(1,1);
    gdcm::DataElement ulr = el.GetAsDataElement(); //ulr --> upper left col/row
    ulr.SetTag( gdcm::Tag(0x0048,0x0201) );

    gdcm::Element<gdcm::VR::US, gdcm::VM::VM2> el1;
    el1.SetValue(col1,0);
    el1.SetValue(row1,1);
    gdcm::DataElement brr = el1.GetAsDataElement();
    brr.SetTag( gdcm::Tag(0x0048,0x0202) ); //brr --> bottom right col/row

    gdcm::Item it;
    gdcm::DataSet &nds = it.GetNestedDataSet();
    nds.Insert( rfn );
    nds.Insert(ulr);
    nds.Insert(brr);

    sq->AddItem(it);

    gdcm::DataElement des( gdcm::Tag(0x0048,0x0200) );
    des.SetVR(gdcm::VR::SQ);
    des.SetValue(*sq);
    des.SetVLToUndefined();

    ds.Insert( des);

    theStreamWriter.SetFile(file);

    std::ofstream of;
    of.open( "output.dcm", std::ios::out | std::ios::binary );
    theStreamWriter.SetStream(of);

    if (!theStreamWriter.CanWriteFile()){
        delete [] buffer;
        std::cout << "Not able to write";
        return 0; //this means that the file was unwritable, period.
        //very similar to a ReadImageInformation failure
    }
    else
        std::cout << "\nable to read";

    if (!theStreamWriter.WriteImageInformation()){
        std::cerr << "unable to write image information" << std::endl;
        delete [] buffer;
        return 1; //the CanWrite function should prevent getting here, else,
        //that's a test failure
    }

```



```

    }

    std::vector<unsigned int> extent =
        gdcmm::ImageHelper::GetDimensionsValue(file);

    unsigned short xmax = extent[0];
    unsigned short ymax = extent[1];
    unsigned short theChunkSize = 1;
    unsigned short ychunk = extent[1]/theChunkSize; //go in chunk sizes of theChunkSize
    unsigned short zmax = extent[2];

    std::cout << xmax << ymax << zmax;

    if (xmax == 0 || ymax == 0)
    {
        std::cerr << "Image has no size, unable to write zero-sized image." << std::endl;
        return 0;
    }

    int z, y, nexty;
    unsigned long prevLen = 0; //when going through the char buffer, make sure to grab
    //the bytes sequentially. So, store how far you got in the buffer with each iteration.
    for (z = 0; z < zmax; ++z){
        for (y = 0; y < ymax; y += ychunk){
            nexty = y + ychunk;
            if (nexty > ymax) nexty = ymax;
            theStreamWriter.DefinePixelExtent(0, xmax, y, nexty, z, z+1);
            unsigned long len = theStreamWriter.DefineProperBufferLength();
            std::cout << "\n" << len;
            char* finalBuffer = new char[len];
            memcpy(finalBuffer, &(buffer[prevLen]), len);
            std::cout << "\nable to write";
            if (!theStreamWriter.Write(finalBuffer, len)){
                std::cerr << "writing failure:" << "output.dcm" << " at y = " << y << " and z = " << z << std::endl;
                delete [] buffer;
                delete [] finalBuffer;
                return 1;
            }
            delete [] finalBuffer;
            prevLen += len;
        }
    }
    delete buffer;

    uint16_t firstTag1 = 0xfffe;
    uint16_t secondTag1 = 0xe0dd;
    uint32_t thirdTag1 = 0x00000000;
    //uint16_t fourthTag1 = 0xffff;
    const int theBufferSize1 = 2*sizeof(uint16_t)+sizeof(uint32_t);
    char* tmpBuffer2 = new char[theBufferSize1];
    memcpy(&(tmpBuffer2[0]), &firstTag1, sizeof(uint16_t));
    memcpy(&(tmpBuffer2[sizeof(uint16_t)]), &secondTag1, sizeof(uint16_t));
    memcpy(&(tmpBuffer2[2*sizeof(uint16_t)]), &thirdTag1, sizeof(uint32_t));
    //memcpy(&(tmpBuffer2[3*sizeof(uint16_t)]), &fourthTag1, sizeof(uint16_t));
    assert( of && !of.eof() && of.good() );
    of.write(tmpBuffer2, theBufferSize1);
    of.flush();
    assert( of );

    return 0;
}

```

## 12.67 FixBrokenJ2K.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====

```

```

=====*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfFragments.h"
#include "gdcmFile.h"

// http://www.lost.in.ua/dicom/c.dcm
//
// -> BuggyJ2Kvvvua-fixed2-j2k.dcm

/*
 * This program attempts to fix a broken J2K/DICOM:
 * It contains 2 bugs:
 * 1. The first 8 bytes seems to be random bytes: remove them
 * 2. YCC is set to 1, while image is grayscale need to set it back to 0
 *
 * Ref:
 * It's a software from http://rentgenprom.ru/ , shipped with universal digital radiographic units
 * "ProScan-2000". The Ukrainian manufacturer developed own digital radiographic unit and it is
 * compatible with software from "ProScan-2000".
 * Information found in DICOM file is:
 *
 * (0008,0070) LO [ZAO "Renthenprom" (JSC Rentgenprom) ]          # 36,1 Manufacturer
 * (0018,1020) LO [2.13.1.7]                                     # 8,1-n Software Version(s)
 *
 */
int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcm::File &file = reader.GetFile();
    const gdcm::DataElement &pixeldata0 = file.GetDataSet().GetDataElement( gdcm::Tag(0x7fe0,0x0010) );
    const gdcm::SequenceOfFragments *sqf = pixeldata0.GetSequenceOfFragments();
    if( !sqf )
    {
        return 1;
    }
    const gdcm::Fragment &frag0 = sqf->GetFragment(0);

    gdcm::ByteValue *bv = const_cast<gdcm::ByteValue*>(frag0.GetByteValue());
    char *ptr = (char*)bv->GetVoidPointer();
    size_t len = bv->GetLength();

    static const unsigned char sig[] = {0,0,0,0,0x6A,0x70,0x32,0x63};
    if( memcmp(ptr, sig, sizeof(sig)) != 0 )
    {
        std::cerr << "magic random signature not found" << std::endl;
        return 1;
    }

    // Apparently the flag to enable a color transform on 3 color components is set in
    // the COD marker. (YCC is byte[6] in the COD marker)
    // we need to disable this flag;
    char *cod_marker = ptr + 0x35; /* 0x2d + 0x8 */ // FIXME
    if( cod_marker[0] == (char)0xff && cod_marker[1] == 0x52 )
    {
        // found start of COD
        if( cod_marker[6+2] == 1 )
        {
            // Change in place:
            *((char*)cod_marker + 6+2) = 0;
            // Prepare a new DataElement:
            gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
            pixeldata.SetVR( gdcm::VR::OB );
            gdcm::SmartPointer<gdcm::SequenceOfFragments> sq = new gdcm::SequenceOfFragments;

            gdcm::Fragment frag;
            // remove 8 first bytes:

```

```

        frag.SetByteValue( ptr + 8, (uint32_t)(len - 8) );
        sq->AddFragment( frag );
        pixeldata.SetValue( *sq );
        file.GetDataSet().Replace( pixeldata );
    }
    else
    {
        return 1;
    }
}
else
{
    std::cerr << "COD not found" << (int)cod_marker[0] << std::endl;
    return 1;
}

gdcm::Writer writer;
writer.SetFile( reader.GetFile() );
writer.SetFileName( outfilename );
writer.CheckFileMetaInformationOff();
if( !writer.Write() )
{
    std::cerr << "Could not write" << std::endl;
}

// paranoid check:
gdcm::ImageReader ireader;
ireader.SetFileName( outfilename );
if( !ireader.Read() )
{
    std::cerr << "file written is still not valid, please report" << std::endl;
    return 1;
}

return 0;
}

```

## 12.68 FixJAIBugJPEGLS.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmImageReader.h"

#include <fstream>

#include "gdcm_charls.h"

/*
 * This small example should show how one can handle the famous JAI-JPEGLS bug
 * It will take in as invalid DICOM/JAI-JPEG-LS and write out as Explicit Little
 * Endian. One can use 'gdcmconv --jpegl's' to recompress properly
 *
 * References:
 * http://charls.codeplex.com/discussions/230307?ProjectName=charls
 * http://charls.codeplex.com/workitem/7297
 * http://www.dcm4che.org/jira/browse/DCM-442
 * http://www.dcm4che.org/jira/browse/DCMEE-1144
 * http://java.net/jira/browse/JAI_IMAGEIO_CORE-183
 *
 * Explanation of the issue:
 *
 * Seems, the error is in the calculation of the default values for thresholds T1,
 * T2, T3, in particular min(MAXVAL, 4095) is not applied in

```

```

*
* FACTOR = (min(MAXVAL, 4095) + 128)/256
*
* as specified in http://www.itu.int/rec/T-REC-T.87-199806-I/en .
*
*/
int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    gdcm::FileMetaInformation::SetSourceApplicationEntityTitle( "FixJAIBugJPEGs" );

    gdcm::ImageReader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcm::Image &image = reader.GetImage();
    //unsigned long len = image.GetBufferLength();
    const gdcm::DataElement &in =
        reader.GetFile().GetDataSet().GetDataElement( gdcm::Tag(0x7fe0,0x0010) );
    const gdcm::SequenceOfFragments *sf = in.GetSequenceOfFragments();
    if( !sf )
    {
        std::cerr << "No pixel data (or not encapsulated)" << std::endl;
        return 1;
    }
    const unsigned int *dims = image.GetDimensions();
    if ( sf->GetNumberOfFragments() != dims[2] )
    {
        std::cerr << "Unsupported" << std::endl;
        return 1;
    }

    // unsigned long totalLen = sf->ComputeByteLength();
    std::vector<unsigned char> rgbyteOutall;
    for(unsigned int i = 0; i < sf->GetNumberOfFragments(); ++i)
    {
        const gdcm::Fragment &frag = sf->GetFragment(i);
        if( frag.IsEmpty() ) return 1;
        const gdcm::ByteValue *bv = frag.GetByteValue();
        if( !bv ) return 1;
        unsigned long totalLen = bv->GetLength();

        std::vector<char> vbuffer;
        vbuffer.resize( totalLen );
        char *buffer = vbuffer.data();
        bv->GetBuffer(buffer, totalLen);
        const unsigned char* pbyteCompressed0 = (const unsigned char*)buffer;
        while( totalLen > 0 && pbyteCompressed0[totalLen-1] != 0xd9 )
        {
            totalLen--;
        }

        JlsParameters metadata;
        char errorMsg[256+1]={'\0'};
        if (JpegLsReadHeader(buffer, totalLen, &metadata, errorMsg) != charls::ApiResult::OK)
        {
            std::cerr << "Can't parse jpegls: " << errorMsg << std::endl;
            return 1;
        }

        std::cout << metadata.width << std::endl;
        std::cout << metadata.height << std::endl;
        std::cout << metadata.bitsPerSample << std::endl;

        gdcm::PixelFormat const &pf = image.GetPixelFormat();
        std::cout << pf << std::endl;

        // http://charls.codeplex.com/discussions/230307?ProjectName=charls
        unsigned char marker_lse_13[] = {
            0xFF, 0xF8, 0x00, 0x0D,
            0x01,
            0x1F, 0xFF,

```

```

        0x00, 0x22, // T1 = 34
        0x00, 0x83, // T2 = 131
        0x02, 0x24, // T3 = 548
        0x00, 0x40
    };

    unsigned char marker_lse_14[] = {
        0xFF, 0xF8, 0x00, 0x0D,
        0x01,
        0x3F, 0xFF,
        0x00, 0x42, // T1 = 66
        0x01, 0x03, // T2 = 259
        0x04, 0x44, // T3 = 1092
        0x00, 0x40
    };

    unsigned char marker_lse_15[] = {
        0xFF, 0xF8, 0x00, 0x0D,
        0x01,
        0x7F, 0xFF,
        0x00, 0x82, // T1 = 130
        0x02, 0x03, // T2 = 515
        0x08, 0x84, // T3 = 2180
        0x00, 0x40
    };

    unsigned char marker_lse_16[] = {
        0xFF, 0xF8, 0x00, 0x0D,
        0x01,
        0xFF, 0xFF,
        0x01, 0x02, // T1 = 258
        0x04, 0x03, // T2 = 1027
        0x11, 0x04, // T3 = 4356
        0x00, 0x40
    };

    const unsigned char *marker_lse = nullptr;
    switch( metadata.bitsPerSample )
    {
    case 13:
        marker_lse = marker_lse_13;
        break;
    case 14:
        marker_lse = marker_lse_14;
        break;
    case 15:
        marker_lse = marker_lse_15;
        break;
    case 16:
        marker_lse = marker_lse_16;
        break;
    }
    if( !marker_lse )
    {
        std::cerr << "Can't handle: " << metadata.bitsPerSample << std::endl;
        return 1;
    }

    // FIXME: One should recompute the value for 0x0F
    vbuffer.insert( vbuffer.begin() + 0x0F, marker_lse, marker_lse+15);

#ifdef 0
    std::ofstream of( "/tmp/d.jls", std::ios::binary );
    of.write( &vbuffer[0], vbuffer.size() );
    of.close();
#endif

    const char *pbyteCompressed = vbuffer.data();
    size_t cbyteCompressed = vbuffer.size(); // updated length

    JlsParameters params;
    JpegLsReadHeader( pbyteCompressed, cbyteCompressed, &params, nullptr);

    std::vector<unsigned char> rgbyteOut;
    //rgbyteOut.resize( image.GetBufferLength() );
    rgbyteOut.resize( params.height * params.width * ((params.bitsPerSample + 7)
        / 8) * params.components);

    CharlsApiResultType result =
        JpegLsDecode( rgbyteOut.data(), rgbyteOut.size(), pbyteCompressed, cbyteCompressed, &params, errorMsg );
    if (result != charls::ApiResult::OK)

```

```

    {
        std::cerr << "Could not patch JAI-JPEGLS: " << errorMsg << std::endl;
        return 1;
    }
    rgbyteOutall.insert( rgbyteOutall.end(), rgbyteOut.begin(), rgbyteOut.end() );
}

gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
pixeldata.SetVR( gdcm::VR::OW );
pixeldata.SetByteValue( (char*)rgbyteOutall.data(), (uint32_t)rgbyteOutall.size() );

// Add the pixel data element
reader.GetFile().GetDataSet().Replace( pixeldata );
reader.GetFile().GetHeader().SetDataSetTransferSyntax(
    gdcm::TransferSyntax::ExplicitVRLittleEndian);

gdcm::Writer writer;
writer.SetFileName( outfilename );
writer.SetFile( reader.GetFile() );
writer.Write();

std::cout << "Success !" << std::endl;

return 0;
}

```

## 12.69 FixOrientation.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmFile.h"
#include "gdcmOrientation.h"
#include "gdcmAttribute.h"

// Very simple orientation changer, fix invalid dataset
int main(int argc, char* argv[] )
{
    // assume AXIAL input for now
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    gdcm::Reader reader;
    reader.SetFileName( filename );
    if (! reader.Read() )
    {
        return 1;
    }

    const double axial[] = { 1,0,0, 0,1,0 };
    (void)axial;
    const double coronal[] = { 0,0,1, 1,0,0 };
    (void)coronal;
    const double sagittal[] = { 0,1,0, 0,0,1 };
    (void)sagittal;
    gdcm::Attribute<0x0020,0x0032> at1; // IPP
    (void)at1;
    gdcm::Attribute<0x0020,0x0037> at2; // IOP

```

```

(void)at2;

gdcmm::File & f = reader.GetFile();
gdcmm::DataSet & ds = f.GetDataSet();
at1.SetFromDataSet( ds );
#ifdef 0
at2.SetFromDataSet( ds );
const double * iop = at2.GetValues();
if( !std::equal(iop, iop + 6, axial ) )
{
    gdcmm::Orientation::OrientationType type = gdcmm::Orientation::GetType ( iop );
    std::cerr << "Wrong orientation: " << gdcmm::Orientation::GetLabel( type ) << std::endl;
    return 1;
}
at2.SetValues( sagittal );
ds.Replace( at2.GetAsDataElement() );
#endif

// for sagittal: swap element 0 & 2
const double tmp0 = at1.GetValue(0);
const double tmp2 = at1.GetValue(2);
(void)tmp2;
//at1.SetValue(tmp2, 0);
//at1.SetValue(tmp0, 2);
at1.SetValue( - tmp0 );
ds.Replace( at1.GetAsDataElement() );

gdcmm::Writer writer;
writer.SetFile( f );
writer.SetFileName( outfilename );
if ( !writer.Write() )
{
    return 1;
}

return 0;
}

```

## 12.70 GenAllVR.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmmReader.h"
#include "gdcmmGlobal.h"
#include "gdcmmDummyValueGenerator.h"
#include "gdcmmMediaStorage.h"
#include "gdcmmWriter.h"
#include "gdcmmItem.h"
#include "gdcmmImageReader.h"
#include "gdcmmSequenceOfItems.h"
#include "gdcmmFile.h"
#include "gdcmmTag.h"
#include "gdcmmDict.h"
#include "gdcmmDictEntry.h"
#include "gdcmmDicts.h"
#include "gdcmmTransferSyntax.h"
#include "gdcmmUIDGenerator.h"
#include "gdcmmFileExplicitFilter.h"

#include <cstdlib>
#include <cstring>

gdcmm::Tag FindTagFromVR(gdcmm::Dict const &dict, gdcmm::VR const &vr)
{
    using gdcmm::Dict;

```

```

Dict::ConstIterator beg = dict.Begin();
Dict::ConstIterator end = dict.End();
Dict::ConstIterator it;
for( it = beg; it != end; ++it)
{
    const gdc::Tag &t = it->first;
    const gdc::DictEntry &de = it->second;
    const gdc::VR &vr_de = de.GetVR();
    if( vr == vr_de && !de.GetRetired() && t.GetGroup() >= 0x8 )
    {
        return t;
    }
}
return gdc::Tag(0xffff,0xffff);
}

struct rnd_gen {
    rnd_gen(char const* r = "abcdefghijklmnopqrstuvwxyz0123456789")
        : range(r), len(std::strlen(r)) { }

    char operator ()() const {
        return range[static_cast<std::size_t>(std::rand() * (1.0 / ((double)RAND_MAX + 1.0)) * (double)len)];
    }
private:
    char const* range;
    std::size_t len;
};

/*
*/
int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " output.dcm" << std::endl;
        return 1;
    }
    const char *outfilename = argv[1];
    static const gdc::Global &g = gdc::Global::GetInstance();
    static const gdc::Dicts &dicts = g.GetDicts();
    static const gdc::Dict &pubdict = dicts.GetPublicDict();
    using gdc::VR;
    using gdc::Tag;

    gdc::Writer w;

    gdc::File &f = w.GetFile();
    gdc::DataSet &ds = f.GetDataSet();

    gdc::FileExplicitFilter fef;
    //fef.SetChangePrivateTags( true );
    fef.SetFile( w.GetFile() );
    if( !fef.Change() )
    {
        std::cerr << "Failed to change" << std::endl;
        return 1;
    }

    gdc::SmartPointer<gdc::SequenceOfItems> sq = new gdc::SequenceOfItems();
    sq->SetLengthToUndefined();

    // gdc::DummyValueGenerator dv;

    const std::size_t len = 10;
    char ss[len+1];
    ss[len] = '\0';

    const char owner_str[] = "GDCM CONFORMANCE TESTS";
    gdc::DataElement owner( gdc::Tag(0x4d4d, 0x10) );
    owner.SetByteValue(owner_str, (uint32_t)strlen(owner_str));
    owner.SetVR( gdc::VR::LO );

    // Create an item
    gdc::Item it;
    it.SetVLToUndefined();
    gdc::DataSet &nds = it.GetNestedDataSet();
    // nds.Insert(owner);
    // nds.Insert(de);

    // Insert sequence into data set
    gdc::DataElement des( gdc::Tag(0x4d4d, 0x1001) );

```



```

des.SetVR(gdcm::VR::SQ);
des.SetValue(*sq);
des.SetVLToUndefined();

ds.Insert(owner);
ds.Insert(des);

// avoid INVALID = 0
for(int i = 1; i < 27; ++i)
{
    VR vr = (VR::VRType)(1LL « i);
    Tag t = FindTagFromVR( pubdict, vr );
    if( vr != VR::UN && vr != VR::SQ )
    {
        assert( t != Tag(0xffff,0xffff) );
        gdcm::DataElement de( t );
        std::generate_n(ss, len, rnd_gen());
        de.SetVR( vr );
        de.SetByteValue( ss, (uint32_t)std::strlen( ss ) );
        nds.Insert( de );
    }
}
sq->AddItem(it);

// Make sure to override any UID stuff
gdcm::UIDGenerator uid;
gdcm::DataElement de( Tag(0x8,0x18) ); // SOP Instance UID
de.SetVR( VR::UI );
const char *u = uid.Generate();
de.SetByteValue( u, (uint32_t)strlen(u) );
ds.Insert( de );

de.SetTag( Tag(0x8,0x16) ); // SOP Class UID
de.SetVR( VR::UI );
gdcm::MediaStorage ms( gdcm::MediaStorage::RawDataStorage );
de.SetByteValue( ms.GetString(), (uint32_t)strlen(ms.GetString()) );
ds.Insert( de );

gdcm::FileMetaInformation &fmi = f.GetHeader();
//fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ImplicitVRLittleEndian );
fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );

w.SetCheckFileMetaInformation( true );
w.SetFileName( outfilename );
if (!w.Write() )
{
    return 1;
}

return 0;
}

```

## 12.71 GenFakelIdentifyFile.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmGlobal.h"
#include "gdcmDummyValueGenerator.h"
#include "gdcmMediaStorage.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmAttribute.h"
#include "gdcmFile.h"

```

```

#include "gdcmtag.h"
#include "gdcmdict.h"
#include "gdcmdictentry.h"
#include "gdcmdicts.h"
#include "gdcmtansferSyntax.h"
#include "gdcmeidGenerator.h"
#include "gdcmanonymizer.h"

#include <cstdlib>
#include <cstring>

gdcmt::DataElement CreateFakeElement(gdcmt::Tag const &tag, bool toremove)
{
    static const gdcmt::Global &g = gdcmt::Global::GetInstance();
    static const gdcmt::Dicts &dicts = g.GetDicts();
    static const gdcmt::Dict &pubdict = dicts.GetPublicDict();
    static size_t countglobal = 0;
    static std::vector<gdcmt::Tag> balcptags =
        gdcmt::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes();
    size_t count = countglobal % balcptags.size();

    const gdcmt::DictEntry &dictentry = pubdict.GetDictEntry(tag);

    gdcmt::DataElement de;
    de.SetTag( tag );
    using gdcmt::VR;
    const VR &vr = dictentry.GetVR();
    //if( vr != VR::INVALID )
    if( vr.IsDual() )
    {
        if( vr == VR::US_SS )
        {
            de.SetVR( VR::US );
        }
        else if( vr == VR::US_SS_OW )
        {
            de.SetVR( VR::OW );
        }
        else if( vr == VR::OB_OW )
        {
            de.SetVR( VR::OB );
        }
    }
    else
    {
        de.SetVR( vr );
    }
    const char str[] = "BasicApplicationLevelConfidentialityProfileAttributes";
    const char safe[] = "This is safe to keep";
    if( de.GetVR() != VR::SQ )
    {
        if( toremove )
            de.SetByteValue( str, (uint32_t)strlen(str) );
        else
            de.SetByteValue( safe, (uint32_t)strlen(safe) );
    }
    else
    {
        // Create an item
        gdcmt::Item it;
        it.SetVLToUndefined();
        gdcmt::DataSet &nds = it.GetNestedDataSet();
        // Insert sequence into data set
        assert(de.GetVR() == gdcmt::VR::SQ );
        gdcmt::SmartPointer<gdcmt::SequenceOfItems> sq = new gdcmt::SequenceOfItems();
        sq->SetLengthToUndefined();
        de.SetValue(*sq);
        de.SetVLToUndefined();
        //ds.Insert( de );

        if( !toremove )
        {
            nds.Insert( CreateFakeElement( balcptags[count], true ) );
            countglobal++;
        }
        else
        {
            gdcmt::Attribute<0x0008,0x0000> at1 = { 0 }; // This element has no reason to be 'anonymized'...
            nds.Insert( at1.GetAsDataElement() );
            gdcmt::Attribute<0x000a,0x0000> at2 = { 0 };
            nds.Insert( at2.GetAsDataElement() );
        }
    }
}

```

```

    }
    sq->AddItem(it);
    }
    return de;
}

/*
*/
int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " output.dcm" << std::endl;
        return 1;
    }
    using gdcm::Tag;
    using gdcm::VR;
    const char *outfilename = argv[1];

    std::vector<gdcm::Tag> balcptags =
        gdcm::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes();

    gdcm::Writer w;
    gdcm::File &f = w.GetFile();
    gdcm::DataSet &ds = f.GetDataSet();

    // Add attribute that need to be anonymized:
    std::vector<gdcm::Tag>::const_iterator it = balcptags.begin();
    for(; it != balcptags.end(); ++it)
    {
        ds.Insert( CreateFakeElement( *it, true ) );
    }

    // Add attribute that do NOT need to be anonymized:
    static const gdcm::Global &g = gdcm::Global::GetInstance();
    static const gdcm::Dicts &dicts = g.GetDicts();
    static const gdcm::Dict &pubdict = dicts.GetPublicDict();

    using gdcm::Dict;
    Dict::ConstIterator dictit = pubdict.Begin();
    for(; dictit != pubdict.End(); ++dictit)
    {
        const gdcm::Tag &dicttag = dictit->first;
        if( dicttag == Tag(0x6e65,0x6146) ) break;
        //const gdcm::DictEntry &dictentry = dictit->second;
        ds.Insert( CreateFakeElement( dicttag, false ) );
    }
    ds.Remove( gdcm::Tag(0x400,0x500) );
    ds.Remove( gdcm::Tag(0x12,0x62) );
    ds.Remove( gdcm::Tag(0x12,0x63) );

    // Make sure to override any UID stuff
    gdcm::UIDGenerator uid;
    gdcm::DataElement de( Tag(0x8,0x18) ); // SOP Instance UID
    de.SetVR( VR::UI );
    const char *u = uid.Generate();
    de.SetByteValue( u, (uint32_t)strlen(u) );
    //ds.Insert( de );
    ds.Replace( de );

    de.SetTag( Tag(0x8,0x16) ); // SOP Class UID
    de.SetVR( VR::UI );
    gdcm::MediaStorage ms( gdcm::MediaStorage::RawDataStorage );
    de.SetByteValue( ms.GetString(), (uint32_t)strlen(ms.GetString()) );
    ds.Replace( de ); // replace !

    gdcm::FileMetaInformation &fmi = f.GetHeader();
    //fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ImplicitVRLittleEndian );
    fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );

    w.SetCheckFileMetaInformation( true );
    w.SetFileName( outfile );
    if (!w.Write() )
    {
        return 1;
    }

    return 0;
}

```

## 12.72 GenLongSeqs.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmFile.h"
#include "gdcmTag.h"

/*
 * This example is used to generate the file:
 *
 *
 * There is a flaw in the DICOM design where it is assumed that Sequence can be
 * either represented as undefined length or defined length. This should work
 * in most case, but the undefined length is a little more general and can
 * store sequence of items that a defined length cannot.
 * We need to make sure that we can store numerous Item in a SQ
 *
 * Warning: do not try to compute the group length elements !
 * Warning: You may need a 64bits machine for this example to work.
 */
int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();

    // Create a Sequence
    gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new gdcm::SequenceOfItems();
    sq->SetLengthToUndefined();

    const char owner_str[] = "GDCM CONFORMANCE TESTS";
    gdcm::DataElement owner( gdcm::Tag(0x4d4d, 0x10) );
    owner.SetByteValue( owner_str, (uint32_t)strlen(owner_str));
    owner.SetVR( gdcm::VR::LO );

    size_t nitems = 1000;
    nitems += std::numeric_limits<uint32_t>::max();
    for(unsigned int idx = 0; idx < nitems; ++idx)
    {
        // Create a dataelement
        //gdcm::DataElement de( gdcm::Tag(0x4d4d, 0x1002) );
        //de.SetByteValue(ptr, ptr_len);
        //de.SetVR( gdcm::VR::OB );

        // Create an item
        gdcm::Item it;
        it.SetVLToUndefined();
        //gdcm::DataSet &nds = it.GetNestedDataSet();
        //nds.Insert(owner);
    }
}

```

```

        //nds.Insert(de);

        sq->AddItem(it);
    }

    // Insert sequence into data set
    gdcm::DataElement des( gdcm::Tag(0x4d4d,0x1001) );
    des.SetVR(gdcm::VR::SQ);
    des.SetValue(*sq);
    des.SetVLToUndefined();

    ds.Insert(owner);
    ds.Insert(des);

    gdcm::Writer w;
    w.SetFile( file );
    //w.SetCheckFileMetaInformation( true );
    w.SetFileName( outfilename );
    if (!w.Write() )
    {
        return 1;
    }

    return 0;
}

```

## 12.73 GenSeqs.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmFile.h"
#include "gdcmTag.h"

/*
 * This example is used to generate the file:
 *
 * gdcmConformanceTests/SequenceWithUndefinedLengthNotConvertibleToDefinedLength.dcm
 *
 * There is a flaw in the DICOM design where it is assumed that Sequence can be
 * either represented as undefined length or defined length. This should work
 * in most case, but the undefined length is a little more general and can
 * store sequence of items that a defined length cannot.
 * Deflated syntax was used in this case since this synthetic example can be
 * nicely compressed using this transfer syntax.
 *
 * Warning: do not try to compute the group length elements !
 * Warning: You may need a 64bits machine for this example to work.
 */
int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {

```

```

    return 1;
}

gdcmm::File &file = reader.GetFile();
gdcmm::DataSet &ds = file.GetDataSet();

//const unsigned int nitems = 1000;
const unsigned int ptr_len = 42; /*94967296 / nitems; */
//assert( ptr_len == 42949672 );
char *ptr = new char[ptr_len];
memset(ptr,0,ptr_len);

// Create a Sequence
gdcmm::SmartPointer<gdcmm::SequenceOfItems> sq = new gdcmm::SequenceOfItems();
sq->SetLengthToUndefined();

const char owner_str[] = "GDCM CONFORMANCE TESTS";
gdcmm::DataElement owner( gdcmm::Tag(0x4d4d, 0x10) );
owner.SetByteValue(owner_str, (uint32_t)strlen(owner_str));
owner.SetVR( gdcmm::VR::LO );

for(unsigned int idx = 0; idx < 10/* nitems*/; ++idx)
{
    // Create a dataelement
    gdcmm::DataElement de( gdcmm::Tag(0x4d4d, 0x1002) );
    de.SetByteValue(ptr, ptr_len);
    de.SetVR( gdcmm::VR::OB );

    // Create an item
    gdcmm::Item it;
    it.SetVLToUndefined();
    gdcmm::DataSet &nds = it.GetNestedDataSet();
    nds.Insert(owner);
    nds.Insert(de);

    sq->AddItem(it);
}

// Insert sequence into data set
gdcmm::DataElement des( gdcmm::Tag(0x4d4d,0x1001) );
des.SetVR(gdcmm::VR::SQ);
des.SetValue(*sq);
des.SetVLToUndefined();

ds.Insert(owner);
ds.Insert(des);

gdcmm::Writer w;
w.SetFile( file );
//w.SetCheckFileMetaInformation( true );
w.SetFileName( outfilename );
if (!w.Write() )
{
    return 1;
}

return 0;
}

```

## 12.74 GenerateStandardSOPClasses.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
*/

#include "gdcmmDefs.h"

```

```

#include "gdcmUIDs.h"
#include "gdcmGlobal.h"
#include "gdcmMediaStorage.h"
#include "gdcmSOPClassUIDToIOD.h"

int main(int , char *[])
{
    using gdcm::MediaStorage;
    gdcm::Global& g = gdcm::Global::GetInstance();
    if( !g.LoadResourcesFiles() )
    {
        std::cerr << "Could not LoadResourcesFiles" << std::endl;
        return 1;
    }

    const gdcm::Defs &defs = g.GetDefs();

    int ret = 0;

    //std::cout << "Table B.5-1 STANDARD SOP CLASSES" << std::endl;
    std::cout << "SOP Class Name,SOP Class UID,IOD Specification (defined in PS 3.3)" << std::endl;

    gdcm::MediaStorage::MSType mst;
    for ( mst = gdcm::MediaStorage::MediaStorageDirectoryStorage; mst < gdcm::MediaStorage::MS_END;
          mst = (gdcm::MediaStorage::MSType)(mst + 1) )
    {
        const char *iod = defs.GetIODNameFromMediaStorage(mst);
        gdcm::UIDs uid;
        uid.SetFromUID( gdcm::MediaStorage::GetMSString(mst) /*mst.GetString()*/ );
        if( iod )
        {
            const char *iod_ref = gdcm::SOPClassUIDToIOD::GetIOD(uid);
            if( iod_ref )
            {
                std::string iod_ref_str = iod_ref;
                //iod_ref_str += " IOD Modules";
                //if( iod_ref_str != iod )
                {
                    //std::cout << "UID: " << uid << " ";
                    std::cout << "'" << uid.GetName() << "' << " << "'" << uid.GetString() << "' << " << "'" << iod << "' <<
                    std::endl;
                    //std::cout << "Incompatible IODs: [" << iod << "] versus ref= [" << iod_ref_str << "]" << std::endl;
                    ++ret;
                }
            }
        }
    }

    return 0;
}

```

## 12.75 GetJPEGSamplePrecision.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * This example is a little helper to detect the famous SIEMENS JPEG lossless compressed image
 * where DICOM is declared as:
 *
 * (0028,0100) US 16 # 2,1 Bits Allocated
 * (0028,0101) US 12 # 2,1 Bits Stored
 * (0028,0102) US 11 # 2,1 High Bit
 * (0028,0103) US 0 # 2,1 Pixel Representation
 *
 */

```

```

* But where JPEG is:
*
*       JPEG_SOF_Parameters:
*           SamplePrecision = 16
*           nLines = 192
*           nSamplesPerLine = 192
*           nComponentsInFrame = 1
*           component 0
*               ComponentIdentifier = 1
*               HorizontalSamplingFactor = 1
*               VerticalSamplingFactor = 1
*               QuantizationTableDestinationSelector = 0
*
*
* This case is valid. One simply has to use the 16bits jpeg decoder to decode the 12bits stored image.
* This used to be an issue in GDCM 1.2.x (fixed in GDCM 1.2.5)
*
* The main return 0 (no error) when the file read is actually a potential problem. At the end of the main
* function, the jpeg stream is stored in the filename specified as second argument
*/

#include "gdcmImageReader.h"
#include "gdcmSequenceOfFragments.h"
#include "gdcmJPEGCodec.h"

#include <iostream>
#include <fstream>

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.jpg" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    gdcm::ImageReader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Could not read: " << filename << std::endl;
        return 1;
    }

    // The output of gdcm::Reader is a gdcm::File
    const gdcm::File &file = reader.GetFile();
    const gdcm::Image &image = reader.GetImage();

    const gdcm::TransferSyntax &ts = file.GetHeader().GetDataSetTransferSyntax();

    if( ts != gdcm::TransferSyntax::JPEGLosslessProcess14 && ts != gdcm::TransferSyntax::JPEGLosslessProcess14_1 )
    {
        std::cerr << "Input is not a lossless JPEG" << std::endl;
        return 1;
    }

    // the dataset is the the set of element we are interested in:
    const gdcm::DataSet &ds = file.GetDataSet();

    const gdcm::Tag rawTag(0x7fe0, 0x0010); // Default to Pixel Data
    const gdcm::DataElement &pdde = ds.GetDataElement( rawTag );
    const gdcm::SequenceOfFragments *sf = pdde.GetSequenceOfFragments();
    if( sf )
    {
        std::ofstream output(outfilename, std::ios::binary);
        sf->WriteBuffer(output);
    }
    else
    {
        std::cerr << "Error" << std::endl;
        return 1;
    }

    gdcm::JPEGCodec jpeg;
    std::ifstream is(outfilename, std::ios::binary);
    gdcm::PixelFormat pf ( gdcm::PixelFormat::UINT8 ); // let's pretend it's a 8bits jpeg
    jpeg.SetPixelFormat( pf );
    gdcm::TransferSyntax ts_jpg;
    bool b = jpeg.GetHeaderInfo( is, ts_jpg );

```



```

if( !b )
{
    return 1;
}

//jpeg.Print( std::cout );
if( jpeg.GetPixelFormat().GetBitsAllocated() != image.GetPixelFormat().GetBitsAllocated()
|| jpeg.GetPixelFormat().GetBitsStored() != image.GetPixelFormat().GetBitsStored() )
{
    std::cerr << "There is a mismatch in between DICOM declared Pixel Format and Sample Precision used in the
    JPEG stream" << std::endl;
    return 0;
}

std::cout << jpeg.GetPixelFormat() << std::endl;
std::cout << image.GetPixelFormat() << std::endl;

return 1;
}

```

## 12.76 GetSequenceUltrasound.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
#include "gdcmReader.h"
#include "gdcmAttribute.h"

bool Region ( char* nomefile, unsigned int* X_min, unsigned int* Y_min, unsigned int* X_max, unsigned int* Y_max
);

int main(int argc, char* argv[] )
{
    // Controllo del numero di argomenti introdotti da riga di comando
    if( argc < 2 )
    {
        std::cerr << "Usage: " << std::endl;
        std::cerr << argv[0] << " inputImageFile " << std::endl;
        return EXIT_FAILURE;
    }

    unsigned int x_min = 1;
    unsigned int y_min = 1;
    unsigned int x_max = 1;
    unsigned int y_max = 1;

    if( Region ( argv[1], &x_min, &y_min, &x_max, &y_max ) )
    {
        std::cout << "x_min = " << x_min << std::endl;
        std::cout << "y_min = " << y_min << std::endl;
        std::cout << "x_max = " << x_max << std::endl;
        std::cout << "y_max = " << y_max << std::endl;
    }

    else
    {
        std::cout << "no\n";
    }
}

bool Region ( char* nomefile, unsigned int* X_min, unsigned int* Y_min, unsigned int* X_max, unsigned int* Y_max
)
{
    gdcm::Reader reader;
    reader.SetFileName( nomefile );

```

```

if( !reader.Read() )
{
    std::cerr << "Could not read: " << nomefile << std::endl;
    return false;
}

gdcm::File &file = reader.GetFile();
gdcm::DataSet &ds = file.GetDataSet();

gdcm::Tag tsqr(0x0018,0x6011);
if( !ds.FindDataElement( tsqr ) )
{
    return false;
}

const gdcm::DataElement &sqr= ds.GetDataElement( tsqr );
//std::cout << sqr << std::endl;
const gdcm::SequenceOfItems *sqi = sqr.GetValueAsSQ();
if( !sqi || !sqi->GetNumberOfItems() )
{
    return false;
}
//std::cout << sqi << std::endl;

const gdcm::Item &item = sqi->GetItem(1);
//std::cout << item << std::endl;
const gdcm::DataSet& nestedds = item.GetNestedDataSet();
//std::cout << nestedds << std::endl;

gdcm::Tag tX0(0x0018,0x6018);
gdcm::Tag tY0(0x0018,0x601a);
gdcm::Tag tX1(0x0018,0x601c);
gdcm::Tag tY1(0x0018,0x601e);

if( (!nestedds.FindDataElement( tX0 ))||(!nestedds.FindDataElement( tY0 ))||(!nestedds.FindDataElement( tX1
))||(!nestedds.FindDataElement( tY1 )) )
{
    return false;
}

const gdcm::DataElement& deX0 = nestedds.GetDataElement( tX0 );
const gdcm::DataElement& deY0 = nestedds.GetDataElement( tY0 );
const gdcm::DataElement& deX1 = nestedds.GetDataElement( tX1 );
const gdcm::DataElement& deY1 = nestedds.GetDataElement( tY1 );
//std::cout << deX0 << std::endl << deY0 << std::endl << deX1 << std::endl << deY1 << std::endl;

//const gdcm::ByteValue *bvX0 = deX0.GetByteValue();
//const gdcm::ByteValue *bvY0 = deY0.GetByteValue();
//const gdcm::ByteValue *bvX1 = deX1.GetByteValue();
//const gdcm::ByteValue *bvY1 = deY1.GetByteValue();
//std::cout << bvX0 << std::endl << bvY0 << std::endl << bvX1 << std::endl << bvY1 << std::endl;

gdcm::Attribute<0x0018,0x6018> atX0;
gdcm::Attribute<0x0018,0x601a> atY0;
gdcm::Attribute<0x0018,0x601c> atX1;
gdcm::Attribute<0x0018,0x601e> atY1;
atX0.SetFromDataElement( deX0 );
atY0.SetFromDataElement( deY0 );
atX1.SetFromDataElement( deX1 );
atY1.SetFromDataElement( deY1 );
uint32_t X0 = atX0.GetValue();
uint32_t Y0 = atY0.GetValue();
uint32_t X1 = atX1.GetValue();
uint32_t Y1 = atY1.GetValue();
std::cout << X0 << std::endl << Y0 << std::endl << X1 << std::endl << Y1 << std::endl;

*X_min = static_cast<unsigned int>(X0);
*Y_min = static_cast<unsigned int>(Y0);
*X_max = static_cast<unsigned int>(X1);
*Y_max = static_cast<unsigned int>(Y1);

//std::cout << "X_min = " << *X_min << std::endl;
//std::cout << "Y_min = " << *Y_min << std::endl;
//std::cout << "X_max = " << *X_max << std::endl;
//std::cout << "Y_max = " << *Y_max << std::endl;

return true;
}

```

## 12.77 GetSubSequenceData.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmImage.h"
#include "gdcmImageWriter.h"
#include "gdcmDataElement.h"
#include "gdcmPrivateTag.h"
#include "gdcmUIDGenerator.h"

#include <iostream>
#include <string>

#include <map>

/*
 * This example will extract the Movie from the private group of
 * GEMS_Ultrasound_MovieGroup_001 See Attribute
 * (7fel,60,GEMS_Ultrasound_MovieGroup_001)
 *
 * The output file will be stored in `outvid.dcm` as
 * MultiframeGrayscaleByteSecondaryCaptureImageStorage
 */
int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    using namespace gdcm;
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    reader.Read();

    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();
    const PrivateTag tseq(0x7fel,0x1,"GEMS_Ultrasound_MovieGroup_001");

    if( !ds.FindDataElement( tseq ) ) return 1;
    const DataElement& seq = ds.GetDataElement( tseq );

    SmartPointer<SequenceOfItems> sqi = seq.GetValueAsSQ();
    assert( sqi->GetNumberOfItems() == 1 );
    Item &item = sqi->GetItem(1);
    DataSet &subds = item.GetNestedDataSet();

    const PrivateTag tseq1(0x7fel,0x10,"GEMS_Ultrasound_MovieGroup_001");

    if( !subds.FindDataElement( tseq1 ) ) return 1;
    const DataElement& seq1 = subds.GetDataElement( tseq1 );

    SmartPointer<SequenceOfItems> sqi2 = seq1.GetValueAsSQ();
    //int n = sqi2->GetNumberOfItems();
    int index = 1;
    Item &item2 = sqi2->GetItem(index);
    DataSet &subds2 = item2.GetNestedDataSet();

    const PrivateTag tseq2(0x7fel,0x20,"GEMS_Ultrasound_MovieGroup_001");

    if( !subds2.FindDataElement( tseq2 ) ) return 1;
    const DataElement& seq2 = subds2.GetDataElement( tseq2 );

    //    std::cout << seq2 << std::endl;

    SmartPointer<SequenceOfItems> sqi3 = seq2.GetValueAsSQ();
    size_t ni3 = sqi3->GetNumberOfItems(); (void)ni3;
    assert( sqi3->GetNumberOfItems() >= 1 );
    Item &item3 = sqi3->GetItem(1);

```

```

DataSet &subds3 = item3.GetNestedDataSet();

const PrivateTag tseq6(0x7fel,0x26,"GEMS_Ultrasound_MovieGroup_001");
if( !subds3.FindDataElement( tseq6 ) ) return true;
const DataElement& seq6 = subds3.GetDataElement( tseq6 );
SmartPointer<SequenceOfItems> sqi6 = seq6.GetValueAsSQ();
size_t ni6= sqi6->GetNumberOfItems();
assert( sqi6->GetNumberOfItems() >= 1 );
const PrivateTag tseq7(0x7fel,0x86,"GEMS_Ultrasound_MovieGroup_001");
int dimx = 0, dimy = 0;
for( size_t i6 = 1; i6 <= ni6; ++i6 )
{
    Item &item6 = sqi6->GetItem(i6);
    DataSet &subds6 = item6.GetNestedDataSet();

    if( subds6.FindDataElement( tseq7 ) )
    {
        Element<VR::SL, VM::VM4> el;
        el.SetFromDataElement( subds6.GetDataElement( tseq7 ) );
        std::cout << "El= " << el.GetValue() << std::endl;
        dimx = el.GetValue(0);
        dimy = el.GetValue(1);
    }
}

const PrivateTag tseq3(0x7fel,0x36,"GEMS_Ultrasound_MovieGroup_001");
if( !subds3.FindDataElement( tseq3 ) ) return true;
const DataElement& seq3 = subds3.GetDataElement( tseq3 );

//      std::cout << seq3 << std::endl;

SmartPointer<SequenceOfItems> sqi4 = seq3.GetValueAsSQ();
size_t ni4= sqi4->GetNumberOfItems();
assert( sqi4->GetNumberOfItems() >= 1 );
const PrivateTag tseq8(0x7fel,0x37,"GEMS_Ultrasound_MovieGroup_001");
const PrivateTag tseq4(0x7fel,0x43,"GEMS_Ultrasound_MovieGroup_001");
const PrivateTag tseq5(0x7fel,0x60,"GEMS_Ultrasound_MovieGroup_001");

std::vector<char> imbuffer;
int dimz = 0;
for( size_t i4 = 1; i4 <= ni4; ++i4 )
{
    Item &item4 = sqi4->GetItem(i4);
    DataSet &subds4 = item4.GetNestedDataSet();

    if( !subds4.FindDataElement( tseq8 ) ) return true;
    const DataElement& de8 = subds4.GetDataElement( tseq8 );
    Element<VR::UL, VM::VM1> ldimz;
    ldimz.SetFromDataElement( de8 );
    dimz += ldimz.GetValue();
    if( !subds4.FindDataElement( tseq4 ) ) return true;
    const DataElement& seq4 = subds4.GetDataElement( tseq4 );
    if( !subds4.FindDataElement( tseq5 ) ) return true;
    const DataElement& seq5 = subds4.GetDataElement( tseq5 );

    //      std::cout << seq4 << std::endl;
    //      std::cout << seq5 << std::endl;

    const ByteValue *bv4 = seq4.GetByteValue();
    (void)bv4;
#ifdef 0
    {
        std::ofstream out( "/tmp/mo4", std::ios::binary );
        out.write( bv4->GetPointer(), bv4->GetLength());
        out.close();
    }
#endif
    const ByteValue *bv5 = seq5.GetByteValue();
#ifdef 0
    {
        std::ofstream out( "/tmp/mo5", std::ios::binary );
        out.write( bv5->GetPointer(), bv5->GetLength());
        out.close();
    }
#endif

    std::cout << bv5->GetLength() << std::endl;
    imbuffer.insert( imbuffer.begin(), bv5->GetPointer(), bv5->GetPointer() + bv5->GetLength() );
}
DataElement fakedata;

```

```

fakedata.SetByteValue( imbuffer.data(), (uint32_t)imbuffer.size() );

gdcmm::SmartPointer<gdcmm::Image> im = new gdcmm::Image;
im->SetNumberOfDimensions( 3 );

im->SetDimension(0, dimx );
im->SetDimension(1, dimy );
im->SetDimension(2, dimz );
size_t l1 = imbuffer.size();
(void)l1;
size_t l2 = im->GetBufferLength();
(void)l2;
assert( im->GetBufferLength() == imbuffer.size() );
im->SetPhotometricInterpretation( gdcmm::PhotometricInterpretation::MONOCHROME2 );

im->SetDataElement( fakedata );

gdcmm::ImageWriter w;
w.SetImage( *im );
DataSet &dataset = w.GetFile().GetDataSet();

gdcmm::UIDGenerator uid;
gdcmm::DataElement de( Tag(0x8,0x18) ); // SOP Instance UID
de.SetVR( VR::UI );
const char *u = uid.Generate();
de.SetByteValue( u, (uint32_t)strlen(u) );
//ds.Insert( de );
dataset.Replace( de );

de.SetTag( Tag(0x8,0x16) ); // SOP Class UID
de.SetVR( VR::UI );
gdcmm::MediaStorage ms(
    gdcmm::MediaStorage::MultiframeGrayscaleByteSecondaryCaptureImageStorage );
de.SetByteValue( ms.GetString(), (uint32_t)strlen(ms.GetString()) );
dataset.Replace( de ); // replace !

w.SetFileName( "outvid.dcm" );
if( !w.Write() )
{
    return 1;
}

return 0;
}

```

## 12.78 HelloVizWorld.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * Basic example for dealing with a DICOM file that contains an Image
 * (read: Pixel Data element)
 */

#include "gdcmmImageReader.h"
#include "gdcmmImageWriter.h"
#include "gdcmmImage.h"
#include "gdcmmPhotometricInterpretation.h"

#include <iostream>

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {

```

```

    std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
    return 1;
}
const char *filename = argv[1];
const char *outfilename = argv[2];

// Instantiate the image reader:
gdcm::ImageReader reader;
reader.SetFileName( filename );
if( !reader.Read() )
{
    std::cerr << "Could not read: " << filename << std::endl;
    return 1;
}
// If we reach here, we know for sure 2 things:
// 1. It is a valid DICOM
// 2. And it contains an Image !

// The output of superclass gdcm::Reader is a gdcm::File
//gdcm::File &file = reader.GetFile();

// The other output of gdcm::ImageReader is a gdcm::Image
const gdcm::Image &image = reader.GetImage();

// Let's get some property from the image:
unsigned int ndim = image.GetNumberOfDimensions();
// Dimensions of the image:
const unsigned int *dims = image.GetDimensions();
// Origin
const double *origin = image.GetOrigin();
const gdcm::PhotometricInterpretation &pi = image.GetPhotometricInterpretation();
for(unsigned int i = 0; i < ndim; ++i)
{
    std::cout << "Dim(" << i << "): " << dims[i] << std::endl;
}
for(unsigned int i = 0; i < ndim; ++i)
{
    std::cout << "Origin(" << i << "): " << origin[i] << std::endl;
}
std::cout << "PhotometricInterpretation: " << pi << std::endl;

// Write the modified DataSet back to disk
gdcm::ImageWriter writer;
writer.SetImage( image );
writer.SetFileName( outfile );
//writer.SetFile( file ); // We purposely NOT copy the meta information from the input
// file, and instead only pass the image
if( !writer.Write() )
{
    std::cerr << "Could not write: " << outfile << std::endl;
    return 1;
}

return 0;
}

```

## 12.79 HelloWorld.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * This example is ... guess what this is for :)
 */
#include "gdcmReader.h"
#include "gdcmWriter.h"

```

```

#include "gdcmAttribute.h"

#include <iostream>

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    // Instantiate the reader:
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Could not read: " << filename << std::endl;
        return 1;
    }

    // If we reach here, we know for sure only 1 thing:
    // It is a valid DICOM file (potentially an old ACR-NEMA 1.0/2.0 file)
    // (Maybe, it's NOT a Dicom image -could be a DICOMDIR, a RTSTRUCT, etc-)

    // The output of gdcm::Reader is a gdcm::File
    gdcm::File &file = reader.GetFile();

    // the dataset is the the set of element we are interested in:
    gdcm::DataSet &ds = file.GetDataSet();

    // Construct a static(*) type for Image Comments :
    gdcm::Attribute<0x0020,0x4000> imagecomments;
    imagecomments.SetValue( "Hello, World !" );

    // Now replace the Image Comments from the dataset with our:
    ds.Replace( imagecomments.GetAsDataElement() );

    // Write the modified DataSet back to disk
    gdcm::Writer writer;
    writer.CheckFileMetaInformationOff(); // Do not attempt to reconstruct the file meta to preserve the file
                                         // as close to the original as possible.
    writer.SetFileName( outfile );
    writer.SetFile( file );
    if( !writer.Write() )
    {
        std::cerr << "Could not write: " << outfile << std::endl;
        return 1;
    }

    return 0;
}

/*
 * (*) static type, means that extra DICOM information VR & VM are computed at compilation time.
 * The compiler is deducing those values from the template arguments of the class.
 */

```

## 12.80 LargeVRDSExplicit.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmWriter.h"

```

```

#include "gdcmAttribute.h"
#include "gdcmFileExplicitFilter.h"
#include "gdcmSequenceOfItems.h"

bool interpolate(const double * pts, size_t npts, std::vector<double> &out )
{
    out.clear();
    for(size_t i = 0; i < 2*npts; ++i )
    {
        const size_t j = i / 2;
        if( i % 2 )
        {
            if( j != npts - 1 )
            {
                assert( 3*j+5 < 3*npts );
                const double midpointx = (pts[3*j+0] + pts[3*j+3]) / 2;
                const double midpointy = (pts[3*j+1] + pts[3*j+4]) / 2;
                const double midpointz = (pts[3*j+2] + pts[3*j+5]) / 2;
                out.push_back( midpointx );
                out.push_back( midpointy );
                out.push_back( midpointz );
            }
        }
        else
        {
            assert( j < npts );
            out.push_back( pts[3*j+0] );
            out.push_back( pts[3*j+1] );
            out.push_back( pts[3*j+2] );
        }
    }
    assert( out.size() == 2 * npts * 3 - 3 );
    return true;
}

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();

    gdcm::FileExplicitFilter fef;
    //fef.SetChangePrivateTags( changeprivatetags );
    fef.SetFile( reader.GetFile() );
    if( !fef.Change() )
    {
        std::cerr << "Failed to change: " << filename << std::endl;
        return 1;
    }

    // (3006,0039) SQ (Sequence with undefined length #=4)      # u/1, 1 ROIContourSequence
    gdcm::Tag tag(0x3006,0x0039);

    const gdcm::DataElement &roicsq = ds.GetDataElement( tag );
    gdcm::SmartPointer<gdcm::SequenceOfItems> sqi = roicsq.GetValueAsSQ();
    //sqi->SetNumberOfItems( 1 );
    const gdcm::Item & item = sqi->GetItem(1); // Item start at #1
    const gdcm::DataSet& nestedds = item.GetNestedDataSet();

    gdcm::Tag tcsq(0x3006,0x0040);
    if( !nestedds.FindDataElement( tcsq ) )
    {
        return 0;
    }
    const gdcm::DataElement& csq = nestedds.GetDataElement( tcsq );
    gdcm::SmartPointer<gdcm::SequenceOfItems> sqi2 = csq.GetValueAsSQ();
    if( !sqi2 || !sqi2->GetNumberOfItems() )
    {

```



```

    return 0;
}
//unsigned int nitems = sqi2->GetNumberOfItems();
gdcm::Item & item2 = sqi2->GetItem(1); // Item start at #1

gdcm::DataSet& nestedds2 = item2.GetNestedDataSet();
//item2.SetVLTToUndefined();
//std::cout << nestedds2 << std::endl;
// (3006,0050) DS [43.57636\65.52504\10.0\46.043102\62.564945\10.0\49.126537\60.714... # 398,48 ContourData
gdcm::Tag tcontourdata(0x3006,0x0050);
const gdcm::DataElement & contourdata = nestedds2.GetDataElement( tcontourdata );
//std::cout << contourdata << std::endl;

//const gdcm::ByteValue *bv = contourdata.GetByteValue();
gdcm::Attribute<0x3006,0x0046> ncontourpoints;
ncontourpoints.Set( nestedds2 );

gdcm::Attribute<0x3006,0x0050> at;
at.SetFromDataElement( contourdata );
const double* pts = at.GetValues();
unsigned int npts = at.GetNumberOfValues() / 3;

std::vector<double> out( pts, pts + npts * 3 );
std::vector<double> out2;

//const unsigned int niter = 7;
const unsigned int niter = 8;
for( unsigned int i = 0; i < niter; ++i)
{
    //bool b =
    interpolate(out.data(), out.size() / 3, out2);
    //const double *pout = &out[0];
    out = out2;
    out2.clear();
}
assert( out.size() % 3 == 0 );

gdcm::Attribute<0x3006,0x0050> at_interpolate;
at_interpolate.SetNumberOfValues( (unsigned int) out.size() / 3 );
at_interpolate.SetValues( out.data(), (uint32_t)out.size() );

ncontourpoints.SetValue( at_interpolate.GetNumberOfValues() / 3 );
nestedds2.Replace( at_interpolate.GetAsDataElement() );
nestedds2.Replace( ncontourpoints.GetAsDataElement() );

//assert(0);

// Let's take item one and subdivide it

gdcm::TransferSyntax ts = gdcm::TransferSyntax::ImplicitVRLittleEndian;
ts = gdcm::TransferSyntax::ExplicitVRLittleEndian;

gdcm::FileMetaInformation &fmi = file.GetHeader();
const char *tsuid = gdcm::TransferSyntax::GetTSString( ts );
// const char * is ok since padding is \0 anyway...
gdcm::DataElement de( gdcm::Tag(0x0002,0x0010) );
de.SetByteValue( tsuid, (uint32_t)strlen(tsuid) );
de.SetVR( gdcm::Attribute<0x0002, 0x0010>::GetVR() );
fmi.Replace( de );
fmi.Remove( gdcm::Tag(0x0002,0x0012) ); // will be regenerated
fmi.Remove( gdcm::Tag(0x0002,0x0013) ); // ' ' ' '
fmi.SetDataSetTransferSyntax(ts);

gdcm::Writer w;
w.SetFile( file );
w.SetFileName( outfilename );
if ( !w.Write() )
{
    return 1;
}

return 0;
}

```

## 12.81 MakeTemplate.cxx

```

/*=====

```

```

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmFileAnonymizer.h"
#include "gdcmReader.h"
#include "gdcmWriter.h"

int main(int argc, char *argv[])
{
    if( argc < 3 ) return 1;
    const char* filename = argv[1];
    const char* outfilename = argv[2];

    //gdcm::Trace::DebugOn();

    // Remove Pixel Data element:
    gdcm::FileAnonymizer fa;
    fa.SetInputFileName( filename );
    fa.SetOutputFileName( outfilename );

    fa.Empty( gdcm::Tag(0x7fe0,0x10) );
    // cannot replace in-place DICOM header:
    //fa.Replace( gdcm::Tag(0x2,0x2), "1.2.840.10008.5.1.4.1.1.7" );

    if( !fa.Write() )
    {
        std::cerr << "impossible to remove Pixel Data attribute" << std::endl;
        return 1;
    }

    // Update the DICOM Header:
    gdcm::Reader reader;
    reader.SetFileName( outfilename );
    if( !reader.Read() )
    {
        std::cerr << "could not read back" << std::endl;
        return 1;
    }

    gdcm::File & file = reader.GetFile();
    gdcm::FileMetaInformation &fmi = file.GetHeader();
    gdcm::TransferSyntax ts = gdcm::TransferSyntax::ImplicitVRLittleEndian;
    ts = gdcm::TransferSyntax::ExplicitVRLittleEndian;
    fmi.SetDataSetTransferSyntax(ts);

    gdcm::Writer writer;
    writer.SetFile( file );
    writer.SetFileName( outfilename ); // warning overwrite file !
    if( !writer.Write() )
    {
        std::cerr << "could not write back" << std::endl;
        return 1;
    }

    return 0;
}

```

## 12.82 MergeTwoFiles.cxx

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even

```

```

    the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
    PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * This example will show how one can read in two DICOM files, use the dataset
 * from file1 and use image from file2 to save it in a 3rd file.
 *
 * Eg:
 * MergeTwoFiles gdcmData/012345.002.050.dcm gdcmData/test.acr merge.dcm
 */

#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmWriter.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        return 1;
    }
    const char *file1 = argv[1];
    const char *file2 = argv[2];
    const char *file3 = argv[3];

    // Read file1
    gdcm::ImageReader reader1;
    reader1.SetFileName( file1 );
    if( !reader1.Read() )
    {
        return 1;
    }

    // Read file2
    gdcm::ImageReader reader2;
    reader2.SetFileName( file2 );
    if( !reader2.Read() )
    {
        return 1;
    }

    // Ok now let's take the DataSet from file1 and the Image from file2
    // Warning: if file2 is -for example- a Secondary Capture Storage, then it has no
    // Image Orientation (Patient) thus any Image Orientation (Patient) from file1
    // will be discarded...

    // let's be fancy. In case reader2 contains explicit, but reader1 is implicit
    // we would rather see an implicit output
    if( reader1.GetFile().GetHeader().GetDataSetTransferSyntax() == gdcm::TransferSyntax::ImplicitVRLittleEndian )
    {
        reader2.GetImage().SetTransferSyntax( gdcm::TransferSyntax::ImplicitVRLittleEndian );
    }

    gdcm::ImageWriter writer;
    writer.SetFileName( file3 );
    writer.SetFile( reader1.GetFile() );
    // ImageWriter will always use all of gdcm::Image information an override anything wrong from
    // reader1.GetFile(), including the Transfer Syntax
    writer.SetImage( reader2.GetImage() );

    gdcm::DataSet &ds = reader1.GetFile().GetDataSet();

    // Make sure that SOPInstanceUID are different
    // Simply removing it is sufficient as gdcm::ImageWriter will generate one by default
    // if not found.
    ds.Remove( gdcm::Tag(0x0008,0x0018) );
    if( !writer.Write() )
    {
        return 1;
    }

    return 0;
}

```

## 12.83 MrProtocol.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 *
 */

/*
28 - 'MrProtocol' VM 1, VR UN, SyngoDT 0, NoOfItems 6, Data '### ASCCONV BEGIN ###
ulVersion = 0xbee332
tSequenceFileName = "%SiemensSeq%\fl_fq_shphs"
tProtocolName = "flash+AF8-100+AF8-through-plane+AF8-V"
tReferenceImage0 = "1.3.12.2.1107.5.2.9.16041.30000007062106100181200004658"
tReferenceImage1 = "1.3.12.2.1107.5.2.9.16041.30000007062106100181200004635"
tReferenceImage2 = "1.3.12.2.1107.5.2.9.16041.30000007062106100181200004683"
ucScanRegionPosValid = 0x1
sProtConsistencyInfo.tBaselineString = "N4_VB11A_LATEST_20031004"
sProtConsistencyInfo.flNominalB0 = 1.494
sProtConsistencyInfo.flGMax = 22
sProtConsistencyInfo.flRiseTime = 10
sGRADSPEC.sEddyCompensationX.aflAmplitude[0] = 0.0141111
sGRADSPEC.sEddyCompensationX.aflAmplitude[1] = 0.057038
sGRADSPEC.sEddyCompensationX.aflAmplitude[2] = -0.00986504
sGRADSPEC.sEddyCompensationX.aflAmplitude[3] = 0.00247627
sGRADSPEC.sEddyCompensationX.aflAmplitude[4] = 0.0026377
sGRADSPEC.sEddyCompensationX.aflTimeConstant[0] = 1.53826
sGRADSPEC.sEddyCompensationX.aflTimeConstant[1] = 0.746617
sGRADSPEC.sEddyCompensationX.aflTimeConstant[2] = 0.339236
sGRADSPEC.sEddyCompensationX.aflTimeConstant[3] = 0.0309809
sGRADSPEC.sEddyCompensationX.aflTimeConstant[4] = 0.00067694
sGRADSPEC.sEddyCompensationY.aflAmplitude[0] = 0.0156411
sGRADSPEC.sEddyCompensationY.aflAmplitude[1] = 0.0440623
sGRADSPEC.sEddyCompensationY.aflAmplitude[2] = -0.00782663
sGRADSPEC.sEddyCompensationY.aflAmplitude[3] = 0.00186828
sGRADSPEC.sEddyCompensationY.aflAmplitude[4] = 0.00154504
sGRADSPEC.sEddyCompensationY.aflTimeConstant[0] = 1.47145
sGRADSPEC.sEddyCompensationY.aflTimeConstant[1] = 0.750538
sGRADSPEC.sEddyCompensationY.aflTimeConstant[2] = 0.339397
sGRADSPEC.sEddyCompensationY.aflTimeConstant[3] = 0.0312962
sGRADSPEC.sEddyCompensationY.aflTimeConstant[4] = 0.000895133
sGRADSPEC.sEddyCompensationZ.aflAmplitude[0] = 0.00618504
sGRADSPEC.sEddyCompensationZ.aflAmplitude[1] = 0.00313121
sGRADSPEC.sEddyCompensationZ.aflAmplitude[2] = 0.000289346
sGRADSPEC.sEddyCompensationZ.aflAmplitude[3] = -0.00019677
sGRADSPEC.sEddyCompensationZ.aflAmplitude[4] = 7.66445e-005
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[0] = 3.37462
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[1] = 0.999351
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[2] = 0.0174646
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[3] = 0.0110094
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[4] = 0.00199922
sGRADSPEC.bEddyCompensationValid = 1
sGRADSPEC.sB0CompensationX.aflAmplitude[0] = 0.307474
sGRADSPEC.sB0CompensationX.aflAmplitude[1] = 0.029337
sGRADSPEC.sB0CompensationX.aflAmplitude[2] = -0.187118
sGRADSPEC.sB0CompensationX.aflTimeConstant[0] = 0.98583
sGRADSPEC.sB0CompensationX.aflTimeConstant[1] = 0.0308443
sGRADSPEC.sB0CompensationX.aflTimeConstant[2] = 0.000466792
sGRADSPEC.sB0CompensationY.aflAmplitude[0] = 0.365257
sGRADSPEC.sB0CompensationY.aflAmplitude[1] = -0.318647
sGRADSPEC.sB0CompensationY.aflAmplitude[2] = -0.0118978
sGRADSPEC.sB0CompensationY.aflTimeConstant[0] = 0.61535
sGRADSPEC.sB0CompensationY.aflTimeConstant[1] = 0.488831
sGRADSPEC.sB0CompensationY.aflTimeConstant[2] = 0.00199991
sGRADSPEC.sB0CompensationZ.aflAmplitude[0] = -0.44647
sGRADSPEC.sB0CompensationZ.aflAmplitude[1] = -0.0455154
sGRADSPEC.sB0CompensationZ.aflAmplitude[2] = -0.0304901
sGRADSPEC.sB0CompensationZ.aflTimeConstant[0] = 0.959231
sGRADSPEC.sB0CompensationZ.aflTimeConstant[1] = 0.0720189

```

```
sGRADSPEC.sB0CompensationZ.aflTimeConstant[2] = 0.00190141
sGRADSPEC.sB0CompensationValid = 1
sGRADSPEC.sCrossTermCompensationXY.aflAmplitude[0] = 0.00105046
sGRADSPEC.sCrossTermCompensationXY.aflTimeConstant[0] = 0.842014
sGRADSPEC.sCrossTermCompensationXZ.aflAmplitude[0] = -0.00150189
sGRADSPEC.sCrossTermCompensationXZ.aflTimeConstant[0] = 0.736169
sGRADSPEC.sCrossTermCompensationYX.aflAmplitude[0] = -5.5278e-005
sGRADSPEC.sCrossTermCompensationYX.aflTimeConstant[0] = 0.228697
sGRADSPEC.sCrossTermCompensationYZ.aflAmplitude[0] = 0.000307999
sGRADSPEC.sCrossTermCompensationYZ.aflTimeConstant[0] = 1.19431
sGRADSPEC.sCrossTermCompensationZX.aflAmplitude[0] = -0.000286868
sGRADSPEC.sCrossTermCompensationZX.aflTimeConstant[0] = 0.665979
sGRADSPEC.sCrossTermCompensationZY.aflAmplitude[0] = 0.000355175
sGRADSPEC.sCrossTermCompensationZY.aflTimeConstant[0] = 0.844189
sGRADSPEC.bCrossTermCompensationValid = 1
sGRADSPEC.lOffsetX = 25
sGRADSPEC.lOffsetY = 84
sGRADSPEC.lOffsetZ = 47
sGRADSPEC.bOffsetValid = 1
sGRADSPEC.lDelayX = 12
sGRADSPEC.lDelayY = 11
sGRADSPEC.lDelayZ = 9
sGRADSPEC.bDelayValid = 1
sGRADSPEC.flSensitivityX = 0.000264087
sGRADSPEC.flSensitivityY = 0.000272009
sGRADSPEC.flSensitivityZ = 0.000272677
sGRADSPEC.bSensitivityValid = 1
sGRADSPEC.alShimCurrent[0] = 183
sGRADSPEC.alShimCurrent[1] = -25
sGRADSPEC.alShimCurrent[2] = -85
sGRADSPEC.alShimCurrent[3] = 378
sGRADSPEC.alShimCurrent[4] = 82
sGRADSPEC.bShimCurrentValid = 1
sGRADSPEC.ucMode = 0x2
sTXSPEC.asNucleusInfo[0].tNucleus = "1H"
sTXSPEC.asNucleusInfo[0].lFrequency = 63684693
sTXSPEC.asNucleusInfo[0].bFrequencyValid = 1
sTXSPEC.asNucleusInfo[0].flReferenceAmplitude = 359.734
sTXSPEC.asNucleusInfo[0].bReferenceAmplitudeValid = 1
sTXSPEC.asNucleusInfo[0].flAmplitudeCorrection = 1
sTXSPEC.asNucleusInfo[0].bAmplitudeCorrectionValid = 1
sTXSPEC.asNucleusInfo[1].bFrequencyValid = 1
sTXSPEC.asNucleusInfo[1].bReferenceAmplitudeValid = 1
sTXSPEC.asNucleusInfo[1].bAmplitudeCorrectionValid = 1
sTXSPEC.arFPULSE[0].tName = "03GreFCE"
sTXSPEC.arFPULSE[0].bAmplitudeValid = 0x1
sTXSPEC.arFPULSE[0].flAmplitude = 147.095
sTXSPEC.arFPULSE[1].tName = "02GreFCE"
sTXSPEC.arFPULSE[1].bAmplitudeValid = 0x1
sTXSPEC.arFPULSE[1].flAmplitude = 147.095
sTXSPEC.arFPULSE[2].tName = "01GreFCE"
sTXSPEC.arFPULSE[2].bAmplitudeValid = 0x1
sTXSPEC.arFPULSE[2].flAmplitude = 147.095
sTXSPEC.lNoOfTraPulses = 3
sTXSPEC.lBTB1ParallelCapacity = 2
sTXSPEC.lBTB1SerialCapacity = 24
sTXSPEC.lBTB2ParallelCapacity = 2
sTXSPEC.lBTB2SerialCapacity = 26
sTXSPEC.bBTBValid = 1
sTXSPEC.flKDynMagnitudeMin = 0.5
sTXSPEC.flKDynMagnitudeMax = 1.5
sTXSPEC.flKDynMagnitudeClipLow = 0.96
sTXSPEC.flKDynMagnitudeClipHigh = 1.04
sTXSPEC.flKDynPhaseMax = 0.698132
sTXSPEC.flKDynPhaseClip = 0.174533
sTXSPEC.bKDynValid = 1
sTXSPEC.ucRFPulseType = 0x1
sTXSPEC.ucExcitMode = 0x1
sTXSPEC.ucSimultaneousExcitation = 0x1
sRXSPEC.lGain = 1
sRXSPEC.bGainValid = 1
sRXSPEC.aFFT_SCALE[0].lRxChannel = 1
sRXSPEC.aFFT_SCALE[0].flFactor = 1.06857
sRXSPEC.aFFT_SCALE[0].bValid = 1
sRXSPEC.aFFT_SCALE[1].lRxChannel = 2
sRXSPEC.aFFT_SCALE[1].flFactor = 1.07454
sRXSPEC.aFFT_SCALE[1].bValid = 1
sRXSPEC.aFFT_SCALE[2].lRxChannel = 3
sRXSPEC.aFFT_SCALE[2].flFactor = 1.06622
sRXSPEC.aFFT_SCALE[2].bValid = 1
sRXSPEC.aFFT_SCALE[3].lRxChannel = 4
```

```

sRXSPEC.aFFT_SCALE[3].flFactor      = 1.06524
sRXSPEC.aFFT_SCALE[3].bValid        = 1
sRXSPEC.aFFT_SCALE[4].lRxChannel    = 5
sRXSPEC.aFFT_SCALE[4].flFactor      = 0.982692
sRXSPEC.aFFT_SCALE[4].bValid        = 1
sRXSPEC.aFFT_SCALE[5].lRxChannel    = 6
sRXSPEC.aFFT_SCALE[5].flFactor      = 0.988603
sRXSPEC.aFFT_SCALE[5].bValid        = 1
sRXSPEC.aFFT_SCALE[6].lRxChannel    = 7
sRXSPEC.aFFT_SCALE[6].flFactor      = 0.981538
sRXSPEC.aFFT_SCALE[6].bValid        = 1
sRXSPEC.aFFT_SCALE[7].lRxChannel    = 8
sRXSPEC.aFFT_SCALE[7].flFactor      = 1.00856
sRXSPEC.aFFT_SCALE[7].bValid        = 1
sRXSPEC.bVariCapVoltagesValid      = 1
sRXSPEC.alDwellTime[0]              = 8500
sAdjFreSpec.ulMode                   = 0x1
sAdjFreSpec.ucAdjWithBC              = 0x1
sAdjTraSpec.ucAdjWithBC              = 0x1
sAdjShimSpec.ulMode                  = 0x1
sAdjShimSpec.ucAdjWithBC              = 0x1
sAdjWatSupSpec.ulMode                = 0x1
sAdjWatSupSpec.ucAdjWithBC              = 0x1
alTR[0]                              = 37000
lContrasts                           = 1
alTE[0]                              = 4000
acFlowComp[0]                       = 1
lCombinedEchoes                      = 1
sSliceArray.asSlice[0].sPosition.dSag = 35.31199581
sSliceArray.asSlice[0].sPosition.dCor = -8.387765754
sSliceArray.asSlice[0].sPosition.dTra = -23.13178296
sSliceArray.asSlice[0].sNormal.dSag   = 0.771051253
sSliceArray.asSlice[0].sNormal.dCor   = 0.5863890019
sSliceArray.asSlice[0].sNormal.dTra   = -0.2482496801
sSliceArray.asSlice[0].dThickness     = 6
sSliceArray.asSlice[0].dPhaseFOV      = 187.5
sSliceArray.asSlice[0].dReadoutFOV    = 250
sSliceArray.lSize                     = 1
sSliceArray.lSag                      = 1
sSliceArray.lConc                     = 1
sSliceArray.ucMode                     = 0x1
sSliceArray.sTSat.dThickness          = 40
sSliceArray.sTSat.dGap                 = 10
sGroupArray.asGroup[0].nSize          = 1
sGroupArray.asGroup[0].dDistFact      = 0.2
sGroupArray.anMember[1]               = -1
sGroupArray.lSize                     = 1
sGroupArray.sPSat.dThickness          = 50
sGroupArray.sPSat.dGap                 = 10
sAutoAlign.dAAMatrix[0]               = 1
sAutoAlign.dAAMatrix[5]               = 1
sAutoAlign.dAAMatrix[10]              = 1
sAutoAlign.dAAMatrix[15]              = 1
sNavigatorPara.ucRespComp              = 0x4
sPrepPulses.ucFatSat                  = 0x4
sPrepPulses.ucWaterSat                = 0x4
sPrepPulses.ucInversion                = 0x4
sPrepPulses.ucSatRecovery              = 0x1
sPrepPulses.ucFatSatMode               = 0x2
sKSpace.lBaseResolution                = 256
sKSpace.lPhaseEncodingLines            = 192
sKSpace.dPhaseResolution                = 1
sKSpace.lPartitions                    = 32
sKSpace.lImagesPerSlab                 = 32
sKSpace.dSliceResolution                = 1
sKSpace.ucPhasePartialFourier           = 0x10
sKSpace.ucSlicePartialFourier           = 0x10
sKSpace.ucAveragingMode                 = 0x2
sKSpace.ucMultiSliceMode                = 0x1
sKSpace.ucDimension                     = 0x2
sKSpace.ucAsymmetricEchoAllowed         = 0x1
sKSpace.unReordering                    = 0x1
sFastImaging.lEPIFactor                 = 1
sFastImaging.lTurboFactor               = 1
sFastImaging.lSegments                  = 3
sFastImaging.ulEnableRFSpoiling         = 0x1
sPhysioImaging.lSignal1                 = 2
sPhysioImaging.lMethod1                 = 2
sPhysioImaging.lSignal2                 = 1
sPhysioImaging.lMethod2                 = 1
sPhysioImaging.lPhases                  = 21

```

```

sPhysioImaging.lRetroGatedImages      = 16
sPhysioImaging.sPhysioECG.lScanWindow = 805
sPhysioImaging.sPhysioECG.lTriggerPulses = 1
sPhysioImaging.sPhysioECG.lTriggerWindow = 5
sPhysioImaging.sPhysioECG.lArrhythmiaDetection = 1
sPhysioImaging.sPhysioECG.lCardiacGateOnThreshold = 100000
sPhysioImaging.sPhysioECG.lCardiacGateOffThreshold = 700000
sPhysioImaging.sPhysioPulse.lTriggerPulses = 1
sPhysioImaging.sPhysioPulse.lTriggerWindow = 5
sPhysioImaging.sPhysioPulse.lCardiacGateOnThreshold = 100000
sPhysioImaging.sPhysioPulse.lCardiacGateOffThreshold = 700000
sPhysioImaging.sPhysioExt.lTriggerPulses = 1
sPhysioImaging.sPhysioExt.lTriggerWindow = 5
sPhysioImaging.sPhysioExt.lCardiacGateOnThreshold = 100000
sPhysioImaging.sPhysioExt.lCardiacGateOffThreshold = 700000
sPhysioImaging.sPhysioResp.lRespGateThreshold = 20
sPhysioImaging.sPhysioResp.lRespGatePhase = 2
sPhysioImaging.sPhysioResp.dGatingRatio = 0.3
sSpecPara.lPhaseCyclingType           = 1
sSpecPara.lPhaseEncodingType          = 1
sSpecPara.lRFExcitationBandwidth      = 1
sSpecPara.ucRemoveOversampling         = 0x1
sSpecPara.lDecouplingType             = 1
sSpecPara.lNOEType                    = 1
sSpecPara.lExcitationType              = 1
sSpecPara.lSpectralSuppression         = 1
sDiffusion.ulMode                     = 0x1
sAngio.sFlowArray.asElm[0].nVelocity  = 100
sAngio.sFlowArray.asElm[0].nDir       = 0x4
sAngio.sFlowArray.lSize               = 1
sAngio.ucPCFlowMode                   = 0x2
sAngio.ucTOFInflow                    = 0x4
sAngio.ucRephasedImage                = 0x1
sAngio.ucPhaseImage                   = 0x1
sEllipticalFilter.ucMode              = 0x1
sPat.lAccelFactPE                     = 1
sPat.lAccelFact3D                     = 1
sPat.ucPATMode                        = 0x1
sPat.ucRefScanMode                    = 0x1
ucAutoMovie                           = 0x1
ucDisableChangeStoreImages            = 0x1
ucReconstructionMode                  = 0x1
ucPHAPSMODE                           = 0x1
ucDixon                               = 0x1
lAverages                             = 2
adFlipAngleDegree[0]                  = 30
lScanTimeSec                           = 103
lTotalScanTimeSec                     = 112
dRefSNR                               = 165404.1473
dRefSNR_VOI                           = 165404.1473
tdefaultEVAProt                       = "%SiemensEvaDefProt%\Inline\Inline.evp"
tcurrentEVAProt                       = "%CURRENTEVAProt%\EVA2A5.tmp"
sCOIL_SELECT_MEAS.asList[0].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[0].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[0].sCoilElementID.tElement = "PP6"
sCOIL_SELECT_MEAS.asList[0].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[0].lRxChannelConnected = 1
sCOIL_SELECT_MEAS.asList[1].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[1].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[1].sCoilElementID.tElement = "PP5"
sCOIL_SELECT_MEAS.asList[1].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[1].lRxChannelConnected = 1
sCOIL_SELECT_MEAS.asList[2].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[2].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[2].sCoilElementID.tElement = "PP3"
sCOIL_SELECT_MEAS.asList[2].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[2].lRxChannelConnected = 2
sCOIL_SELECT_MEAS.asList[3].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[3].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[3].sCoilElementID.tElement = "PP4"
sCOIL_SELECT_MEAS.asList[3].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[3].lRxChannelConnected = 3
sCOIL_SELECT_MEAS.asList[4].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[4].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[4].sCoilElementID.tElement = "PP2"
sCOIL_SELECT_MEAS.asList[4].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[4].lRxChannelConnected = 4
sCOIL_SELECT_MEAS.asList[5].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[5].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[5].sCoilElementID.tElement = "PP1"
sCOIL_SELECT_MEAS.asList[5].lElementSelected = 1

```

```

sCOIL_SELECT_MEAS.asList[5].lRxChannelConnected = 4
sCOIL_SELECT_MEAS.asList[6].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[6].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[6].sCoilElementID.tElement = "PA6"
sCOIL_SELECT_MEAS.asList[6].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[6].lRxChannelConnected = 5
sCOIL_SELECT_MEAS.asList[7].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[7].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[7].sCoilElementID.tElement = "PA5"
sCOIL_SELECT_MEAS.asList[7].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[7].lRxChannelConnected = 5
sCOIL_SELECT_MEAS.asList[8].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[8].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[8].sCoilElementID.tElement = "PA3"
sCOIL_SELECT_MEAS.asList[8].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[8].lRxChannelConnected = 6
sCOIL_SELECT_MEAS.asList[9].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[9].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[9].sCoilElementID.tElement = "PA4"
sCOIL_SELECT_MEAS.asList[9].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[9].lRxChannelConnected = 7
sCOIL_SELECT_MEAS.asList[10].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[10].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[10].sCoilElementID.tElement = "PA2"
sCOIL_SELECT_MEAS.asList[10].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[10].lRxChannelConnected = 8
sCOIL_SELECT_MEAS.asList[11].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[11].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[11].sCoilElementID.tElement = "PA1"
sCOIL_SELECT_MEAS.asList[11].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[11].lRxChannelConnected = 8
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[0] = 0xff
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[1] = 0x76
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[2] = 0x78
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[3] = 0x87
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[4] = 0x67
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[0] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[1] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[2] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[3] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[4] = 0x2
sEFISPEC.bEFIDataValid = 1
### ASCCONV END ###
,
*/

/*
 * Table of equivalence:
 *
ulVersion = 0xbee332
<=>
27 - 'MrProtocolVersion' VM 1, VR IS, SyngoDT 6, NoOfItems 6, Data '12510002'
*/

#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmCSAHeader.h"
#include "gdcmAttribute.h"
#include "gdcmGlobal.h"
#include "gdcmDicts.h"

#include <map>

#include <math.h>

int main(int argc, char *argv [])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcm::ImageReader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }

    gdcm::CSAHeader csa;

```



```

const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

//const gdcm::PrivateTag &t1 = csa.GetCSAImageHeaderInfoTag();
const gdcm::PrivateTag &t2 = csa.GetCSASeriesHeaderInfoTag();

if( ds.FindDataElement( t2 ) )
{
    csa.LoadFromDataElement( ds.GetDataElement( t2 ) );
    //csa.Print( std::cout );
}

if( !csa.FindCSAElementByName( "MrProtocol" ) )
{
    return 1;
}

const gdcm::CSAElement &csael = csa.GetCSAElementByName( "MrProtocol" );
//std::cout << csael << std::endl;

const gdcm::ByteValue *bv = csael.GetByteValue();
if( !bv )
{
    return 1;
}

std::string str(bv->GetPointer(), bv->GetLength());
std::istream is(str);
std::string s;
typedef std::map< std::string, std::string > MyMapType;
MyMapType mymap;
while( std::getline(is, s) )
{
    std::string::size_type pos = s.find( '=' );
    if( pos != std::string::npos )
    {
        std::string sub1 = s.substr(0, pos);
        sub1.erase( sub1.find_last_not_of(' ') + 1);
        std::string sub2 = s.substr(pos+1); // skip the '=' char
        sub2.erase( 0, sub2.find_first_not_of(' '));
        //std::cout << sub1 << std::endl;
        mymap.insert( MyMapType::value_type(sub1, sub2) );
    }
    else
    {
        // ### ASCCONV BEGIN ###
        // ### ASCCONV END ###
    }
}

const char fourierstr[] = "sKSpace.ucSlicePartialFourier";
const gdcm::CSAHeaderDict &csadict = gdcm::Global::GetInstance().GetDicts().GetCSAHeaderDict();
const gdcm::CSAHeaderDictEntry &fourier = csadict.GetCSAHeaderDictEntry( fourierstr );
std::cout << fourier << std::endl;
MyMapType::const_iterator it = mymap.find( fourierstr );
if( it == mymap.end() ) return 1;
//std::cout << it->second << std::endl;
const std::string &partial_fourier = it->second;
if( partial_fourier == "0x1" )
{
    std::cout << "partial fourier is 4/8" << std::endl;
}
else if( partial_fourier == "0x2" )
{
    std::cout << "partial fourier is 5/8" << std::endl;
}
else if( partial_fourier == "0x4" )
{
    std::cout << "partial fourier is 6/8" << std::endl;
}
else if( partial_fourier == "0x8" )
{
    std::cout << "partial fourier is 7/8" << std::endl;
}
else if( partial_fourier == "0x10" )
{
    std::cout << "partial fourier is 8/8" << std::endl;
}
else
{
    std::cerr << "Impossible: " << partial_fourier << std::endl;
    return 1;
}
}

/*
This is the Flip Angle:

```

```
adFlipAngleDegree[0] = 30
```

One can find it also in the protocol:

```
...
    <ParamFuncutor."<TlmapFuncutor">
    {
        <Class> "<TlmapFuncutor@IceImagePostProcFuncutors">

        <ParamBool."<EXECUTE"> { }
        <ParamDouble."<Flip1_deg"> { <Precision> 16 14.7378520000000000 }
    }
...

*/
// Below is an attempt to play with the CSAHeader dict:
#ifdef 0
const char gspec[] = "sGRADSPEC.flSensitivityX";
it = mymap.find( gspec );
if( it == mymap.end() ) return 1;
const std::string &dummy = it->second;
std::cout << dummy << std::endl;

const gdcm::CSAHeaderDictEntry &csaentry = csadict.GetCSAHeaderDictEntry( gspec );
std::cout << csaentry << std::endl;
#endif

/*
    sSliceArray.ucMode -- should be in (1, 2, 4)
    enum SeriesMode
    {
        ASCENDING    = 0x01,
        DESCENDING    = 0x02,
        INTERLEAVED    = 0x04
    };
*/
const char sliceorderstr[] = "sSliceArray.ucMode";
const gdcm::CSAHeaderDictEntry &sliceorder = csadict.GetCSAHeaderDictEntry( sliceorderstr );
std::cout << sliceorder << std::endl;

it = mymap.find ( sliceorderstr );
if( it == mymap.end() ) return 1;
const std::string &slice_order = it->second;
if( slice_order == "0x1" )
{
    std::cout << "slice_order: ASCENDING" << std::endl;
}
else if( slice_order == "0x2" )
{
    std::cout << "slice_order: DESCENDING" << std::endl;
}
else if( slice_order == "0x4" )
{
    std::cout << "slice_order: INTERLEAVED" << std::endl;
}
else
{
    std::cerr << "Impossible: " << slice_order << std::endl;
    return 1;
}

gdcm::MrProtocol mrprot;
if( csa.GetMrProtocol(ds, mrprot) )
{
    std::cout << mrprot << std::endl;
}

return 0;
}
```

## 12.84 PrintLUT.cxx

```
/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
```

```

All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
*/

#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmImage.h"
#include "gdcmPhotometricInterpretation.h"

#include <iostream>

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " input.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];

    // Instantiate the image reader:
    gdcm::ImageReader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Could not read: " << filename << std::endl;
        return 1;
    }
    const gdcm::Image &image = reader.GetImage();

    const gdcm::LookupTable & lut = image.GetLUT();
    lut.Print( std::cout );

    return 0;
}

```

## 12.85 PublicDict.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
* Dummy example to show GDCM Dict(s) API (Part 6) + Collected Private Attributes:
*/

#include "gdcmGlobal.h"
#include "gdcmDicts.h"
#include "gdcmDict.h"
#include "gdcmCSAHeader.h"
#include "gdcmPrivateTag.h"

int main(int , char *[])
{
    const gdcm::Global& g = gdcm::Global::GetInstance(); // sum of all knowledge !
    const gdcm::Dicts &dicts = g.GetDicts();

```

```

const gdcm::Dict &pub = dicts.GetPublicDict(); // Part 6

//std::cout << pub << std::endl;

// 3 different ways to access the same information

// 1. From the public dict only:
gdcm::Tag patient_name(0x10,0x10);
const gdcm::DictEntry &entry1 = pub.GetDictEntry(patient_name);
std::cout << entry1 << std::endl;

// 2. From all dicts:
const gdcm::DictEntry &entry2 = dicts.GetDictEntry(patient_name);
std::cout << entry2 << std::endl;

// 3. This solution is the most flexible solution as you can request using the same
// API either a public tag or a private tag
const char *strowner = nullptr;
const gdcm::DictEntry &entry3 = dicts.GetDictEntry(patient_name,strowner);
std::cout << entry3 << std::endl;

// Private attributes:

// try with a private tag now:
const gdcm::PrivateTag &private_tag = gdcm::CSAHeader::GetCSAImageHeaderInfoTag();
//std::cout << private_tag << std::endl;
const gdcm::DictEntry &entry4 = dicts.GetDictEntry(private_tag,private_tag.GetOwner());
std::cout << entry4 << std::endl;

// Let's pretend that private lookup is on 0x10xx elements:
gdcm::PrivateTag dummy = private_tag;
dummy.SetElement( (uint16_t)(0x1000 + dummy.GetElement()) );
const gdcm::DictEntry &entry5 = dicts.GetDictEntry(dummy,dummy.GetOwner());
std::cout << entry5 << std::endl;

return 0;
}

```

## 12.86 QIDO-RS.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmJSON.h"

/*
 * Simple QIDO-RS round-trip to test implementation of gdcm::JSON
 * See Supl66 for details
 */
int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    using namespace gdcm;
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() ) return 1;

    gdcm::JSON json;
    json.PrettyPrintOn();
    std::stringstream ss;
    const gdcm::File & f = reader.GetFile();

```

```

json.Code( f.GetDataSet(), ss);

std::cout << ss.str() << std::endl;

gdcmm::Writer w;
gdcmm::File & ff = w.GetFile();
ff.GetHeader().SetDataSetTransferSyntax( gdcmm::TransferSyntax::ExplicitVRLittleEndian );
if( !json.Decode(ss, ff.GetDataSet() ) )
{
    std::cerr << "Could not decode" << std::endl;
    return 1;
}
w.SetFileName( "/tmp/debug.dcm" );
if( !w.Write() ) return 1;

return 0;
}

```

## 12.87 ReadAndDumpDICOMDIR.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * This example shows how to read and dump a DICOMDIR File
 *
 * Thanks:
 * Tom Marynowski (lordglub gmail) for contributing this example
 */
#include "gdcmmReader.h"
#include "gdcmmMediaStorage.h"

typedef std::set<gdcmm::DataElement> DataElementSet;
typedef DataElementSet::const_iterator ConstIterator;

int main(int argc, char *argv [])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];

    gdcmm::Reader reader;
    reader.SetFileName( filename);
    if( !reader.Read() )
    {
        std::cerr << "Could not read: " << filename << std::endl;
        return 1;
    }
    std::stringstream strm;

    gdcmm::File &file = reader.GetFile();
    gdcmm::DataSet &ds = file.GetDataSet();
    gdcmm::FileMetaInformation &fmi = file.GetHeader();

    gdcmm::MediaStorage ms;
    ms.SetFromFile(file);
    if( ms != gdcmm::MediaStorage::MediaStorageDirectoryStorage )
    {
        std::cout << "This file is not a DICOMDIR" << std::endl;
        return 1;
    }

    if (fmi.FindDataElement( gdcmm::Tag (0x0002, 0x0002)))
    {
        strm.str("");
        fmi.GetDataElement( gdcmm::Tag (0x0002, 0x0002) ).GetValue().Print(strm);
    }
    else
    {

```

```

std::cerr << " Media Storage Sop Class UID not present" << std::endl;
}

//TODO il faut trimer strm.str() avant la comparaison au cas ou...
if ("1.2.840.10008.1.3.10"!=strm.str())
{
std::cout << "This file is not a DICOMDIR" << std::endl;
return 1;
}

ConstIterator it = ds.GetDES().begin();

for( ; it != ds.GetDES().end(); ++it)
{
if (it->GetTag()==gdcm::Tag (0x0004, 0x1220))
{
const gdcm::DataElement &de = (*it);
// ne pas utiliser GetSequenceOfItems pour extraire les items
gdcm::SmartPointer<gdcm::SequenceOfItems> sqi =de.GetValueAsSQ();
unsigned int itemused = 1;
while (itemused<=sqi->GetNumberOfItems())

{
strm.str("");

if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1430)))
sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);

//TODO il faut trimer strm.str() avant la comparaison
while((strm.str()=="PATIENT")||((strm.str()=="PATIENT ")))
{
std::cout << strm.str() << std::endl;
strm.str("");
if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0010, 0x0010)))
sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0010, 0x0010)).GetValue().Print(strm);
std::cout << "PATIENT NAME : " << strm.str() << std::endl;

//PATIENT ID
strm.str("");
if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0010, 0x0020)))
sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0010, 0x0020)).GetValue().Print(strm);
std::cout << "PATIENT ID : " << strm.str() << std::endl;

/*ADD TAG TO READ HERE*/
std::cout << "===== " << std::endl;
itemused++;
strm.str("");
if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1430)))
sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);

//TODO il faut trimer strm.str() avant la comparaison
while((strm.str()=="STUDY")||((strm.str()=="STUDY ")))
{
std::cout << " " << strm.str() << std::endl;
//UID
strm.str("");
if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0020, 0x000d)))
sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0020, 0x000d)).GetValue().Print(strm);
std::cout << " STUDY UID : " << strm.str() << std::endl;

//STUDY DATE
strm.str("");
if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0008, 0x0020)))
sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0008, 0x0020)).GetValue().Print(strm);
std::cout << " STUDY DATE : " << strm.str() << std::endl;

//STUDY DESCRIPTION
strm.str("");
if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0008, 0x1030)))
sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0008, 0x1030)).GetValue().Print(strm);
std::cout << " STUDY DESCRIPTION : " << strm.str() << std::endl;

/*ADD TAG TO READ HERE*/
std::cout << " " << "===== " << std::endl;

itemused++;
strm.str("");
if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1430)))

```

```

        sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);

//TODO il faut trimer strm.str() avant la comparaison
while((strm.str()=="SERIES")||((strm.str()=="SERIES ")))
{
    std::cout << "          " << strm.str() << std::endl;
    strm.str("");
    if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0020, 0x000e)))
        sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0020, 0x000e)).GetValue().Print(strm);
    std::cout << "          SERIE UID" << strm.str() << std::endl;

//SERIE MODALITY
    strm.str("");
    if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0008, 0x0060)))
        sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0008, 0x0060)).GetValue().Print(strm);
    std::cout << "          SERIE MODALITY" << strm.str() << std::endl;

//SERIE DESCRIPTION
    strm.str("");
    if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0008, 0x103e)))
        sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0008, 0x103e)).GetValue().Print(strm);
    std::cout << "          SERIE DESCRIPTION" << strm.str() << std::endl;

/*ADD TAG TO READ HERE*/

    std::cout << "          " << "===== " << std::endl;
    itemused++;
    strm.str("");
    if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1430)))
        sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);

//TODO il faut trimer strm.str() avant la comparaison
while ((strm.str()=="IMAGE")||((strm.str()=="IMAGE ")))
// if(tmp=="IMAGE")
{
    std::cout << "          " << strm.str() << std::endl;

//UID
    strm.str("");
    if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1511)))
        sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1511)).GetValue().Print(strm);
    std::cout << "          IMAGE UID : " << strm.str() << std::endl;

//PATH de l'image
    strm.str("");
    if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1500)))
        sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1500)).GetValue().Print(strm);
    std::cout << "          IMAGE PATH : " << strm.str() << std::endl;
/*ADD TAG TO READ HERE*/

    if(itemused < sqi->GetNumberOfItems())
    {itemused++;
    }else{break;}

    strm.str("");

    if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1430)))
        sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);

    }
    }
    }
    itemused++;
}
}
}
return 0;
}

```

## 12.88 ReadAndDumpDICOMDIR2.cxx

/\*=====

```

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2017 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * This example shows how to read and dump a DICOMDIR File
 *
 * Thanks:
 *   Tom Marynowski (lordglub gmail) for contributing the original
 *   ReadAndDumpDICOMDIR.cxx example
 *   Mihail Isakov for contributing offset calculation code here:
 *   https://sourceforge.net/p/gdcm/mailman/gdcm-developers/?viewmonth=201707&viewday=15
 *   Tod Baudais for combining the above and cleaning up this example
 */

#include <string>
#include <unordered_map>
#include <iostream>
#include <memory>

#include "gdcmReader.h"
#include "gdcmAttribute.h"
#include "gdcmDirectory.h"

//=====
//=====

#define TAG_MEDIA_STORAGE_SOP_CLASS_UID 0x0002,0x0002
#define TAG_DIRECTORY_RECORD_SEQUENCE 0x0004,0x1220
#define TAG_DIRECTORY_RECORD_TYPE 0x0004,0x1430
#define TAG_PATIENTS_NAME 0x0010,0x0010
#define TAG_PATIENT_ID 0x0010,0x0020
#define TAG_STUDY_DATE 0x0008,0x0020
#define TAG_STUDY_DESCRIPTION 0x0008,0x1030
#define TAG_MODALITY 0x0008,0x0060
#define TAG_SERIES_DESCRIPTION 0x0008,0x103E
#define TAG_REFERENCED_FILE_ID 0x0004,0x1500
#define TAG_REFERENCED_LOWER_LEVEL_DIRECTORY_ENTITY_OFFSET 0x0004,0x1420
#define TAG_NEXT_DIRECTORY_RECORD_OFFSET 0x0004,0x1400

//=====
// Some handy utility functions
//=====

std::string left_trim(const std::string &s) {
    std::string ss(s);
    ss.erase(ss.begin(), std::find_if(ss.begin(), ss.end(), std::not1(std::ptr_fun<int, int>(std::isspace))));
    return ss;
}

std::string right_trim(const std::string &s) {
    std::string ss(s);
    ss.erase(std::find_if(ss.rbegin(), ss.rend(), std::not1(std::ptr_fun<int, int>(std::isspace))).base(),
              ss.end());
    return ss;
}

std::string trim(const std::string &s) {
    return left_trim(right_trim(s));
}

//=====
// This code could be put in a header file somewhere
//=====

class DICOMDIRReader {
public:
    DICOMDIRReader & operator = (const DICOMDIRReader &rhs) = delete;
    DICOMDIRReader & operator = (DICOMDIRReader &&rhs) = delete;
    DICOMDIRReader & operator = (const DICOMDIRReader &rhs) = delete;
    DICOMDIRReader & operator = (DICOMDIRReader &&rhs) = delete;
    DICOMDIRReader() {}
    DICOMDIRReader(const DICOMDIRReader &rhs) = delete;
    DICOMDIRReader(DICOMDIRReader &&rhs) = delete;
    DICOMDIRReader(const DICOMDIRReader &rhs) = delete;
    DICOMDIRReader(DICOMDIRReader &&rhs) = delete;

```



```

    virtual ~DICOMDIRReader      (void) {}

public:
    struct Common {
        int64_t child_offset;
        int64_t sibling_offset;
    };

    struct Image: public Common {
        std::string path;
    };

    struct Series: public Common {
        std::string modality;
        std::string description;

        std::vector<std::shared_ptr<Image>> children;
    };

    struct Study: public Common {
        std::string date;
        std::string description;

        std::vector<std::shared_ptr<Series>> children;
    };

    struct Patient: public Common {
        std::string name;
        std::string id;

        std::vector<std::shared_ptr<Study>> children;
    };

    struct Other: public Common {
    };

    const std::vector<std::shared_ptr<Patient>>&    load      (const std::string &path);

    const std::vector<std::shared_ptr<Patient>>&    patients  (void) { return _patients; }

private:
    template <class T>
    std::string get_string      (const T &ds, const gdcm::Tag &tag)
    {
        std::stringstream strm;
        if (ds.FindDataElement(tag)) {
            auto &de = ds.GetDataElement(tag);
            if (!de.IsEmpty() && !de.IsUndefinedLength())
                de.GetValue().Print(strm);
        }
        return trim(strm.str());
    }

    template <class P, class C, class O>
    void reassemble_hierarchy  (P &parent_offsets, C &child_offsets, O &other_offsets)
    {
        for (auto &parent : parent_offsets) {
            int64_t sibling_offset;
            auto c = child_offsets[parent.second->child_offset];
            if (!c) {
                auto o = other_offsets[parent.second->child_offset];
                if (!o) {
                    continue;
                } else {
                    sibling_offset = o->sibling_offset;
                }
            } else {
                parent.second->children.push_back(c);
                sibling_offset = c->sibling_offset;
            }

            // Get all siblings
            while (sibling_offset) {
                c = child_offsets[sibling_offset];
                if (!c) {
                    auto o = other_offsets[sibling_offset];
                    if (!o) {
                        break;
                    } else {
                        sibling_offset = o->sibling_offset;
                    }
                }
            }
        }
    }

```

```

        }
        } else {
            parent.second->children.push_back(c);
            sibling_offset = c->sibling_offset;
        }
    }
}

std::vector<std::shared_ptr<Patient> > _patients;
};

//=====
// This code could be put in an implementation file somewhere
//=====

const std::vector<std::shared_ptr<DICOMDIRReader::Patient>> & DICOMDIRReader::load (const std::string &path)
{
    _patients.clear();

    //
    // Read the dataset from the DICOMDIR file
    //

    gdcm::Reader reader;
    reader.SetFileName(path.c_str());
    if(!reader.Read()) {
        throw std::runtime_error("Unable to read file");
    }

    // Retrieve information from file
    auto &file = reader.GetFile();
    auto &data_set = file.GetDataSet();
    auto &file_meta_information = file.GetHeader();

    // Retrieve and check the Media Storage class from file
    gdcm::MediaStorage media_storage;
    media_storage.SetFromFile(file);
    if(media_storage != gdcm::MediaStorage::MediaStorageDirectoryStorage) {
        throw std::runtime_error("This file is not a DICOMDIR");
    }

    auto media_storage_sop_class_uid = get_string(file_meta_information,
        gdcm::Tag(TAG_MEDIA_STORAGE_SOP_CLASS_UID));

    // Make sure we have a DICOMDIR file
    if (media_storage_sop_class_uid != "1.2.840.10008.1.3.10") {
        throw std::runtime_error("This file is not a DICOMDIR");
    }

    //
    // Offset to first item courtesy of Mihail Isakov
    //

    gdcm::VL first_item_offset = 0;
    auto it = data_set.Begin();
    for(; it != data_set.End() && it->GetTag() != gdcm::Tag(TAG_DIRECTORY_RECORD_SEQUENCE); ++it) {
        first_item_offset += it->GetLength<gdcm::ExplicitDataElement>();
    }
    // Tag (4 bytes)
    first_item_offset += it->GetTag().GetLength();
    // VR field
    first_item_offset += it->GetVR().GetLength();
    // VL field
    // For Explicit VR: adventitiously VL field length = VR field length,
    // for SQ 4 bytes:
    // http://dicom.nema.org/medical/dicom/current/output/html/part05.html#table_7.1-1
    first_item_offset += it->GetVR().GetLength();

    //
    // Iterate all data elements
    //

    // For each item in data set
    for(auto data_element : data_set.GetDES()) {

        // Only look at Directory sequence
        if (data_element.GetTag() != gdcm::Tag(TAG_DIRECTORY_RECORD_SEQUENCE))
            continue;

        auto item_sequence = data_element.GetValueAsSQ();
    }
}

```

```

auto num_items = item_sequence->GetNumberOfItems();

//
// Compute an offset table
//

// Start calculation of offset to each item courtesy of Mihail Isakov
std::vector<int64_t> item_offsets(num_items+1);
item_offsets[0] = file_meta_information.GetFullLength() + static_cast<int64_t>(first_item_offset);

//
// Extract out all of the items
//

std::unordered_map<int64_t, std::shared_ptr<Patient>> patient_offsets;
std::unordered_map<int64_t, std::shared_ptr<Study>> study_offsets;
std::unordered_map<int64_t, std::shared_ptr<Series>> series_offsets;
std::unordered_map<int64_t, std::shared_ptr<Image>> image_offsets;
std::unordered_map<int64_t, std::shared_ptr<Other>> other_offsets;

for (uint32_t item_index = 1; item_index <= num_items; ++item_index) {
    auto &item = item_sequence->GetItem(item_index);

    // Add offset for item to offset table
    item_offsets[item_index] = item_offsets[item_index-1] + item.GetLength<gdcm::ExplicitDataElement>();

    // Child offset
    gdcm::Attribute<TAG_REFERENCED_LOWER_LEVEL_DIRECTORY_ENTITY_OFFSET> child_offset;
    child_offset.SetFromDataElement(item.GetDataElement(gdcm::Tag
(TAG_REFERENCED_LOWER_LEVEL_DIRECTORY_ENTITY_OFFSET)));

    // Sibling offset
    gdcm::Attribute<TAG_NEXT_DIRECTORY_RECORD_OFFSET> sibling_offset;
    sibling_offset.SetFromDataElement(item.GetDataElement(gdcm::Tag
(TAG_NEXT_DIRECTORY_RECORD_OFFSET)));

    // Record Type
    auto record_type = trim(get_string(item, gdcm::Tag (TAG_DIRECTORY_RECORD_TYPE)));

    // std::cout << "record_type " << record_type << " at " << item_offsets[item_index-1] << std::endl;
    // std::cout << " child_offset " << child_offset.GetValue() << std::endl;
    // std::cout << " sibling_offset " << sibling_offset.GetValue() << std::endl;

    // Extract patient information
    if (record_type == "PATIENT") {
        auto patient = std::make_shared<Patient>();
        patient->name = get_string(item, gdcm::Tag (TAG_PATIENTS_NAME));
        patient->id = get_string(item, gdcm::Tag (TAG_PATIENT_ID));

        patient->child_offset = child_offset.GetValue();
        patient->sibling_offset = sibling_offset.GetValue();
        patient_offsets[item_offsets[item_index-1]] = patient;

    // Extract study information
    } else if (record_type == "STUDY") {
        auto study = std::make_shared<Study>();
        study->date = get_string(item, gdcm::Tag (TAG_STUDY_DATE));
        study->description = get_string(item, gdcm::Tag (TAG_STUDY_DESCRIPTION));

        study->child_offset = child_offset.GetValue();
        study->sibling_offset = sibling_offset.GetValue();
        study_offsets[item_offsets[item_index-1]] = study;

    // Extract series information
    } else if (record_type == "SERIES") {
        auto series = std::make_shared<Series>();
        series->modality = get_string(item, gdcm::Tag (TAG_MODALITY));
        series->description = get_string(item, gdcm::Tag (TAG_SERIES_DESCRIPTION));

        series->child_offset = child_offset.GetValue();
        series->sibling_offset = sibling_offset.GetValue();
        series_offsets[item_offsets[item_index-1]] = series;

    // Extract image information
    } else if (record_type == "IMAGE") {
        auto image = std::make_shared<Image>();
        image->path = get_string(item, gdcm::Tag (TAG_REFERENCED_FILE_ID));

        image->child_offset = child_offset.GetValue();
        image->sibling_offset = sibling_offset.GetValue();
        image_offsets[item_offsets[item_index-1]] = image;
    }
}

```

```

    } else {
        auto other = std::make_shared<Other>();

        other->child_offset = child_offset.GetValue();
        other->sibling_offset = sibling_offset.GetValue();
        other_offsets[item_offsets[item_index-1]] = other;
    }
}

// Check validity
if (patient_offsets.size() == 0)
    throw std::runtime_error("Unable to find patient record");

reassemble_hierarchy(series_offsets, image_offsets, other_offsets);
reassemble_hierarchy(study_offsets, series_offsets, other_offsets);
reassemble_hierarchy(patient_offsets, study_offsets, other_offsets);

// Set the new root
for (auto &patient : patient_offsets) {
    _patients.push_back(patient.second);
}

return _patients;
}

//=====
// Quick test
//=====

int main(int argc, const char *argv[]) {
    DICOMDIRReader reader;

    try {
        if (argc != 2)
            throw std::runtime_error("Wrong number of arguments");

        auto &patients = reader.load(argv[1]);

        for (auto &patient : patients) {

            std::cout << "PATIENT" << std::endl;
            std::cout << "NAME: " << patient->name << std::endl;
            std::cout << "ID: " << patient->id << std::endl;

            int x = 0;
            for (auto &study : patient->children) {
                std::cout << "    STUDY" << std::endl;
                std::cout << "        DESCRIPTION: " << study->description << std::endl;
                std::cout << "        DATE: " << study->date << std::endl;

                for (auto &series : study->children) {
                    x+=1;
                    std::cout << "            SERIES " << x << std::endl;
                    std::cout << "            DESCRIPTION: " << series->description << std::endl;
                    std::cout << "            MODALITY: " << series->modality << std::endl;

                    for (auto &image : series->children) {
                        std::cout << "                IMAGE PATH: " << image->path << std::endl;
                    }
                }
            }
        }
    }
    catch (...) {
        // TODO handle this
        return EXIT_FAILURE;
    }

    return EXIT_SUCCESS;
}

```

## 12.89 ReadAndPrintAttributes.cxx

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

```

Copyright (c) 2006-2011 Mathieu Malaterre  
 All rights reserved.  
 See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even  
 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR  
 PURPOSE. See the above copyright notice for more information.

```

=====*/
/*
 * This small example will show how one can read and print
 * a DICOM Attribute using different technique (by tag or by name)
 */

#include "gdcmReader.h"
#include "gdcmGlobal.h"
#include "gdcmDicts.h"
#include "gdcmDict.h"
#include "gdcmAttribute.h"
#include "gdcmStringFilter.h"

#include <iostream>

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " input.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];

    // Instantiate the reader:
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Could not read: " << filename << std::endl;
        return 1;
    }

    // The output of gdcm::Reader is a gdcm::File
    gdcm::File &file = reader.GetFile();

    // the dataset is the the set of element we are interested in:
    gdcm::DataSet &ds = file.GetDataSet();

    const gdcm::Global& g = gdcm::Global::GetInstance();
    const gdcm::Dicts &dicts = g.GetDicts();
    const gdcm::Dict &pubdict = dicts.GetPublicDict();

    using namespace gdcm;

    // In this example we will show why using name to lookup attribute can be
    // dangerous.
    Tag tPatientName(0x00,0x00);
    //const DictEntry &de1 =
    pubdict.GetDictEntryByName("Patient Name", tPatientName);

    std::cout << "Found: " << tPatientName << std::endl;

    // Indeed the attribute could not be found. Since DICOM 2003, Patient Name
    // has become Patient's Name.

    Tag tPatientsName;
    //const DictEntry &de2 =
    pubdict.GetDictEntryByName("Patient's Name", tPatientsName);

    std::cout << "Found: " << tPatientsName << std::endl;

    // Let's try to read an arbitrary DICOM Attribute:
    Tag tDoseGridScaling;
    //const DictEntry &de3 =
    pubdict.GetDictEntryByName("Dose Grid Scaling", tDoseGridScaling);

    std::cout << "Found: " << tDoseGridScaling << std::endl;

    if( ds.FindDataElement( tDoseGridScaling ) )
    {
        gdcm::StringFilter sf;

```

```

sf.SetFile(file);
std::cout << "Attribute Value as String: " << sf.ToString( tDoseGridScaling ) << std::endl;

// Let's check the name again:
std::pair<std::string, std::string> pss
    = sf.ToStringPair( tDoseGridScaling );
std::cout << "Attribute Name Checked: " << pss.first << std::endl;
std::cout << "Attribute Value (string): " << pss.second << std::endl;

//const DataElement &dgs = ds.GetDataElement( tDoseGridScaling );

// Let's assume for a moment we knew the tag number:
Attribute<0x3004,0x000e> at;
assert( at.GetTag() == tDoseGridScaling );
at.SetFromDataSet( ds );
// For the sake of long term maintenance, we will not write
// that this particular attribute is stored as a double. What if
// a user made a mistake. It is much safer to rely on GDCM internal
// mechanism to deduce the VR::DS type (represented as a ieee double)
Attribute<0x3004,0x000e>::ArrayType v = at.GetValue();
std::cout << "DoseGridScaling=" << v << std::endl;
}

return 0;
}

```

## 12.90 ReadExplicitLengthSQIVR.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmlByteValue.h"
#include "gdcmlDataSet.h"
#include "gdcmlImplicitDataElement.h"
#include "gdcmlPrivateTag.h"
#include "gdcmlReader.h"
#include "gdcmlSequenceOfItems.h"

using namespace gdcml;

int main(int argc, char *argv[])
{
    if ( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcml::Reader r;
    r.SetFileName( filename );
    r.Read();

    //gdcml::PrivateTag pt(0x01,0x42,"ELSCINT1");
    //gdcml::Tag pt(0x88,0x200);
    gdcml::Tag pt(0x8,0x1140);
    DataSet &ds = r.GetFile().GetDataSet();
    const DataElement &de = ds.GetDataElement( pt );

    std::cout << de << std::endl;
    const ByteValue *bv = de.GetByteValue();
    SmartPointer<SequenceOfItems> sqi = new SequenceOfItems;
    sqi->SetLength( bv->GetLength() );
    std::stringstream ss;
    ss.str( std::string( bv->GetPointer(), bv->GetLength() ) );
    sqi->Read<ImplicitDataElement,SwapperNoOp>( ss );

    std::cout << *sqi << std::endl;

    return 0;
}

```

## 12.91 ReadGEMSSDO.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmDataElement.h"
#include "gdcmPrivateTag.h"

#include <iostream>
#include <string>

using namespace gdcm;

struct SDOElement
{
    typedef std::vector<std::string>::size_type SizeType;
    const char *GetData(SizeType index) const {
        return Data[index].c_str();
    }
    SizeType GetNumberOfData() const {
        return Data.size();
    }
    void SetData(SizeType index, const char *data) {
        Data[index] = data;
    }
    const char *GetDataFormat() const {
        return DataFormat.c_str();
    }
    void SetDataFormat(const char *dataformat, SizeType num) {
        DataFormat = dataformat;
        Data.resize( num );
    }
    void Print( std::ostream &os ) const {
        os << DataFormat << ":" << std::endl;
        std::vector<std::string>::const_iterator it = Data.begin();
        size_t s = 0;
        for( ; it != Data.end(); ++it )
        {
            os << "  (" << s++ << ") " << *it << std::endl;
        }
    }
private:
    std::string DataFormat;
    std::vector<std::string> Data;
};

class SDOHeader
{
public:
    typedef std::vector<SDOElement> SDOElements;
    typedef SDOElements::size_type SizeType;
    SizeType GetNumberOfSDOElements() const {
        return InternalSDODataset.size();
    }
    void AddSDOElement(SDOElement const &sdoelement) {
        InternalSDODataset.push_back( sdoelement );
    }
    const SDOElement &GetSDOElement(SizeType index) const {
        return InternalSDODataset[index];
    }
    const SDOElement &GetSDOElementByName(const char *) const {
        return InternalSDODataset[0];
    }
    void LoadFromAttributes(std::string const &s1, std::string const &s2)
    {
        std::string tok;
        std::string tok2;
        std::stringstream strstr(s1);

```

```

std::stringstream strstr2(s2);

SDOElement element;
// Do format
size_t count = 0;
while ( std::getline ( strstr2, tok, '\\\' ) )
{
    //std::cout << tok << " ";
    std::getline ( strstr2, tok2, '\\\' );
    //std::cout << tok2 << std::endl;
    count += atoi( tok2.c_str() );
    element.SetDataFormat( tok.c_str(), atoi( tok2.c_str() ) );
    for( size_t t = 0; t < element.GetNumberOfData(); ++t )
    {
        std::getline ( strstr, tok, '\\\' );
        element.SetData(t, tok.c_str() );
    }
    AddSDOElement( element );
}
//while ( std::getline ( strstr, tok, '^\' ) )
// while ( std::getline ( strstr, tok, '\\\' ) )
// {
//     std::cout << tok << std::endl;
//     count++;
// }
// std::cout << "Count: " << count << std::endl;
// count = 0;

// std::cout << "Count: " << count << std::endl;

}

void Print( std::ostream &os ) const {
    SDOElements::const_iterator it = InternalSDODataset.begin();
    for( ; it != InternalSDODataset.end(); ++it )
    {
        it->Print ( os );
    }
}

private:
    SDOElements InternalSDODataset;
};

bool sdo_decode( DataElement const &stringdata, DataElement const &stringdataformat )
{
    const char *sd = stringdata.GetByteValue()->GetPointer();
    const size_t len_sd = stringdata.GetByteValue()->GetLength();

    std::string s1 = std::string( sd, len_sd );

    const char *sdf = stringdataformat.GetByteValue()->GetPointer();
    const size_t len_sdf = stringdataformat.GetByteValue()->GetLength();

    std::string s2 = std::string( sdf, len_sdf );

    // std::cout << s1 << std::endl;
    // std::cout << s2 << std::endl;

    SDOHeader header;
    header.LoadFromAttributes( s1, s2 );

    header.Print( std::cout );

    return true;
}

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " input.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    File &file = reader.GetFile();

```



```

DataSet &ds = file.GetDataSet();

// StringData (0033,xx1F) 3 "GEMS_GENIE_1" List of SDO parameters stored as
// list of strings
const PrivateTag tstringdata(0x33,0x1f,"GEMS_GENIE_1");
// StringDataFormat (0033,xx23) 3 "GEMS_GENIE_1" Format of string parameters;
// contains information about name and number of strings in list
const PrivateTag tstringdataformat(0x33,0x23,"GEMS_GENIE_1");

if( !ds.FindDataElement( tstringdata ) ) return 1;
const DataElement& stringdata = ds.GetDataElement( tstringdata );
if( !ds.FindDataElement( tstringdataformat ) ) return 1;
const DataElement& stringdataformat = ds.GetDataElement( tstringdataformat );

sdo_decode( stringdata, stringdataformat );

return 0;
}

```

## 12.92 ReadMultiTimesException.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
// The intention of this sample program is to provoke bad_alloc exceptions in gdcm code

#include "gdcmImageReader.h"

int main(int argc, char* argv[])
{
    // We pre-allocate some memory (about 1Gb) to help the issue to show up earlier
    char *dummyBuffer = new char[1024*1024*1100]; (void)dummyBuffer;
    // Check the number of parameters given
    if (argc < 3)
    {
        std::cerr << "Usage: " << argv[0] << " Filename numberOfTries" << std::endl;
        return 1;
    }

    std::cout << "We are going to read the file: " << argv[1] << " " << argv[2] << " times" << std::endl;
    // We hold the pointers in an array to avoid the memory to be released
    // We read the input file n-times
    for (int i = 0; i < atoi(argv[2]); ++i)
    {
        gdcm::ImageReader reader;
        std::cout << "Reading try: " << i << std::endl;
        // Read files
        reader.SetFileName(argv[1]);
        try
        {
            reader.Read();
            gdcm::Image &img = reader.GetImage();
            unsigned long len = img.GetBufferLength();
            char *buffer = new char[ len ];
            img.GetBuffer( buffer ); // do NOT de-allocate buffer !
        }
        catch (std::bad_alloc &ba)
        {
            (void)ba;
            std::cerr << "BAD ALLOC Exception caught!" << std::endl;
        }
        catch (...)
        {
            std::cerr << "Exception caught!" << std::endl;
        }
    }

    return 0;
}

```

```
}
```

## 12.93 ReadUTF8QtDir.cxx

```
/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/

/*
 * GDCM API expect a const char * as input for SetFileName
 * In order to use this API from Qt, here is a simple test that
 * shows how to do it in a portable manner:
 *
 * http://doc.qt.nokia.com/latest/qdir.html#navigation-and-directory-operations
 */

#include "gdcmReader.h"
#include "gdcmDirectory.h"

#include <QDir>
#include <QString>
#include <QCoreApplication>

#include <string>
#include <fstream>

#include <stdio.h> // fopen

static int TestBothFuncs(const char *info , const char *ba_str)
{
    int res = 0;
    FILE *f = fopen( ba_str, "r" );
    if( f )
    {
        std::cout << info << " fopen: " << ba_str << std::endl;
        fclose(f);
        ++res;
    }
    gdcm::Reader reader;
    std::ifstream is( ba_str, std::ios::binary );
    if( is.is_open() )
    {
        std::cout << info << " is_open: " << ba_str << std::endl;
        ++res;
    }
    reader.SetStream( is );
    if( reader.CanRead() == true )
    {
        std::cout << info << " SetStream/CanRead:" << ba_str << std::endl;
        ++res;
    }
    is.close();
    reader.SetFileName( ba_str );
    if( reader.CanRead() == true )
    {
        std::cout << info << " SetFileName/CanRead:" << ba_str << std::endl;
        ++res;
    }
    return 4 - res;
}

static int scanFolder(const char dirname[])
{
    int res = 0;
    gdcm::Directory dir;
    unsigned int nfiles = dir.Load( dirname, true );
    const gdcm::Directory::FileNamesType &filenames = dir.GetFilesNames();

    for( unsigned int i = 0; i < nfiles; ++i )
```

```

    {
        const char *ba_str = filenames[i].c_str();
        res += TestBothFuncs("GDCM", ba_str);
    }
    return res;
}

static int scanFolderQt(QDir const &dir, QStringList& files)
{
    int res = 0;
    QFileInfoList children = dir.entryInfoList(QDir::AllEntries|QDir::NoDotAndDotDot);
    for ( int i=0; i<children.count(); i++ ) {
        QFileInfo file = children.at(i);
        if ( file.isDir() == true ) {
            res += scanFolderQt(QDir(file.absoluteFilePath()), files);
            continue;
        }
        // Convert back from the internal representation to 8bits
        // toLocal8Bit() returns by copy. Need to store explicitly the QByteArray
        QByteArray str = file.absoluteFilePath().toLocal8Bit();
        const char *ba_str1 = str.constData();
        res += TestBothFuncs("QString", ba_str1);
    }
    return res;
}

int main(int argc, char *argv[])
{
    // very important:
    QApplication qCoreApp( argc , argv );
    if( argc < 2 )
    {
        std::cerr << argv[0] << " dir " << std::endl;
        return 1;
    }

    int res = 0;
    const char *dirname = argv[1];
    res += scanFolder( dirname );

    QDir dir( QString::fromLocal8Bit(dirname) );
    QStringList files;
    res += scanFolderQt( dir, files);

    if( res )
        std::cerr << "Problem with UTF-8" << std::endl;
    else
        std::cerr << "Success with UTF-8" << std::endl;

    return res;
}

```

## 12.94 SimpleScanner.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/

/*
 * Simple example to show how to use Scanner API.
 * It exposes the three different cases:
 * - DICOM Attribute is present and has a value
 * - DICOM Attribute is present and has no value
 * - DICOM Attribute is not present at all
 * It also shows the purpose of the function 'IsKey' to detect whether or
 * not the file has been read by the gdcm::Scanner. Technically most of the time
 * if a file is not a 'Key' this is because it is not a DICOM file. You need to use
 * gdcm::System::FileExists to decide whether or not the file actually exist on the disk.
 */

```

```

*
* It was tested on this particular image:
* ./SimpleScanner gdcmData/012345.002.050.dcm
*/

#include "gdcmStrictScanner.h"
#include "gdcmSimpleSubjectWatcher.h"
#include "gdcmFileNameEvent.h"

class MyFileWatcher : public gdcm::SimpleSubjectWatcher
{
public:
    MyFileWatcher(gdcm::Subject * s, const char *comment = ""):
        gdcm::SimpleSubjectWatcher(s,comment){}
    void ShowFileName(gdcm::Subject *, const gdcm::Event &evt) override
    {
        const gdcm::FileNameEvent &pe = dynamic_cast<const gdcm::FileNameEvent&>(evt);
        const char *fn = pe.GetFileName();
        std::cout << "FileName: " << fn << " FileSize: " << gdcm::System::FileSize( fn ) << std::endl;
    }
};

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        return 1;
    }
    const char *filename = argv[1];
    const char filename_invalid[] = "this is a file that may not exist on this disk.dcm";

    gdcm::SmartPointer<gdcm::StrictScanner> sp = new gdcm::StrictScanner;
    gdcm::StrictScanner &s = *sp;
    //gdcm::SimpleSubjectWatcher w(&s, "TestFileName" );
    MyFileWatcher w(&s, "TestFileName" );

    const gdcm::Tag tag_array[] = {
        gdcm::Tag(0x8,0x50),
        gdcm::Tag(0x8,0x51),
        gdcm::Tag(0x8,0x60),
        gdcm::Tag(0x8,0x80),
    };
    s.AddTag( tag_array[0] );
    s.AddTag( tag_array[1] );
    s.AddTag( tag_array[2] );
    s.AddTag( tag_array[3] );

    gdcm::Directory::FileNamesType filenames;
    filenames.push_back( filename );
    filenames.push_back( filename_invalid );

    if( !s.Scan( filenames ) )
    {
        return 1;
    }

    //s.Print( std::cout );

    for(gdcm::Directory::FileNamesType::const_iterator it = filenames.begin();
        it != filenames.end(); ++it )
    {
        if( s.IsKey( it->c_str() ) )
        {
            std::cout << "INFO:" << it->c_str() << " is a proper Key for the Scanner (this is a DICOM file)" << std::endl;
        }
        else
        {
            std::cout << "INFO:" << it->c_str() << " is not a proper Key for the Scanner (this is either not a DICOM file
            or file does not exist)" << std::endl;
        }
    }

    gdcm::StrictScanner::TagToValue const &tvt = s.GetMapping(filename);

    const gdcm::Tag *ptag = tag_array;
    for( ; ptag != tag_array + 3; ++ptag )
    {
        gdcm::StrictScanner::TagToValue::const_iterator it = tvt.find( *ptag );
        if( it != tvt.end() )

```

```

    {
        std::cout << *ptag << " was properly found in this file" << std::endl;
        // it contains a pair of value. the first one is the actual tag, so the following is always true:
        // *ptag == it->first
        // The second part is the actual value (stored as RAW strings). You will have to reinterpret this string
        // if VR for *ptag is not VR::VRASCII !
        const char *value = it->second;
        if( *value )
        {
            std::cout << " It has the value: " << value << std::endl;
        }
        else
        {
            std::cout << " It has no value (empty)" << std::endl;
        }
    }
    else
    {
        std::cout << "Sorry " << *ptag << " could not be found in this file" << std::endl;
    }
}

return 0;
}

```

## 12.95 SortImage.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
*/
#include "gdcmSorter.h"
#include "gdcmScanner.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"

bool mysort(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
    //gdcm::Attribute<0x0020,0x0013> at1; // Instance Number
    gdcm::Attribute<0x0018,0x1060> at1; // Trigger Time
    gdcm::Attribute<0x0020,0x0032> at11; // Image Position (Patient)
    at1.Set( ds1 );
    at11.Set( ds1 );
    //gdcm::Attribute<0x0020,0x0013> at2;
    gdcm::Attribute<0x0018,0x1060> at2;
    gdcm::Attribute<0x0020,0x0032> at22;
    at2.Set( ds2 );
    at22.Set( ds2 );
    if( at11 == at22 )
    {
        return at1 < at2;
    }
    return at11 < at22;
}

bool mysort_part1(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
    gdcm::Attribute<0x0018,0x1060> at1;
    at1.Set( ds1 );
    gdcm::Attribute<0x0018,0x1060> at2;
    at2.Set( ds2 );
    return at1 < at2;
}

bool mysort_part2(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )

```

```

{
    gdc::Attribute<0x0020,0x0032> at1;
    at1.Set( ds1 );
    gdc::Attribute<0x0020,0x0032> at2;
    at2.Set( ds2 );
    return at1 < at2;
}

// technically all files are in the same Frame of Reference, so this function
// should be a no-op
bool mysort_dummy(gdc::DataSet const & ds1, gdc::DataSet const & ds2 )
{
    gdc::Attribute<0x0020,0x0052> at1; // FrameOfReferenceUID
    at1.Set( ds1 );
    gdc::Attribute<0x0020,0x0052> at2;
    at2.Set( ds2 );
    return at1 < at2;
}

int main(int argc, char *argv[])
{
    if (argc < 2 ) return 1;
    const char *dirname = argv[1];
    gdc::Directory dir;
    unsigned int nfiles = dir.Load( dirname );

    dir.Print( std::cout );

    gdc::Sorter sorter;
    sorter.SetSortFunction( mysort );
    sorter.Sort( dir.GetFilesNames() );

    std::cout << "Sorter:" << std::endl;
    sorter.Print( std::cout );

    gdc::Sorter sorter2;
    sorter2.SetSortFunction( mysort_part1 );
    sorter2.StableSort( dir.GetFilesNames() );
    sorter2.SetSortFunction( mysort_part2 );
    sorter2.StableSort( sorter2.GetFilesNames() ); // IMPORTANT
    sorter2.SetSortFunction( mysort_dummy );
    sorter2.StableSort( sorter2.GetFilesNames() ); // IMPORTANT

    std::cout << "Sorter2:" << std::endl;
    sorter2.Print( std::cout );

    gdc::Scanner s;
    s.AddTag( gdc::Tag(0x20,0x32) ); // Image Position (Patient)
    //s.AddTag( gdc::Tag(0x20,0x37) ); // Image Orientation (Patient)
    s.Scan( dir.GetFilesNames() );

    //s.Print( std::cout );

    // Count how many different IPP there are:
    const gdc::Scanner::ValuesType &values = s.GetValues();
    size_t nvalues = values.size();
    std::cout << "There are " << nvalues << " different type of values" << std::endl;

    //std::cout << "nfiles=" << nfiles << std::endl;
    if( nfiles % nvalues != 0 )
    {
        std::cerr << "Impossible: this is a not a proper series" << std::endl;
        return 1;
    }
    std::cout << "Series is composed of " << (nfiles/nvalues) << " different 3D volumes" << std::endl;

    return 0;
}

```

## 12.96 StreamImageReaderTest.cxx

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.

```

See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the above copyright notice for more information.

```

=====*/
// This work was realised during the GSOC 2011 by Manoj Alwani

#include "gdcmStreamImageReader.h"
#include "gdcmFileMetaInformation.h"
#include "gdcmSystem.h"
#include "gdcmFilename.h"
#include "gdcmByteSwap.h"
#include "gdcmTrace.h"
#include "gdcmTesting.h"
#include "gdcmImageHelper.h"
#include "gdcmImageReader.h"
#include "gdcmImage.h"
#include "gdcmMediaStorage.h"
#include "gdcmRAWCodec.h"
#include "gdcmJPEGCodec.h"
#include "gdcmUIDGenerator.h"
#include "gdcmStreamImageWriter.h"
#include "gdcmAttribute.h"
#include "gdcmFile.h"
#include "gdcmTag.h"

bool StreamImageRead(gdcm::StreamImageWriter & theStreamWriter,
    const char* filename, const char* outfileName, int resolution)
{
    gdcm::StreamImageReader reader;

    reader.SetFileName( filename );

    if (!reader.ReadImageInformation())
    {
        std::cerr << "unable to read image information" << std::endl;
        return 1; //unable to read tags as expected.
    }

    //let's be tricky; each image will be read in portions, first the top half, then the bottom
    //that way, we can test how the stream handles fragmentation of the data
    //we could also loop this to get various different size combinations, but I'm not sure
    //that's useful, yet.
    std::vector<unsigned int> extent =
        gdcm::ImageHelper::GetDimensionsValue(reader.GetFile());
    // std::cout << extent[0];
    //at this point, these values aren't used, but may be in the future
    //unsigned short xmin = 0;
    //unsigned short xmax = extent[0];
    //unsigned short ymin = 0;
    //unsigned short ymax = extent[1];
    //unsigned short zmin = 0;
    //unsigned short zmax = extent[2];

    std::cout << "\n Row: " << extent[0] << "\n Col : " << extent[1] << "\n Resolution : " << extent[2] << std::endl;

    int a = 1;
    for (int i=1; i<=(extent[2]-resolution);++i)
        a = a*2;

    reader.DefinePixelExtent(0, extent[0]/a, 0, extent[1]/a, resolution-1, resolution);

    unsigned long len = reader.DefineProperBufferLength();
    char* finalBuffer = new char[len];
    memset(finalBuffer, 0, sizeof(char)*len);

    if (reader.CanReadImage())
    {
        bool result = reader.Read(finalBuffer, len);
        if( !result )
        {
            std::cout << "res2 failure:" << filename << std::endl;
            delete [] finalBuffer;
            return 1;
        }
    }
    else
    {
        std::cout << "Able to read";
    }
}

```

```

else
{
    std::cerr<< "Not able to put in buffer"<< std::endl;
}
/*
//now, read in smaller buffer extents
reader.DefinePixelExtent(xmin, xmax, ymin, ymax);
len = reader.DefineProperBufferLength();

char* buffer = new char[len];
bool res2 = reader.Read(buffer, len);
if( !res2 ){
    std::cerr<< "res2 failure:" << filename << std::endl;
    return 1;
}
//copy the result into finalBuffer
memcpy(finalBuffer, buffer, len);

//now read the next half of the image
ymin = ymax;
ymax = extent[1];

reader.DefinePixelExtent(xmin, xmax, ymin, ymax);

//std::cerr<< "Success to read image from file: " << filename << std::endl;
unsigned long len2 = reader.DefineProperBufferLength();

char* buffer2 = new char[len2];
bool res3 = reader.Read(buffer2, len2);
if( !res3 ){
    std::cerr<< "res3 failure:" << filename << std::endl;
    return 1;
}
//copy the result into finalBuffer
memcpy(&(finalBuffer[len]), buffer2, len2);

delete [] buffer;
delete [] buffer2;
*/

gdcm::Writer w;
gdcm::File &file = w.GetFile();
gdcm::DataSet &ds = file.GetDataSet();

file.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );

gdcm::UIDGenerator uid;
gdcm::DataElement de( gdcm::Tag(0x8,0x18) ); // SOP Instance UID
de.SetVR( gdcm::VR::UI );
const char *u = uid.Generate();
de.SetByteValue( u, strlen(u) );
ds.Insert( de );

gdcm::DataElement del( gdcm::Tag(0x8,0x16) );
del.SetVR( gdcm::VR::UI );
gdcm::MediaStorage ms( gdcm::MediaStorage::VLWholeSlideMicroscopyImageStorage );
del.SetByteValue( ms.GetString(), strlen(ms.GetString()) );
ds.Insert( del );

const char mystr[] = "MONOCHROME2 ";
gdcm::DataElement de2( gdcm::Tag(0x28,0x04) );
//de.SetTag(gdcm::Tag(0x28,0x04));
de2.SetVR( gdcm::VR::CS );
de2.SetByteValue(mystr, strlen(mystr));
ds.Insert( de2 );

gdcm::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
ds.Insert( Number_Of_Frames.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0010> row = {extent[0]/a};
ds.Insert( row.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0011> col = {extent[1]/a};
ds.Insert( col.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0100> at = {8};
ds.Insert( at.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0002> at1 = {1};

```



```

ds.Insert( at1.GetAsDataElement() );

gdcM::Attribute<0x0028,0x0101> at2 = {8};
ds.Insert( at2.GetAsDataElement() );

gdcM::Attribute<0x0028,0x0102> at3 = {7};
ds.Insert( at3.GetAsDataElement() );
/*
ds1.Remove( gdcM::Tag(0x0028,0x0008) );

gdcM::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
ds1.Insert( Number_Of_Frames.GetAsDataElement() );
*/
theStreamWriter.SetFile(file);

if (!theStreamWriter.WriteImageInformation())
{
    std::cerr << "unable to write image information" << std::endl;
    return 1; //the CanWrite function should prevent getting here, else,
    //that's a test failure
}
std::vector<unsigned int> extent1 = gdcM::ImageHelper::GetDimensionsValue(file);

unsigned short xmax = extent1[0];
unsigned short ymax = extent1[1];
unsigned short theChunkSize = 1;
unsigned short ychunk = extent1[1]/theChunkSize; //go in chunk sizes of theChunkSize
unsigned short zmax = 1;

std::cout << "\n Row: " << extent1[0] << "\n Col : " << extent1[1] << "\n Resolution : " << extent1[2] << std::endl;

if (xmax == 0 || ymax == 0)
{
    std::cerr << "Image has no size, unable to write zero-sized image." << std::endl;
    return 0;
}

int z, y, nexty;
unsigned long prevLen = 0; //when going through the char buffer, make sure to grab
//the bytes sequentially. So, store how far you got in the buffer with each iteration.

for (z = 0; z < zmax; ++z){
    for (y = 0; y < ymax; y += ychunk){
        nexty = y + ychunk;
        if (nexty > ymax) nexty = ymax;
        theStreamWriter.DefinePixelExtent(0, xmax, y, nexty, z, z+1);
        unsigned long len = theStreamWriter.DefineProperBufferLength();
        std::cout << "\n" << len;
        char* finalBuffer1 = new char[len];
        memcpy(finalBuffer1, &(finalBuffer[prevLen]), len);
        std::cout << "\nable to write";

        if (!theStreamWriter.Write(finalBuffer1, len)){
            std::cerr << "writing failure:" << "output.dcm" << " at y = " << y << " and z = " << z << std::endl;
            delete [] finalBuffer1;
            delete [] finalBuffer;
            return 1;
        }
        delete [] finalBuffer1;
        prevLen += len;
    }
}
delete [] finalBuffer;
std::cout << "all is set";

return true;
}

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm Resolution" << std::endl;
        return 1;
    }

    const char *filename = argv[1];
    const char *outfilename = argv[2];

```

```

char *res = argv[3];

int resolution = atoi(res);

gdcm::StreamImageWriter theStreamWriter;

std::ofstream of;
of.open( outfilename, std::ios::out | std::ios::binary );
theStreamWriter.SetStream(of);

// else
// First of get rid of warning/debug message
gdcm::Trace::DebugOn();
gdcm::Trace::WarningOn();

if(!StreamImageRead( theStreamWriter, filename, outfilename, resolution))
    return 1;

uint16_t firstTag1 = 0xfffe;
uint16_t secondTag1 = 0xe0dd;
uint32_t thirdTag1 = 0x00000000;
//uint16_t fourthTag1 = 0xffff;
const int theBufferSize1 = 2*sizeof(uint16_t)+sizeof(uint32_t);
char* tmpBuffer2 = new char[theBufferSize1];
memcpy(&(tmpBuffer2[0]), &firstTag1, sizeof(uint16_t));
memcpy(&(tmpBuffer2[sizeof(uint16_t)]), &secondTag1, sizeof(uint16_t));
memcpy(&(tmpBuffer2[2*sizeof(uint16_t)]), &thirdTag1, sizeof(uint32_t));
//memcpy(&(tmpBuffer2[3*sizeof(uint16_t)]), &fourthTag1, sizeof(uint16_t));
assert( of && !of.eof() && of.good() );
of.write(tmpBuffer2, theBufferSize1);
of.flush();
assert( of );

return 0;
}

```

## 12.97 TemplateEmptyImage.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmFileStreamer.h"
#include "gdcmTag.h"
#include "gdcmTrace.h"
#include "gdcmImageRegionReader.h"
#include "gdcmImageHelper.h"
#include "gdcmWriter.h"
#include "gdcmImageWriter.h"
#include "gdcmTagKeywords.h"
#include "gdcmUIDGenerator.h"

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    const char * filename = argv[1];
    gdcm::ImageRegionReader irr;
    irr.SetFileName( filename );
    const bool b3 = irr.ReadInformation();
    std::cout << b3 << std::endl;
    gdcm::Image & img = irr.GetImage();
    std::cout << img << std::endl;
    // const gdcm::Region & r = irr.GetRegion();
    // std::cout << r << std::endl;
    gdcm::ImageWriter w;
    gdcm::File & file = w.GetFile();
    gdcm::DataSet & ds = file.GetDataSet();

    gdcm::UIDGenerator uid;

```

```

namespace kwd = gdcm::Keywords;
kwd::FrameOfReferenceUID frameref;
frameref.SetValue( uid.Generate() );
// ContentDate
char date[22];
const size_t datelen = 8;
int res = gdcm::System::GetCurrentDateTime( date );
(void)res;
kwd::ContentDate contentdate;
// Do not copy the whole cstring:
contentdate.SetValue( gdcm::DAComp( date, datelen ) );
ds.Insert( contentdate.GetAsDataElement() );
// ContentTime
const size_t timelen = 6 + 1 + 6; // time + milliseconds
kwd::ContentTime contenttime;
// Do not copy the whole cstring:
contenttime.SetValue( gdcm::TMComp( date+datelen, timelen ) );
ds.Insert( contenttime.GetAsDataElement() );
gdcm::MediaStorage ms0 = w.ComputeTargetMediaStorage();
std::cout << ms0 << std::endl;
kwd::SeriesNumber seriesnumber = { 1 };
kwd::InstanceNumber instancenum = { 1 };
kwd::StudyID studyid = { "St1" };
kwd::PatientID patientid = { "P1" };
kwd::SOPClassUID sopclassuid;
kwd::PositionReferenceIndicator pri;
//kwd::Laterality lat;
//kwd::BodyPartExamined bodypartex = { "HEAD" };
kwd::BodyPartExamined bodypartex = { "ANKLE" };
kwd::PatientOrientation pator;
kwd::BurnedInAnnotation bia = { "NO" };
kwd::ConversionType convtype = { "SYN" };
kwd::PresentationLUTShape plutshape = { "IDENTITY" }; // MONOCHROME2
// gdcm will pick the Word in case Byte class is not compatible:
gdcm::MediaStorage ms = gdcm::MediaStorage::MultiframeGrayscaleByteSecondaryCaptureImageStorage;
sopclassuid.SetValue( ms.GetString() );
ds.Insert( instancenum.GetAsDataElement() );
ds.Insert( sopclassuid.GetAsDataElement() );
ds.Insert( seriesnumber.GetAsDataElement() );
ds.Insert( patientid.GetAsDataElement() );
ds.Insert( studyid.GetAsDataElement() );
ds.Insert( frameref.GetAsDataElement() );
ds.Insert( pri.GetAsDataElement() );
//ds.Insert( lat.GetAsDataElement() );
ds.Insert( bodypartex.GetAsDataElement() );
ds.Insert( pator.GetAsDataElement() );
ds.Insert( bia.GetAsDataElement() );
ds.Insert( convtype.GetAsDataElement() );
ds.Insert( plutshape.GetAsDataElement() );
// gdcm::MediaStorage ms1 = w.ComputeTargetMediaStorage();
// std::cout << ms1 << std::endl;
std::cout << ds << std::endl;
gdcm::PixelFormat & pf = img.GetPixelFormat();
pf.SetPixelRepresentation(0); // always overwrite
img.SetSlope(1);
img.SetIntercept(0);
w.SetImage( img );
w.SetFileName( "TemplateImage.dcm" );
if( !w.Write() )
{
    return 1;
}

return 0;
}

```

## 12.98 TraverseModules.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

```

This software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the above copyright notice for more information.

```

=====*/
/*
*/

#include "gdcDefs.h"
#include "gdcGlobal.h"
#include "gdcIODs.h"
#include "gdcIOD.h"
#include "gdcMacros.h"
#include "gdcIODEntry.h"
#include "gdcModules.h"
#include "gdcModule.h"
#include "gdcAnonymizer.h"
#include "gdcDicts.h"

int main(int , char *[])
{
    using namespace gdc;
    static Global &g = Global::GetInstance();

    if( !g.LoadResourcesFiles() )
    {
        return 1;
    }

    static const Defs &defs = g.GetDefs();
    static const Modules &modules = defs.GetModules();
    static const IODs &iods = defs.GetIODs();
    static const Macros &macros = defs.GetMacros();
    static const Dicts &dicts = g.GetDicts();

    std::vector<Tag> tags = gdc::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes();
    for( std::vector<Tag>::const_iterator tit = tags.begin(); tit != tags.end(); ++tit )
    {
        const Tag &tag = *tit;
        const DictEntry &dictentry = dicts.GetDictEntry(tag);
        std::cout << "Processing Attribute: " << tag << " " << dictentry << std::endl;

        IODs::IODMapTypeConstIterator it = iods.Begin();
        for( ; it != iods.End(); ++it )
        {
            const IODs::IODName &name = it->first;
            const IOD &iod = it->second;

            const size_t niods = iod.GetNumberOfIODs();
            // Iterate over each iod entry in order:
            for(unsigned int idx = 0; idx < niods; ++idx)
            {
                const IODEntry &iodentry = iod.GetIODEntry(idx);
                const char *ref = iodentry.GetRef();
                //Usage::UsageType ut = iodentry.GetUsageType();

                const Module &module = modules.GetModule( ref );
                if( module.FindModuleEntryInMacros(macros, tag) )
                {
                    const ModuleEntry &module_entry = module.GetModuleEntryInMacros(macros,tag);
                    Type type = module_entry.GetType();
                    std::cout << "IOD Name: " << name << std::endl;
                    std::cout << "Type: " << type << std::endl;
                }
            }
        }
    }

    return 0;
}

```

## 12.99 VolumeSorter.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

```

Copyright (c) 2006-2011 Mathieu Malaterre  
 All rights reserved.  
 See Copyright.txt or <http://gdcms.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even  
 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR  
 PURPOSE. See the above copyright notice for more information.

```

=====*/
/*
#include "gdcmsorter.h"
#include "gdcmsippsorter.h"
#include "gdcmscanner.h"
#include "gdcmdataset.h"
#include "gdcmsattribute.h"
#include "gdcms-testing.h"

bool mysort1(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
    gdcm::Attribute<0x0020,0x000d> at1;
    at1.Set( ds1 );
    gdcm::Attribute<0x0020,0x000d> at2;
    at2.Set( ds2 );
    return at1 < at2;
}

bool mysort2(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
    gdcm::Attribute<0x0020,0x000e> at1;
    at1.Set( ds1 );
    gdcm::Attribute<0x0020,0x000e> at2;
    at2.Set( ds2 );
    return at1 < at2;
}

bool mysort3(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
    // This is a floating point number is the comparison ok ?
    gdcm::Attribute<0x0020,0x0037> at1;
    at1.Set( ds1 );
    gdcm::Attribute<0x0020,0x0037> at2;
    at2.Set( ds2 );
    return at1 < at2;
}

bool mysort4(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
    // Do the IPP sorting here
    gdcm::Attribute<0x0020,0x0032> iop1;
    gdcm::Attribute<0x0020,0x0037> iop1;
    iop1.Set( ds1 );
    iop1.Set( ds1 );
    gdcm::Attribute<0x0020,0x0032> iop2;
    gdcm::Attribute<0x0020,0x0037> iop2;
    iop2.Set( ds2 );
    iop2.Set( ds2 );
    if( iop1 != iop2 )
    {
        return false;
    }

    // else
    double normal[3];
    normal[0] = iop1[1]*iop1[5] - iop1[2]*iop1[4];
    normal[1] = iop1[2]*iop1[3] - iop1[0]*iop1[5];
    normal[2] = iop1[0]*iop1[4] - iop1[1]*iop1[3];
    double dist1 = 0;
    for (int i = 0; i < 3; ++i) dist1 += normal[i]*iop1[i];
    double dist2 = 0;
    for (int i = 0; i < 3; ++i) dist2 += normal[i]*iop2[i];

    std::cout << dist1 << "," << dist2 << std::endl;
    return dist1 < dist2;
}

int main(int argc, char *argv[])

```

```

{
    const char *extradataroot = gdcm::Testing::GetDataExtraRoot();
    std::string dir1;
    if( argc < 2 )
    {
        if( !extradataroot )
        {
            return 1;
        }
        dir1 = extradataroot;
        dir1 += "/gdcmSampleData/ForSeriesTesting/VariousIncidences/ST1";
    }
    else
    {
        dir1 = argv[1];
    }

    gdcm::Directory d;
    d.Load( dir1, true ); // recursive !
    const gdcm::Directory::FileNamesType &l1 = d.GetFilesNames();
    const size_t nfiles = l1.size();
    std::cout << nfiles << std::endl;

    //if( nfiles != 280 )
    // {
    //     return 1;
    // }

    //d.Print( std::cout );

    gdcm::Scanner s0;
    const gdcm::Tag t1(0x0020,0x000d); // Study Instance UID
    const gdcm::Tag t2(0x0020,0x000e); // Series Instance UID
    //const gdcm::Tag t3(0x0010,0x0010); // Patient's Name
    s0.AddTag( t1 );
    s0.AddTag( t2 );
    //s0.AddTag( t3 );
    //s0.AddTag( t4 );
    //s0.AddTag( t5 );
    //s0.AddTag( t6 );
    bool b = s0.Scan( d.GetFilesNames() );
    if( !b )
    {
        std::cerr << "Scanner failed" << std::endl;
        return 1;
    }

    //s0.Print( std::cout );

    // Only get the DICOM files:
    gdcm::Directory::FileNamesType l2 = s0.GetKeys();
    const size_t nfiles2 = l2.size();
    std::cout << nfiles2 << std::endl;

    if ( nfiles2 > nfiles )
    {
        return 1;
    }

    gdcm::Sorter sorter;
    sorter.SetSortFunction( mysort1 );
    sorter.StableSort( l2 );

    sorter.SetSortFunction( mysort2 );
    sorter.StableSort( sorter.GetFilesNames() );

    sorter.SetSortFunction( mysort3 );
    sorter.StableSort( sorter.GetFilesNames() );

    sorter.SetSortFunction( mysort4 );
    sorter.StableSort( sorter.GetFilesNames() );

    //sorter.Print( std::cout );

    // Let's try to check our result:
    // assume that IPP is precise enough so that we can test floating point equality:
    size_t nvalues = 0;
}
{
    gdcm::Scanner s;
    s.AddTag( gdcm::Tag(0x20,0x32) ); // Image Position (Patient)

```

```

//s.AddTag( gdcm::Tag(0x20,0x37) ); // Image Orientation (Patient)
s.Scan( d.GetFileNames() );

//s.Print( std::cout );

const gdcm::Scanner::ValueType &values = s.GetValues();
nvalues = values.size();
std::cout << "There are " << nvalues << " different type of values" << std::endl;
assert( nfiles2 % nvalues == 0 );
std::cout << "Series is composed of " << (nfiles/nvalues) << " different 3D volumes" << std::endl;
}

gdcm::Directory::FileNamesType sorted_files = sorter.GetFileNames();

// Which means we can take nvalues files at a time and execute gdcm::IPPSorter on it:
gdcm::IPPSorter ippsorter;
gdcm::Directory::FileNamesType sub( sorted_files.begin(), sorted_files.begin() + nvalues);
std::cout << sub.size() << std::endl;
std::cout << sub[0] << std::endl;
std::cout << sub[nvalues-1] << std::endl;
ippsorter.SetComputeZSpacing( false );
if( !ippsorter.Sort( sub ) )
{
    std::cerr << "Could not sort" << std::endl;
    return 1;
}

std::cout << "IPPSorter:" << std::endl;
ippsorter.Print( std::cout );

return 0;
}

```

## 12.100 csa2img.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
/*
 * I do not know what the format is, just guessing from info found on the net:
 *
 * http://atonal.ucdavis.edu/matlab/fmri/spm5/spm_dicom_convert.m
 *
 * This example is an attempt at understanding the format used by SIEMENS
 * their "SIEMENS CSA NON-IMAGE" DICOM file (1.3.12.2.1107.5.9.1)
 *
 * Everything done in this code is for the sole purpose of writing interoperable
 * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
 * If you believe anything in this code violates any law or any of your rights,
 * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
 * find a solution.
 */
#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmCSAHeader.h"
#include "gdcmAttribute.h"
#include "gdcmPrivateTag.h"

#include <math.h>

int main(int argc, char *argv [])
{
    if( argc < 2 ) return 1;
    // gdcmDataExtra/gdcmNonImageData/exCSA_Non-Image_Storage.dcm

```

```

// PHANTOM.MR.CARDIO_COEUR_S_QUENCE_DE_REP_RAGE.9.257.2008.03.20.14.53.25.578125.43151705.IMA
const char *filename = argv[1];

gdcm::Reader reader; // Do not use ImageReader
reader.SetFileName( filename );
if( !reader.Read() )
{
    std::cerr << "Failed to read: " << filename << std::endl;
    return 1;
}

gdcm::CSAHeader csa;
const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

const gdcm::PrivateTag &t1 = csa.GetCSAImageHeaderInfoTag();
//std::cout << t1 << std::endl;
//const gdcm::PrivateTag &t2 = csa.GetCSASeriesHeaderInfoTag();

if( ds.FindDataElement( t1 ) )
{
    csa.LoadFromDataElement( ds.GetDataElement( t1 ) );
    csa.Print( std::cout );
}
int dims[2] = {};
if( csa.FindCSAElementByName( "Columns" ) )
{
    const gdcm::CSAElement &csael = csa.GetCSAElementByName( "Columns" );
    std::cout << csael << std::endl;
    //const gdcm::ByteValue *bv = csael.GetByteValue();
    gdcm::Element<gdcm::VR::IS, gdcm::VM::VM1> el;
    el.Set( csael.GetValue() );
    dims[0] = el.GetValue();
    std::cout << "Columns:" << el.GetValue() << std::endl;
}

if( csa.FindCSAElementByName( "Rows" ) )
{
    const gdcm::CSAElement &csael2 = csa.GetCSAElementByName( "Rows" );
    std::cout << csael2 << std::endl;
    gdcm::Element<gdcm::VR::IS, gdcm::VM::VM1> el2;
    el2.Set( csael2.GetValue() );
    dims[1] = el2.GetValue();
    std::cout << "Rows:" << el2.GetValue() << std::endl;
}

double spacing[2] = { 1., 1. };
bool spacingfound = false;
if( csa.FindCSAElementByName( "PixelSpacing" ) )
{
    const gdcm::CSAElement &csael3 = csa.GetCSAElementByName( "PixelSpacing" );
    if( !csael3.IsEmpty() )
    {
        std::cout << csael3 << std::endl;
        gdcm::Element<gdcm::VR::DS, gdcm::VM::VM2> el3;
        el3.Set( csael3.GetValue() );
        spacing[0] = el3.GetValue(0);
        spacing[1] = el3.GetValue(1);
        std::cout << "PixelSpacing:" << el3.GetValue() << "," << el3.GetValue(1) << std::endl;
        spacingfound = true;
    }
}

if( !spacingfound )
{
    std::cerr << "Problem with PixelSpacing" << std::endl;
    //return 1;
}

if( !dims[0] || !dims[1] )
{
    std::cerr << "Problem with dims" << std::endl;
    return 1;
}

gdcm::ImageWriter writer;

gdcm::Image &image = writer.GetImage();
image.SetNumberOfDimensions( 2 ); // good default
image.SetDimension(0, dims[0] );
image.SetDimension(1, dims[1] );
image.SetSpacing(0, spacing[0] );
image.SetSpacing(1, spacing[1] );

```



```

gdcmm::PixelFormat pixeltype = gdcmm::PixelFormat::INT16; // bytewidth = spm_type('int16','bits')/8;

//unsigned long l = image.GetBufferLength();
//const int p = 1 / (dims[0] * dims[1]);

//image.SetNumberOfDimensions( 3 );
//image.SetDimension(2, p / pixeltype.GetPixelSize() );

gdcmm::PhotometricInterpretation pi;
pi = gdcmm::PhotometricInterpretation::MONOCHROME2;
//pixeltype.SetSamplesPerPixel( );
image.SetPhotometricInterpretation( pi );
image.SetPixelFormat( pixeltype );
//image.SetIntercept( inputimage.GetIntercept() );
//image.SetSlope( inputimage.GetSlope() );

//gdcmm::DataElement pixeldata( gdcmm::Tag(0x7fe1,0x1010) );
//pixeldata.SetByteValue( &outbuf[0], outbuf.size() );
gdcmm::PrivateTag csananimaget(0x7fe1,0x10,"SIEMENS CSA NON-IMAGE");
const gdcmm::DataElement &pixeldata = ds.GetDataElement( csananimaget );
image.SetDataElement( pixeldata );

std::string outfilename = "outcsa.dcm";
//writer.SetFile( reader.GetFile() );
writer.SetFileName( outfilename.c_str() );
if( !writer.Write() )
{
    std::cerr << "could not write: " << outfilename << std::endl;
    return 1;
}

return 0;
}

```

## 12.101 iU22tomultisc.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * iU22 Raw Data extractor
 */
#include "gdcmmReader.h"
#include "gdcmmImageWriter.h"
#include "gdcmmAttribute.h"
#include "gdcmmPrivateTag.h"

#include <math.h>

int main(int argc, char *argv [])
{
    if( argc < 2 ) return 1;
    // IM_001
    const char *filename = argv[1];

    gdcmm::Reader reader; // Do not use ImageReader
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }

    // * The data is simply 8-bit unsigned in the obvious x/y/z order
    // * 200D,300B contains the data
    // * 200D,3001 contains the no. of voxels (416,412,256 in this case)

```

```

// * 200D,3003 contains the voxel sizes (0.156184527398215 /
// 0.1223749613981957 / 0.328479990704639 in this case)

const gdcm::File &file = reader.GetFile();
const gdcm::DataSet &ds = file.GetDataSet();
const gdcm::PrivateTag trawdataus( 0x200d, 0x0b, "Philips US Imaging DD 033" );
const gdcm::DataElement &rawdataus = ds.GetDataElement( trawdataus );

const gdcm::PrivateTag tcolsrowsframes( 0x200d, 0x01, "Philips US Imaging DD 036" );
const gdcm::DataElement &colsrowsframes = ds.GetDataElement( tcolsrowsframes );
// const gdcm::PrivateTag tcolsrowsframes( 0x200d, 0x02, "Philips US Imaging DD 036" );
// this is just a duplicate previous tag.
const gdcm::PrivateTag tvoxelspacing( 0x200d, 0x03, "Philips US Imaging DD 036" );
const gdcm::DataElement &voxelspacing = ds.GetDataElement( tvoxelspacing );

gdcm::Element<gdcm::VR::DS, gdcm::VM::VM3> dims; // Use DS to interpret value stored in LO
dims.SetFromDataElement( colsrowsframes );

gdcm::Element<gdcm::VR::DS, gdcm::VM::VM3> spacing;
spacing.SetFromDataElement( voxelspacing );

gdcm::ImageWriter writer;

gdcm::Image &image = writer.GetImage();
image.SetNumberOfDimensions( 3 ); // good default
image.SetDimension(0, (unsigned int)dims[0] );
image.SetDimension(1, (unsigned int)dims[1] );
image.SetDimension(2, (unsigned int)dims[2] );
image.SetSpacing(0, spacing[0] );
image.SetSpacing(1, spacing[1] );
image.SetSpacing(2, spacing[2] );
gdcm::PixelFormat pixeltype = gdcm::PixelFormat::UINT8;

gdcm::PhotometricInterpretation pi;
pi = gdcm::PhotometricInterpretation::MONOCHROME2;
image.SetPhotometricInterpretation( pi );
image.SetPixelFormat( pixeltype );

image.SetDataElement( rawdataus );

std::string outfilename = "outiu22.dcm";

gdcm::DataElement de( gdcm::Tag(0x8,0x16) ); // SOP Class UID
de.SetVR( gdcm::VR::UI );
gdcm::MediaStorage ms(
    gdcm::MediaStorage::UltrasoundMultiFrameImageStorage );
// gdcm::MediaStorage::MultiframeGrayscaleByteSecondaryCaptureImageStorage );
de.SetByteValue( ms.GetString(), (uint32_t)strlen(ms.GetString()));
writer.GetFile().GetDataSet().Replace( de );

writer.SetFileName( outfilename.c_str() );
if( !writer.Write() )
{
    std::cerr << "could not write: " << outfilename << std::endl;
    return 1;
}

return 0;
}

```

## 12.102 pmsct\_rgb1.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*

```

```

* This example shows how to rewrite a ELSCINT1/PMSCT_RGB1 compressed
* image so that it is readable by most 3rd party software (DICOM does
* not specify this particular encoding).
* This is required for the sake of interoperability with any standard
* conforming DICOM system.
*
* Everything done in this code is for the sole purpose of writing interoperable
* software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
* If you believe anything in this code violates any law or any of your rights,
* please contact us (gdcm-developers@lists.sourceforge.net) so that we can
* find a solution.
*
* Everything you do with this code is at your own risk, since decompression
* algorithm was not written from specification documents.
*
* Special thanks to:
* Jean-Pierre Roux for providing the sample datasets
*/
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmAttribute.h"
#include "gdcmImageWriter.h"

void delta_decode(const unsigned char *data_in, size_t data_size,
  std::vector<unsigned char> &new_stream, unsigned short pc, size_t w, size_t h)
{
    const size_t plane_size = h * w;
    const size_t outputlen = 3 * plane_size;
    new_stream.resize( outputlen );

    assert( data_size != outputlen );
    if( data_size == outputlen )
    {
        return;
    }
    typedef unsigned char byte;
    enum {
        COLORMODE = 0x81,
        ESCMODE = 0x82,
        REPEATMODE = 0x83
    };

    const byte* src = (const byte*)data_in;
    byte* dest = (byte*)new_stream.data();
    union { byte gray; byte rgb[3]; } pixel;
    pixel.rgb[0] = pixel.rgb[1] = pixel.rgb[2] = 0;
    // always start in grayscale mode
    bool graymode = true;
    size_t dx = 1;
    size_t dy = 3;
    // algorithm works with both planar configuration
    // It does produce surprising greenish background color for planar
    // configuration is 0, while the nested Icon SQ display a nice black
    // background
    if (pc)
    {
        dx = plane_size;
        dy = 1;
    }
    size_t ps = plane_size;

    // The following is highly unoptimized as we have nested if statement in a while loop
    // we need to switch from one algorithm to the other (RGB <-> GRAY)
    while (ps)
    {
        // next byte:
        byte b = *src++;
        assert( src < data_in + data_size );
        // mode selection:
        switch ( b )
        {
            case ESCMODE:
                // Used to treat a byte 81/82/83 as a normal byte
                if (graymode)
                {
                    pixel.gray += *src++;
                    dest[0*dx] = pixel.gray;
                    dest[1*dx] = pixel.gray;
                    dest[2*dx] = pixel.gray;
                }
                else

```

```

        {
            pixel.rgb[0] += *src++;
            pixel.rgb[1] += *src++;
            pixel.rgb[2] += *src++;
            dest[0*dx] = pixel.rgb[0];
            dest[1*dx] = pixel.rgb[1];
            dest[2*dx] = pixel.rgb[2];
        }
        dest += dy;
        ps--;
        break;
    case REPEATMODE:
        // repeat mode (RLE)
        b = *src++;
        ps -= b;
        if (graymode)
        {
            while (b-- > 0)
            {
                dest[0*dx] = pixel.gray;
                dest[1*dx] = pixel.gray;
                dest[2*dx] = pixel.gray;
                dest += dy;
            }
        }
        else
        {
            while (b-- > 0)
            {
                dest[0*dx] = pixel.rgb[0];
                dest[1*dx] = pixel.rgb[1];
                dest[2*dx] = pixel.rgb[2];
                dest += dy;
            }
        }
        break;
    case COLORMODE:
        // We are switching from one mode to the other. The stream contains an intermixed
        // compression of RGB codec and GRAY codec. Each one not knowing of the other
        // reset old value to 0.
        if (graymode)
        {
            graymode = false;
            pixel.rgb[0] = pixel.rgb[1] = pixel.rgb[2] = 0;
        }
        else
        {
            graymode = true;
            pixel.gray = 0;
        }
        break;
    default:
        // This is identical to ESCMODE, it would be nicer to use fall-through
        if (graymode)
        {
            pixel.gray += b;
            dest[0*dx] = pixel.gray;
            dest[1*dx] = pixel.gray;
            dest[2*dx] = pixel.gray;
        }
        else
        {
            pixel.rgb[0] += b;
            pixel.rgb[1] += *src++;
            pixel.rgb[2] += *src++;
            dest[0*dx] = pixel.rgb[0];
            dest[1*dx] = pixel.rgb[1];
            dest[2*dx] = pixel.rgb[2];
        }
        dest += dy;
        ps--;
        break;
    } // end switch
} // end while
}

int main(int argc, char *argv [])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcm::Reader reader;

```

```

reader.SetFileName( filename );
if( !reader.Read() )
{
    std::cerr << "Failed to read: " << filename << std::endl;
    return 1;
}
const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

// (07a1,1011) CS [PMSCT_RGB1] # 10,1 Tamar Compression Type
const gdcm::PrivateTag tcompressiontype(0x07a1,0x0011,"ELSCINT1");
if( !ds.FindDataElement( tcompressiontype ) ) return 1;
const gdcm::DataElement& compressiontype = ds.GetDataElement( tcompressiontype );
if( compressiontype.IsEmpty() ) return 1;
const gdcm::ByteValue * bv = compressiontype.GetByteValue();
std::string comprle = "PMSCT_RLE1";
std::string comprgb = "PMSCT_RGB1";
bool isrle = false;
bool isrgb = false;
if( strcmp( bv->GetPointer(), comprle.c_str(), comprle.size() ) == 0 )
{
    isrle = true;
    return 1;
}
if( strcmp( bv->GetPointer(), comprgb.c_str(), comprgb.size() ) == 0 )
{
    isrgb = true;
}
if( !isrgb && !isrle ) return 1;

const gdcm::PrivateTag tcompressedpixeldata(0x07a1,0x000a,"ELSCINT1");
if( !ds.FindDataElement( tcompressedpixeldata ) ) return 1;
const gdcm::DataElement& compressionpixeldata = ds.GetDataElement( tcompressedpixeldata );
if( compressionpixeldata.IsEmpty() ) return 1;
const gdcm::ByteValue * bv2 = compressionpixeldata.GetByteValue();

gdcm::Attribute<0x0028,0x0006> at0;
at0.SetFromDataSet( ds );
gdcm::Attribute<0x0028,0x0010> at1;
at1.SetFromDataSet( ds );
gdcm::Attribute<0x0028,0x0011> at2;
at2.SetFromDataSet( ds );

std::vector<unsigned char> buffer;
delta_decode((const unsigned char*)bv2->GetPointer(), bv2->GetLength(), buffer,
    at0.GetValue(), at1.GetValue(), at2.GetValue() );

gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
pixeldata.SetVR( gdcm::VR::OW );
pixeldata.SetByteValue( (char*)buffer.data(), (uint32_t)buffer.size() );
// TODO we should check that decompress byte buffer match the expected size (row*col*...)

// Add the pixel data element
reader.GetFile().GetDataSet().Replace( pixeldata );

reader.GetFile().GetHeader().SetDataSetTransferSyntax(
    gdcm::TransferSyntax::ExplicitVRLittleEndian);
gdcm::Writer writer;
writer.SetFile( reader.GetFile() );

// Cleanup stuff:
// remove the compressed pixel data:
// FIXME: should I remove more private tags ? all of them ?
// oh well this is just an example
// use gdcm::Anonymizer::RemovePrivateTags if needed...
writer.GetFile().GetDataSet().Remove( compressionpixeldata.GetTag() );
std::string outfilename;
if( argc > 2 )
    outfilename = argv[2];
else
    outfilename = "outrgb.dcm";
writer.SetFileName( outfilename.c_str() );
if( !writer.Write() )
{
    std::cerr << "Failed to write" << std::endl;
    return 1;
}

std::cout << "success !" << std::endl;

return 0;
}

```

## 12.103 rle2img.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
/*
 * This example shows how to rewrite a ELSCINT1/PMSCT_RLE1 compressed
 * image so that it is readable by most 3rd party software (DICOM does
 * not specify this particular encoding).
 * This is required for the sake of interoperability with any standard
 * conforming DICOM system.
 *
 * Everything done in this code is for the sole purpose of writing interoperable
 * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
 * If you believe anything in this code violates any law or any of your rights,
 * please contact us (gdcml-developers@lists.sourceforge.net) so that we can
 * find a solution.
 *
 * Everything you do with this code is at your own risk, since decompression
 * algorithm was not written from specification documents.
 *
 * Special thanks to:
 * Mauro Maiorca for bringing to our attention on this new ELSCINT1
 * compression algorithm : PMSCT_RLE1 (different from the 'LOSSLESS RICE')
 * See post at:
 * http://groups.google.com/group/comp.protocols.dicom/msg/f2b99bf706a7f8ca
 *
 * Thanks to Jesus Spinola, for more datasets,
 * http://www.itk.org/pipermail/insight-users/2008-April/025571.html
 *
 * And last but not least, a very big thank to Ivo van Poorten, without
 * whom we would still be looking at this compressed byte stream as if
 * it was RLE compressed.
 */
#include "gdcmlReader.h"
#include "gdcmlPrivateTag.h"
#include "gdcmlAttribute.h"
#include "gdcmlImageWriter.h"

/* FIXME: Why is PhilipsLosslessRice.dcm a 512x512 image ... */
void delta_decode(const char *inbuffer, size_t length, std::vector<unsigned short> &output)
{
    // RLE pass
    std::vector<char> temp;
    for(size_t i = 0; i < length; ++i)
    {
        if( inbuffer[i] == (char)0xa5 )
        {
            //unsigned char repeat = (unsigned char)inbuffer[i+1] + 1;
            //assert( (unsigned char)inbuffer[i+1] != 255 );
            int repeat = (unsigned char)inbuffer[i+1] + 1;
            char value = inbuffer[i+2];
            while(repeat)
            {
                temp.push_back( value );
                --repeat;
            }
            i+=2;
        }
        else
        {
            temp.push_back( inbuffer[i] );
        }
    }

    // Delta encoding pass
    unsigned short delta = 0;
    for(size_t i = 0; i < temp.size(); ++i)
    {
        if( temp[i] == 0x5a )

```

```

        {
            unsigned char v1 = (unsigned char)temp[i+1];
            unsigned char v2 = (unsigned char)temp[i+2];
            unsigned short value = (unsigned short)(v2 * 256 + v1);
            output.push_back( value );
            delta = value;
            i+=2;
        }
        else
        {
            unsigned short value = (unsigned short)(temp[i] + delta);
            output.push_back( value );
            delta = value;
        }
        //assert( output[output.size()-1] == ref[output.size()-1] );
    }

    if ( output.size() % 2 )
    {
        output.resize( output.size() - 1 );
    }
    std::cout << length << " -> " << output.size() * 2 << std::endl;
}

int main(int argc, char *argv [])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << "input.dcm [output.dcm]" << std::endl;
        std::cerr << "will default to 'outrle.dcm' unless output.dcm is specified."
            << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }
    const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

    // (07a1,1011) CS [PMSCT_RLE1] # 10,1 Tamar Compression Type
    const gdcm::PrivateTag tcompressiontype(0x07a1,0x0011,"ELSCINT1");
    if( !ds.FindDataElement( tcompressiontype ) ) return 1;
    const gdcm::DataElement& compressiontype = ds.GetDataElement( tcompressiontype );
    if ( compressiontype.IsEmpty() ) return 1;
    const gdcm::ByteValue * bv = compressiontype.GetByteValue();
    std::string comprle = "PMSCT_RLE1";
    std::string comprgb = "PMSCT_RGB1";
    bool isrle = false;
    bool isrgb = false;
    if( strncmp( bv->GetPointer(), comprle.c_str(), comprle.size() ) == 0 )
    {
        isrle = true;
    }
    if( strncmp( bv->GetPointer(), comprgb.c_str(), comprgb.size() ) == 0 )
    {
        isrgb = true;
        std::cerr << "See: pmsct_rgb1.cxx instead" << std::endl;
        return 1;
    }
    if( !isrgb && !isrle ) return 1;

    // check if compressed pixel data reside in private or standard tag
    const gdcm::PrivateTag tprivatepixeldata(0x07a1,0x100a,"ELSCINT1");
    const gdcm::Tag tstandardpixeldata(0x7fe0, 0x0010);
    gdcm::Tag tpixeldata;
    if(ds.FindDataElement(tprivatepixeldata)) tpixeldata = tprivatepixeldata;
    else if(ds.FindDataElement(tstandardpixeldata)) tpixeldata = tstandardpixeldata;
    if(!ds.FindDataElement(tpixeldata)) return 1;

    const gdcm::DataElement& compressionpixeldata = ds.GetDataElement( tpixeldata );
    if ( compressionpixeldata.IsEmpty() ) return 1;
    const gdcm::ByteValue * bv2 = compressionpixeldata.GetByteValue();

    gdcm::Attribute<0x0028,0x0010> at1;
    at1.SetFromDataSet( ds );
    gdcm::Attribute<0x0028,0x0011> at2;

```

```

at2.SetFromDataSet( ds );

gdcmm::DataElement pixeldata;
// if standard voxel data element does not exist, create it
if( !reader.GetFile().GetDataSet().FindDataElement( tpixeldata ) )
{
    pixeldata = gdcmm::DataElement( tpixeldata, 0, gdcmm::VR::OW );
}
else{
    pixeldata = reader.GetFile().GetDataSet().GetDataElement( tpixeldata );
}

pixeldata.SetVR( gdcmm::VR::OW );
gdcmm::VL bv2l = bv2->GetLength();
gdcmm::VL at1l = at1.GetValue() * 2; /* sizeof(unsigned short) == 2 */
// Handle special case that is not compressed:
if( bv2l == at1l )
{
    pixeldata.SetByteValue( bv2->GetPointer(), bv2->GetLength() );
}
else
{
    std::vector<unsigned short> buffer;
    delta_decode(bv2->GetPointer(), bv2->GetLength(), buffer);
    pixeldata.SetByteValue( (char*)buffer.data(), (uint32_t)(buffer.size() * sizeof( unsigned short )) );
}
// TODO we should check that decompress byte buffer match the expected size (row*col*...)

// Add the pixel data element
if( reader.GetFile().GetDataSet().FindDataElement( tpixeldata ) )
{
    reader.GetFile().GetDataSet().Replace( pixeldata );
}
else
{
    reader.GetFile().GetDataSet().ReplaceEmpty( pixeldata );
}

reader.GetFile().GetHeader().SetDataSetTransferSyntax(
    gdcmm::TransferSyntax::ExplicitVRLittleEndian);
gdcmm::Writer writer;
writer.SetFile( reader.GetFile() );

// Cleanup stuff:
// This makes the code equivalent to Philips workstation IntelliSpace Portal
if( writer.GetFile().GetDataSet().FindDataElement( tcompressiontype ) )
{
    writer.GetFile().GetDataSet().Remove( gdcmm::Tag(0x07a1,0x1011) );
}
if( writer.GetFile().GetDataSet().FindDataElement( tprivatepixeldata ) )
{
    writer.GetFile().GetDataSet().Remove( gdcmm::Tag(0x07a1,0x100a) );
}

std::string outfilename;
if (argc > 2)
    outfilename = argv[2];
else
    outfilename = "outrlc.dcm";
writer.SetFileName( outfilename.c_str() );
if( !writer.Write() )
{
    std::cerr << "Failed to write" << std::endl;
    return 1;
}

std::cout << "success !" << std::endl;

return 0;
}

```

## 12.104 uid\_unique.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

```



Copyright (c) 2006-2011 Mathieu Malaterre  
 All rights reserved.  
 See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even  
 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR  
 PURPOSE. See the above copyright notice for more information.

```

=====*/
#include "gdcmUIDGenerator.h"

#include <iostream>
#include <string>
#include <set>

int main()
{
    gdcm::UIDGenerator uid;
    //const char myroot[] = "9876543210.9876543210.9876543210.9876543210.9876543210"; // fails in ~40000 tries
    const char myroot[] = "9876543210.9876543210.9876543210";
    uid.SetRoot( myroot );
    std::set<std::string> uids;
    uint64_t wrap = 0;
    uint64_t c = 0;
    while(true)
    {
        const char *unique = uid.Generate();
        //std::cout << unique << std::endl;
        if( c % 10000 == 0 )
        {
            std::cout << "wrap=" << wrap << ",c=" << c << std::endl;
        }
        ++c;
        if( c == 0 )
        {
            wrap++;
        }
        if ( uids.count(unique) == 1 )
        {
            std::cerr << "Failed with: " << unique << std::endl;
            return 1;
        }
        uids.insert( unique );
    }
}

```

## 12.105 DecompressImage.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * This example will take in a DICOM file, and tries to decompress it (actually write it
 * as ImplicitVRLittleEndian Transfer Syntax).
 *
 * Compilation:
 * $ CLASSPATH=gdcm.jar javac ../../gdcm/Examples/Java/DecompressImage.java -d .
 *
 * Usage:
 * $ LD_LIBRARY_PATH=. CLASSPATH=gdcm.jar:. java DecompressImage gdcmData/012345.002.050.dcm out.dcm
 */
import gdcm.*;

public class DecompressImage
{

```

```

public static void main(String[] args) throws Exception
{
    String file1 = args[0];
    String file2 = args[1];
    ImageReader reader = new ImageReader();
    reader.SetFileName( file1 );
    boolean ret = reader.Read();
    if( !ret )
    {
        throw new Exception("Could not read: " + file1 );
    }

    ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
    change.SetTransferSyntax( new TransferSyntax(TransferSyntax.TSType.ImplicitVRLittleEndian) );
    change.SetInput( reader.GetImage() );
    if( !change.Change() )
    {
        throw new Exception("Could not change: " + file1 );
    }

    Image out = change.GetOutput();
    System.out.println( out.toString() );

    // Set the Source Application Entity Title
    FileMetaInformation.SetSourceApplicationEntityTitle( "Just For Fun" );

    ImageWriter writer = new ImageWriter();
    writer.SetFileName( file2 );
    writer.SetFile( reader.GetFile() );
    writer.SetImage( out );
    ret = writer.Write();
    if( !ret )
    {
        throw new Exception("Could not write: " + file2 );
    }
}
}

```

## 12.106 DecompressPixmap.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * This example will take in a DICOM file, and tries to decompress it (actually write it
 * as ImplicitVRLittleEndian Transfer Syntax).
 *
 * Compilation:
 * $ CLASSPATH=gdcm.jar javac ../../gdcm/Examples/Java/DecompressPixmap.java -d .
 *
 * Usage:
 * $ LD_LIBRARY_PATH=. CLASSPATH=gdcm.jar:. java DecompressPixmap gdcmData/012345.002.050.dcm out.dcm
 */
import gdcm.*;

public class DecompressPixmap
{
    public static void main(String[] args) throws Exception
    {
        String file1 = args[0];
        String file2 = args[1];
        PixmapReader reader = new PixmapReader();
        reader.SetFileName( file1 );
        boolean ret = reader.Read();
        if( !ret )

```

```

        {
            throw new Exception("Could not read: " + file1 );
        }

        ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
        change.SetTransferSyntax( new TransferSyntax(TransferSyntax.TSType.ImplicitVRLittleEndian) );
        PixmapToPixmapFilter filter = (PixmapToPixmapFilter)change;
        filter.SetInput( reader.GetPixmap() );
        if( !change.Change() )
        {
            throw new Exception("Could not change: " + file1 );
        }

        // The following does not work in Java/swig 2.0.7
        //Pixmap p = ((PixmapToPixmapFilter)change).GetOutput();
        Pixmap p = change.GetOutputAsPixmap(); // be explicit
        //System.out.println( p.toString() );

        // Set the Source Application Entity Title
        FileMetaInformation.SetSourceApplicationEntityTitle( "Just For Fun" );

        PixmapWriter writer = new PixmapWriter();
        writer.SetFileName( file2 );
        writer.SetFile( reader.GetFile() );
        writer.SetImage( p );
        ret = writer.Write();
        if( !ret )
        {
            throw new Exception("Could not write: " + file2 );
        }
    }
}

```

## 12.107 ExtractImageRegion.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * This small code shows how to use the gdcm.ImageRegionReader API
 * In this example we are taking each frame by frame and dump them to
 * /tmp/frame.raw.
 *
 * Usage:
 * $ LD_LIBRARY_PATH=. CLASSPATH=gdcm.jar:. java ExtractImageRegion input.dcm
 */
import gdcm.*;
import java.io.FileOutputStream;

public class ExtractImageRegion
{
    public static void main(String[] args) throws Exception
    {
        String filename = args[0];

        // instantiate the reader:
        ImageRegionReader reader = new ImageRegionReader();
        reader.SetFileName( filename );

        // pull DICOM info:
        if (!reader.ReadInformation()) return;
        // Get file infos
        File f = reader.GetFile();

        // get some info about image
    }
}

```

```

UIntArrayType dims = ImageHelper.GetDimensionsValue(f);
PixelFormat pf = ImageHelper.GetPixelFormatValue (f);
int pixelsize = pf.GetPixelSize();

// buffer to get the pixels
long buffer_length = dims.get(0) * dims.get(1) * pixelsize;
byte[] buffer = new byte[ (int)buffer_length ];

// define a simple box region.
BoxRegion box = new BoxRegion();
for (int z = 0; z < dims.get(2); z++)
{
    // Define that I want the image 0, full size (dimx x dimy pixels)
    // and do that for each z:
    box.SetDomain(0, dims.get(0) - 1, 0, dims.get(1) - 1, z, z);
    //System.Console.WriteLine( box.toString() );
    reader.SetRegion( box );

    // reader will try to load the uncompressed image region into buffer.
    // the call returns an error when buffer.Length is too small. For instance
    // one can call:
    // long buf_len = reader.ComputeBufferLength(); // take into account pixel size
    // to get the exact size of minimum buffer
    if (reader.ReadIntoBuffer(buffer, buffer_length))
    {
        FileOutputStream fos = new FileOutputStream("/tmp/frame.raw");
        fos.write(buffer);
        fos.close();
    }
    else
    {
        throw new Exception("can't read pixels error");
    }
}
}

```

## 12.108 FileAnonymize.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/

import gdcm.*;

public class FileAnonymize
{
    public static class MyWatcher extends SimpleSubjectWatcher
    {
        public MyWatcher(Subject s) { super(s,"Override String"); }
        protected void ShowProgress(Subject caller, Event evt)
        {
            ProgressEvent pe = ProgressEvent.Cast(evt);
            System.out.println( "This is my progress: " + pe.GetProgress() );
        }
    }

    public static void main(String[] args) throws Exception
    {
        String input = args[0];
        String output = args[1];

        FileAnonymizer fa = new FileAnonymizer();
        fa.SetInputFileName( input );
        fa.SetOutputFileName( output );

        // Empty Operations

```

```

// It will create elements, since those tags are non-registered public elements (2011):
fa.Empty( new Tag(0x0008,0x1313) );
fa.Empty( new Tag(0x0008,0x1317) );
// Remove Operations
// The following Tag are actually carefully chosen, since they refer to SQ:
fa.Remove( new Tag(0x0008,0x2112) );
fa.Remove( new Tag(0x0008,0x9215) );
// Replace Operations
// do not call replace operation on SQ attribute !
fa.Replace( new Tag(0x0018,0x5100), "MYVALUE " );
fa.Replace( new Tag(0x0008,0x1160), "MYOTHERVAL" );

if( !fa.Write() )
{
    System.out.println( "Could not write" );
    return;
}

System.out.println( "success" );
}

```

## 12.109 HelloSimple.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
/*
 * Compilation:
 * $ CLASSPATH=gdcm.jar javac ../../gdcm/Examples/Java/HelloSimple.java -d .
 *
 * Usage:
 * $ LD_LIBRARY_PATH=. CLASSPATH=gdcm.jar:. java HelloSimple gdcmData/012345.002.050.dcm
 */
import gdcm.*;

public class HelloSimple
{
    public static void main(String[] args) throws Exception
    {
        String filename = args[0];
        Reader reader = new Reader();
        reader.SetFileName( filename );
        boolean ret = reader.Read();
        if( !ret )
        {
            throw new Exception("Could not read: " + filename );
        }
        File f = reader.GetFile();
        DataSet ds = f.GetDataSet();

        System.out.println( ds.toString() );

        System.out.println("Success reading: " + filename );
    }
}

```

## 12.110 ReadFiles.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

```

Copyright (c) 2006-2011 Mathieu Malaterre  
 All rights reserved.  
 See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even  
 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR  
 PURPOSE. See the above copyright notice for more information.

```

=====*/
import gdcm.*;
import java.io.File;

public class ReadFiles
{
    static int i = 0;
    public static void process(String path)
    {
        //String path = file.getPath();
        assert PosixEmulation.FileExists(path) : "Problem converting to 8bits";

        System.out.println("Reading: " + path );
        System.out.println("File: " + i++);
        Reader r = new Reader();
        try
        {
            r.SetFileName( path );
            TagSetType skip = new TagSetType();
            skip.insert( new Tag(0x7fe0,0x10) );
            boolean b = r.ReadUpToTag( new Tag(0x88,0x200), skip );
            //System.out.println("DS:\n" + r.GetFile().GetDataSet().toString() );
        }
        finally
        {
            r.delete(); // will properly call C++ destructor and close file descriptor
        }
    }

    // Process only files under dir
    public static void visitAllFiles(File dir)
    {
        if (dir.isDirectory())
        {
            String[] children = dir.list();
            for (int i=0; i<children.length; i++)
            {
                visitAllFiles(new File(dir, children[i]));
            }
        }
        else
        {
            process(dir.getPath());
        }
    }

    public static void waiting (int n)
    {
        long t0, t1;
        t0 = System.currentTimeMillis();
        do
        {
            t1 = System.currentTimeMillis();
        }
        while ((t1 - t0) < (n * 1000));
    }

    public static void main(String[] args) throws Exception
    {
        String directory = args[0];

        Directory gdir = new Directory();
        long n = gdir.Load( directory, true );
        System.out.println( gdir.toString() );
        FilenamesType files = gdir.GetFilenames();
        for( long i = 0; i < n; ++i )
        {
            String path = files.get( (int)i );
            process( path );
        }
    }
}

```

```

    }

    System.out.println( "Java API" );

    //waiting( 10 );
    for( int i = 0; i < 2; ++i )
    {
        File dir = new File(directory);
        visitAllFiles(dir);
    }
}

```

## 12.111 ScanDirectory.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/

import gdcm.*;
import gdcm.Reader;
import gdcm.LookupTable;
import java.io.File;
import java.io.*;
import java.awt.image.*;
import javax.imageio.ImageIO;

public class ScanDirectory
{
    public static class MyWatcher extends SimpleSubjectWatcher
    {
        public MyWatcher(Subject s) { super(s,"Override String"); }
        protected void ShowProgress(Subject caller, Event evt)
        {
            ProgressEvent pe = ProgressEvent.Cast(evt);
            System.out.println( "This is my progress: " + pe.GetProgress() );
        }
    }

    public static byte[] GetAsByte(Bitmap input)
    {
        long len = input.GetBufferLength();
        byte[] buffer = new byte[ (int)len ];
        PhotometricInterpretation pi = input.GetPhotometricInterpretation();
        if( pi.GetType() == PhotometricInterpretation.PIType.MONOCHROME1 )
        {
            ImageChangePhotometricInterpretation icpi = new ImageChangePhotometricInterpretation();
            icpi.SetInput( input );
            icpi.SetPhotometricInterpretation(
                new PhotometricInterpretation(
                    PhotometricInterpretation.PIType.MONOCHROME2 ) );
            if( icpi.Change() )
            {
                Bitmap output = icpi.GetOutput();
                output.GetArray( buffer );
            }
            return buffer;
        }
        else
        {
            input.GetArray( buffer );
            return buffer;
        }
    }

    public static short[] GetAsShort(Bitmap input)
    {
        long len = input.GetBufferLength(); // length in bytes

```

```

short[] buffer = new short[ (int)len / 2 ];
PhotometricInterpretation pi = input.GetPhotometricInterpretation();
if( pi.GetType() == PhotometricInterpretation.PIType.MONOCHROME1 )
{
    ImageChangePhotometricInterpretation icpi = new ImageChangePhotometricInterpretation();
    icpi.SetInput( input );
    icpi.SetPhotometricInterpretation(
        new PhotometricInterpretation(
            PhotometricInterpretation.PIType.MONOCHROME2 ) );
    if( icpi.Change() )
    {
        Bitmap output = icpi.GetOutput();
        output.GetArray( buffer );
    }
    return buffer;
}
else
{
    input.GetArray( buffer );
    return buffer;
}
}

public static boolean WritePNG(Bitmap input, String outfilename )
{
    int imageType = BufferedImage.TYPE_CUSTOM;
    PixelFormat pf = input.GetPixelFormat();
    PhotometricInterpretation pi = input.GetPhotometricInterpretation();
    // We need to handle both public and private icon
    // It could well be that we are getting an RGB Icon or 16 bits Icon:
    ColorModel colorModel = null;
    if( pf.GetSamplesPerPixel() == 1 )
    {
        if( pi.GetType() == PhotometricInterpretation.PIType.MONOCHROME1
            || pi.GetType() == PhotometricInterpretation.PIType.MONOCHROME2 )
        {
            if( pf.GetScalarType() == PixelFormat.ScalarType.UINT8 )
            {
                imageType = BufferedImage.TYPE_BYTE_GRAY;
            }
            else if( pf.GetScalarType() == PixelFormat.ScalarType.UINT12 )
            {
                imageType = BufferedImage.TYPE_USHORT_GRAY;
            }
            else if( pf.GetScalarType() == PixelFormat.ScalarType.UINT16 )
            {
                imageType = BufferedImage.TYPE_USHORT_GRAY;
            }
        }
        else if( pi.GetType() == PhotometricInterpretation.PIType.PALETTE_COLOR )
        {
            LookupTable lut = input.GetLUT();
            long rl = lut.GetLUTLength( LookupTable.LookupTableType.RED );
            byte[] rbuf = new byte[ (int)rl ];
            long r12 = lut.GetLUT( LookupTable.LookupTableType.RED, rbuf );
            assert rl == r12;
            long gl = lut.GetLUTLength( LookupTable.LookupTableType.GREEN );
            byte[] gbuf = new byte[ (int)gl ];
            long g12 = lut.GetLUT( LookupTable.LookupTableType.GREEN, gbuf );
            assert gl == g12;
            long bl = lut.GetLUTLength( LookupTable.LookupTableType.BLUE );
            byte[] bbuf = new byte[ (int)bl ];
            long b12 = lut.GetLUT( LookupTable.LookupTableType.BLUE, bbuf );
            assert bl == b12;
            colorModel = new IndexColorModel(8, (int)rl, rbuf, gbuf, bbuf);
            // For code below
            imageType = BufferedImage.TYPE_BYTE_GRAY;
        }
    }
    else if( pf.GetSamplesPerPixel() == 3 )
    {
        if( pf.GetScalarType() == PixelFormat.ScalarType.UINT8 )
        {
            // FIXME should be TYPE_3BYTE_RGB
            imageType = BufferedImage.TYPE_3BYTE_BGR;
        }
    }
}

//System.out.println( "pf: " + pf.toString() );
//System.out.println( "pi: " + pi.toString() );
long width = input.GetDimension(0);
long height = input.GetDimension(0);
BufferedImage bi;

```



```

if( pi.GetType() == PhotometricInterpretation.PITYPE.PALETTE_COLOR )
{
    bi = new BufferedImage(colorModel,
        colorModel.createCompatibleWritableRaster((int)width, (int)height),
        false, null);
}
else
{
    bi = new BufferedImage((int)width, (int)height, imageType);
}
WritableRaster wr = bi.getRaster();
//System.out.println( "imagetype: " + imageType );
if( imageType == BufferedImage.TYPE_BYTE_GRAY
    || imageType == BufferedImage.TYPE_3BYTE_BGR )
{
    byte[] buffer = GetAsByte( input );
    wr.setDataElements (0, 0, (int)width, (int)height, buffer);
}
else if( imageType == BufferedImage.TYPE_USHORT_GRAY )
{
    short[] buffer = GetAsShort( input );
    wr.setDataElements (0, 0, (int)width, (int)height, buffer);
}

File outputfile = new File( outfilename );
try {
    ImageIO.write(bi, "png", outputfile);
} catch (IOException e) {
    return false;
}
return true;
}

public static void main(String[] args) throws Exception
{
    String directory = args[0];

    Directory d = new Directory();
    long nfiles = d.Load( directory, true );
    if(nfiles == 0)
    {
        throw new Exception("No files found");
    }
    // System.out.println( "Files:\n" + d.toString() );
    FilenamesType fns = d.GetFilenames();

    //Scanner s = new Scanner();
    SmartPtrScan sscan = Scanner.New();
    Scanner s = sscan.__ref__();
    //SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(s, "MySimple");
    MyWatcher watcher = new MyWatcher(s);
    Tag[] tagarray = {
        new Tag(0x0010, 0x0010), // PatientName
        new Tag(0x0010, 0x0020), // PatientID
        new Tag(0x0010, 0x0030), // PatientBirthDate
        new Tag(0x0010, 0x0040), // PatientSex
        new Tag(0x0010, 0x1010), // PatientAge
        new Tag(0x0020, 0x000d), // StudyInstanceUID
        new Tag(0x0020, 0x0010), // StudyID
        new Tag(0x0008, 0x0020), // StudyDate
        new Tag(0x0008, 0x1030), // StudyDescription
        new Tag(0x0020, 0x000e), // SeriesInstanceUID
        new Tag(0x0020, 0x0011), // SeriesNumber
        new Tag(0x0008, 0x0021), // SeriesDate
        new Tag(0x0008, 0x103e), // SeriesDescription
        new Tag(0x0008, 0x0090), // ReferringPhysicianName
        new Tag(0x0008, 0x0060), // Modality
        new Tag(0x0054, 0x0400), // ImageID ?? Should be Instance number ??
        new Tag(0x0008, 0x0018), // SOPInstanceUID
        new Tag(0x0008, 0x0032), // AcquisitionTime
        new Tag(0x0008, 0x0033), // ContentTime
        new Tag(0x0020, 0x0013), // InstanceNumber
        new Tag(0x0020, 0x1041), // SliceLocation
        new Tag(0x0018, 0x0050), // SliceThickness ?? Eg. Enhanced MR Image Storage
        new Tag(0x0008, 0x0080), // InstitutionName
        new Tag(0x0028, 0x1050), // WindowCenter
        new Tag(0x0028, 0x1051), // WindowWidth
    };
    for( Tag t : tagarray ) {
        //System.out.println( "Tag: " + t.toString() );
        s.AddTag( t );
    }
}

```

```

    }
    boolean b = s.Scan( fns );
    if(!b)
    {
        throw new Exception("Could not scan");
    }
    String fn0 = fns.get(0);
    TagToValue mappings = s.GetMapping( fn0 );
    System.out.println( "mappings size: " + mappings.size() );
    for( Tag tag : tagarray ) {
        if( mappings.has_key( tag ) ) {
            String val = mappings.get( tag );
            System.out.println( "tag/val: " + tag + "->" + val );
        }
    }

    for( long idx = 0; idx < fns.size(); ++idx )
    {
        Reader r = new Reader();
        String fn = fns.get( (int)idx );
        String outfn = fn + ".png";
        r.SetFileName( fn );
        TagSetType tst = new TagSetType();
        tst.insert( new Tag(0x7fe0,0x10) );
        b = r.ReadUpToTag( new Tag(0x88,0x200), tst );
        UIntArrayType dims = ImageHelper.GetDimensionsValue( r.GetFile() );
        if( b )
        {
            IconImageFilter iif = new IconImageFilter();
            System.out.println( "Processing: " + fn );

            iif.SetFile( r.GetFile() );
            b = iif.Extract();
            if( b )
            {
                Bitmap icon = iif.GetIconImage(0);
                WritePNG(icon, outfn);
            }
            else
            {
                ImageReader ir = new ImageReader();
                ir.SetFileName( fn );
                if( ir.Read() )
                {
                    Image img = ir.GetImage();
                    StringFilter sf = new StringFilter();
                    sf.SetFile( r.GetFile() );
                    String strval = sf.ToString( new Tag(0x0028,0x0120) );
                    IconImageGenerator iig = new IconImageGenerator();
                    iig.SetPixmap( img );
                    iig.AutoPixelMinMax( true );
                    try {
                        double val = Double.parseDouble( strval );
                        iig.SetOutsideValuePixel( val );
                    }
                    catch ( NumberFormatException e ) {
                    }
                    iig.ConvertRGBToPaletteColor( false );
                    long idims[] = { 128, 128 };
                    iig.SetOutputDimensions( idims );
                    iig.Generate();
                    Bitmap icon = iig.GetIconImage();
                    WritePNG(icon, outfn);
                }
            }
        }
    }

    System.out.println( "Scan:\n" + s.toString() );

    System.out.println( "success" );
}
}

```

## 12.112 SimplePrint.java

```

/*=====

```

```

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * Compilation:
 * $ CLASSPATH=gdcm.jar javac ../../gdcm/Examples/Java/SimplePrint.java -d .
 *
 * Usage:
 * $ LD_LIBRARY_PATH=. CLASSPATH=gdcm.jar:. java SimplePrint gdcmData/012345.002.050.dcm
 */
import gdcm.*;

public class SimplePrint
{
    public static void RecurseDataSet(File f, DataSet ds, String indent)
    {
        JavaDataSet cds = new JavaDataSet(ds);
        while(!cds.IsAtEnd())
        {
            DataElement de = cds.GetCurrent();
            // Compute VR from the toplevel file, and the currently processed dataset:
            VR vr = DataSetHelper.ComputeVR(f, ds, de.GetTag() );

            if( vr.Compatible( new VR(VR.VRType.SQ) ) )
            {
                long uvl = de.GetVL().GetValueLength(); // Test cast is ok
                System.out.println( indent + de.GetTag().toString() + ":" + uvl ); // why not ?
                //SequenceOfItems sq = de.GetSequenceOfItems();
                // GetValueAsSQ handle more cases than GetSequenceOfItems
                SmartPtrSQ sq = de.GetValueAsSQ();
                long n = sq.GetNumberOfItems();
                for( long i = 1; i <= n; i++) // item starts at 1, not 0
                {
                    Item item = sq.GetItem( i );
                    DataSet nested = item.GetNestedDataSet();
                    RecurseDataSet( f, nested, indent + " " );
                }
            }
            else
            {
                System.out.println( indent + de.toString() );
            }
            cds.Next();
        }
    }

    public static void main(String[] args) throws Exception
    {
        String filename = args[0];
        Reader reader = new Reader();
        reader.SetFileName( filename );
        boolean ret = reader.Read();
        if( !ret )
        {
            throw new Exception("Could not read: " + filename );
        }
        File f = reader.GetFile();
        DataSet ds = f.GetDataSet();

        RecurseDataSet( f, ds, "" );
    }
}

```

## 12.113 AddPrivateAttribute.py

```

00001
00014
00015 """

```

```

00016 Usage:
00017
00018     python AddPrivateAttribute.py input.dcm output.dcm
00019
00020
00021 """
00022
00023 import sys
00024 import gdcm
00025
00026 if __name__ == "__main__":
00027
00028     file1 = sys.argv[1]
00029     file2 = sys.argv[2]
00030
00031     r = gdcm.Reader()
00032     r.SetFileName( file1 )
00033     if not r.Read():
00034         sys.exit(1)
00035
00036     f = r.GetFile()
00037     ds = f.GetDataSet()
00038
00039     # Create a dataelement
00040     de = gdcm.DataElement(gdcm.Tag(0x0051, 0x1011))
00041     de.SetByteStringValue("p2")
00042     de.SetVR(gdcm.VR(gdcm.VR.SH))
00043
00044     ds.Insert(de)
00045
00046     w = gdcm.Writer()
00047     w.SetFile( f )
00048     w.SetFileName( file2 )
00049     if not w.Write():
00050         sys.exit(1)

```

## 12.114 ConvertMPL.py

```

00001
00014
00015 """
00016 display a DICOM image with matplotlib via numpy
00017
00018 Caveats:
00019 - Does not support UINT12/INT12
00020
00021 Usage:
00022
00023     python ConvertNumpy.py "IM000000"
00024
00025 Thanks:
00026     plotting example - Ray Schumacher 2009
00027 """
00028
00029 import gdcm
00030 import numpy
00031 from pylab import *
00032
00033
00034 def get_gdcm_to_numpy_typemap():
00035     """Returns the GDCM Pixel Format to numpy array type mapping."""
00036     _gdcm_np = {gdcm.PixelFormat.UINT8 :numpy.int8,
00037                 gdcm.PixelFormat.INT8  :numpy.uint8,
00038                 gdcm.PixelFormat.UINT16:numpy.uint16,
00039                 gdcm.PixelFormat.INT16 :numpy.int16,
00040                 gdcm.PixelFormat.UINT32:numpy.uint32,
00041                 gdcm.PixelFormat.INT32 :numpy.int32,
00042                 gdcm.PixelFormat.FLOAT32:numpy.float32,
00043                 gdcm.PixelFormat.FLOAT64:numpy.float64 }
00044     return _gdcm_np
00045
00046 def get_numpy_array_type(gdcm_pixel_format):
00047     """Returns a numpy array typecode given a GDCM Pixel Format."""
00048     return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]
00049
00050 def gdcm_to_numpy(image):
00051     """Converts a GDCM image to a numpy array.

```

```

00052     """
00053     pf = image.GetPixelFormat().GetScalarType()
00054     print 'pf', pf
00055     print image.GetPixelFormat().GetScalarTypeAsString()
00056     assert pf in get_gdcm_to_numpy_typemap().keys(), \
00057         "Unsupported array type %s"%pf
00058     d = image.GetDimension(0), image.GetDimension(1)
00059     print 'Image Size: %d x %d' % (d[0], d[1])
00060     dtype = get_numpy_array_type(pf)
00061     gdcm_array = image.GetBuffer()
00062
00063     result = numpy.frombuffer(gdcm_array, dtype=dtype).astype(float)
00064
00065     result.shape = d
00066     return result
00067
00068 if __name__ == "__main__":
00069     import sys
00070     r = gdcm.ImageReader()
00071     filename = sys.argv[1]
00072     r.SetFileName( filename )
00073     if not r.Read(): sys.exit(1)
00074     numpy_array = gdcm_to_numpy( r.GetImage() )
00075
00076     subplot(111)# one plot, on left
00077     title(filename)
00078
00079     imshow(numpy_array, interpolation='bilinear', cmap=cm.jet)
00080
00081     subplots_adjust(bottom=0.1, right=0.8, top=0.9)
00082     cax = axes([0.85, 0.1, 0.075, 0.8])
00083     colorbar(cax=cax)
00084     title('values')
00085     get_current_fig_manager().window.title('plot')
00086     show()

```

## 12.115 ConvertNumpy.py

```

00001
00002
00003 """
00004 This module add support for converting a gdcm.Image to a numpy array.
00005
00006 Caveats:
00007 - Does not support UINT12/INT12
00008
00009 Removed:
00010 - float16 is defined in GDCM API but no implementation exist for it ...
00011 """
00012
00013 import gdcm
00014 import numpy
00015
00016 def get_gdcm_to_numpy_typemap():
00017     """Returns the GDCM Pixel Format to numpy array type mapping."""
00018     _gdcm_np = {gdcm.PixelFormat.UINT8 :numpy.uint8,
00019                 gdcm.PixelFormat.INT8 :numpy.int8,
00020                 #gdcm.PixelFormat.UINT12 :numpy.uint12,
00021                 #gdcm.PixelFormat.INT12 :numpy.int12,
00022                 gdcm.PixelFormat.UINT16 :numpy.uint16,
00023                 gdcm.PixelFormat.INT16 :numpy.int16,
00024                 gdcm.PixelFormat.UINT32 :numpy.uint32,
00025                 gdcm.PixelFormat.INT32 :numpy.int32,
00026                 #gdcm.PixelFormat.FLOAT16:numpy.float16,
00027                 gdcm.PixelFormat.FLOAT32:numpy.float32,
00028                 gdcm.PixelFormat.FLOAT64:numpy.float64 }
00029     return _gdcm_np
00030
00031 def get_numpy_array_type(gdcm_pixel_format):
00032     """Returns a numpy array typecode given a GDCM Pixel Format."""
00033     return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]
00034
00035 def gdcm_to_numpy(image):
00036     """Converts a GDCM image to a numpy array.
00037     """
00038     pf = image.GetPixelFormat()
00039
00040

```

```

00052     assert pf.GetScalarType() in get_gdcm_to_numpy_typemap().keys(), \
00053         "Unsupported array type %s"%pf
00054
00055     shape = image.GetDimension(0) * image.GetDimension(1), pf.GetSamplesPerPixel()
00056     if image.GetNumberOfDimensions() == 3:
00057         shape = shape[0] * image.GetDimension(2), shape[1]
00058
00059     dtype = get_numpy_array_type(pf.GetScalarType())
00060     gdcm_array = image.GetBuffer()
00061     result = numpy.frombuffer(gdcm_array, dtype=dtype)
00062     result.shape = shape
00063     return result
00064
00065 if __name__ == "__main__":
00066     import sys
00067     r = gdcm.ImageReader()
00068     filename = sys.argv[1]
00069     r.SetFileName( filename )
00070     if not r.Read():
00071         sys.exit(1)
00072
00073     numpy_array = gdcm_to_numpy( r.GetImage() )
00074     print numpy_array

```

## 12.116 ConvertPIL.py

```

00001
00014
00015 """
00016 save a DICOM image with PIL via numpy
00017
00018 Caveats:
00019 - Does not support UINT12/INT12
00020
00021 Usage:
00022
00023 python ConvertNumpy.py "IM000000"
00024
00025 Thanks:
00026 plotting example - Ray Schumacher 2009
00027 """
00028
00029 import gdcm
00030 import numpy
00031 from PIL import Image, ImageOps
00032
00033
00034 def get_gdcm_to_numpy_typemap():
00035     """Returns the GDCM Pixel Format to numpy array type mapping."""
00036     _gdcm_np = {gdcm.PixelFormat.UINT8 :numpy.int8,
00037                 gdcm.PixelFormat.INT8 :numpy.uint8,
00038                 gdcm.PixelFormat.UINT16 :numpy.uint16,
00039                 gdcm.PixelFormat.INT16 :numpy.int16,
00040                 gdcm.PixelFormat.UINT32 :numpy.uint32,
00041                 gdcm.PixelFormat.INT32 :numpy.int32,
00042                 gdcm.PixelFormat.FLOAT32 :numpy.float32,
00043                 gdcm.PixelFormat.FLOAT64 :numpy.float64 }
00044     return _gdcm_np
00045
00046 def get_numpy_array_type(gdcm_pixel_format):
00047     """Returns a numpy array typecode given a GDCM Pixel Format."""
00048     return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]
00049
00050 def gdcm_to_numpy(image):
00051     """Converts a GDCM image to a numpy array.
00052     """
00053     pf = image.GetPixelFormat().GetScalarType()
00054     print 'pf', pf
00055     print image.GetPixelFormat().GetScalarTypeAsString()
00056     assert pf in get_gdcm_to_numpy_typemap().keys(), \
00057         "Unsupported array type %s"%pf
00058     d = image.GetDimension(0), image.GetDimension(1)
00059     print 'Image Size: %d x %d' % (d[0], d[1])
00060     dtype = get_numpy_array_type(pf)
00061     gdcm_array = image.GetBuffer()
00062     result = numpy.frombuffer(gdcm_array, dtype=dtype)
00063     maxV = float(result[result.argmax()])

```

```

00064
00065     result = numpy.log(result+50)
00066     maxV = float(result[result.argmax()])
00067     result = result*(2.**8/maxV)
00068     result.shape = d
00069     return result
00070
00071 if __name__ == "__main__":
00072     import sys
00073     r = gdcmm.ImageReader()
00074     filename = sys.argv[1]
00075     r.SetFileName( filename )
00076     if not r.Read(): sys.exit(1)
00077     numpy_array = gdcmm_to_numpy( r.GetImage() )
00078
00079     pilImage = Image.frombuffer('L',
00080                                numpy_array.shape,
00081                                numpy_array.astype(numpy.uint8),
00082                                'raw','L',0,1)
00083
00084     pilImage = ImageOps.autocontrast(pilImage, cutoff=.1)
00085     pilImage.save(sys.argv[1]+'.jpg')

```

## 12.117 CreateRAWStorage.py

```

00001
00014
00015 """
00016     <uid value="1.2.840.10008.5.1.4.1.1.66" name="Raw Data Storage" type="SOP Class" part="PS 3.4"
00017     retired="false"/>
00018 """
00019 import gdcmm
00020 import sys,os
00021
00022 if __name__ == "__main__":
00023     r = gdcmm.Reader()
00024     # Will require Testing...
00025     dataroot = gdcmm.Testing.GetDataRoot()
00026     filename = os.path.join( dataroot, '012345.002.050.dcm' )
00027     r.SetFileName( filename )
00028     r.Read()
00029     f = r.GetFile()
00030     ds = f.GetDataSet()
00031
00032     uid = "1.2.840.10008.5.1.4.1.1.66"
00033     # f = gdcmm.File()
00034     # ds = f.GetDataSet()
00035     de = gdcmm.DataElement( gdcmm.Tag(0x0008,0x0016) )
00036     de.SetByteStringValue( uid )
00037     vr = gdcmm.VR( gdcmm.VR.UI )
00038     de.SetVR( vr )
00039     ds.Replace( de )
00040
00041     ano = gdcmm.Anonymizer()
00042     ano.SetFile( r.GetFile() )
00043     ano.RemovePrivateTags()
00044     ano.RemoveGroupLength()
00045     taglist = [
00046         gdcmm.Tag(0x0008,0x0008),
00047         gdcmm.Tag(0x0008,0x0022),
00048         gdcmm.Tag(0x0008,0x0032),
00049         gdcmm.Tag(0x0008,0x2111),
00050         gdcmm.Tag(0x0008,0x1150),
00051         gdcmm.Tag(0x0008,0x1155),
00052         gdcmm.Tag(0x0008,0x0100),
00053         gdcmm.Tag(0x0008,0x0102),
00054         gdcmm.Tag(0x0008,0x0104),
00055         gdcmm.Tag(0x0040,0xa170),
00056         gdcmm.Tag(0x0008,0x2112),
00057         gdcmm.Tag(0x0008,0x0100),
00058         gdcmm.Tag(0x0008,0x0102),
00059         gdcmm.Tag(0x0008,0x0104),
00060         gdcmm.Tag(0x0008,0x9215),
00061         gdcmm.Tag(0x0018,0x0010),
00062         gdcmm.Tag(0x0018,0x0022),
00063         gdcmm.Tag(0x0018,0x0050),

```

```

00064     gdcM.Tag(0x0018,0x0060),
00065     gdcM.Tag(0x0018,0x0088),
00066     gdcM.Tag(0x0018,0x0090),
00067     gdcM.Tag(0x0018,0x1040),
00068     gdcM.Tag(0x0018,0x1100),
00069     gdcM.Tag(0x0018,0x1110),
00070     gdcM.Tag(0x0018,0x1111),
00071     gdcM.Tag(0x0018,0x1120),
00072     gdcM.Tag(0x0018,0x1130),
00073     gdcM.Tag(0x0018,0x1150),
00074     gdcM.Tag(0x0018,0x1151),
00075     gdcM.Tag(0x0018,0x1152),
00076     gdcM.Tag(0x0018,0x1160),
00077     gdcM.Tag(0x0018,0x1190),
00078     gdcM.Tag(0x0018,0x1210),
00079     gdcM.Tag(0x0020,0x0012),
00080     gdcM.Tag(0x0020,0x0032),
00081     gdcM.Tag(0x0020,0x0037),
00082     gdcM.Tag(0x0020,0x1041),
00083     gdcM.Tag(0x0020,0x4000),
00084     gdcM.Tag(0x0028,0x0002),
00085     gdcM.Tag(0x0028,0x0004),
00086     gdcM.Tag(0x0028,0x0010),
00087     gdcM.Tag(0x0028,0x0011),
00088     gdcM.Tag(0x0028,0x0030),
00089     gdcM.Tag(0x0028,0x0100),
00090     gdcM.Tag(0x0028,0x0101),
00091     gdcM.Tag(0x0028,0x0102),
00092     gdcM.Tag(0x0028,0x0103),
00093     gdcM.Tag(0x0028,0x1052),
00094     gdcM.Tag(0x0028,0x1053),
00095     gdcM.Tag(0x0028,0x2110),
00096     gdcM.Tag(0x0028,0x2112),
00097     gdcM.Tag(0x7fe0,0x0010),
00098     gdcM.Tag(0x0018,0x0020),
00099     gdcM.Tag(0x0018,0x0021),
00100     gdcM.Tag(0x0018,0x0023),
00101     gdcM.Tag(0x0018,0x0025),
00102     gdcM.Tag(0x0018,0x0080),
00103     gdcM.Tag(0x0018,0x0081),
00104     gdcM.Tag(0x0018,0x0083),
00105     gdcM.Tag(0x0018,0x0084),
00106     gdcM.Tag(0x0018,0x0085),
00107     gdcM.Tag(0x0018,0x0086),
00108     gdcM.Tag(0x0018,0x0087),
00109     gdcM.Tag(0x0018,0x0091),
00110     gdcM.Tag(0x0018,0x0093),
00111     gdcM.Tag(0x0018,0x0094),
00112     gdcM.Tag(0x0018,0x0095),
00113     gdcM.Tag(0x0018,0x1088),
00114     gdcM.Tag(0x0018,0x1090),
00115     gdcM.Tag(0x0018,0x1094),
00116     gdcM.Tag(0x0018,0x1250),
00117     gdcM.Tag(0x0018,0x1251),
00118     gdcM.Tag(0x0018,0x1310),
00119     gdcM.Tag(0x0018,0x1312),
00120     gdcM.Tag(0x0018,0x1314),
00121     gdcM.Tag(0x0018,0x1315),
00122     gdcM.Tag(0x0018,0x1316),
00123     gdcM.Tag(0x0020,0x0110),
00124     gdcM.Tag(0x0028,0x0120),
00125     gdcM.Tag(0x0028,0x1050),
00126     gdcM.Tag(0x0028,0x1051)
00127 ]
00128 for tag in taglist:
00129     #print tag
00130     ano.Remove( tag )
00131
00132 # special handling
00133 gen = gdcM.UIDGenerator()
00134 ano.Replace( gdcM.Tag(0x0008,0x9123), gen.Generate() )
00135 #ano.Empty( gdcM.Tag(0x0040,0x0555) )
00136
00137
00138 #
00139 # uid = gen.Generate()
00140 # de.SetTag( gdcM.Tag(0x0008,0x0018) )
00141 # de.SetByteStringValue( uid )
00142 # ds.Insert( de )
00143
00144 # init FMI now:

```



```

00145     #fmi = f.GetHeader()
00146     #ts = gdcm.TransferSyntax()
00147     #print ts
00148     #fmi.SetDataSetTransferSyntax( ts ) # default
00149     #print fmi.GetDataSetTransferSyntax()
00150     #de.SetTag( gdcm.Tag(0x0002,0x0010) )
00151     #uid = "1.2.840.10008.1.2"
00152     #de.SetByteStringValue( uid )
00153     #fmi.Insert( de )
00154 #   f.SetHeader( r.GetFile().GetHeader() )
00155
00156     writer = gdcm.Writer()
00157     writer.SetFile( ano.GetFile() )
00158     writer.SetFileName( "rawstorage.dcm" );
00159     writer.Write()

```

## 12.118 DecompressImage.py

```

00001
00014
00015 """
00016 Usage:
00017
00018     python DecompressImage.py gdcmData/012345.002.050.dcm decompress.dcm
00019 """
00020
00021 import gdcm
00022 import sys
00023
00024 if __name__ == "__main__":
00025
00026     file1 = sys.argv[1]
00027     file2 = sys.argv[2]
00028
00029     r = gdcm.ImageReader()
00030     r.SetFileName( file1 )
00031     if not r.Read():
00032         sys.exit(1)
00033
00034     # check GetFragment API:
00035     pd = r.GetFile().GetDataSet().GetDataElement( gdcm.Tag(0x7fe0, 0x0010) )
00036     frags = pd.GetSequenceOfFragments();
00037     frags.GetFragment(0);
00038
00039     ir = r.GetImage()
00040     w = gdcm.ImageWriter()
00041     image = w.GetImage()
00042
00043     image.SetNumberOfDimensions( ir.GetNumberOfDimensions() );
00044     dims = ir.GetDimensions();
00045     print ir.GetDimension(0);
00046     print ir.GetDimension(1);
00047     print "Dims:", dims
00048
00049     # Just for fun:
00050     dircos = ir.GetDirectionCosines()
00051     t = gdcm.Orientation.GetType( tuple(dircos) )
00052     l = gdcm.Orientation.GetLabel( t )
00053     print "Orientation label:", l
00054
00055     image.SetDimension(0, ir.GetDimension(0) );
00056     image.SetDimension(1, ir.GetDimension(1) );
00057
00058     pixeltype = ir.GetPixelFormat();
00059     image.SetPixelFormat( pixeltype );
00060
00061     pi = ir.GetPhotometricInterpretation();
00062     image.SetPhotometricInterpretation( pi );
00063
00064     pixeldata = gdcm.DataElement( gdcm.Tag(0x7fe0,0x0010) )
00065     str1 = ir.GetBuffer()
00066     #print ir.GetBufferLength()
00067     pixeldata.SetByteStringValue( str1 )
00068     image.SetDataElement( pixeldata )
00069
00070     w.SetFileName( file2 )
00071     w.SetFile( r.GetFile() )

```

```

00072 w.SetImage( image )
00073 if not w.Write():
00074     sys.exit(1)

```

## 12.119 DumbAnonymizer.py

```

00001
00014
00015 """
00016 This example shows how one can use the gdcm.Anonymizer in 'dumb' mode.
00017 This class becomes really handy when one knows which particular tag to fill in.
00018
00019 Usage:
00020
00021     python DumbAnonymizer.py gdcmData/012345.002.050.dcm out.dcm
00022
00023 """
00024
00025 import gdcm
00026
00027 # http://www.oid-info.com/get/1.3.6.1.4.17434
00028 THERALYS_ORG_ROOT = "1.3.6.1.4.17434"
00029
00030 tag_rules={
00031     # Value
00032     (0x0012,0x0010): ("Value", "MySponsorName"),
00033     (0x0012,0x0020): ("Value", "MyProtocolID"),
00034     (0x0012,0x0021): ("Value", "MyProtocolName"),
00035     (0x0012,0x0062): ("Value", "YES"),
00036     (0x0012,0x0063): ("Value", "MyDeidentificationMethod"),
00037
00038     # Method
00039     # (0x0002,0x0003): ("Method", "GenerateMSOPIId"),
00040     # (0x0008,0x1155): ("Method", "GenerateMSOPIId"),
00041     (0x0008,0x0018): ("Method", "GenerateMSOPIId"),
00042     (0x0010,0x0010): ("Method", "GetSponsorInitials"),
00043     (0x0010,0x0020): ("Method", "GetSponsorId"),
00044     (0x0012,0x0030): ("Method", "GetSiteId"),
00045     (0x0012,0x0031): ("Method", "GetSiteName"),
00046     (0x0012,0x0040): ("Method", "GetSponsorId"),
00047     (0x0012,0x0050): ("Method", "GetTPIId"),
00048     (0x0018,0x0022): ("Method", "KeepIfExist"),
00049     (0x0018,0x1315): ("Method", "KeepIfExist"),
00050     (0x0020,0x000d): ("Method", "GenerateStudyId"),
00051     (0x0020,0x000e): ("Method", "GenerateSeriesId"),
00052     (0x0020,0x1002): ("Method", "GetNumberOfFrames"),
00053     (0x0020,0x0020): ("Method", "GetPatientOrientation"),
00054     # Other:
00055     (0x0012,0x0051): ("Patient Field", "Type Examen"),
00056     (0x0018,0x1250): ("Sequence Field", "Receive Coil"),
00057     (0x0018,0x0088): ("Sequence Field", "Spacing Between Slice"),
00058     (0x0018,0x0095): ("Sequence Field", "Pixel Bandwidth"),
00059     (0x0018,0x0082): ("Sequence Field", "Inversion Time"),
00060 }
00061
00062 class MyAnon:
00063     def __init__(self):
00064         self.studyuid = None
00065         self.seriesuid = None
00066         generator = gdcm.UIDGenerator()
00067         if not self.studyuid:
00068             self.studyuid = generator.Generate()
00069         if not self.seriesuid:
00070             self.seriesuid = generator.Generate()
00071     def GetSponsorInitials(self):
00072         return "dummy^foobar"
00073     def GenerateStudyId(self):
00074         return self.studyuid
00075     def GenerateSeriesId(self):
00076         return self.seriesuid
00077     #def GenerateMSOPIId(self):
00078     def GenerateMSOPIId(self):
00079         generator = gdcm.UIDGenerator()
00080         return generator.Generate()
00081     def GetSiteId(self):
00082         return "MySiteId"
00083     def GetSiteName(self):

```

```

00084     return "MySiteName"
00085 def GetSponsorId(self):
00086     return "MySponsorId"
00087 def GetTPId(self):
00088     return "MyTP"
00089
00090 if __name__ == "__main__":
00091     import sys
00092     gdcmm.FileMetaInformation.SetSourceApplicationEntityTitle( "DumbAnonymizer" )
00093     gdcmm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT )
00094
00095     r = gdcmm.Reader()
00096     filename = sys.argv[1]
00097     r.SetFileName( filename )
00098     if not r.Read(): sys.exit(1)
00099
00100     obj = MyAnon()
00101
00102     w = gdcmm.Writer()
00103     ano = gdcmm.Anonymizer()
00104     ano.SetFile( r.GetFile() )
00105     ano.RemoveGroupLength()
00106     for tag,rule in tag_rules.items():
00107         if rule[0] == 'Value':
00108             print tag,rule
00109             ano.Replace( gdcmm.Tag( tag[0], tag[1] ), rule[1] )
00110         elif rule[0] == 'Method':
00111             print tag,rule
00112             # result = locals()[rule[1]]()
00113             methodname = rule[1]
00114             if hasattr(obj, methodname):
00115                 _member = getattr(obj, methodname)
00116                 result = _member()
00117                 ano.Replace( gdcmm.Tag( tag[0], tag[1] ), result )
00118             else:
00119                 print "Problem with: ", methodname
00120
00121     outfilename = sys.argv[2]
00122     w.SetFileName( outfilename )
00123     w.SetFile( ano.GetFile() )
00124     if not w.Write(): sys.exit(1)

```

## 12.120 ExtractImageRegion.py

```

00001
00014
00015 """
00016
00017 This small code shows how to use the gdcmm.ImageRegionReader API
00018 In this example we are taking each frame by frame and dump them to
00019 /tmp/frame.raw.
00020
00021 Usage:
00022 $ ExtractImageRegion.py input.dcm
00023
00024 Example:
00025 $ ExtractImageRegion.py gdcmmData/012345.002.050.dcm
00026 $ md5sum /tmp/frame.raw
00027 d594a5e2fde12f32b6633ca859b4d4a6 /tmp/frame.raw
00028 $ gdcmmInfo --md5sum gdcmmData/012345.002.050.dcm
00029 [...]
00030 md5sum: d594a5e2fde12f32b6633ca859b4d4a6
00031 """
00032
00033 import gdcmm
00034
00035 if __name__ == "__main__":
00036     import sys
00037     filename = sys.argv[1]
00038
00039     file_size = gdcmm.System.FileSize(filename);
00040
00041     # instantiate the reader:
00042     reader = gdcmm.ImageRegionReader();
00043     reader.SetFileName( filename );
00044
00045     # pull DICOM info:

```

```

00046     if not reader.ReadInformation():
00047         sys.exit(1)
00048
00049     # store current offset:
00050     cur_pos = reader.GetStreamCurrentPosition();
00051
00052     remaining = file_size - cur_pos;
00053
00054     print("Remaining bytes to read (Pixel Data): %d" % remaining );
00055
00056     # Get file infos
00057     f = reader.GetFile();
00058
00059     # get some info about image
00060     dims = gdcm.ImageHelper.GetDimensionsValue(f);
00061     print(dims)
00062     pf = gdcm.ImageHelper.GetPixelFormatValue (f);
00063     pixelsize = pf.GetPixelSize();
00064     pi = gdcm.ImageHelper.GetPhotometricInterpretationValue(f);
00065     print( pi );
00066
00067     # buffer to get the pixels
00068     buffer = bytearray( dims[0] * dims[1] * pixelsize )
00069
00070     # define a simple box region.
00071     box = gdcm.BoxRegion();
00072     for z in range(0, dims[2]):
00073         # Define that I want the image 0, full size (dimx x dimy pixels)
00074         # and do that for each z:
00075         box.SetDomain(0, dims[0] - 1, 0, dims[1] - 1, z, z);
00076         #print( box.toString() );
00077         reader.SetRegion( box );
00078
00079         # reader will try to load the uncompressed image region into buffer.
00080         # the call returns an error when buffer.Length is too small. For instance
00081         # one can call:
00082         # uint buf_len = reader.ComputeBufferLength(); // take into account pixel size
00083         # to get the exact size of minimum buffer
00084         if reader.ReadIntoBuffer(buffer):
00085             open('/tmp/frame.raw', 'wb').write(buffer)
00086         else:
00087             #throw new Exception("can't read pixels error");
00088             sys.exit(1)

```

## 12.121 FindAllPatientName.py

```

00001 """
00014 """
00015 This example shows how one can use the gdcm.CompositeNetworkFunctions class
00016 for executing a C-FIND query
00017 It will print the list of patient name found
00018
00019 Usage:
00020
00021 python FindAllPatientName.py
00022
00023 """
00024
00025 import gdcm
00026
00027 # Patient Name
00028 tag = gdcm.Tag(0x10,0x10)
00029 de = gdcm.DataElement(tag)
00030
00031 # Search all patient name where string match 'F*'
00032 de.SetByteStringValue('F*')
00033
00034 ds = gdcm.DataSet()
00035 ds.Insert(de)
00036
00037 cnf = gdcm.CompositeNetworkFunctions()
00038 theQuery = cnf.ConstructQuery (gdcm.ePatientRootType,gdcm.ePatient,ds)
00039
00040 #print theQuery.ValidateQuery()
00041
00042 # prepare the variable for output
00043 ret = gdcm.DataSetArrayType()

```

```

00044
00045 # Execute the C-FIND query
00046 cnf.CFind('dicom.example.com',11112,theQuery,ret,'GDCM_PYTHON','ANY-SCP')
00047
00048 for i in range(0,ret.size()):
00049     print "Patient #",i
00050     print ret[i]

```

## 12.122 FixCommaBug.py

```

00001
00014
00015 """
00016 Using LC_NUMERIC set to something not compatible with "C" it is possible to write out "," instead of
00017 "." as required by the DICOM standard
00018 Issue is still current (IMHO) with gdcm 2.0.9
00019 """
00020
00021 import gdcm
00022 import sys
00023
00024 filename = sys.argv[1]
00025 outname = sys.argv[2]
00026
00027 # read
00028 r = gdcm.Reader()
00029 r.SetFileName( filename )
00030 if not r.Read():
00031     print "not valid"
00032     sys.exit(1)
00033
00034 file = r.GetFile()
00035 dataset = file.GetDataSet()
00036
00037 ano = gdcm.Anonymizer()
00038 ano.SetFile( file )
00039
00040 tags = [
00041     gdcm.Tag(0x0018,0x1164),
00042     gdcm.Tag(0x0018,0x0088),
00043     gdcm.Tag(0x0018,0x0050),
00044     gdcm.Tag(0x0028,0x0030),
00045 ]
00046
00047 for tag in tags:
00048     print tag
00049     if dataset.FindDataElement( tag ):
00050         pixelpacing = dataset.GetDataElement( tag )
00051         #print pixelpacing
00052         bv = pixelpacing.GetByteValue()
00053         str = bv.GetBuffer()
00054         #print bv.GetLength()
00055         #print len(str)
00056         new_str = str.replace(",",".")
00057         # Need to explicitly pass bv.GetLength() to remove any trailing garbage
00058         ano.Replace( tag, new_str, bv.GetLength() )
00059
00060 #print dataset
00061
00062 w = gdcm.Writer()
00063 w.SetFile( file )
00064 w.SetFileName( outname )
00065 if not w.Write():
00066     print "Cannot write"
00067     sys.exit(1)
00068
00069 # paranoid:
00070 image_reader = gdcm.ImageReader()
00071 image_reader.SetFileName( outname )
00072 if not image_reader.Read():
00073     print "there is still a comma"
00074     sys.exit(1)
00075
00076 print "Success!"
00077 sys.exit(0) # success

```

## 12.123 GetPortionCSAHeader.py

```

00001
00014
00015 """
00016 Usage:
00017
00018 python GetPortionCSAHeader.py input.dcm
00019
00020 Footnote:
00021     SIEMENS is not publishing any information on the CSA header. So any info extracted
00022     is at your own risk.
00023 """
00024
00025 import sys
00026 import gdcm
00027
00028 if __name__ == "__main__":
00029
00030     file = sys.argv[1]
00031
00032     r = gdcm.Reader()
00033     r.SetFileName( file )
00034     if not r.Read():
00035         sys.exit(1)
00036
00037     ds = r.GetFile().GetDataSet()
00038     csa_t1 = gdcm.CSAHeader()
00039     csa_t2 = gdcm.CSAHeader()
00040     #print csa
00041     t1 = csa_t1.GetCSAImageHeaderInfoTag();
00042     print t1
00043     t2 = csa_t2.GetCSASeriesHeaderInfoTag();
00044     print t2
00045     # Let's do it for t1:
00046     if ds.FindDataElement( t1 ):
00047         csa_t1.LoadFromDataElement( ds.GetDataElement( t1 ) )
00048         print csa_t1
00049
00050     # Now let's pretend we are only interested in B_value and DiffusionGradientDirection entries:
00051     bvalues = csa_t1.GetCSAElementByName( "B_value" ) # WARNING: it is case sensitive !
00052     print bvalues
00053
00054     diffgradir = csa_t1.GetCSAElementByName( "DiffusionGradientDirection" ) # WARNING: it is case sensitive
00055     ! print diffgradir
00056
00057     # repeat for t2 if you like it:
00058     if ds.FindDataElement( t2 ):
00059         csa_t2.LoadFromDataElement( ds.GetDataElement( t2 ) )
00060         # print csa_t2
00061
00062     gdt = csa_t2.GetCSAElementByName( "GradientDelayTime" )
00063     print gdt
00064
00065     bv = gdt.GetByteValue();
00066     #print bv
00067     str = bv.GetPointer()
00068     print str.split("\\")

```

## 12.124 HelloWorld.py

```

00001
00014
00015 """
00016 Hello World !
00017 """
00018
00019 import gdcm
00020 import sys
00021
00022 if __name__ == "__main__":
00023
00024     # verbosity:
00025     #gdcm.Trace.DebugOn()
00026     #gdcm.Trace.WarningOn()
00027     #gdcm.Trace.ErrorOn()

```

```

00028
00029 # Get the filename from the command line
00030 filename = sys.argv[1]
00031
00032 # Instantiate a gdcm.Reader
00033 # This is the main class to handle any type of DICOM object
00034 # You should check for gdcm.ImageReader for reading specifically DICOM Image file
00035 r = gdcm.Reader()
00036 r.SetFileName( filename )
00037 # If the reader fails to read the file, we should stop !
00038 if not r.Read():
00039     print "Not a valid DICOM file"
00040     sys.exit(1)
00041
00042 # Get the DICOM File structure
00043 file = r.GetFile()
00044
00045 # Get the DataSet part of the file
00046 dataset = file.GetDataSet()
00047
00048 # Ok let's print it !
00049 print dataset
00050
00051 # Use StringFilter to print a particular Tag:
00052 sf = gdcm.StringFilter()
00053 sf.SetFile(r.GetFile())
00054
00055 # Check if Attribute exist
00056 print dataset.FindElement( gdcm.Tag(0x0028,0x0010) )
00057
00058 # Let's print it as string pair:
00059 print sf.ToStringPair(gdcm.Tag(0x0028,0x0010))

```

## 12.125 ManipulateFile.py

```

00001
00014
00015 """
00016 Usage:
00017
00018 python ManipulateFile.py input.dcm output.dcm
00019
00020 Footnote:
00021 GDCM 1.2.x would create incorrect Multiframe MR Image Storage file. Try to recover from
00022 the issues to recreate a MultiframeGrayscaleByteSecondaryCaptureImageStorage file.
00023 e.g:
00024
00025 python ManipulateFile.py Insight/Testing/Temporary/itkGDCMImageIOTest5-j2k.dcm manipulated.dcm
00026 """
00027
00028 import sys
00029 import gdcm
00030
00031 if __name__ == "__main__":
00032
00033     file1 = sys.argv[1]
00034     file2 = sys.argv[2]
00035
00036     r = gdcm.Reader()
00037     r.SetFileName( file1 )
00038     if not r.Read():
00039         sys.exit(1)
00040
00041     ano = gdcm.Anonymizer()
00042     ano.SetFile( r.GetFile() )
00043     ano.RemovePrivateTags()
00044     ano.Remove( gdcm.Tag(0x0032,0x1030) )
00045     ano.Remove( gdcm.Tag(0x008,0x14) )
00046     ano.Remove( gdcm.Tag(0x008,0x1111) )
00047     ano.Remove( gdcm.Tag(0x008,0x1120) )
00048     ano.Remove( gdcm.Tag(0x008,0x1140) )
00049     ano.Remove( gdcm.Tag(0x10,0x21b0) )
00050     ano.Empty( gdcm.Tag(0x10,0x10) )
00051     ano.Empty( gdcm.Tag(0x10,0x20) )
00052     ano.Empty( gdcm.Tag(0x10,0x30) )
00053     ano.Empty( gdcm.Tag(0x20,0x10) )
00054     ano.Empty( gdcm.Tag(0x32,0x1032) )

```

```

00055 ano.Empty( gdcm.Tag(0x32,0x1033) )
00056 ano.Empty( gdcm.Tag(0x40,0x241) )
00057 ano.Empty( gdcm.Tag(0x40,0x254) )
00058 ano.Empty( gdcm.Tag(0x40,0x253) )
00059 ano.Empty( gdcm.Tag(0x40,0x1001) )
00060 ano.Empty( gdcm.Tag(0x8,0x80) )
00061 ano.Empty( gdcm.Tag(0x8,0x50) )
00062 ano.Empty( gdcm.Tag(0x8,0x1030) )
00063 ano.Empty( gdcm.Tag(0x8,0x103e) )
00064 ano.Empty( gdcm.Tag(0x18,0x1030) )
00065 ano.Empty( gdcm.Tag(0x38,0x300) )
00066 g = gdcm.UIDGenerator()
00067 ano.Replace( gdcm.Tag(0x0008,0x0018), g.Generate() )
00068 ano.Replace( gdcm.Tag(0x0020,0x000d), g.Generate() )
00069 ano.Replace( gdcm.Tag(0x0020,0x000e), g.Generate() )
00070 ano.Replace( gdcm.Tag(0x0020,0x0052), g.Generate() )
00071 #ano.Replace( gdcm.Tag(0x0008,0x0016), "1.2.840.10008.5.1.4.1.1.7.2" )
00072 """
00073 ano.Remove( gdcm.Tag(0x0018,0x0020) ) # ScanningSequence
00074 ano.Remove( gdcm.Tag(0x0018,0x0021) ) # SequenceVariant
00075 ano.Remove( gdcm.Tag(0x0018,0x0022) ) # ScanOptions
00076 ano.Remove( gdcm.Tag(0x0018,0x0023) ) # MRAcquisitionType
00077 ano.Remove( gdcm.Tag(0x0018,0x0050) ) # SliceThickness
00078 ano.Remove( gdcm.Tag(0x0018,0x0080) ) # RepetitionTime
00079 ano.Remove( gdcm.Tag(0x0018,0x0081) ) # EchoTime
00080 ano.Remove( gdcm.Tag(0x0018,0x0088) ) # SpacingBetweenSlices
00081 ano.Remove( gdcm.Tag(0x0018,0x0091) ) # EchoTrainLength
00082 ano.Remove( gdcm.Tag(0x0018,0x1164) ) # ImagerPixelSpacing
00083
00084 ano.Remove( gdcm.Tag(0x0020,0x0032) ) # Image Position (Patient)
00085 ano.Remove( gdcm.Tag(0x0020,0x0037) ) # Image Orientation (Patient)
00086 ano.Remove( gdcm.Tag(0x0020,0x0052) ) # Frame of Reference UID
00087 ano.Remove( gdcm.Tag(0x0020,0x1040) ) # Position Reference Indicator
00088
00089 ano.Replace( gdcm.Tag(0x0028,0x0301), "NO" ) # Burned In Annotation
00090
00091 ano.Empty( gdcm.Tag(0x0020,0x0020) )
00092
00093 ano.Remove( gdcm.Tag(0x7fe0,0x0000) )
00094
00095 #ano.Empty( gdcm.Tag(0x0028,0x0009) ) # Frame Increment Pointer
00096
00097 #ano.Empty( gdcm.Tag(0x0028,0x1052) ) #<entry group="0028" element="1052" vr="DS" vm="1" name="Rescale
Intercept"/>
00098 #ano.Empty( gdcm.Tag(0x0028,0x1053) ) #<entry group="0028" element="1053" vr="DS" vm="1" name="Rescale
Slope"/>
00099 #ano.Replace( gdcm.Tag(0x0028,0x1054), "US" ) #<entry group="0028" element="1054" vr="LO" vm="1"
name="Rescale Type"/>
00100
00101 ano.Replace( gdcm.Tag(0x2050, 0x0020), "IDENTITY")
00102 """
00103
00104 w = gdcm.Writer()
00105 w.SetFile( ano.GetFile() )
00106 w.SetFileName( file2 )
00107 if not w.Write():
00108     sys.exit(1)

```

## 12.126 ManipulateSequence.py

```

00001
00014
00015 """
00016 Usage:
00017
00018 python ManipulateSequence.py input.dcm output.dcm
00019
00020 This was tested using:
00021
00022 python ManipulateSequence.py gdcmData/D_CLUNIE_CT1_J2KI.dcm myoutput.dcm
00023
00024 This is a dummy example on how to modify a value set in a nested-nested dataset
00025
00026 WARNING:
00027 Do not use as-is in production, this is just an example
00028 This example works in an undefined length Item only (you need to explicitly recompute the length
otherwise)

```



```

00029 """
00030
00031 import sys
00032 import gdcm
00033
00034 if __name__ == "__main__":
00035
00036     file1 = sys.argv[1]
00037     file2 = sys.argv[2]
00038
00039     r = gdcm.Reader()
00040     r.SetFileName( file1 )
00041     if not r.Read():
00042         sys.exit(1)
00043
00044     f = r.GetFile()
00045     ds = f.GetDataSet()
00046     tsis = gdcm.Tag(0x0008,0x2112) # SourceImageSequence
00047     if ds.FindDataElement( tsis ):
00048         sis = ds.GetDataElement( tsis )
00049         #sqsis = sis.GetSequenceOfItems()
00050         # GetValueAsSQ handle more cases
00051         sqsis = sis.GetValueAsSQ()
00052         if sqsis.GetNumberOfItems():
00053             item1 = sqsis.GetItem(1)
00054             nestedds = item1.GetNestedDataSet()
00055             tprcs = gdcm.Tag(0x0040,0x170) # PurposeOfReferenceCodeSequence
00056             if nestedds.FindDataElement( tprcs ):
00057                 prcs = nestedds.GetDataElement( tprcs )
00058                 sqprcs = prcs.GetSequenceOfItems()
00059                 if sqprcs.GetNumberOfItems():
00060                     item2 = sqprcs.GetItem(1)
00061                     nestedds2 = item2.GetNestedDataSet()
00062                     # (0008,0104) LO [Uncompressed predecessor] # 24, 1 CodeMeaning
00063                     tcm = gdcm.Tag(0x0008,0x0104)
00064                     if nestedds2.FindDataElement( tcm ):
00065                         cm = nestedds2.GetDataElement( tcm )
00066                         mystr = "GDCM was here"
00067                         cm.SetByteStringValue( mystr )
00068
00069     w = gdcm.Writer()
00070     w.SetFile( f )
00071     w.SetFileName( file2 )
00072     if not w.Write():
00073         sys.exit(1)

```

## 12.127 MergeFile.py

```

00001
00014
00015 """
00016 Usage:
00017
00018 python MergeFile.py input1.dcm input2.dcm
00019
00020 It will produce a 'merge.dcm' output file, which contains all meta information from input1.dcm
00021 and copy the Stored Pixel values from input2.dcm
00022 This script even works when input2.dcm is a Secondary Capture and does not contains information
00023 such as IOP and IPP...
00024 """
00025
00026 import sys
00027 import gdcm
00028
00029 if __name__ == "__main__":
00030
00031     file1 = sys.argv[1]
00032     file2 = sys.argv[2]
00033
00034     r1 = gdcm.ImageReader()
00035     r1.SetFileName( file1 )
00036     if not r1.Read():
00037         sys.exit(1)
00038
00039     r2 = gdcm.ImageReader()
00040     r2.SetFileName( file2 )
00041     if not r2.Read():

```

```

00042     sys.exit(1)
00043
00044     # Image from r2 could be Secondary Capture and thus would not contains neither IPP nor IOP
00045     # Instead always prefer to only copy the Raw Data Element.
00046     # Warning ! Image need to be identical ! Only the value of Stored Pixel can be different.
00047     r1.GetImage().SetDataElement( r2.GetImage().GetDataElement() )
00048
00049     w = gdcm.ImageWriter()
00050     w.SetFile( r1.GetFile() )
00051     #w.SetImage( r2.GetImage() ) # See comment above
00052     w.SetImage( r1.GetImage() )
00053
00054     w.SetFileName( "merge.dcm" )
00055     if not w.Write():
00056         sys.exit(1)
00057
00058     sys.exit(0)

```

## 12.128 NewSequence.py

```

00001
00014
00015 """
00016 Usage:
00017
00018     python NewSequence.py input.dcm output.dcm
00019
00020
00021 Thanks to Robert Irie for code
00022 """
00023
00024 import sys
00025 import gdcm
00026
00027 if __name__ == "__main__":
00028
00029     file1 = sys.argv[1]
00030     file2 = sys.argv[2]
00031
00032     r = gdcm.Reader()
00033     r.SetFileName( file1 )
00034     if not r.Read():
00035         sys.exit(1)
00036
00037     f = r.GetFile()
00038     ds = f.GetDataSet()
00039     #t sis = gdcm.Tag(0x0008,0x2112) # SourceImageSequence
00040
00041     # Create a dataelement
00042     de = gdcm.DataElement(gdcm.Tag(0x0010, 0x2180))
00043     de.SetByteStringValue("Occupation")
00044     de.SetVR(gdcm.VR(gdcm.VR.SH))
00045
00046     # Create an item
00047     it=gdcm.Item()
00048     it.SetVLToUndefined() # Needed to not popup error message
00049     #it.InsertDataElement(de)
00050     nds=it.GetNestedDataSet()
00051     nds.Insert(de)
00052
00053     # Create a Sequence
00054     sq=gdcm.SequenceOfItems().New()
00055     sq.SetLengthToUndefined()
00056     sq.AddItem(it)
00057
00058     # Insert sequence into data set
00059     des=gdcm.DataElement(gdcm.Tag(0x0400,0x0550))
00060     des.SetVR(gdcm.VR(gdcm.VR.SQ))
00061     des.SetValue(sq.__ref__())
00062     des.SetVLToUndefined()
00063
00064     ds.Insert(des)
00065
00066     w = gdcm.Writer()
00067     w.SetFile( f )
00068     w.SetFileName( file2 )
00069     if not w.Write():
00070         sys.exit(1)

```

## 12.129 PhilipsPrivateRescaleInterceptSlope.py

```

00001
00014
00015 """
00016 Usage:
00017
00018     python
00019     """
00020
00021 import gdcm
00022 import sys
00023
00024 filename = sys.argv[1]
00025 tmpfile = "/tmp/philips_rescaled.dcm"
00026
00027
00028 # Need to access some private tags, read the file :
00029 reader = gdcm.Reader()
00030 reader.SetFileName( filename )
00031 if not reader.Read():
00032     sys.exit(1)
00033
00034 ds = reader.GetFile().GetDataSet()
00035
00036 #print ds
00037 # (2005,1409)      DS      4      0.0
00038 # (2005,140a)      DS      16     1.52283272283272
00039
00040 # (2005,0014)      LO      26     Philips MR Imaging DD 005
00041 tag1 = gdcm.PrivateTag(0x2005,0x09,"Philips MR Imaging DD 005")
00042 tag2 = gdcm.PrivateTag(0x2005,0x0a,"Philips MR Imaging DD 005")
00043 print tag1
00044 print tag2
00045
00046 # make sure to do a copy, we want the private tag to remain
00047 # otherwise gdcm gives us a reference
00048 el1 = gdcm.DataElement( ds.GetDataElement( tag1 ) )
00049 print el1
00050 el2 = gdcm.DataElement( ds.GetDataElement( tag2 ) )
00051 print el2
00052
00053 # (0028,1052) DS [-1000]          # 6, 1 RescaleIntercept
00054 # (0028,1053) DS [1]             # 2, 1 RescaleSlope
00055
00056 el1.SetTag( gdcm.Tag(0x0028,0x1052) )
00057 el2.SetTag( gdcm.Tag(0x0028,0x1053) )
00058
00059 ds.Insert( el1 )
00060 ds.Insert( el2 )
00061
00062 w = gdcm.Writer()
00063 w.SetCheckFileMetaInformation( False )
00064 w.SetFileName( tmpfile )
00065 w.SetFile( reader.GetFile() )
00066 if not w.Write():
00067     sys.exit(1)
00068
00069 print "success"

```

## 12.130 PlaySound.py

```

00001
00014
00015 """
00016 Usage:
00017
00018     python PlaySound.py input.dcm
00019     """
00020
00021 import gdcm
00022 import sys
00023
00024 #filename = "/home/mmalaterre/Creatis/gdcmDataExtra/gdcmNonImageData/audio_from_rafael_sanguinetti.dcm"
00025 filename = sys.argv[1]
00026 print filename
00027

```

```

00028 r = gdcM.Reader()
00029 r.SetFileName( filename )
00030 if not r.Read():
00031     sys.exit(1)
00032
00033 ds = r.GetFile().GetDataSet()
00034
00035 waveformtag = gdcM.Tag(0x5400,0x0100)
00036 waveformsq = ds.GetDataElement( waveformtag )
00037 #print waveformsq
00038
00039 #print dir(waveformsq)
00040
00041 items = waveformsq.GetSequenceOfItems()
00042
00043 if not items.GetNumberOfItems():
00044     sys.exit(1)
00045
00046 item = items.GetItem(1)
00047 #print item
00048
00049 waveformds = item.GetNestedDataSet()
00050 #print waveformds
00051
00052 waveformdatatag = gdcM.Tag(0x5400,0x0101)
00053 waveformdata = waveformds.GetDataElement( waveformdatatag )
00054
00055 #print waveformdata.GetPointer()
00056 bv = waveformdata.GetByteValue()
00057 print dir(bv)
00058
00059 #print bv.GetPointer()
00060 print bv.GetLength()
00061 l = 116838
00062
00063 file='test.wav'
00064 myfile = open(file, "wb")
00065 s = bv.GetPointer()
00066 for i in range(0, l):
00067     myfile.write(s[i])
00068 myfile.close()
00069
00070 # http://mail.python.org/pipermail/python-list/2004-October/288905.html
00071 if sys.platform.startswith('win'):
00072     from winsound import PlaySound, SND_FILENAME, SND_ASYNC
00073     PlaySound(file, SND_FILENAME|SND_ASYNC)
00074 elif sys.platform.find('linux')>-1:
00075     from wave import open as waveOpen
00076     from ossaudiodev import open as ossOpen
00077     s = waveOpen(file,'rb')
00078     (nc,sw,fr,nf,comptype, compname) = s.getparams()
00079     dsp = ossOpen('/dev/dsp','w')
00080     try:
00081         from ossaudiodev import AFMT_S16_NE
00082     except ImportError:
00083         if byteorder == "little":
00084             AFMT_S16_NE = ossaudiodev.AFMT_S16_LE
00085         else:
00086             AFMT_S16_NE = ossaudiodev.AFMT_S16_BE
00087     dsp.setparameters(AFMT_S16_NE, nc, fr)
00088     data = s.readframes(nf)
00089     s.close()
00090     dsp.write(data)
00091     dsp.close()

```

## 12.131 PrivateDict.py

```

00001
00014
00015 """
00016 """
00017
00018 import gdcM
00019 import sys,os
00020
00021 if __name__ == "__main__":
00022     #gdcM.Trace.DebugOn()

```

```

00023 globInst = gdcm.Global.GetInstance()
00024 # Try to load Part3.xml file
00025 # This file is too big for being accessible directly at runtime.
00026 globInst.LoadResourcesFiles()
00027
00028
00029 # Get a private tag from the runtime dicts. LoadResourcesFiles could
00030 # have failed but this has no impact on the private dict
00031
00032 d = globInst.GetDicts()
00033 print d.GetDictEntry( gdcm.Tag(0x0029,0x0010) ,"SIEMENS CSA HEADER" )
00034 pd = d.GetPrivateDict()
00035 print pd.GetDictEntry( gdcm.PrivateTag(0x0029,0x0010,"SIEMENS CSA HEADER" ) )

```

## 12.132 ReWriteSCAsMR.py

```

00001
00014
00015 """
00016 GDCM 1.x would write out MR Image Storage as Secondary Capture Object while still setting Rescale
00017 Slope/Intercept
00018 and saving the Pixel Spacing in (0028,0030)
00019 """
00020 import gdcm
00021 import sys,os
00022
00023 def CheckSecondaryCaptureObjectIsMRImageStorage(r):
00024     ds = r.GetFile().GetDataSet()
00025     # Check Source Image Sequence
00026     if ds.FindDataElement( gdcm.Tag(0x0008,0x2112) ):
00027         sis = ds.GetDataElement( gdcm.Tag(0x0008,0x2112) )
00028         sqsis = sis.GetSequenceOfItems()
00029         if sqsis.GetNumberOfItems():
00030             item1 = sqsis.GetItem(1)
00031             nestedds = item1.GetNestedDataSet()
00032             if nestedds.FindDataElement( gdcm.Tag(0x0008,0x1150) ):
00033                 ReferencedSOPClassUID = nestedds.GetDataElement( gdcm.Tag(0x0008,0x1150) )
00034                 raw = ReferencedSOPClassUID.GetByteValue().GetPointer()
00035                 uids = gdcm.UIDs()
00036                 # what is the actual object we are looking at ?
00037                 ms = gdcm.MediaStorage()
00038                 ms.SetFromDataSet(ds)
00039                 msuid = ms.GetString()
00040                 uids.SetFromUID( msuid )
00041                 msuidname = uids.GetName() # real Media Storage Name
00042                 uids.SetFromUID( raw )
00043                 sqmsuidname = uids.GetName() # Source Image Sequence Media Storage Name
00044                 # If object is SC and Source derivation is MRImageStorage then we can assume 'Pixel Spacing' is
00045                 correct
00046                 if( sqmsuidname == 'MR Image Storage' and msuidname == 'Secondary Capture Image Storage' ):
00047                     return True
00048                 # in all other case simply return the currentspacing:
00049                 return False
00050
00051 if __name__ == "__main__":
00052     r = gdcm.ImageReader()
00053     filename = sys.argv[1]
00054     r.SetFileName( filename )
00055     if not r.Read():
00056         sys.exit(1)
00057     f = r.GetFile()
00058
00059     if( CheckSecondaryCaptureObjectIsMRImageStorage(r) ):
00060         # Special handling of the spacing:
00061         # GDCM 1.2.0 would not rewrite correctly DICOM Object and would always set them as 'Secondary Capture
00062         Image Storage'
00063         # while we would rather have 'MR Image Storage'
00064         gdcm.ImageHelper.SetForcePixelSpacing( True )
00065         mrspacing = gdcm.ImageHelper.GetSpacingValue( r.GetFile() )
00066         # TODO: I cannot do simply the following:
00067         #image.SetSpacing( mrspacing )
00068         image.SetSpacing(0, mrspacing[0] )
00069         image.SetSpacing(1, mrspacing[1] )
00070         image.SetSpacing(2, mrspacing[2] )
00071         gdcm.ImageHelper.SetForceRescaleInterceptSlope( True )
00072         ris = gdcm.ImageHelper.GetRescaleInterceptSlopeValue( r.GetFile() )

```

```

00071     image.SetIntercept( ris[0] )
00072     image.SetSlope( ris[1] )
00073
00074     outfilename = sys.argv[2]
00075     w = gdcM.ImageWriter()
00076     w.SetFileName( outfilename )
00077     w.SetFile( r.GetFile() )
00078     w.SetImage( image )
00079     if not w.Write():
00080         sys.exit(1)
00081
00082     sys.exit(0)

```

## 12.133 ReadAndDumpDICOMDIR.py

```

00001
00023
00024
00025
00026 import sys
00027 import gdcM
00028
00029 if __name__ == "__main__":
00030     # Check arguments
00031     if (len(sys.argv) < 2):
00032         # No filename passed
00033         print "No input filename found"
00034         quit()
00035
00036     filename = sys.argv[1]
00037
00038
00039     # Read file
00040     reader = gdcM.Reader()
00041     reader.SetFileName(filename)
00042     if (not reader.Read()):
00043         print "Unable to read %s" % (filename)
00044         quit()
00045
00046     file = reader.GetFile()
00047
00048     # Retrieve header information
00049     fileMetaInformation = file.GetHeader()
00050     print fileMetaInformation
00051
00052     # Retrieve data set
00053     dataSet = file.GetDataSet()
00054     #print dataSet
00055
00056     # Check media storage
00057     mediaStorage = gdcM.MediaStorage()
00058     mediaStorage.SetFromFile(file)
00059     if (gdcM.MediaStorage.GetMSType(str(mediaStorage)) != gdcM.MediaStorage.MediaStorageDirectoryStorage):
00060         # File is not a DICOMDIR
00061         print "This file is not a DICOMDIR (Media storage type: %s)" % (str(mediaStorage))
00062         quit()
00063
00064     # Check Media Storage SOP Class
00065     if (fileMetaInformation.FindDataElement(gdcM.Tag(0x0002, 0x0002))):
00066         sopClassUid = str(fileMetaInformation.GetDataElement(gdcM.Tag(0x0002, 0x0002)).GetValue())
00067         # Check SOP UID
00068         if (sopClassUid != "1.2.840.10008.1.3.10"):
00069             # File is not a DICOMDIR
00070             print "This file is not a DICOMDIR"
00071     else:
00072         # Not present
00073         print "Media Storage SOP Class not present"
00074         quit()
00075
00076     # Iterate through the DICOMDIR data set
00077     iterator = dataSet.GetDES().begin()
00078     while (not iterator.equal(dataSet.GetDES().end())):
00079         dataElement = iterator.next()
00080
00081         # Check the element tag
00082         if (dataElement.GetTag() == gdcM.Tag(0x004, 0x1220)):
00083             # The 'Directory Record Sequence' element

```

```

00084         sequence = dataElement.GetValueAsSQ()
00085
00086         # Loop through the sequence items
00087         itemNr = 1
00088         while (itemNr < sequence.GetNumberOfItems()):
00089             item = sequence.GetItem(itemNr)
00090
00091             # Check the element tag
00092             if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
00093                 # The 'Directory Record Type' element
00094                 value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())
00095
00096                 # PATIENT
00097                 while (value.strip() == "PATIENT"):
00098                     print value.strip()
00099                     # Print patient name
00100                     if (item.FindDataElement(gdcm.Tag(0x0010, 0x0010))):
00101                         value = str(item.GetDataElement(gdcm.Tag(0x0010, 0x0010)).GetValue())
00102                         print value
00103
00104                     # Print patient ID
00105                     if (item.FindDataElement(gdcm.Tag(0x0010, 0x0020))):
00106                         value = str(item.GetDataElement(gdcm.Tag(0x0010, 0x0020)).GetValue())
00107                         print value
00108
00109                     # Next
00110                     itemNr = itemNr + 1
00111                     item = sequence.GetItem(itemNr)
00112                     if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
00113                         value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())
00114
00115                 # STUDY
00116                 while (value.strip() == "STUDY"):
00117                     print value.strip()
00118
00119                     # Print study UID
00120                     if (item.FindDataElement(gdcm.Tag(0x0020, 0x000d))):
00121                         value = str(item.GetDataElement(gdcm.Tag(0x0020, 0x000d)).GetValue())
00122                         print value
00123
00124                     # Print study date
00125                     if (item.FindDataElement(gdcm.Tag(0x0008, 0x0020))):
00126                         value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x0020)).GetValue())
00127                         print value
00128
00129                     # Print study description
00130                     if (item.FindDataElement(gdcm.Tag(0x0008, 0x1030))):
00131                         value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x1030)).GetValue())
00132                         print value
00133
00134                     # Next
00135                     itemNr = itemNr + 1
00136                     item = sequence.GetItem(itemNr)
00137                     if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
00138                         value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())
00139
00140                 # SERIES
00141                 while (value.strip() == "SERIES"):
00142                     print value.strip()
00143
00144                     # Print series UID
00145                     if (item.FindDataElement(gdcm.Tag(0x0020, 0x000e))):
00146                         value = str(item.GetDataElement(gdcm.Tag(0x0020, 0x000e)).GetValue())
00147                         print value
00148
00149                     # Print series modality
00150                     if (item.FindDataElement(gdcm.Tag(0x0008, 0x0060))):
00151                         value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x0060)).GetValue())
00152                         print "Modality"
00153                         print value
00154
00155                     # Print series description
00156                     if (item.FindDataElement(gdcm.Tag(0x0008, 0x103e))):
00157                         value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x103e)).GetValue())
00158                         print "Description"
00159                         print value
00160
00161                     # Next
00162                     itemNr = itemNr + 1
00163                     item = sequence.GetItem(itemNr)
00164                     if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):

```

```

00165         value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())
00166
00167         # IMAGE
00168         while (value.strip() == "IMAGE"):
00169             print value.strip()
00170
00171         # Print image UID
00172         if (item.FindDataElement(gdcm.Tag(0x0004, 0x1511))):
00173             value = str(item.GetDataElement(gdcm.Tag(0x0004,
00174             0x1511)).GetValue())
00175             print value
00176
00177         # Next
00178         if (itemNr < sequence.GetNumberOfItems()):
00179             itemNr = itemNr + 1
00180         else:
00181             break
00182
00183         item = sequence.GetItem(itemNr)
00184         if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
00185             value = str(item.GetDataElement(gdcm.Tag(0x0004,
00186             0x1430)).GetValue())
00187         # Next
00188         itemNr = itemNr + 1

```

## 12.134 RemovePrivateTags.py

```

00001
00014
00015 """
00016 Usage:
00017
00018 python RemovePrivateTags.py input.dcm output.dcm
00019 """
00020
00021 import sys
00022 import gdcm
00023
00024 if __name__ == "__main__":
00025
00026     file1 = sys.argv[1]
00027     file2 = sys.argv[2]
00028
00029     # Instantiate the reader.
00030     r = gdcm.Reader()
00031     r.SetFileName( file1 )
00032     if not r.Read():
00033         sys.exit(1)
00034
00035     # Remove private tags
00036     ano = gdcm.Anonymizer()
00037     ano.SetFile( r.GetFile() )
00038     if not ano.RemovePrivateTags():
00039         sys.exit(1)
00040
00041     # Write DICOM file
00042     w = gdcm.Writer()
00043     w.SetFile( ano.GetFile() )
00044     #w.CheckFileMetaInformationOff() # Do not attempt to check meta header
00045     w.SetFileName( file2 )
00046     if not w.Write():
00047         sys.exit(1)
00048
00049
00050     # It is usually a good idea to exit the script with an error, as gdcm does not remove partial
00051     # (incorrect) DICOM file
00052     # (application level)

```

## 12.135 ScanDirectory.py

```

00001
00014

```



```

00015 import gdcmm
00016 import sys,os
00017
00018 class ProgressWatcher(gdcm.SimpleSubjectWatcher):
00019     def ShowProgress(self, sender, event):
00020         pe = gdcm.ProgressEvent.Cast(event)
00021         print pe.GetProgress()
00022     def EndFilter(self):
00023         print "Yay ! I am done"
00024
00025 if __name__ == "__main__":
00026     directory = sys.argv[1]
00027
00028     # Define the set of tags we are interested in
00029     t1 = gdcm.Tag(0x8,0x8);
00030     t2 = gdcm.Tag(0x10,0x10);
00031
00032     # Iterate over directory
00033     d = gdcm.Directory();
00034     nfiles = d.Load( directory );
00035     if(nfiles == 0): sys.exit(1);
00036     # System.Console.WriteLine( "Files:\n" + d.toString() );
00037
00038     filenames = d.GetFilenames()
00039
00040     # Get rid of any Warning while parsing the DICOM files
00041     gdcm.Trace.WarningOff()
00042
00043     # instantiate Scanner:
00044     sp = gdcm.Scanner.New();
00045     s = sp.__ref__()
00046     w = ProgressWatcher(s, 'Watcher')
00047
00048     s.AddTag( t1 );
00049     s.AddTag( t2 );
00050     b = s.Scan( filenames );
00051     if(not b): sys.exit(1);
00052
00053     print "success" ;
00054     #print s
00055
00056     pttv = gdcm.PythonTagToValue( s.GetMapping( filenames[1] ) )
00057     pttv.Start()
00058     # iterate until the end:
00059     while( not pttv.IsAtEnd() ):
00060         # get current value for tag and associated value:
00061         # if tag was not found, then it was simply not added to the internal std::map
00062         # Warning value can be None
00063         tag = pttv.GetCurrentTag()
00064         value = pttv.GetCurrentValue()
00065         print tag,"->",value
00066         # increment iterator
00067         pttv.Next()
00068
00069     sys.exit(0)

```

## 12.136 SortImage.py

```

00001
00014
00015 """
00016 Usage:
00017
00018     python SortImage.py dirname
00019 """
00020
00021 import gdcm
00022 import sys
00023
00024 def PrintProgress(object, event):
00025     assert event == "ProgressEvent"
00026     print "Progress:", object.GetProgress()
00027
00028 def MySort(ds1, ds2):
00029     # compare ds1
00030     return False
00031

```

```

00032 if __name__ == "__main__":
00033
00034     dirname = sys.argv[1]
00035     d = gdcmm.Directory()
00036     d.Load( dirname )
00037
00038     print d
00039
00040     sorter = gdcmm.Sorter()
00041     sorter.SetSortFunction( MySort )
00042     #sorter.AddObserver( "ProgressEvent", PrintProgress )
00043     sorter.Sort( d.GetFilenames() )
00044
00045     print "Sorter:"
00046     print sorter

```

## 12.137 WriteBuffer.py

```

00001
00014
00015 """
00016 Usage:
00017
00018 http://chuckhahm.com/Ischem/Zurich/XX_0134
00019
00020 (2005,1132) SQ (Sequence with undefined length #=8) # u/1, 1 Unknown Tag & Data
00021 (ffff,e000) na (Item with undefined length #=9) # u/1, 1 Item
00022 (2005,0011) LO [Philips MR Imaging DD 002] # 26, 1 PrivateCreator
00023 (2005,1137) PN [PDF_CONTROL_GEN_PARS] # 20, 1 Unknown Tag & Data
00024 (2005,1138) PN (no value available) # 0, 0 Unknown Tag & Data
00025 (2005,1139) PN [IEEE_PDF] # 8, 1 Unknown Tag & Data
00026 (2005,1140) PN (no value available) # 0, 0 Unknown Tag & Data
00027 (2005,1141) PN (no value available) # 0, 0 Unknown Tag & Data
00028 (2005,1143) SL 3103 # 4, 1 Unknown Tag & Data
00029 (2005,1144) OW 0566\0000\013b\0000\0a4a\0000\000e\0000\0a7a\0000\0195\0000\0008... # 3104, 1 Unknown
Tag & Data
00030 (2005,1147) CS [Y] # 2, 1 Unknown Tag & Data
00031 (ffff,e00d) na (ItemDelimitationItem) # 0, 0 ItemDelimitationItem
00032 (ffff,e000) na (Item with undefined length #=9) # u/1, 1 Item
00033 (2005,0011) LO [Philips MR Imaging DD 002] # 26, 1 PrivateCreator
00034 (2005,1137) PN [PDF_CONTROL_PREP_PARS] # 22, 1 Unknown Tag & Data
00035 (2005,1138) PN (no value available) # 0, 0 Unknown Tag & Data
00036 (2005,1139) PN [IEEE_PDF] # 8, 1 Unknown Tag & Data
00037 (2005,1140) PN (no value available) # 0, 0 Unknown Tag & Data
00038 (2005,1141) PN (no value available) # 0, 0 Unknown Tag & Data
00039 (2005,1143) SL 7934 # 4, 1 Unknown Tag & Data
00040 (2005,1144) OW 19b6\0000\005f\0000\1b2a\0000\00f3\0000\1eee\0000\0000\0000\0008... # 7934, 1 Unknown
Tag & Data
00041 (2005,1147) CS [Y] # 2, 1 Unknown Tag & Data
00042 (ffff,e00d) na (ItemDelimitationItem) # 0, 0 ItemDelimitationItem
00043 ...
00044 """
00045
00046 import sys
00047 import gdcmm
00048
00049 if __name__ == "__main__":
00050
00051     file1 = sys.argv[1]
00052     file2 = sys.argv[2]
00053
00054     r = gdcmm.Reader()
00055     r.SetFileName( file1 )
00056     if not r.Read():
00057         sys.exit(1)
00058
00059     fg = gdcmm.FileNameGenerator()
00060     f = r.GetFile()
00061     ds = f.GetDataSet()
00062     tsis = gdcmm.Tag(0x2005,0x1132) #
00063     if ds.FindDataElement( tsis ):
00064         sis = ds.GetDataElement( tsis )
00065         #sqsis = sis.GetSequenceOfItems()
00066         # GetValueAsSQ handle more cases
00067         sqsis = sis.GetValueAsSQ()
00068         if sqsis.GetNumberOfItems():
00069             nitems = sqsis.GetNumberOfItems();

```

```

00070         fg.SetNumberOfFileNames( nitems )
00071         fg.SetPrefix( file2 )
00072         if not fg.Generate():
00073             print "problem"
00074             sys.exit(1)
00075         for i in range(0,nitems):
00076             item1 = sqsis.GetItem(i+1) # Item start at 1
00077             nestedds = item1.GetNestedDataSet()
00078             tprcs = gdcm.Tag(0x2005,0x1144) #
00079             if nestedds.FindDataElement( tprcs ):
00080                 prcs = nestedds.GetDataElement( tprcs )
00081                 bv = prcs.GetByteValue()
00082                 print bv
00083                 f = open( fg.GetFilename(i) , "w" )
00084                 f.write( bv.WriteBuffer() )

```

## 12.138 HelloActiviz.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
using vtkgdcm;
using Kitware.VTK;
using System;
using System.Runtime.InteropServices;

/*
 * This example shows how vtkgdcm can be connected to Kitware.VTK Activiz product.
 * Three (3) arguments are required:
 * 1. Input DICOM file (SWIG)
 * 2. Temporary PNG (intermediate) file (Activiz)
 * 3. Final DICOM file (SWIG)
 *
 * $ export MONO_PATH=/usr/lib/cli/Activiz.NET:/usr/lib/cli/Kitware.mummy.Runtime-1.0
 * $ mono ./bin/HelloActiviz.exe ~/Creatis/gdcmData/test.acr out.png toto.dcm
 *
 * Footnote:
 * this test originally used vtkBMPWriter / vtkBMPReader combination to store intermediate
 * image file, but BMP file are 24bits by default. Instead use PNG format which supports seems
 * to be closer to what was expected in this simple test.
 */
public class HelloActiviz
{
    // Does not work with Activiz.NET-5.4.0.455-Linux-x86_64-Personal
    /*
    static void ConnectSWIGToActiviz(Kitware.VTK.vtkImageExport imgin, Kitware.VTK.vtkImageImport imgout)
    {
        imgout.SetUpdateInformationCallback(imgin.GetUpdateInformationCallback());
        imgout.SetPipelineModifiedCallback(imgin.GetPipelineModifiedCallback());
        imgout.SetWholeExtentCallback(imgin.GetWholeExtentCallback());
        imgout.SetSpacingCallback(imgin.GetSpacingCallback());
        imgout.SetOriginCallback(imgin.GetOriginCallback());
        imgout.SetScalarTypeCallback(imgin.GetScalarTypeCallback());
        imgout.SetNumberOfComponentsCallback(imgin.GetNumberOfComponentsCallback());
        imgout.SetPropagateUpdateExtentCallback(imgin.GetPropagateUpdateExtentCallback());
        imgout.SetUpdateDataCallback(imgin.GetUpdateDataCallback());
        imgout.SetDataExtentCallback(imgin.GetDataExtentCallback());
        imgout.SetBufferPointerCallback(imgin.GetBufferPointerCallback());
        imgout.SetCallbackUserData(imgin.GetCallbackUserData());
    }
    */

    static Kitware.VTK.vtkImageData ConnectSWIGToActiviz(vtkgdcm.vtkImageData imgin)
    {
        HandleRef rawCppThis = imgin.GetCppThis();
        Kitware.VTK.vtkImageData imgout = new Kitware.VTK.vtkImageData( rawCppThis.Handle, false, false);
        return imgout;
    }
}

```

```

    }

    static vtkgdcml.vtkImageData ConnectActivizToSWIG(Kitware.VTK.vtkImageData imgin)
    {
        HandleRef rawCppThis = imgin.GetCppThis();
        vtkgdcml.vtkImageData imgout = new vtkgdcml.vtkImageData( rawCppThis );
        return imgout;
    }

    public static int Main(string[] args)
    {
        string filename = args[0];
        string outfilename = args[1];

        // Step 1. Test SWIG -> Activiz
        vtkGDCMImageReader reader = vtkGDCMImageReader.New();
        reader.SetFileName( filename );
        //reader.Update(); // DO NOT call Update to check pipeline execution

        Kitware.VTK.vtkImageData imgout = ConnectSWIGToActiviz(reader.GetOutput());

        System.Console.WriteLine( imgout.ToString() ); // not initialized as expected

        vtkPNGWriter writer = new vtkPNGWriter();
        writer.SetInput( imgout );
        writer.SetFileName( outfilename );
        writer.Write();

        // Step 2. Test Activiz -> SWIG
        vtkPNGReader bmpreader = new vtkPNGReader();
        bmpreader.SetFileName( outfilename );
        //bmpreader.Update(); // DO NOT update to check pipeline execution

        System.Console.WriteLine( bmpreader.GetOutput().ToString() ); // not initialized as expected

        vtkgdcml.vtkImageData imgout2 = ConnectActivizToSWIG(bmpreader.GetOutput());

        System.Console.WriteLine( imgout2.ToString() ); // not initialized as expected

        Kitware.VTK.vtkMedicalImageProperties prop = new Kitware.VTK.vtkMedicalImageProperties();
        prop.SetModality( "MR" );

        string outfilename2 = args[2];
        vtkGDCMImageWriter writer2 = vtkGDCMImageWriter.New();
        writer2.SetMedicalImageProperties( prop.CastToActiviz() );
        writer2.SetFileName( outfilename2 );
        writer2.SetInput( imgout2 );
        writer2.Write();

        return 0;
    }
}

```

## 12.139 HelloActiviz2.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
using Kitware.VTK;
using Kitware.VTK.GDCM;

/*
 * Usage:
 * $ export MONO_PATH=/usr/lib/cli/Activiz.NET:/usr/lib/cli/Kitware.mummy.Runtime-1.0
 * $ mono ./bin/HelloActiviz2.exe gdcmlData/test.acr bla.png bla2.dcm

```

```

*/
/*
 * From the outside view, no-one can detect that object pass to/from
 * vtkGDCMImageWriter/vtkGDCMImageReader are not Activiz object.
 *
 * TODO: Test Command/Observer
 */
public class HelloActiviz2
{
    public static int Main(string[] args)
    {
        string filename = args[0];
        string outfilename = args[1];
        string outfilename2 = args[2];

        vtkGDCMImageReader reader = new Kitware.VTK.GDCM.vtkGDCMImageReader();
        reader.SetFileName( filename );

        // When calling multiple times creation of C# object from the same C++ object it triggers a:
        //error: potential refcounting error: Duplicate rawCppThis - weak reference that is still alive. Attempting to
        //      add '0x00b2dc10' again.
        //      Allowing new wrapped object to take over table key...
        //      Original object should *not* have been destroyed while we still had it in our table without notifying
        //      us...
        //reader.GetOutput();
        //reader.GetOutput();

        System.Console.WriteLine( reader.ToString() ); // Test the ToString compat with Activiz

        vtkGDCMImageWriter writer = new vtkGDCMImageWriter();
        writer.SetInput( reader.GetOutput() );
        writer.SetFileName( outfilename2 );
        writer.Write();

        System.Console.WriteLine( reader.GetOutput().ToString() ); // Test the ToString compat with Activiz

        System.Console.WriteLine( writer.ToString() ); // Test the ToString compat with Activiz

        vtkPNGWriter pngwriter = new vtkPNGWriter();
        pngwriter.SetInput( reader.GetOutput() );
        pngwriter.SetFileName( outfilename );
        pngwriter.Write();

        // at that point the .Write() should have triggered an Update() on the reader:
        if( reader.GetImageFormat() == vtkgdc.VTK_LUMINANCE ) // MONOCHROME2
        {
            System.Console.WriteLine( "Image is MONOCHROME2" ); //
        }

        vtkPNGReader bmpreader = new vtkPNGReader();
        bmpreader.SetFileName( outfilename );

        vtkMedicalImageProperties prop = new vtkMedicalImageProperties();
        prop.SetModality( "MR" );

        vtkMatrix4x4 dircos = reader.GetDirectionCosines();
        dircos.Invert();

        vtkGDCMImageWriter writer2 = new vtkGDCMImageWriter();
        writer2.SetFileName( outfilename2 );
        writer2.SetDirectionCosines( dircos );
        writer2.SetMedicalImageProperties( prop );
        writer2.SetInput( bmpreader.GetOutput() );
        writer2.Write();

        return 0;
    }
}

```

## 12.140 HelloActiviz3.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre

```

```

All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

    This software is distributed WITHOUT ANY WARRANTY; without even
    the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
    PURPOSE. See the above copyright notice for more information.

=====*/
using Kitware.VTK;
using Kitware.VTK.GDCM;

/*
 * $ export MONO_PATH=/usr/lib/cli/Activiz.NET:/usr/lib/cli/Kitware.mummy.Runtime-1.0
 * $ mono ./bin/HelloActiviz3.exe ~/Creatis/gdcmData/test.acr
 */
public class HelloActiviz3
{
    public static int Main(string[] args)
    {
        string filename = args[0];

        vtkGDCMImageReader reader = vtkGDCMImageReader.New();
        vtkStringArray array = vtkStringArray.New();
        array.InsertNextValue(filename);

        reader.SetFileNames(array);
        reader.Update();

        //System.Console.Write(reader.GetOutput());

        vtkRenderWindowInteractor iren = vtkRenderWindowInteractor.New();

        vtkImageViewer2 viewer = vtkImageViewer2.New();
        viewer.SetInput(reader.GetOutput());
        viewer.SetupInteractor(iren);
        viewer.SetSize(600, 600);
        viewer.Render();

        iren.Initialize();
        iren.Start();

        return 0;
    }
}

```

## 12.141 HelloActiviz4.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

    This software is distributed WITHOUT ANY WARRANTY; without even
    the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
    PURPOSE. See the above copyright notice for more information.

=====*/
using Kitware.VTK;
using Kitware.VTK.GDCM;

/*
 * $ export MONO_PATH=/usr/lib/cli/Activiz.NET:/usr/lib/cli/Kitware.mummy.Runtime-1.0
 * $ mono ./bin/HelloActiviz4.exe ~/Creatis/gdcmData/test.acr
 */
public class HelloActiviz4
{
    public static int Main(string[] args)
    {
        string filename = args[0];

        vtkGDCMImageReader reader = new vtkGDCMImageReader();
        vtkStringArray array = vtkStringArray.New();
        array.InsertNextValue(filename);

        reader.SetFileNames(array);
    }
}

```

```

        reader.Update();

        //System.Console.WriteLine(reader.GetOutput());

        vtkRenderWindowInteractor iren = vtkRenderWindowInteractor.New();

        vtkImageViewer viewer = vtkImageViewer.New();
        viewer.SetInput(reader.GetOutput());
        viewer.SetupInteractor(iren);
        viewer.SetSize(600, 600);
        viewer.Render();

        iren.Initialize();
        iren.Start();

        return 0;
    }
}

```

## 12.142 HelloActiviz5.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
using Kitware.VTK;
using Kitware.VTK.GDCM;

// The command line arguments are:
// -I      => run in interactive mode; unless this is used, the program will
//          not allow interaction and exit
// -D <path> => path to the data; the data should be in <path>/Data/

/*
 * $ export MONO_PATH=/usr/lib/cli/Activiz.NET:/usr/lib/cli/Kitware.mummy.Runtime-1.0
 * $ mono ./bin/HelloActiviz5.exe -I
 */
public class HelloActiviz5
{
    public static int Main(string[] args)
    {
        vtkTesting testHelper = vtkTesting.New();
        for (int cc = 0; cc < args.Length; cc++)
        {
            //testHelper.AddArguments(argc, const_cast<const char **>(argv));
            //System.Console.WriteLine( "args: " + args[cc] + "\n" );
            testHelper.AddArgument( args[cc] );
        }
        if ( testHelper.IsFlagSpecified("-D") != 0 )
        {
            string VTK_DATA_ROOT = vtkGDCMTesting.GetVTKDataRoot();
            if( VTK_DATA_ROOT != null )
            {
                //System.Console.WriteLine( "VTK_DATA_ROOT: " + VTK_DATA_ROOT + "\n" );
                testHelper.SetDataRoot( VTK_DATA_ROOT );
                testHelper.AddArgument( "-D" );
                testHelper.AddArgument( VTK_DATA_ROOT );
            }
        }

        string dataRoot = testHelper.GetDataRoot();
        string filename = dataRoot;
        filename += "/Data/mr.001";

        vtkDirectory dir = vtkDirectory.New();
        if( dir.FileIsDirectory( dataRoot ) == 0 )
        {
            filename = vtkGDCMTesting.GetGDCMDataRoot() + "/test.acr";

```

```

    }
    //System.Console.Write( "dataRoot: " + dataRoot + "\n" );
    System.Console.Write( "filename being used is: " + filename + "\n" );

    vtkGDCMImageReader reader = vtkGDCMImageReader.New();
    vtkStringArray array = vtkStringArray.New();
    array.InsertNextValue(filename);
    reader.SetFileNames(array);
    reader.Update();

    System.Console.Write(reader.GetOutput());

    vtkRenderWindowInteractor iren = vtkRenderWindowInteractor.New();

    vtkRenderer ren1 = vtkRenderer.New();
    vtkRenderWindow renWin = vtkRenderWindow.New();
    renWin.AddRenderer(ren1);

    vtkImageActor actor = vtkImageActor.New();

    vtkImageMapToWindowLevelColors coronalColors = vtkImageMapToWindowLevelColors.New();
    coronalColors.SetInput(reader.GetOutput());

    actor.SetInput(coronalColors.GetOutput());

    ren1.AddActor(actor);
    iren.SetRenderWindow(renWin);

    iren.Initialize();

    renWin.Render();

    int retVal = testHelper.IsInteractiveModeSpecified();

    if( retVal != 0 )
    {
        iren.Start();
    }

    return 0;
}

```

## 12.143 HelloVTKWorld.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
using vtkgdcm;

/*
 * This test only test the SWIG/VTK part, you do not need Activiz
 */
public class HelloVTKWorld
{
    public static int Main(string[] args)
    {
        string filename = args[0];
        vtkGDCMImageReader reader = vtkGDCMImageReader.New();
        reader.SetFileName( filename );
        reader.Update();

        vtkMedicalImageProperties prop = reader.GetMedicalImageProperties();
        System.Console.WriteLine( prop.GetPatientName() ); //

        if( reader.GetImageFormat() == vtkgdcm.vtkgdcm.VTK_LUMINANCE ) // MONOCHROME2
        {

```



```

        System.Console.WriteLine( "Image is MONOCHROME2" ); //
    }

    // Just for fun, invert the direction cosines, output should reflect that:
    vtkMatrix4x4 dircos = reader.GetDirectionCosines();
    dircos.Invert();

    string outfilename = args[1];
    vtkGDCMImageWriter writer = vtkGDCMImageWriter.New();
    writer.SetMedicalImageProperties( reader.GetMedicalImageProperties() );
    writer.SetDirectionCosines( dircos );
    writer.SetShift( reader.GetShift() );
    writer.SetScale( reader.GetScale() );
    writer.SetImageFormat( reader.GetImageFormat() );
    writer.SetFileName( outfilename );
    writer.SetInputConnection( reader.GetOutputPort() );
    writer.Write();

    return 0;
}

```

## 12.144 HelloVTKWorld2.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
using vtkgdcms;

/*
 * This test only test the SWIG/VTK part, you do not need Activiz
 */
public class HelloVTKWorld2
{
    public static int Main(string[] args)
    {
        string VTK_DATA_ROOT = vtkGDCMTesting.GetVTKDataRoot();

        vtkVolumel6Reader reader = vtkVolumel6Reader.New();
        reader.SetDataDimensions(64, 64);
        reader.SetDataByteOrderToLittleEndian();
        reader.SetFilePrefix(VTK_DATA_ROOT + "/Data/headsq/quarter");
        reader.SetImageRange(1, 93);
        reader.SetDataSpacing(3.2, 3.2, 1.5);

        vtkImageCast cast = vtkImageCast.New();
        cast.SetInputConnection( reader.GetOutputPort() );
        cast.SetOutputScalarTypeToUnsignedChar();

        // By default this is creating a Multiframe Grayscale Word Secondary Capture Image Storage
        vtkGDCMImageWriter writer = vtkGDCMImageWriter.New();
        writer.SetFileName( "headsq.dcm" );
        writer.SetInputConnection( reader.GetOutputPort() );
        // cast -> Multiframe Grayscale Byte Secondary Capture Image Storage
        // writer.SetInputConnection( cast.GetOutputPort() );
        writer.SetFileDimensionality( 3 );
        writer.Write();

        return 0;
    }
}

```

## 12.145 MetalImageMD5Activiz.cs

```

/*=====

```

```

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
using Kitware.VTK;
using Kitware.VTK.GDCM;
using gdcm;

/*
 * $ export MONO_PATH=/usr/lib/cli/Activiz.NET:/usr/lib/cli/Kitware.mummy.Runtime-1.0
 * $ mono ./bin/MetaImageMD5Activiz.exe gdcmData/012345.002.050.dcm
 */
public class MetaImageMD5Activiz
{
    public static int ProcessOneMHDMD5(string filename)
    {
        vtkGDCMImageReader reader = vtkGDCMImageReader.New();
        reader.FileLowerLeftOn();
        reader.DebugOff();
        int canread = reader.CanReadFile( filename );
        if( canread == 0 )
        {
            string refms = gdcm.Testing.GetMediaStorageFromFile(filename);
            if( gdcm.MediaStorage.IsImage( gdcm.MediaStorage.GetMSType(refms) ) )
            {
                System.Console.Write( "Problem with file: " + filename + "\n" );
                return 1;
            }
            // not an image
            return 0;
        }

        reader.SetFileName( filename );
        reader.Update();

        // System.Console.Write(reader.GetOutput());

        vtkMetaImageWriter writer = vtkMetaImageWriter.New();
        writer.SetCompression( false );
        writer.SetInput( reader.GetOutput() );
        string subdir = "MetaImageMD5Activiz";
        string tmpdir = gdcm.Testing.GetTempDirectory( subdir );
        if( !gdcm.PosixEmulation.FileIsDirectory( tmpdir ) )
        {
            gdcm.PosixEmulation.MakeDirectory( tmpdir );
        }
        string mhdfile = gdcm.Testing.GetTempFilename( filename, subdir );

        string rawfile = mhdfile;
        mhdfile += ".mhd";
        rawfile += ".raw";
        writer.SetFileName( mhdfile );
        writer.Write();

        string digestmhd = gdcm.Testing.ComputeFileMD5( mhdfile );
        string digestraw = gdcm.Testing.ComputeFileMD5( rawfile );

        string mhdref = vtkGDCMTesting.GetMHDMD5FromFile(filename);
        string rawref = vtkGDCMTesting.GetRAWMD5FromFile(filename);

        if( mhdref != digestmhd )
        {
            System.Console.Write( "Problem with mhd file: " + filename + "\n" );
            System.Console.Write( digestmhd );
            System.Console.Write( "\n" );
            System.Console.Write( mhdref );
            System.Console.Write( "\n" );
            return 1;
        }
        if( rawref != digestraw )
        {
            System.Console.Write( "Problem with raw file: " + filename + "\n" );
            System.Console.Write( digestraw );

```

```

        System.Console.Write( "\n" );
        System.Console.Write( rawref );
        System.Console.Write( "\n" );
        return 1;
    }

    return 0;
}

public static int Main(string[] args)
{
    if ( args.Length == 1 )
    {
        string filename = args[0];
        return ProcessOneMHDMD5( filename );
    }

    // Loop over all gdcmData
    gdcm.Trace.DebugOff();
    gdcm.Trace.WarningOff();
    gdcm.Trace.ErrorOff();

    uint n = gdcm.Testing.GetNumberOfFileNames();
    int ret = 0;
    for( uint i = 0; i < n; ++i )
    {
        string filename = gdcm.Testing.GetFileName( i );
        ret += ProcessOneMHDMD5( filename );
    }
    return ret;
}
}

```

## 12.146 RefCounting.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
using Kitware.VTK;
using Kitware.VTK.GDCM;

/*
 * this is not so much an example but simply a test to make sure ctor / dtor work as expected
 * and call the ::New and ->Delete() of VTK style.
 */
public class RefCounting
{
    public static int Main(string[] args)
    {
        {
            vtkGDCMTesting testing1 = vtkGDCMTesting.New();
            vtkGDCMTesting testing2 = new vtkGDCMTesting(); // just in case people do not read STYLE documentation

            vtkGDCMImageReader reader1 = vtkGDCMImageReader.New();
            vtkGDCMImageReader reader2 = new vtkGDCMImageReader();

            vtkGDCMImageWriter writer1 = vtkGDCMImageWriter.New();
            vtkGDCMImageWriter writer2 = new vtkGDCMImageWriter();

            using (vtkGDCMTesting testing3 = new vtkGDCMTesting())
            {
                System.Console.Write( "GetReferenceCount: " + testing1.GetReferenceCount() + "\n");
                System.Console.Write( "GetReferenceCount: " + testing2.GetReferenceCount() + "\n");
                System.Console.Write( "GetReferenceCount: " + testing3.GetReferenceCount() + "\n");
            }

            using (vtkGDCMImageReader reader3 = new vtkGDCMImageReader())
            {
                System.Console.Write( "GetReferenceCount: " + reader3.GetReferenceCount() + "\n");
            }
        }
    }
}

```

```

    }

    using (vtkGDCMImageWriter writer3 = vtkGDCMImageWriter.New())
    {
        System.Console.Write( "GetReferenceCount: " + writer3.GetReferenceCount() + "\n");
    }

    // C# destructor will call ->Delete on all C++ object as expected.
    return 0;
}
}

```

## 12.147 Compute3DSpacing.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader2.h"
#include "vtkImageChangeInformation.h"
#include "vtkStringArray.h"
#include "vtkVersion.h"
#include "gdcmIPPSorter.h"

#ifdef vtkFloatingPointType
#define vtkFloatingPointType double
#endif

/*
 * Simple example to check computation of spacing within vtkGDCMImageReader2
 * This is a direct implementation of:
 *
 *      http://gdcm.sourceforge.net/wiki/index.php/Using_GDCM_API#Automatic_ordering_of_slices_for_vtkGDCMImageReader.SetFileNames
 *
 * For more advanced information on how 3D spacing is being computed see:
 *
 * - http://gdcm.sourceforge.net/html/classgdcm_1_1IPPSorter.html
 *
 * Usage:
 *
 * $ Compute3DSpacing SIEMENS_MAGNETOM-12-MONO2-FileSeq0.dcm \
 *   SIEMENS_MAGNETOM-12-MONO2-FileSeq1.dcm \
 *   SIEMENS_MAGNETOM-12-MONO2-FileSeq2.dcm \
 *   SIEMENS_MAGNETOM-12-MONO2-FileSeq3.dcm
 */

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;

    std::vector<std::string> filenames;
    for( int i = 1; i < argc; ++i )
    {
        filenames.push_back( argv[i] );
    }

    gdcm::IPPSorter s;
    s.SetComputeZSpacing( true );
    s.SetZSpacingTolerance( 1e-3 );
    bool b = s.Sort( filenames );
    if( !b )
    {
        std::cerr << "Failed to sort files" << std::endl;
        return 1;
    }
    std::cout << "Sorting succeeded:" << std::endl;

```

```

//s.Print( std::cout );

std::cout << "Found z-spacing:" << std::endl;
std::cout << s.GetZSpacing() << std::endl;
const double ippzspacing = s.GetZSpacing();

const std::vector<std::string> & sorted = s.GetFileNames();
vtkGDCMImageReader2 * reader = vtkGDCMImageReader2::New();
vtkStringArray *files = vtkStringArray::New();
std::vector< std::string >::const_iterator it = sorted.begin();
for( ; it != sorted.end(); ++it)
{
    const std::string &f = *it;
    files->InsertNextValue( f.c_str() );
}
reader->SetFileNames( files );
reader->Update();

const vtkFloatingPointType *spacing = reader->GetOutput()->GetSpacing();
vtkImageChangeInformation *v16 = vtkImageChangeInformation::New();
#if (VTK_MAJOR_VERSION >= 6)
v16->SetInputConnection( reader->GetOutputPort() );
#else
v16->SetInput( reader->GetOutput() );
#endif
v16->SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
v16->Update();

v16->GetOutput()->Print( std::cout );

return 0;
}

```

## 12.148 Convert16BitsTo8Bits.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageCast.h"
#include "vtkVersion.h"

#include "gdcmTesting.h"
// The following file is 16/16/15 but the scalar range of the image is [0,192]
// it could be safely stored as 8bits instead:
// gdcmData/012345.002.050.dcm

int main(int, char *[])
{
    const char *directory = gdcm::Testing::GetDataRoot();
    if(!directory) return 1;
    std::string file = std::string(directory) + "/012345.002.050.dcm";
    std::cout << file << std::endl;

    vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
    reader->SetFileName( file.c_str() );
    reader->Update();
    //reader->GetOutput()->Print( std::cout );

    vtkImageCast *cast = vtkImageCast::New();
    if (VTK_MAJOR_VERSION >= 6)
        cast->SetInputConnection( reader->GetOutputPort() );
    else
        cast->SetInput( reader->GetOutput() );
    #endif
}

```

```

cast->SetOutputScalarTypeToUnsignedChar();

vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
writer->SetFileName( "/tmp/cast.dcm" );
#if (VTK_MAJOR_VERSION >= 6)
writer->SetInputConnection( cast->GetOutputPort() );
#else
writer->SetInput( cast->GetOutput() );
#endif
writer->SetImageFormat( reader->GetImageFormat() );
writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
writer->SetDirectionCosines( reader->GetDirectionCosines() );
writer->SetShift( reader->GetShift() );
writer->SetScale( reader->GetScale() );
writer->Write();

reader->Delete();
cast->Delete();
writer->Delete();

return 0;
}

```

## 12.149 ConvertMultiFrameToSingleFrame.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkStringArray.h"
#include "vtkVersion.h"

#include "gdcmTesting.h"
#include "gdcmFilenameGenerator.h"

int main(int argc, char *argv[])
{
    std::string filename;
    if( argc <= 1 )
    {
        const char *directory = gdcm::Testing::GetDataRoot();
        if(!directory) return 1;
        std::string file = std::string(directory) + "/US-PAL-8-10x-echo.dcm";
        filename = file;
    }
    else
    {
        filename = argv[1];
    }
    std::cout << "file: " << filename << std::endl;

    vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
    reader->SetFileName( filename.c_str() );
    reader->Update();
    //reader->GetOutput()->Print( std::cout );

    int dims[3];
    reader->GetOutput()->GetDimensions( dims );

    std::ostringstream os;
    os << "singleframe";
    os << "%04d.dcm";
    gdcm::FilenameGenerator fg;
    fg.SetPattern( os.str().c_str() );

```

```

unsigned int nfiles = dims[2];
fg.SetNumberOfFileNames( nfiles );
bool b = fg.Generate();
if( !b )
{
    std::cerr << "FilenameGenerator::Generate() failed" << std::endl;
    return 1;
}
if( !fg.GetNumberOfFileNames() )
{
    std::cerr << "FilenameGenerator::Generate() failed somehow..." << std::endl;
    return 1;
}

// By default write them as Secondary Capture (for portability)
vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
vtkStringArray *filenames = vtkStringArray::New();
for(unsigned int i = 0; i < fg.GetNumberOfFileNames(); ++i)
{
    filenames->InsertNextValue( fg.GetFileName(i) );
}
assert( filenames->GetNumberOfValues() == (int)fg.GetNumberOfFileNames() );
writer->SetFileNames( filenames );
filenames->Delete();
writer->SetFileDimensionality( 2 );
#if (VTK_MAJOR_VERSION >= 6)
    writer->SetInputConnection( reader->GetOutputPort() );
#else
    writer->SetInput( reader->GetOutput() );
#endif
writer->SetImageFormat( reader->GetImageFormat() );
writer->Write();

reader->Delete();
writer->Delete();

return 0;
}

```

## 12.150 ConvertRGBToLuminance.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageLuminance.h"
#include "vtkVersion.h"

#include "gdcmTesting.h"

// There is no such thing as MR Image Storage + Photometric Interpretation = RGB
// let's rewrite that into a proper single component image:
int main(int, char *[])
{
    const char *directory = gdcm::Testing::GetDataRoot();
    if(!directory) return 1;
    std::string file = std::string(directory) + "/SIEMENS-MR-RGB-16Bits.dcm";
    std::cout << file << std::endl;

    vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
    reader->SetFileName( file.c_str() );
    reader->Update();
    //reader->GetOutput()->Print( std::cout );

    vtkImageLuminance *luminance = vtkImageLuminance::New();
}

```

```

#if (VTK_MAJOR_VERSION >= 6)
    luminance->SetInputConnection( reader->GetOutputPort() );
#else
    luminance->SetInput( reader->GetOutput() );
#endif

    vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
    writer->SetFileName( "/tmp/bla.dcm" );
#if (VTK_MAJOR_VERSION >= 6)
    writer->SetInputConnection( luminance->GetOutputPort() );
#else
    writer->SetInput( luminance->GetOutput() );
#endif
    //writer->SetImageFormat( reader->GetImageFormat() ); // Do NOT pass image format
    writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
    writer->SetDirectionCosines( reader->GetDirectionCosines() );
    writer->SetShift( reader->GetShift() );
    writer->SetScale( reader->GetScale() );
    writer->Write();

    // TODO:
    //vtkImageAppendComponents.h

    reader->Delete();
    luminance->Delete();
    writer->Delete();

    return 0;
}

```

## 12.151 ConvertSingleBitTo8Bits.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageCast.h"
#include "vtkPointData.h"
#include "vtkBitArray.h"
#include "vtkUnsignedCharArray.h"
#include "vtkVersion.h"

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
    reader->SetFileName( filename );
    reader->Update();
    //reader->GetOutput()->Print( std::cout );

    vtkDataArray* array = reader->GetOutput()->GetPointData()->GetScalars();
    vtkBitArray *barray = vtkBitArray::SafeDownCast( array );
    if( !barray ) return false;
    vtkIdType nvalues = array->GetNumberOfTuples();
    vtkUnsignedCharArray *uarray = vtkUnsignedCharArray::New();
    uarray->SetNumberOfTuples( nvalues );
    for(vtkIdType i = 0; i < nvalues; ++i)
    {

```



```

    uarray->SetValue( i, (unsigned char)barray->GetValue(i) );
}

vtkImageData *copy = vtkImageData::New();
// http://www.vtk.org/Wiki/VTK/VTK_6_Migration/Changes_to_Scalars_Manipulation_Functions#AllocateScalars.28.29
copy->SetExtent( reader->GetOutput()->GetExtent() );
#if (VTK_MAJOR_VERSION >= 6)
copy->AllocateScalars(VTK_UNSIGNED_CHAR, 3);
#else
copy->SetScalarType( VTK_UNSIGNED_CHAR );
copy->AllocateScalars();
#endif

//uarray->Print( std::cout );
//copy->GetPointData()->GetScalars()->Print( std::cout );
copy->GetPointData()->SetScalars( uarray );
uarray->Delete();

vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
writer->SetFileName( outfilename );
//writer->SetInput( cast->GetOutput() );
#if (VTK_MAJOR_VERSION >= 6)
writer->SetInputData( copy );
#else
writer->SetInput( copy );
#endif
writer->SetImageFormat( reader->GetImageFormat() );
writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
writer->SetDirectionCosines( reader->GetDirectionCosines() );
writer->SetShift( reader->GetShift() );
writer->SetScale( reader->GetScale() );
writer->SetFileDimensionality( reader->GetFileDimensionality() );
writer->Write();

reader->Delete();
copy->Delete();
writer->Delete();

return 0;
}

```

## 12.152 CreateFakePET.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageWriter.h"
#include "vtkImageReader.h"
#include "vtkImageCast.h"
#include "vtkImageData.h"
#include "vtkPointData.h"
#include "vtkDataArray.h"
#include "vtkMedicalImageProperties.h"
#include "vtkStringArray.h"
#include "vtkVersion.h"

#include "gdcmTrace.h"
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"
#include "gdcmFilenameGenerator.h"

/*
 * Minimal example to create a fake RTDOSE file. The data contains a sphere
 * just for testing.
 * The vtkMedicalImageProperties is not properly filled, but only contains a
 * single field which is required to set the proper SOP Class
 */

```

```

*/
int main(int, char *[])
{
    gdcm::Trace::DebugOn();

    const vtkIdType xSize = 512;
    const vtkIdType ySize = 512;
    const vtkIdType zSize = 512;

    // Create the filenames in advance to supply to the vtkGDCMImageWriter
    std::ostream os;
    os << "PT";
    os << "%03d.dcm";
    gdcm::FilenameGenerator fg;
    fg.SetPattern( os.str().c_str() );
    unsigned int nfiles = zSize;
    fg.SetNumberOfFileNames( nfiles );
    bool b = fg.Generate();
    if( !b )
    {
        std::cerr << "FilenameGenerator::Generate() failed" << std::endl;
        return 1;
    }
    if( !fg.GetNumberOfFileNames() )
    {
        std::cerr << "FilenameGenerator::Generate() failed somehow..." << std::endl;
        return 1;
    }

    vtkStringArray *filenames = vtkStringArray::New();
    for(unsigned int i = 0; i < fg.GetNumberOfFileNames(); ++i)
    {
        filenames->InsertNextValue( fg.GetFilename(i) );
    }

    vtkImageData *image = vtkImageData::New();
    image->SetDimensions(xSize,ySize,zSize);
    image->SetOrigin(-350.684,350.0,890.76);
    image->SetSpacing(5.4688,-5.4688,-3.27);
    #if VTK_MAJOR_VERSION <= 5
        image->SetNumberOfScalarComponents(1);
        image->SetScalarTypeToDouble();
    #else
        image->AllocateScalars(VTK_DOUBLE,1);
    #endif

    double pt[3];
    for( int z = 0; z < zSize; ++z )
        for( int y = 0; y < ySize; ++y )
            for( int x = 0; x < xSize; ++x )
            {
                pt[0] = x;
                pt[1] = y;
                pt[2] = z;
                pt[0] -= xSize / 2;
                pt[1] -= ySize / 2;
                pt[2] -= zSize / 2;
                pt[0] /= xSize / 2;
                pt[1] /= ySize / 2;
                pt[2] /= zSize / 2;
                const double unit = pt[0] * pt[0] + pt[1] * pt[1] + pt[2] * pt[2];
                const double inval = unit <= 1. ? (3 * unit + 7) : 0.; // just for fun => max == 10.
                double* pixel= static_cast<double*>(image->GetScalarPointer(x,y,z));
                pixel[0] = inval;
            }

    vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
    writer->SetFileDimensionality( 2 );
    writer->SetFileNames(filenames);
    #if (VTK_MAJOR_VERSION >= 6)
        writer->SetInputData( image );
    #else
        writer->SetInput( image );
    #endif
    writer->GetMedicalImageProperties()->SetSliceThickness("1.5");
    writer->GetMedicalImageProperties()->SetModality( "PT" );
    writer->SetScale( 0.0042 ); // why not
    writer->Write();

    image->Delete();

```

```

writer->Delete();

return 0;
}

```

## 12.153 CreateFakeRTDOSE.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/

#include "vtkGDCMImageWriter.h"
#include "vtkImageReader.h"
#include "vtkImageCast.h"
#include "vtkImageData.h"
#include "vtkPointData.h"
#include "vtkDataArray.h"
#include "vtkMedicalImageProperties.h"
#include "vtkVersion.h"

#include "gdcmlTrace.h"
#include "gdcmlReader.h"
#include "gdcmlWriter.h"
#include "gdcmlAttribute.h"

/*
 * Minimal example to create a fake RTDOSE file. The data contains a sphere
 * just for testing.
 * The vtkMedicalImageProperties is not properly filled, but only contains a
 * single field which is required to set the proper SOP Class
 */
int main(int, char *[])
{
    //gdcml::Trace::DebugOn();

    const vtkIdType xSize = 512;
    const vtkIdType ySize = 512;
    const vtkIdType zSize = 512;

    vtkImageData *image = vtkImageData::New();
    image->SetDimensions(xSize,ySize,zSize);
    image->SetOrigin(-350.684,350.0,890.76);
    image->SetSpacing(5.4688,-5.4688,-3.27);
    #if VTK_MAJOR_VERSION <= 5
        image->SetNumberOfScalarComponents(1);
        image->SetScalarTypeToDouble();
    #else
        image->AllocateScalars(VTK_DOUBLE,1);
    #endif

    double pt[3];
    for( int z = 0; z < zSize; ++z )
        for( int y = 0; y < ySize; ++y )
            for( int x = 0; x < xSize; ++x )
            {
                pt[0] = x;
                pt[1] = y;
                pt[2] = z;
                pt[0] -= xSize / 2;
                pt[1] -= ySize / 2;
                pt[2] -= zSize / 2;
                pt[0] /= xSize / 2;
                pt[1] /= ySize / 2;
                pt[2] /= zSize / 2;
                const double unit = pt[0] * pt[0] + pt[1] * pt[1] + pt[2] * pt[2];
                const double inval = unit <= 1. ? (3 * unit + 7) : 0.; // just for fun => max == 10.
                double* pixel= static_cast<double*>(image->GetScalarPointer(x,y,z));
                pixel[0] = inval;
            }
}

```

```

    }

    vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
    writer->SetFileDimensionality( 3 );
    writer->SetFileName( "rtdose.dcm" );
#if (VTK_MAJOR_VERSION >= 6)
    writer->SetInputData( image );
#else
    writer->SetInput( image );
#endif
    writer->GetMedicalImageProperties()->SetSliceThickness("1.5");
    writer->GetMedicalImageProperties()->AddUserDefinedValue( "Dose Units", "GY");
    writer->GetMedicalImageProperties()->AddUserDefinedValue( "Dose Summation Type", "PLAN");
    writer->GetMedicalImageProperties()->AddUserDefinedValue( "Dose Type", "PHYSICAL");
    writer->GetMedicalImageProperties()->AddUserDefinedValue( "Frame of Reference UID",
        "1.3.12.2.1107.5.6.1.68100.30270111041215391275000000001");
    writer->GetMedicalImageProperties()->SetModality( "RTDOSE" );
    //writer->GetMedicalImageProperties()->SetModality( "PT" ); // debug
    writer->SetScale( 0.0042 ); // why not
    writer->Write();

    image->Delete();
    writer->Delete();

    // BEGIN HACK
    // In GDCM version 2.4.3 and before, the following tag was missing which caused issue with some RTDose
    // software:

    // Open the DICOM file that was temporarily created. This will allows me to used
    // GDCM to append specific tags that allows the RTDOSE to be associated with the
    // relevant CT images.
    gdcm::Reader reader2;
    reader2.SetFileName("rtdose.dcm" );
    reader2.Read();
    gdcm::File &file = reader2.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();

    // Required by some software and not automatically added by GDCM in old version
    gdcm::Attribute<0x0028,0x0009> framePointer;
    framePointer.SetNumberOfValues(1);
    framePointer.SetValue( gdcm::Tag(0x3004,0x000C) );
    ds.Replace( framePointer.GetAsDataElement() );

    gdcm::Writer writer2;
    writer2.CheckFileMetaInformationOff();
    writer2.SetFileName("rtdose2.dcm");
    writer2.SetFile( file );
    writer2.Write();
    // END HACK

    return 0;
}

```

## 12.154 GenerateRTSTRUCT.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMPolyDataWriter.h"
#include "vtkGDCMPolyDataReader.h"
#include "vtkPolyData.h"
#include "vtkPolyDataReader.h"
#include "vtkMedicalImageProperties.h"
#include "vtkRTStructSetProperties.h"
#include "vtkStringArray.h"

```

```

#include "vtkAppendPolyData.h"
#include "vtkPolyDataWriter.h"
#include "vtkPolyDataMapper.h"
#include "vtkPolyDataMapper2D.h"
#include "vtkActor2D.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkMedicalImageProperties.h"
#include "vtkRenderWindow.h"
#include "vtkRenderer.h"
#include "vtkCamera.h"
#include "vtkProperty.h"
#include "vtkProperty2D.h"
#include "vtkImageData.h"
#include "vtkVersion.h"

#include <algorithm> //for std::find

#include "gdcmDirectoryHelper.h"

using namespace gdcm;

//view each organ independently of the others, to make sure that
//organ names correspond to actual segmentations.
void ShowOrgan(vtkPolyData* inData)
{
    // Now we'll look at it.
    vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
    #if (VTK_MAJOR_VERSION >= 6)
        cubeMapper->SetInputData( inData );
    #else
        cubeMapper->SetInput( inData );
    #endif
    cubeMapper->SetScalarRange(0,7);
    vtkActor *cubeActor = vtkActor::New();
    cubeActor->SetMapper(cubeMapper);
    vtkProperty * property = cubeActor->GetProperty();
    property->SetRepresentationToWireframe();

    vtkRenderer *renderer = vtkRenderer::New();
    vtkRenderWindow *renWin = vtkRenderWindow::New();
    renWin->AddRenderer(renderer);

    vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
    iren->SetRenderWindow(renWin);

    renderer->AddActor(cubeActor);
    renderer->ResetCamera();
    renderer->SetBackground(1,1,1);

    renWin->SetSize(300,300);

    renWin->Render();
    iren->Start();

    cubeMapper->Delete();
    cubeActor->Delete();
    renderer->Delete();
    renWin->Delete();
    iren->Delete();
}

/*
 * Full application which ... RTSTRUCT
 */
int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " directory-with-rtstruct-and-ct-images\n";
        return 1;
    }
    std::string theDirName(argv[1]);
    Directory::FilenameType theRTSeries =
        DirectoryHelper::GetRTStructSeriesUIDs(theDirName);

    gdcm::Directory theDir;
    theDir.Load(argv[1]);

    if (theRTSeries.empty())
    {
        std::cerr << "No RTStructs found for the test, ending." << std::endl;
    }
}

```

```

    return 1;
}

for (size_t q = 0; q < theRTSeries.size(); q++)
{
    Directory::FileNamesType theRTNames =
        DirectoryHelper::GetFileNamesFromSeriesUIDs(theDirName, theRTSeries[q]);

    if (theRTNames.empty()){
        std::cerr << "Unable to load RT Series " << theRTSeries[q] << ", continuing. " << std::endl;
        continue;
    }

    vtkGDCMPolyDataReader * reader = vtkGDCMPolyDataReader::New();
    reader->SetFileName( theRTNames[0].c_str() );
    reader->Update();

    //std::cout << reader->GetMedicalImageProperties()->GetStudyDate() << std::endl;

    vtkGDCMPolyDataWriter * writer = vtkGDCMPolyDataWriter::New();
    int numMasks = reader->GetNumberOfOutputPorts() + 1; //add a blank one in
    writer->SetNumberOfInputPorts( numMasks );
    std::string thePotentialName = theDirName + "/" + "GDCMTestRTStruct." + theRTSeries[q] + ".dcm";
    gdcmm::Directory::FileNamesType theFileNames = theDir.GetFileNames();
    //keep renaming the output until we get something that doesn't overwrite what was there already
    int count = 0;
    while (std::find(theFileNames.begin(), theFileNames.end(), thePotentialName) != theFileNames.end())
    {
        char buff[255];
        snprintf(buff, sizeof(buff), "%d", count);
        thePotentialName = theDirName + "/" + "GDCMTestRTStruct." + buff + "." + theRTSeries[q] + ".dcm";
    }
    writer->SetFileName( thePotentialName.c_str() );
    writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
    //this line is cheating, we won't have the same stuff, and may not have a struct
    //to start with.
    //have to go back to the original data to reconstruct the RTStructureSetProperties
    //writer->SetRTStructSetProperties( reader->GetRTStructSetProperties() );
    //writer->Write();

    //loop through the outputs in order to write them out as if they had been created and appended
    vtkStringArray* roiNames = vtkStringArray::New();
    vtkStringArray* roiAlgorithms = vtkStringArray::New();
    vtkStringArray* roiTypes = vtkStringArray::New();
    roiNames->SetNumberOfValues(numMasks);
    roiAlgorithms->SetNumberOfValues(numMasks);
    roiTypes->SetNumberOfValues(numMasks);
    vtkAppendPolyData* append = vtkAppendPolyData::New();

    //ok, now we'll add a blank organ
    //the blank organ is to test to ensure that blank organs work; there have been crash reports
    //this code is added at the beginning to ensure that the blank organs are read
    //and preserved as individual organs.
    vtkPolyData* blank = vtkPolyData::New();
    #if (VTK_MAJOR_VERSION >= 6)
        writer->SetInputData(0, blank);
    #else
        writer->SetInput(0, blank);
    #endif
    roiNames->InsertValue(0, "blank");
    roiAlgorithms->InsertValue(0, "blank");
    roiTypes->InsertValue(0, "ORGAN");

    //note the offsets used to place the blank rtstruct at the beginning of the newly generated RT.
    //the idea is to run the program twice; first to generate an rtstruct with a blank mask (making
    //sure that that functionality works), and then a second time to make sure that everything is
    //being read properly. Multiple organs with the same name could cause some strangenesses.
    for (int i = 1; i < numMasks; ++i)
    {
        #if (VTK_MAJOR_VERSION >= 6)
            writer->SetInputConnection(i, reader->GetOutputPort(i-1));
            append->AddInputConnection(reader->GetOutputPort(i-1));
        #else
            writer->SetInput(i, reader->GetOutput(i-1));
            append->AddInput(reader->GetOutput(i-1));
        #endif
        std::string theString = reader->GetRTStructSetProperties()->GetStructureSetROIName(i-1);
        roiNames->InsertValue(i, theString);
        theString = reader->GetRTStructSetProperties()->GetStructureSetROIGenerationAlgorithm(i-1);
        roiAlgorithms->InsertValue(i, theString);
        theString = reader->GetRTStructSetProperties()->GetStructureSetRTROIInterpretedType(i-1);

```

```

        roiTypes->InsertValue(i, theString);

        ShowOrgan(reader->GetOutput(i-1));
    }

    vtkRTStructSetProperties* theProperties = vtkRTStructSetProperties::New();
    writer->SetRTStructSetProperties(theProperties);
    writer->InitializeRTStructSet(theDirName,
        reader->GetRTStructSetProperties()->GetStructureSetLabel(),
        reader->GetRTStructSetProperties()->GetStructureSetName(),
        roiNames, roiAlgorithms, roiTypes);

    writer->SetRTStructSetProperties(theProperties);
    writer->Write();

    // print reader output:
    reader->Print( std::cout );
    // print first output:
    reader->GetOutput()->Print( std::cout );

    reader->Delete();
    append->Delete();
    roiNames->Delete();
    roiTypes->Delete();
    theProperties->Delete();
    roiAlgorithms->Delete();
    blank->Delete();

    writer->Delete();
}
return 0;
}

```

## 12.155 MagnifyFile.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageMagnify.h"
#include "vtkImageCast.h"
#include "vtkVersion.h"

#include "gdcmTesting.h"
#include "gdcmSystem.h"

// This is a simple test to magnify an image that is known to give excellent
// compression ratio. This will be our test for those large image
int main(int, char *[])
{
    const char *directory = gdcm::Testing::GetDataRoot();
    if(!directory) return 1;
    std::string file = std::string(directory) + "/test.acr";
    std::cout << file << std::endl;
    if( !gdcm::System::FileExists( file.c_str() ) ) return 1;

    vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
    reader->SetFileName( file.c_str() );
    reader->Update();
    //reader->GetOutput()->Print( std::cout );

    vtkImageCast *cast = vtkImageCast::New();
    #if (VTK_MAJOR_VERSION >= 6)
        cast->SetInputConnection( reader->GetOutputPort() );
    #else

```

```

    cast->SetInput( reader->GetOutput() );
#endif
    cast->SetOutputScalarTypeToUnsignedShort();

    vtkImageMagnify *magnify = vtkImageMagnify::New();
    #if (VTK_MAJOR_VERSION >= 6)
        magnify->SetInputConnection( cast->GetOutputPort() );
    #else
        magnify->SetInput( cast->GetOutput() );
    #endif
    magnify->SetInterpolate( 1 );
    magnify->SetInterpolate( 0 );
    int factor = 100;
    magnify->SetMagnificationFactors (factor, factor, 1);

    vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
    writer->SetFileName( "/tmp/bla.dcm" );
    #if (VTK_MAJOR_VERSION >= 6)
        writer->SetInputConnection( magnify->GetOutputPort() );
    #else
        writer->SetInput( magnify->GetOutput() );
    #endif
    writer->SetImageFormat( reader->GetImageFormat() );
    writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
    writer->SetDirectionCosines( reader->GetDirectionCosines() );
    writer->SetShift( reader->GetShift() );
    writer->SetScale( reader->GetScale() );
    writer->Write();

    // TODO:
    //vtkImageAppendComponents.h

    reader->Delete();
    magnify->Delete();
    writer->Delete();

    return 0;
}

```

## 12.156 gdcmorthoplanes.cxx

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

#include "vtkActor.h"
#include "vtkCamera.h"
#include "vtkMatrix4x4.h"
#include "vtkTransform.h"
#include "vtkAssembly.h"
#include "vtkCellPicker.h"
#include "vtkCommand.h"
#include "vtkImageActor.h"
#include "vtkImageMapToColors.h"
#include "vtkImageOrthoPlanes.h"
#include "vtkImagePlaneWidget.h"
#include "vtkImageReader.h"
#include "vtkInteractorEventRecorder.h"
#include "vtkLookupTable.h"
#include "vtkOutlineFilter.h"
#include "vtkPolyDataMapper.h"
#include "vtkProperty.h"
#include "vtkRenderWindow.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkRenderer.h"
#include "vtkVolume16Reader.h"
#include "vtkImageData.h"

```



```

#include "vtkImageChangeInformation.h"
#include "vtkOrientationMarkerWidget.h"
#include "vtkAnnotatedCubeActor.h"
#include "vtkAxesActor.h"
#include "vtkCaptionActor2D.h"
#include "vtkTextProperty.h"
#include "vtkPropAssembly.h"

#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkStringArray.h"
#include "vtkVersion.h"

#include "gdcmsystem.h"
#include "gdcmdirectory.h"
#include "gdcmppsorter.h"

#ifdef vtkFloatingPointType
#define vtkFloatingPointType double
#endif

//-----
class vtkOrthoPlanesCallback : public vtkCommand
{
public:
    static vtkOrthoPlanesCallback *New()
    { return new vtkOrthoPlanesCallback; }

    void Execute( vtkObject *caller, unsigned long vtkNotUsed( event ),
                 void *callData )
    {
        vtkImagePlaneWidget* self =
            reinterpret_cast< vtkImagePlaneWidget* >( caller );
        if(!self) return;

        double* wl = static_cast<double*>( callData );

        if ( self == this->WidgetX )
        {
            this->WidgetY->SetWindowLevel(wl[0],wl[1],1);
            this->WidgetZ->SetWindowLevel(wl[0],wl[1],1);
        }
        else if( self == this->WidgetY )
        {
            this->WidgetX->SetWindowLevel(wl[0],wl[1],1);
            this->WidgetZ->SetWindowLevel(wl[0],wl[1],1);
        }
        else if (self == this->WidgetZ)
        {
            this->WidgetX->SetWindowLevel(wl[0],wl[1],1);
            this->WidgetY->SetWindowLevel(wl[0],wl[1],1);
        }
    }

    vtkOrthoPlanesCallback():WidgetX( 0 ), WidgetY( 0 ), WidgetZ ( 0 ) {}

    vtkImagePlaneWidget* WidgetX;
    vtkImagePlaneWidget* WidgetY;
    vtkImagePlaneWidget* WidgetZ;
};

int main( int argc, char *argv[] )
{
    //char* fname = vtkTestUtilities::ExpandDataFileName(argc, argv, "Data/headsq/quarter");

    //vtkVolume16Reader* v16 = vtkVolume16Reader::New();
    // v16->SetDataDimensions( 64, 64);
    // v16->SetDataByteOrderToLittleEndian();
    // v16->SetImageRange( 1, 93);
    // v16->SetDataSpacing( 3.2, 3.2, 1.5);
    // v16->SetFilePrefix( fname );
    // v16->SetDataMask( 0x7fff);
    // v16->Update();
    std::vector<std::string> filenames;
    if( argc < 2 )
    {
        std::cerr << argv[0] << " filename1.dcm [filename2.dcm ...]\n";
        return 1;
    }
    else
    {
        {

```

```

// Is it a single directory ? If so loop over all files contained in it:
const char *filename = argv[1];
if( argc == 2 && gdcm::System::FileIsDirectory( filename ) )
{
    std::cout << "Loading directory: " << filename << std::endl;
    bool recursive = false;
    gdcm::Directory d;
    d.Load(filename, recursive);
    gdcm::Directory::FileNamesType const &files = d.GetFilesNames();
    for( gdcm::Directory::FileNamesType::const_iterator it = files.begin(); it != files.end(); ++it )
    {
        filenames.push_back( it->c_str() );
    }
}
else // list of files passed directly on the cmd line:
    // discard non-existing or directory
{
    for(int i=1; i < argc; ++i)
    {
        filename = argv[i];
        if( gdcm::System::FileExists( filename ) )
        {
            if( gdcm::System::FileIsDirectory( filename ) )
            {
                std::cerr << "Discarding directory: " << filename << std::endl;
            }
            else
            {
                filenames.push_back( filename );
            }
        }
        else
        {
            std::cerr << "Discarding non existing file: " << filename << std::endl;
        }
    }
}
//names->Print( std::cout );
}

vtkGDCMImageReader * reader = vtkGDCMImageReader::New();
double ippzspacing;
if( filenames.size() > 1 )
{
    //gdcm::Trace::DebugOn();
    //gdcm::Trace::WarningOn();
    gdcm::IPPSorter s;
    s.SetComputeZSpacing( true );
    s.SetZSpacingTolerance( 1e-3 );
    bool b = s.Sort( filenames );
    if( !b )
    {
        std::cerr << "Failed to sort files" << std::endl;
        return 1;
    }
    std::cout << "Sorting succeeded:" << std::endl;
    s.Print( std::cout );

    std::cout << "Found z-spacing:" << std::endl;
    std::cout << s.GetZSpacing() << std::endl;
    ippzspacing = s.GetZSpacing();

    const std::vector<std::string> & sorted = s.GetFilesNames();
    vtkStringArray *files = vtkStringArray::New();
    std::vector< std::string >::const_iterator it = sorted.begin();
    for( ; it != sorted.end(); ++it )
    {
        const std::string &f = *it;
        files->InsertNextValue( f.c_str() );
    }
    reader->SetFileNames( files );
    //reader->SetFileLowerLeft( 1 );
    reader->Update(); // important
    files->Delete();
}
else
{
    reader->SetFileName( argv[1] );
    reader->Update(); // important
    ippzspacing = reader->GetOutput()->GetSpacing()[2];
    ippzspacing = 4;
}

```

```

    }

    //reader->GetOutput()->Print( std::cout );
    //vtkFloatingPointType range[2];
    //reader->GetOutput()->GetScalarRange(range);
    //std::cout << "Range: " << range[0] << " " << range[1] << std::endl;

    const vtkFloatingPointType *spacing = reader->GetOutput()->GetSpacing();

    vtkImageChangeInformation *v16 = vtkImageChangeInformation::New();
    #if (VTK_MAJOR_VERSION >= 6)
        v16->SetInputConnection( reader->GetOutputPort() );
    #else
        v16->SetInput( reader->GetOutput() );
    #endif
    v16->SetOutputSpacing( spacing[0], spacing[1], ippszspacing );
    v16->Update();

    #if 0
        vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
        writer->SetInput( v16->GetOutput() );
        writer->SetFileLowerLeft( reader->GetFileLowerLeft() );
        writer->SetDirectionCosines( reader->GetDirectionCosines() );
        writer->SetImageFormat( reader->GetImageFormat() );
        writer->SetFileDimensionality( 3); //reader->GetFileDimensionality() );
        writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
        writer->SetShift( reader->GetShift() );
        writer->SetScale( reader->GetScale() );
        writer->SetFileName( "out.dcm" );
        writer->Write();
    #endif

    vtkOutlineFilter* outline = vtkOutlineFilter::New();
    outline->SetInputConnection(v16->GetOutputPort());

    vtkPolyDataMapper* outlineMapper = vtkPolyDataMapper::New();
    outlineMapper->SetInputConnection(outline->GetOutputPort());

    vtkActor* outlineActor = vtkActor::New();
    outlineActor->SetMapper( outlineMapper);

    vtkRenderer* ren1 = vtkRenderer::New();
    vtkRenderer* ren2 = vtkRenderer::New();

    vtkRenderWindow* renWin = vtkRenderWindow::New();
    renWin->AddRenderer(ren2);
    renWin->AddRenderer(ren1);

    vtkRenderWindowInteractor* iren = vtkRenderWindowInteractor::New();
    iren->SetRenderWindow(renWin);

    vtkCellPicker* picker = vtkCellPicker::New();
    picker->SetTolerance(0.005);

    vtkProperty* ipwProp = vtkProperty::New();
    //assign default props to the ipw's texture plane actor

    vtkImagePlaneWidget* planeWidgetX = vtkImagePlaneWidget::New();
    planeWidgetX->SetInteractor( iren);
    planeWidgetX->SetKeyPressActivationValue('x');
    planeWidgetX->SetPicker(picker);
    planeWidgetX->RestrictPlaneToVolumeOn();
    planeWidgetX->GetPlaneProperty()->SetColor(1,0,0);
    planeWidgetX->SetTexturePlaneProperty(ipwProp);
    planeWidgetX->TextureInterpolateOff();
    planeWidgetX->SetResliceInterpolateToNearestNeighbour();
    #if (VTK_MAJOR_VERSION >= 6)
        planeWidgetX->SetInputConnection(v16->GetOutputPort());
    #else
        planeWidgetX->SetInput(v16->GetOutput());
    #endif
    #endif
    planeWidgetX->SetPlaneOrientationToXAxes();
    //planeWidgetX->SetSliceIndex(32);
    planeWidgetX->DisplayTextOn();
    planeWidgetX->On();
    planeWidgetX->InteractionOff();
    planeWidgetX->InteractionOn();

    vtkImagePlaneWidget* planeWidgetY = vtkImagePlaneWidget::New();
    planeWidgetY->SetInteractor( iren);

```

```

    planeWidgetY->SetKeyPressActivationValue('y');
    planeWidgetY->SetPicker(picker);
    planeWidgetY->GetPlaneProperty()->SetColor(1,1,0);
    planeWidgetY->SetTexturePlaneProperty(ipwProp);
    planeWidgetY->TextureInterpolateOn();
    planeWidgetY->SetResliceInterpolateToLinear();
    #if (VTK_MAJOR_VERSION >= 6)
        planeWidgetY->SetInputConnection(vl6->GetOutputPort());
    #else
        planeWidgetY->SetInput(vl6->GetOutput());
    #endif
    planeWidgetY->SetPlaneOrientationToYAxes();
    //planeWidgetY->SetSlicePosition(102.4);
    planeWidgetY->SetLookupTable(planeWidgetX->GetLookupTable());
    planeWidgetY->DisplayTextOn();
    planeWidgetY->UpdatePlacement();
    planeWidgetY->On();

    vtkImagePlaneWidget* planeWidgetZ = vtkImagePlaneWidget::New();
    planeWidgetZ->SetInteractor(iren);
    planeWidgetZ->SetKeyPressActivationValue('z');
    planeWidgetZ->SetPicker(picker);
    planeWidgetZ->GetPlaneProperty()->SetColor(0,0,1);
    planeWidgetZ->SetTexturePlaneProperty(ipwProp);
    planeWidgetZ->TextureInterpolateOn();
    planeWidgetZ->SetResliceInterpolateToCubic();
    #if (VTK_MAJOR_VERSION >= 6)
        planeWidgetZ->SetInputConnection(vl6->GetOutputPort());
    #else
        planeWidgetZ->SetInput(vl6->GetOutput());
    #endif
    planeWidgetZ->SetPlaneOrientationToZAxes();
    //planeWidgetZ->SetSliceIndex(25);
    planeWidgetZ->SetLookupTable(planeWidgetX->GetLookupTable());
    planeWidgetZ->DisplayTextOn();
    planeWidgetZ->On();

    vtkImageOrthoPlanes* orthoPlanes = vtkImageOrthoPlanes::New();
    orthoPlanes->SetPlane(0, planeWidgetX);
    orthoPlanes->SetPlane(1, planeWidgetY);
    orthoPlanes->SetPlane(2, planeWidgetZ);
    orthoPlanes->ResetPlanes();

    vtkOrthoPlanesCallback* cbk = vtkOrthoPlanesCallback::New();
    cbk->WidgetX = planeWidgetX;
    cbk->WidgetY = planeWidgetY;
    cbk->WidgetZ = planeWidgetZ;
    planeWidgetX->AddObserver(vtkCommand::EndWindowLevelEvent, cbk);
    planeWidgetY->AddObserver(vtkCommand::EndWindowLevelEvent, cbk);
    planeWidgetZ->AddObserver(vtkCommand::EndWindowLevelEvent, cbk);
    cbk->Delete();

    double wl[2];
    planeWidgetZ->GetWindowLevel(wl);

    // Add a 2D image to test the GetReslice method
    //
    vtkImageMapToColors* colorMap = vtkImageMapToColors::New();
    colorMap->PassAlphaToOutputOff();
    colorMap->SetActiveComponent(0);
    colorMap->SetOutputFormatToLuminance();
    #if (VTK_MAJOR_VERSION >= 6)
        colorMap->SetInputData(planeWidgetZ->GetResliceOutput());
    #else
        colorMap->SetInput(planeWidgetZ->GetResliceOutput());
    #endif
    colorMap->SetLookupTable(planeWidgetX->GetLookupTable());

    vtkImageActor* imageActor = vtkImageActor::New();
    imageActor->PickableOff();
    #if (VTK_MAJOR_VERSION >= 6)
        imageActor->SetInputData(colorMap->GetOutput());
    #else
        imageActor->SetInput(colorMap->GetOutput());
    #endif

    // Add the actors
    //
    ren1->AddActor(outlineActor);

```

```

ren2->AddActor( imageActor);

ren1->SetBackground( 0.1, 0.1, 0.2);
ren2->SetBackground( 0.2, 0.1, 0.2);

renWin->SetSize( 600, 350);

ren1->SetViewport(0,0,0.58333,1);
ren2->SetViewport(0.58333,0,1,1);

// Set the actors' positions
//
renWin->Render();
//iren->SetEventPosition( 175,175);
//iren->SetKeyCode('r');
//iren->InvokeEvent(vtkCommand::CharEvent,NULL);
//iren->SetEventPosition( 475,175);
//iren->SetKeyCode('r');
//iren->InvokeEvent(vtkCommand::CharEvent,NULL);
//renWin->Render();

//ren1->GetActiveCamera()->Elevation(110);
//ren1->GetActiveCamera()->SetViewUp(0, 0, -1);
//ren1->GetActiveCamera()->Azimuth(45);
//ren1->GetActiveCamera()->Dolly(1.15);
ren1->ResetCameraClippingRange();

vtkAnnotatedCubeActor* cube = vtkAnnotatedCubeActor::New();
cube->SetXPlusFaceText ( "R" );
cube->SetXMinusFaceText ( "L" );
cube->SetYPlusFaceText ( "A" );
cube->SetYMinusFaceText ( "P" );
cube->SetZPlusFaceText ( "H" );
cube->SetZMinusFaceText ( "F" );
cube->SetFaceTextScale( 0.666667 );

vtkAxesActor* axes2 = vtkAxesActor::New();

vtkMatrix4x4 *invert = vtkMatrix4x4::New();
invert->DeepCopy( reader->GetDirectionCosines() );
invert->Invert();

// simulate a left-handed coordinate system
//
vtkTransform *transform = vtkTransform::New();
transform->Identity();
//transform->RotateY(90);
transform->Concatenate(invert);
axes2->SetShaftTypeToCylinder();
axes2->SetUserTransform( transform );
cube->GetAssembly()->SetUserTransform( transform );

axes2->SetTotalLength( 1.5, 1.5, 1.5 );
axes2->SetCylinderRadius( 0.500 * axes2->GetCylinderRadius() );
axes2->SetConeRadius ( 1.025 * axes2->GetConeRadius() );
axes2->SetSphereRadius ( 1.500 * axes2->GetSphereRadius() );

vtkTextProperty* tprop = axes2->GetXAxisCaptionActor2D()->
    GetCaptionTextProperty();
tprop->ItalicOn();
tprop->ShadowOn();
tprop->SetFontFamilyToTimes();

axes2->GetYAxisCaptionActor2D()->GetCaptionTextProperty()->ShallowCopy( tprop );
axes2->GetZAxisCaptionActor2D()->GetCaptionTextProperty()->ShallowCopy( tprop );

vtkPropAssembly* assembly = vtkPropAssembly::New();
assembly->AddPart( axes2 );
assembly->AddPart( cube );

vtkOrientationMarkerWidget* widget = vtkOrientationMarkerWidget::New();
widget->SetOutlineColor( 0.9300, 0.5700, 0.1300 );
widget->SetOrientationMarker( assembly );
widget->SetInteractor( iren );
widget->SetViewport( 0.0, 0.0, 0.4, 0.4 );
widget->SetEnabled( 1 );
widget->InteractiveOff();
widget->InteractiveOn();

// Playback recorded events
//

```

```

//vtkInteractorEventRecorder *recorder = vtkInteractorEventRecorder::New();
//recorder->SetInteractor(iren);
//recorder->ReadFromInputStringOn();
//recorder->SetInputString(IOEventLog);

// Interact with data
// Render the image
//
iren->Initialize();
renWin->Render();

// Test SetKeyPressActivationValue for one of the widgets
//
//iren->SetKeyCode('z');
//iren->InvokeEvent(vtkCommand::CharEvent,NULL);
//iren->SetKeyCode('z');
//iren->InvokeEvent(vtkCommand::CharEvent,NULL);

//int retVal = vtkRegressionTestImage( renWin );
//
//if ( retVal == vtkRegressionTester::DO_INTERACTOR)
//{
//    iren->Start();
//}

// Clean up
//
//recorder->Off();
//recorder->Delete();

ipwProp->Delete();
orthoPlanes->Delete();
planeWidgetX->Delete();
planeWidgetY->Delete();
planeWidgetZ->Delete();
colorMap->Delete();
imageActor->Delete();
picker->Delete();
outlineActor->Delete();
outlineMapper->Delete();
outline->Delete();
iren->Delete();
renWin->Delete();
ren1->Delete();
ren2->Delete();
v16->Delete();
reader->Delete();

return 0;
}

```

## 12.157 gdcmmreslice.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"

#include "vtkRenderer.h"
#include "vtkAssembly.h"
#include "vtkImageFlip.h"
#include "vtkImageReslice.h"
#include "vtkRenderWindow.h"
#include "vtkAnnotatedCubeActor.h"
#include "vtkTransform.h"
#include "vtkAxesActor.h"
#include "vtkTextProperty.h"

```

```

#include "vtkCaptionActor2D.h"
#include "vtkPropAssembly.h"
#include "vtkOrientationMarkerWidget.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkImageData.h"
#include "vtkLookupTable.h"
#include "vtkTexture.h"
#include "vtkPlaneSource.h"
#include "vtkVersion.h"

int main( int argc, char *argv[] )
{
    if( argc < 2 ) return 1;
    vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
    reader->SetFileName( argv[1] );
    //reader->FileLowerLeftOn();
    reader->Update();

    vtkImageFlip *flip = vtkImageFlip::New();
    #if (VTK_MAJOR_VERSION >= 6)
        flip->SetInputConnection(reader->GetOutputPort());
    #else
        flip->SetInput(reader->GetOutput());
    #endif
    flip->SetFilteredAxis(0);
    flip->Update();

    vtkImageReslice *reslice = vtkImageReslice::New();
    //reslice->SetInput(reader->GetOutput());
    #if (VTK_MAJOR_VERSION >= 6)
        reslice->SetInputConnection(flip->GetOutputPort());
    #else
        reslice->SetInput(flip->GetOutput());
    #endif
    //reslice->SetResliceAxesDirectionCosines()
    reader->GetDirectionCosines()->Print(std::cout);
    vtkMatrix4x4 *invert = vtkMatrix4x4::New();
    invert->DeepCopy( reader->GetDirectionCosines() );
    invert->Invert();

    //reslice->SetResliceAxes( reader->GetDirectionCosines() );
    reslice->SetResliceAxes( invert );
    reslice->Update();
    vtkImageData* ima = reslice->GetOutput();

    vtkLookupTable* table = vtkLookupTable::New();
    table->SetNumberOfColors(1000);
    table->SetTableRange(0,1000);
    table->SetSaturationRange(0,0);
    table->SetHueRange(0,1);
    table->SetValueRange(0,1);
    table->SetAlphaRange(1,1);
    table->Build();

    // Texture
    vtkTexture* texture = vtkTexture::New();
    #if (VTK_MAJOR_VERSION >= 6)
        texture->SetInputData(ima);
    #else
        texture->SetInput(ima);
    #endif
    texture->InterpolateOn();
    texture->SetLookupTable(table);

    // PlaneSource
    vtkPlaneSource* plane = vtkPlaneSource::New();

    // PolyDataMapper
    vtkPolyDataMapper *planeMapper = vtkPolyDataMapper::New();
    #if (VTK_MAJOR_VERSION >= 6)
        planeMapper->SetInputConnection(plane->GetOutputPort());
    #else
        planeMapper->SetInput(plane->GetOutput());
    #endif

    // Actor
    vtkActor* planeActor = vtkActor::New();
    planeActor->SetTexture(texture);
    planeActor->SetMapper(planeMapper);

```

```

planeActor->PickableOn();

// Final rendering with simple interactor:
vtkRenderer *ren = vtkRenderer::New();
vtkRenderWindow *renwin = vtkRenderWindow::New();
renwin->AddRenderer(ren);
vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
iren->SetRenderWindow(renwin);
ren->AddActor(planeActor);
ren->SetBackground(0,0,0.5);

// DICOM is RAH:
vtkAnnotatedCubeActor* cube = vtkAnnotatedCubeActor::New();
cube->SetXPlusFaceText ( "R" );
cube->SetXMinusFaceText ( "L" );
cube->SetYPlusFaceText ( "A" );
cube->SetYMinusFaceText ( "P" );
cube->SetZPlusFaceText ( "H" );
cube->SetZMinusFaceText ( "F" );

vtkAxesActor* axes2 = vtkAxesActor::New();

vtkTransform *transform = vtkTransform::New();
transform->Identity();
//reader->GetDirectionCosines()->Print(std::cout);
transform->Concatenate(invert);
//axes2->SetShaftTypeToCylinder();
axes2->SetUserTransform( transform );
cube->GetAssembly()->SetUserTransform( transform ); // can't get it to work

vtkPropAssembly* assembly = vtkPropAssembly::New();
assembly->AddPart( axes2 );
assembly->AddPart( cube );

vtkOrientationMarkerWidget* widget = vtkOrientationMarkerWidget::New();
widget->SetOrientationMarker( assembly );
widget->SetInteractor( iren );
widget->SetEnabled( 1 );
widget->InteractiveOff();
widget->InteractiveOn();

renwin->Render();
iren->Start();

// Clean up:
reader->Delete();
table->Delete();
texture->Delete();
plane->Delete();
planeMapper->Delete();
planeActor->Delete();
ren->Delete();
renwin->Delete();
iren->Delete();

return 0;
}

```

## 12.158 gdcmrtionplan.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkImageData.h"
#include "vtkPointData.h"
#include "vtkPolyData.h"
#include "vtkProperty.h"

```



```

#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkRenderer.h"
#include "vtkCellArray.h"
#include "vtkPoints.h"
#include "vtkDoubleArray.h"
#include <vtkXMLImageDataWriter.h>
#include <vtkXMLPolyDataWriter.h>
#include <vtkRenderWindowInteractor.h>
#include <vtkImageColorViewer.h>
#include "vtkVersion.h"

#include "gdcmmReader.h"
#include "gdcmmAttribute.h"

/*
  This example is just for fun. We found a RT Ion Plan Storage and simply extracted the viz stuff for VTK

  RTIonPlanStorage, // 1.2.840.10008.5.1.4.1.1.481.8
*/
int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " filename.dcm outfile.vti\n";
        return 1;
    }
    const char * filename = argv[1];
    const char * outfilename = argv[2];
    const char * outfilename2 = argv[3];

    gdcmm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcmm::MediaStorage ms;
    ms.SetFromFile( reader.GetFile() );
    if( ms != gdcmm::MediaStorage::RTIonPlanStorage )
    {
        return 1;
    }

    /*
    (300a,03a2) SQ                                     # u/1,1 Ion Beam Sequence
      (fffe,e000) na (Item with undefined length)
        (0008,1040) LO [Test]                           # 4,1 Institutional Department Name
        (300a,00b2) SH (no value)                         # 0,1 Treatment Machine Name
        (300a,00b3) CS [MU]                               # 2,1 Primary Dosimeter Unit
        (300a,00c0) IS [1 ]                             # 2,1 Beam Number
        (300a,00c2) LO [1 ]                             # 2,1 Beam Name
        (300a,00c4) CS [STATIC]                          # 6,1 Beam Type
        (300a,00c6) CS [PROTON]                          # 6,1 Radiation Type
        (300a,00ce) CS [TREATMENT ]                     # 10,1 Treatment Delivery Type
        (300a,00d0) IS [0 ]                             # 2,1 Number of Wedges
        (300a,00e0) IS [1 ]                             # 2,1 Number of Compensators
        (300a,00ed) IS [0 ]                             # 2,1 Number of Boli
        (300a,00f0) IS [1 ]                             # 2,1 Number of Blocks
        (300a,0110) IS [2 ]                             # 2,1 Number of Control Points
        (300a,02ea) SQ                                   # u/1,1 Ion Range Compensator Sequence
          (fffe,e000) na (Item with undefined length)
            (300a,00e1) SH [lucite]                       # 6,1 Material ID
            (300a,00e4) IS [1 ]                           # 2,1 Compensator Number
            (300a,00e5) SH [75hdhe5 ]                     # 8,1 Compensator ID
            (300a,00e7) IS [35]                           # 2,1 Compensator Rows
            (300a,00e8) IS [37]                           # 2,1 Compensator Columns
            (300a,00e9) DS [3.679991\4.249288 ]           # 18,2 Compensator Pixel Spacing
            (300a,00ea) DS [-76.00\62.50]                 # 12,2 Compensator Position
            (300a,00ec) DS
            [52.13\52.13\52.13\53.18\54.04\54.04\47.11\40.06\40.06\38.79\34.87\33.28\33.28\33.28\33.28\35.43\35.43\34.54\34.54\34.71\36.
            # 7618,1-n Compensator Thickness Data
            (300a,02e0) CS [ABSENT]                       # 6,1 Compensator Divergence
            (300a,02e1) CS [SOURCE_SIDE ]                 # 12,1 Compensator Mounting Position
            (300a,02e4) FL 39.2                           # 4,1 Isocenter to Compensator Tray Distance
            (300a,02e5) FL 2.12                           # 4,1 Compensator Column Offset
            (300a,02e8) FL 4.76                           # 4,1 Compensator Milling Tool Diameter
          (fffe,e00d)
    */
    const gdcmm::DataSet& ds = reader.GetFile().GetDataSet();

```

```

gdcmm::Tag tbeamsq(0x300a,0x03a2);
if( !ds.FindDataElement( tbeamsq ) )
{
    return 1;
}
const gdcmm::DataElement &beamsq = ds.GetDataElement( tbeamsq );
//std::cout << beamsq << std::endl;
gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqi = beamsq.GetValueAsSQ();
if( !sqi || !sqi->GetNumberOfItems() )
{
    return 1;
}

//for(unsigned int pd = 0; pd < sqi->GetNumberOfItems(); ++pd)
// {
//const gdcmm::Item & item = sqi->GetItem(1); // Item start at #1
const gdcmm::Item & item = sqi->GetItem(1); // Item start at #1
const gdcmm::DataSet& nestedds = item.GetNestedDataSet();
//std::cout << nestedds << std::endl;
gdcmm::Tag tcompensatorsq(0x300a,0x02ea);
if( !nestedds.FindDataElement( tcompensatorsq ) )
{
    return 1;
}
const gdcmm::DataElement &compensatorsq = nestedds.GetDataElement( tcompensatorsq );
//std::cout << compensatorsq << std::endl;
gdcmm::SmartPointer<gdcmm::SequenceOfItems> ssqi = compensatorsq.GetValueAsSQ();
const gdcmm::Item & item2 = ssqi->GetItem(1); // Item start at #1
const gdcmm::DataSet& nestedds2 = item2.GetNestedDataSet();
//std::cout << nestedds2 << std::endl;
gdcmm::Tag tcompensatorthicknessdata(0x300a,0x00ec);
if( !nestedds2.FindDataElement( tcompensatorthicknessdata ) )
{
    return 1;
}
const gdcmm::DataElement &compensatorthicknessdata = nestedds2.GetDataElement( tcompensatorthicknessdata );
// std::cout << compensatorthicknessdata << std::endl;
gdcmm::Attribute<0x300a,0x00ec> at;
at.SetFromDataElement( compensatorthicknessdata );
const double* pts = at.GetValues();
// (300a,00e7) IS [35] # 2,1 Compensator Rows
gdcmm::Attribute<0x300a,0x00e7> at1;
const gdcmm::DataElement &compensatorrows = nestedds2.GetDataElement( at1.GetTag() );
at1.SetFromDataElement( compensatorrows );
std::cout << at1.GetValue() << std::endl;
// (300a,00e8) IS [37] # 2,1 Compensator Columns
gdcmm::Attribute<0x300a,0x00e8> at2;
const gdcmm::DataElement &compensatorcols = nestedds2.GetDataElement( at2.GetTag() );
at2.SetFromDataElement( compensatorcols );
std::cout << at2.GetValue() << std::endl;

// (300a,00e9) DS [3.679991\4.249288 ] # 18,2 Compensator Pixel Spacing
gdcmm::Attribute<0x300a,0x00e9> at3;
const gdcmm::DataElement &compensatorpixelspacing = nestedds2.GetDataElement( at3.GetTag() );
at3.SetFromDataElement( compensatorpixelspacing );
std::cout << at3.GetValue(0) << std::endl;
// (300a,00ea) DS [-76.00\62.50] # 12,2 Compensator Position
gdcmm::Attribute<0x300a,0x00ea> at4;
const gdcmm::DataElement &compensatorposition = nestedds2.GetDataElement( at4.GetTag() );
at4.SetFromDataElement( compensatorposition );
std::cout << at4.GetValue(0) << std::endl;

vtkDoubleArray *d = vtkDoubleArray::New();
d->SetArray( const_cast<double*>(pts) , at1.GetValue() * at2.GetValue() , 0 );

vtkImageData *img = vtkImageData::New();
img->Initialize();
img->SetDimensions( at2.GetValue(), at1.GetValue(), 1 );
//imgb->SetExtent(1, xdim, 1, ydim, 1, zdim);
#if (VTK_MAJOR_VERSION >= 6)
    assert(0);
#else
    img->SetScalarTypeToDouble();
#endif
img->SetSpacing( at3.GetValue(1), at3.GetValue(0), 1); // FIXME image is upside down
img->SetOrigin( at4.GetValue(0), at4.GetValue(1), 1);
#if (VTK_MAJOR_VERSION >= 6)
    assert(0);
#else
    img->SetNumberOfScalarComponents(1);
#endif

```

```

img->GetPointData()->SetScalars(d);

#if (VTK_MAJOR_VERSION >= 6)
#else
    img->Update();
#endif
img->Print(std::cout);

vtkXMLImageDataWriter *writeb= vtkXMLImageDataWriter::New();
#if (VTK_MAJOR_VERSION >= 6)
    writeb->SetInputData( img );
#else
    writeb->SetInput( img );
#endif
writeb->SetFileName( outfilename );
writeb->Write( );

/*
    (300a,03a6) SQ                                # u/l,1 Ion Block Sequence
    (fffe,e000) na (Item with undefined length)
    (300a,00e1) SH [brass ]                        # 6,1 Material ID
    (300a,00f7) FL 95.03                          # 4,1 Isocenter to Block Tray Distance
    (300a,00f8) CS [APERTURE]                      # 8,1 Block Type
    (300a,00fa) CS [ABSENT]                        # 6,1 Block Divergence
    (300a,00fb) CS [SOURCE_SIDE ]                 # 12,1 Block Mounting Position
    (300a,00fc) IS [1 ]                           # 2,1 Block Number
    (300a,0100) DS [50.00 ]                       # 6,1 Block Thickness
    (300a,0104) IS [179 ]                         # 4,1 Block Number of Points
    (300a,0106) DS
    [1.7\50.0\14.3\50.0\16.7\49.4\18.7\48.2\19.4\47.7\20.1\47.1\21.0\47.0\22.3\47.0\23.7\46.8\25.7\46.2\27.0\45.6\27.2\45.4\28.2
2\37.4\43.0\37.1\44.7\36] # 1934,2-2n Block Data
    (fffe,e00d)
    (fffe,e0dd)

*/
gdcmm::Tag tblocksq(0x300a,0x03a6);
if( !nestedds.FindDataElement( tblocksq ) )
{
    return 1;
}
const gdcmm::DataElement &blocksq = nestedds.GetDataElement( tblocksq );
//std::cout << blocksq << std::endl;
gdcmm::SmartPointer<gdcmm::SequenceOfItems> sssqi = blocksq.GetValueAsSQ();
const gdcmm::Item & item3 = sssqi->GetItem(1); // Item start at #1
const gdcmm::DataSet& nestedds3 = item3.GetNestedDataSet();

gdcmm::Tag tblockdata(0x300a,0x0106);
if( !nestedds3.FindDataElement( tblockdata ) )
{
    return 1;
}
const gdcmm::DataElement &blockdata = nestedds3.GetDataElement( tblockdata );
// std::cout << blockdata << std::endl;
gdcmm::Attribute<0x300a,0x0106> at_;
at_.SetFromDataElement( blockdata );

vtkDoubleArray *scalars = vtkDoubleArray::New();
scalars->SetNumberOfComponents(3);

gdcmm::Attribute<0x300a,0x0104> bnpts; // IS [179 ] # 4,1 Block Number
of Points
if( !nestedds3.FindDataElement( bnpts.GetTag() ) )
{
    return 1;
}
const gdcmm::DataElement &blocknpts = nestedds3.GetDataElement( bnpts.GetTag() );
bnpts.SetFromDataElement( blocknpts );
//std::cout << bnpts.GetValue() << std::endl;

vtkPolyData *output = vtkPolyData::New();
vtkPoints *newPts = vtkPoints::New();
vtkCellArray *polys = vtkCellArray::New();
const double *ptr = at_.GetValues();
//unsigned int npts = bnpts.GetNumberOfValues() / 2;
unsigned int npts = bnpts.GetValue();
vtkIdType *ptIds = new vtkIdType[npts];
for(unsigned int i = 0; i < npts; ++i)
{
    float x[3] = {};
    x[0] = (float)ptr[2*i+0];
    x[1] = (float)ptr[2*i+1];
    //x[2] = pts[i+2];

```

```

        vtkIdType ptId = newPts->InsertNextPoint( x );
        //std::cout << x[0] << ", " << x[1] << ", " << x[2] << std::endl;
        ptIds[i] = ptId;
    }
    vtkIdType cellId = polys->InsertNextCell(npts , ptIds);
    (void)cellId;
    delete[] ptIds;

    output->SetPoints(newPts);
    newPts->Delete();
    output->SetPolys(polys);
    polys->Delete();
    //output->GetCellData()->SetScalars(scalars);
    //scalars->Delete();
    #if (VTK_MAJOR_VERSION >= 6)
    #else
        output->Update();
    #endif
    output->Print( std::cout );

    // }

    vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();

    vtkImageColorViewer *viewer = vtkImageColorViewer::New();
    #if (VTK_MAJOR_VERSION >= 6)
        viewer->SetInputData(img);
    #else
        viewer->SetInput(img);
    #endif
    viewer->SetupInteractor(iren);
    viewer->SetSize(600, 600);
    viewer->GetRenderer()->ResetCameraClippingRange();
    viewer->Render();
    viewer->GetRenderer()->ResetCameraClippingRange();

    vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
    //vtkPolyDataMapper2D* cubeMapper = vtkPolyDataMapper2D::New();
    #if (VTK_MAJOR_VERSION >= 6)
        cubeMapper->SetInputData( output );
    #else
        cubeMapper->SetInput( output );
    #endif
    cubeMapper->SetScalarRange(0,7);
    vtkActor *cubeActor = vtkActor::New();
    //vtkActor2D* cubeActor = vtkActor2D::New();
    cubeActor->SetMapper(cubeMapper);
    vtkProperty *property = cubeActor->GetProperty();
    property->SetRepresentationToWireframe();

    viewer->GetRenderer()->AddActor( cubeActor );

    vtkXMLPolyDataWriter *writec= vtkXMLPolyDataWriter::New();
    #if (VTK_MAJOR_VERSION >= 6)
        writec->SetInputData( output );
    #else
        writec->SetInput( output );
    #endif
    writec->SetFileName( outfilename2 );
    writec->Write();

    iren->Initialize();
    iren->Start();

    return 0;
}

```

## 12.159 gdcmrtpplan.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

```

Copyright (c) 2006-2011 Mathieu Malaterre  
 All rights reserved.  
 See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even  
 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR  
 PURPOSE. See the above copyright notice for more information.

```

=====*/
#include "vtkImageData.h"
#include "vtkPointData.h"
#include "vtkPolyData.h"
#include "vtkProperty.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkRenderer.h"
#include "vtkCellArray.h"
#include "vtkPoints.h"
#include "vtkDoubleArray.h"
#include <vtkXMLImageDataWriter.h>
#include <vtkRenderWindowInteractor.h>
#include <vtkImageColorViewer.h>
#include "vtkVersion.h"

#include "gdcmReader.h"
#include "gdcmAttribute.h"

/*
  This example is just for fun. We found a fake RT Ion Plan Storage and simply extracted the viz stuff for VTK
  but this is rather a RT Plan storage
*/
int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " filename.dcm outfile.vti\n";
        return 1;
    }
    const char * filename = argv[1];
    const char * outfile = argv[2];

    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcm::MediaStorage ms;
    ms.SetFromFile( reader.GetFile() );
    if( ms != gdcm::MediaStorage::RTIonPlanStorage )
    {
        return 1;
    }

    /*
    (300a,00b0) SQ                                # u/1,1 Beam Sequence
    (ffff,e000) na (Item with undefined length)
    (300a,00b2) SH (no value)                      # 0,1 Treatment Machine Name
    (300a,00c0) IS [1 ]                             # 2,1 Beam Number
    (300a,00c2) LO [1 ]                             # 2,1 Beam Name
    (300a,00c4) CS [STATIC]                         # 6,1 Beam Type
    (300a,00c6) CS [PROTON]                         # 6,1 Radiation Type
    (300a,00ce) CS [TREATMENT ]                     # 10,1 Treatment Delivery Type
    (300a,00e0) IS [1 ]                             # 2,1 Number of Compensators
    (300a,00e3) SQ                                # u/1,1 Compensator Sequence
    (ffff,e000) na (Item with undefined length)
    (300a,00e1) SH [lucite]                         # 6,1 Material ID
    (300a,00e4) IS [1 ]                             # 2,1 Compensator Number
    (300a,00e5) SH [75hdhe5 ]                       # 8,1 Compensator ID
    (300a,00e7) IS [35]                             # 2,1 Compensator Rows
    (300a,00e8) IS [37]                             # 2,1 Compensator Columns
    (300a,00e9) DS [3.679991\4.249288 ]             # 18,2 Compensator Pixel Spacing
    (300a,00ea) DS [-76.00\62.50]                   # 12,2 Compensator Position
    (300a,00ec) DS
    [52.13\52.13\52.13\53.18\54.04\54.04\47.11\40.06\40.06\38.79\34.87\33.28\33.28\33.28\33.28\35.43\35.43\34.54\34.54\34.71\36.
    # 7618,1-n Compensator Thickness Data
    (300a,02e0) CS [ABSENT]                         # 6,1 Compensator Divergence
    (300a,02e1) CS [SOURCE_SIDE ]                   # 12,1 Compensator Mounting Position
    (ffff,e00d)
    (ffff,e000) na (Item with undefined length)
  
```

```

        (fffe,e00d)
        (fffe,e0dd)
    */
    const gdcm::DataSet& ds = reader.GetFile().GetDataSet();
    gdcm::Tag tbeamsq(0x300a,0x00b0);
    if( !ds.FindDataElement( tbeamsq ) )
    {
        return 1;
    }
    const gdcm::DataElement &tbeamsq = ds.GetDataElement( tbeamsq );
    //std::cout << tbeamsq << std::endl;
    gdcm::SmartPointer<gdcm::SequenceOfItems> sqi = tbeamsq.GetValueAsSQ();
    if( !sqi || !sqi->GetNumberOfItems() )
    {
        return 1;
    }

    //for(unsigned int pd = 0; pd < sqi->GetNumberOfItems(); ++pd)
    // {
    //const gdcm::Item & item = sqi->GetItem(1); // Item start at #1
    const gdcm::Item & item = sqi->GetItem(2); // Item start at #1
    const gdcm::DataSet& nestedds = item.GetNestedDataSet();
    //std::cout << nestedds << std::endl;
    gdcm::Tag tcompensatorsq(0x300a,0x00e3);
    if( !nestedds.FindDataElement( tcompensatorsq ) )
    {
        return 1;
    }
    const gdcm::DataElement &tcompensatorsq = nestedds.GetDataElement( tcompensatorsq );
    //std::cout << tcompensatorsq << std::endl;
    gdcm::SmartPointer<gdcm::SequenceOfItems> ssqi = tcompensatorsq.GetValueAsSQ();
    const gdcm::Item & item2 = ssqi->GetItem(1); // Item start at #1
    const gdcm::DataSet& nestedds2 = item2.GetNestedDataSet();
    //std::cout << nestedds2 << std::endl;
    gdcm::Tag tcompensatorthicknessdata(0x300a,0x00ec);
    if( !nestedds2.FindDataElement( tcompensatorthicknessdata ) )
    {
        return 1;
    }
    const gdcm::DataElement &tcompensatorthicknessdata = nestedds2.GetDataElement( tcompensatorthicknessdata );
    // std::cout << tcompensatorthicknessdata << std::endl;
    gdcm::Attribute<0x300a,0x00ec> at;
    at.SetFromDataElement( tcompensatorthicknessdata );
    const double* pts = at.GetValues();
    //          (300a,00e7) IS [35] # 2,1 Compensator Rows
    gdcm::Attribute<0x300a,0x00e7> at1;
    const gdcm::DataElement &tcompensatorrows = nestedds2.GetDataElement( at1.GetTag() );
    at1.SetFromDataElement( tcompensatorrows );
    std::cout << at1.GetValue() << std::endl;
    //          (300a,00e8) IS [37] # 2,1 Compensator Columns
    gdcm::Attribute<0x300a,0x00e8> at2;
    const gdcm::DataElement &tcompensatorcols = nestedds2.GetDataElement( at2.GetTag() );
    at2.SetFromDataElement( tcompensatorcols );
    std::cout << at2.GetValue() << std::endl;

    // (300a,00e9) DS [3.679991\4.249288 ] # 18,2 Compensator Pixel Spacing
    gdcm::Attribute<0x300a,0x00e9> at3;
    const gdcm::DataElement &tcompensatorpixelspacing = nestedds2.GetDataElement( at3.GetTag() );
    at3.SetFromDataElement( tcompensatorpixelspacing );
    std::cout << at3.GetValue(0) << std::endl;
    // (300a,00ea) DS [-76.00\62.50] # 12,2 Compensator Position
    gdcm::Attribute<0x300a,0x00ea> at4;
    const gdcm::DataElement &tcompensatorposition = nestedds2.GetDataElement( at4.GetTag() );
    at4.SetFromDataElement( tcompensatorposition );
    std::cout << at4.GetValue(0) << std::endl;

    vtkDoubleArray *d = vtkDoubleArray::New();
    d->SetArray( const_cast<double*>(pts) , at1.GetValue() * at2.GetValue() , 0 );

    vtkImageData *img = vtkImageData::New();
    img->Initialize();
    img->SetDimensions( at2.GetValue(), at1.GetValue(), 1 );
    //img->SetExtent(1, xdim, 1, ydim, 1, zdim);
    #if (VTK_MAJOR_VERSION >= 6)
        assert(0);
    #else
        img->SetScalarTypeToDouble();
    #endif
    img->SetSpacing( at3.GetValue(1), at3.GetValue(0), 1); // FIXME image is upside down
    img->SetOrigin( at4.GetValue(0), at4.GetValue(1), 1);
    #if (VTK_MAJOR_VERSION >= 6)

```

```

    assert(0);
#else
    img->SetNumberOfScalarComponents(1);
#endif
    img->GetPointData()->SetScalars(d);

    vtkXMLImageDataWriter *writeb= vtkXMLImageDataWriter::New();
    #if (VTK_MAJOR_VERSION >= 6)
    writeb->SetInputData( img );
    #else
    writeb->SetInput( img );
    #endif
    writeb->SetFileName( outfilename );
    writeb->Write( );
/*
(300a,00f4) SQ                                     # u/1,1 Block Sequence
(fffe,e000) na (Item with undefined length)
(300a,00e1) SH [brass ]                             # 6,1 Material ID
(300a,00f8) CS [APERTURE]                           # 8,1 Block Type
(300a,00fa) CS [ABSENT]                             # 6,1 Block Divergence
(300a,00fb) CS [SOURCE_SIDE ]                       # 12,1 Block Mounting Position
(300a,00fc) IS [1 ]                                 # 2,1 Block Number
(300a,0100) DS [50.00 ]                             # 6,1 Block Thickness
(300a,0104) IS [179 ]                               # 4,1 Block Number of Points
(300a,0106) DS
[1.7\50.0\14.3\50.0\16.7\49.4\18.7\48.2\19.4\47.7\20.1\47.1\21.0\47.0\22.3\47.0\23.7\46.8\25.7\46.2\27.0\45.6\27.2\45.4\28.2
# 1934,2-2n Block Data
(fffe,e00d)
(fffe,e000) na (Item with undefined length)
(fffe,e00d)
(fffe,e0dd)
*/
gdcm::Tag tblocksq(0x300a,0x00f4);
if( !nestedds.FindDataElement( tblocksq ) )
{
    return 1;
}
const gdcm::DataElement &blocksq = nestedds.GetDataElement( tblocksq );
//std::cout << blocksq << std::endl;
gdcm::SmartPointer<gdcm::SequenceOfItems> sssqi = blocksq.GetValueAssSQ();
const gdcm::Item & item3 = sssqi->GetItem(1); // Item start at #1
const gdcm::DataSet& nestedds3 = item3.GetNestedDataSet();

gdcm::Tag tblockdata(0x300a,0x0106);
if( !nestedds3.FindDataElement( tblockdata ) )
{
    return 1;
}
const gdcm::DataElement &blockdata = nestedds3.GetDataElement( tblockdata );
// std::cout << blockdata << std::endl;
gdcm::Attribute<0x300a,0x0106> at_;
at_.SetFromDataElement( blockdata );

vtkDoubleArray *scalars = vtkDoubleArray::New();
scalars->SetNumberOfComponents(3);

gdcm::Attribute<0x300a,0x0104> bnpts; // IS [179 ] # 4,1 Block Number of Points
if( !nestedds3.FindDataElement( bnpts.GetTag() ) )
{
    return 1;
}
const gdcm::DataElement &blocknpts = nestedds3.GetDataElement( bnpts.GetTag() );
bnpts.SetFromDataElement( blocknpts );
std::cout << bnpts.GetValue() << std::endl;

vtkPolyData *output = vtkPolyData::New();
vtkPoints *newPts = vtkPoints::New();
vtkCellArray *polys = vtkCellArray::New();
const double *ptr = at_.GetValues();
//unsigned int npts = bnpts.GetNumberOfValues() / 2;
unsigned int npts = bnpts.GetValue();
vtkIdType *ptIds = new vtkIdType[npts];
for(unsigned int i = 0; i < npts; ++i)
{
    float x[3] = {};
    x[0] = (float)ptr[2*i+0];
    x[1] = (float)ptr[2*i+1];
    //x[2] = ptr[i+2];
    vtkIdType ptId = newPts->InsertNextPoint( x );
    //std::cout << x[0] << " " << x[1] << " " << x[2] << std::endl;
    ptIds[i] = ptId;
}

```

```

    }
    vtkIdType cellId = polys->InsertNextCell(npts , ptIds);
    (void)cellId;
    delete[] ptIds;

    output->SetPoints(newPts);
    newPts->Delete();
    output->SetPolys(polys);
    polys->Delete();
    //output->GetCellData()->SetScalars(scalars);
    //scalars->Delete();
#if (VTK_MAJOR_VERSION >= 6)
#else
    output->Update();
#endif
    output->Print( std::cout );

// }

    vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();

    vtkImageColorViewer *viewer = vtkImageColorViewer::New();
    if (VTK_MAJOR_VERSION >= 6)
        viewer->SetInputData(img);
    else
        viewer->SetInput(img);
    #endif
    viewer->SetupInteractor(iren);
    viewer->SetSize(600, 600);
    viewer->Render();

    vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
    //vtkPolyDataMapper2D* cubeMapper = vtkPolyDataMapper2D::New();
    if (VTK_MAJOR_VERSION >= 6)
        cubeMapper->SetInputData( output );
    else
        cubeMapper->SetInput( output );
    #endif
    cubeMapper->SetScalarRange(0,7);
    vtkActor *cubeActor = vtkActor::New();
    //vtkActor2D* cubeActor = vtkActor2D::New();
    cubeActor->SetMapper(cubeMapper);
    vtkProperty * property = cubeActor->GetProperty();
    property->SetRepresentationToWireframe();

    viewer->GetRenderer()->AddActor( cubeActor );

    iren->Initialize();
    iren->Start();

    return 0;
}

```

## 12.160 gdcmscene.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMPolyDataReader.h"
// #include "vtkGDCMPolyDataWriter.h"

#include "vtkAppendPolyData.h"
#include "vtkPolyDataWriter.h"

```



```

#include "vtkPolyDataMapper.h"
#include "vtkPolyDataMapper2D.h"
#include "vtkActor2D.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkRenderWindow.h"
#include "vtkRenderer.h"
#include "vtkCamera.h"
#include "vtkProperty.h"
#include "vtkProperty2D.h"
#include "vtkVersion.h"

// gdcmDataExtra/gdcmNonImageData/exRT_Structure_Set_Storage.dcm
// gdcmDataExtra/gdcmNonImageData/RTSTRUCT_1.3.6.1.4.1.22213.1.1396.2.dcm
// gdcmDataExtra/gdcmNonImageData/RT/RTStruct.dcm

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " filename1.dcm\n";
        return 1;
    }
    const char * filename = argv[1];

    vtkGDCMPolyDataReader * reader = vtkGDCMPolyDataReader::New();
    reader->SetFileName( filename );
    reader->Update();

    // vtkGDCMPolyDataWriter * writer2 = vtkGDCMPolyDataWriter::New();
    // for(int num = 0; num < reader->GetNumberOfOutputPorts(); ++num )
    //     writer2->SetInput( num, reader->GetOutput( num) );
    // writer2->SetFileName( "rtstruct.dcm" );
    // writer2->Write();

    // print reader output:
    reader->Print( std::cout );
    // print first output:
    reader->GetOutput()->Print( std::cout );

    vtkAppendPolyData *append = vtkAppendPolyData::New();
    int n = reader->GetNumberOfOutputPorts();
    for(int i = 0; i < n; ++i)
    {
#ifdef VTK_MAJOR_VERSION >= 6
        append->AddInputConnection( reader->GetOutputPort(i) );
#else
        append->AddInput( reader->GetOutput(i) );
#endif
    }

    vtkPolyDataWriter * writer = vtkPolyDataWriter::New();
#ifdef VTK_MAJOR_VERSION >= 6
    writer->SetInputConnection( reader->GetOutputPort() );
#else
    writer->SetInput( reader->GetOutput() );
#endif
    writer->SetFileName( "rtstruct.vtk" );
    //writer->Write();

    // Now we'll look at it.
    vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
    //vtkPolyDataMapper2D* cubeMapper = vtkPolyDataMapper2D::New();
    //cubeMapper->SetInput( reader->GetOutput() );
#ifdef VTK_MAJOR_VERSION >= 6
    cubeMapper->SetInputConnection( append->GetOutputPort() );
#else
    cubeMapper->SetInput( append->GetOutput() );
#endif
    cubeMapper->SetScalarRange(0,7);
    vtkActor *cubeActor = vtkActor::New();
    //vtkActor2D* cubeActor = vtkActor2D::New();
    cubeActor->SetMapper(cubeMapper);
    vtkProperty * property = cubeActor->GetProperty();
    property->SetRepresentationToWireframe();
    //cubeActor->GetProperty()->SetColor(1, 0, 0);

    // The usual rendering stuff.
    vtkCamera *camera = vtkCamera::New();
    camera->SetPosition(1,1,1);

```

```
//      camera->SetFocalPoint(0,0,0);

vtkRenderer *renderer = vtkRenderer::New();
vtkRenderWindow *renWin = vtkRenderWindow::New();
renWin->AddRenderer(renderer);

vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
iren->SetRenderWindow(renWin);

renderer->AddActor(cubeActor);
//renderer->AddActor2D(cubeActor);
//renderer->SetActiveCamera(camera);
renderer->ResetCamera();
renderer->SetBackground(1,1,1);

renWin->SetSize(300,300);

// interact with data
renWin->Render();
iren->Start();

reader->Delete();
append->Delete();
cubeMapper->Delete();
cubeActor->Delete();
// camera->Delete();
renderer->Delete();
renWin->Delete();
iren->Delete();

writer->Delete();

return 0;
}
```

## 12.161 gdcmttexture.cxx

```
/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"

#include "vtkRenderer.h"
#include "vtkAssembly.h"
#include "vtkRenderWindow.h"
#include "vtkAnnotatedCubeActor.h"
#include "vtkTransform.h"
#include "vtkAxesActor.h"
#include "vtkTextProperty.h"
#include "vtkCaptionActor2D.h"
#include "vtkPropAssembly.h"
#include "vtkOrientationMarkerWidget.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkImageData.h"
#include "vtkLookupTable.h"
#include "vtkTexture.h"
#include "vtkPlaneSource.h"
#include "vtkVersion.h"

int main( int argc, char *argv[] )
{
    if( argc < 2 ) return 1;
    vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
```

```

reader->SetFileName( argv[1] );

reader->Update();
vtkImageData* ima = reader->GetOutput();

vtkLookupTable* table = vtkLookupTable::New();
table->SetNumberOfColors(1000);
table->SetTableRange(0,1000);
table->SetSaturationRange(0,0);
table->SetHueRange(0,1);
table->SetValueRange(0,1);
table->SetAlphaRange(1,1);
table->Build();

// Texture
vtkTexture* texture = vtkTexture::New();
#if (VTK_MAJOR_VERSION >= 6)
    texture->SetInputData(ima);
#else
    texture->SetInput(ima);
#endif
texture->InterpolateOn();
texture->SetLookupTable(table);

// PlaneSource
vtkPlaneSource* plane = vtkPlaneSource::New();
plane->SetOrigin( -0.5, -0.5, 0.0);
plane->SetPoint1( 0.5, -0.5, 0.0);
plane->SetPoint2( -0.5, 0.5, 0.0);

// PolyDataMapper
vtkPolyDataMapper *planeMapper = vtkPolyDataMapper::New();
#if (VTK_MAJOR_VERSION >= 6)
    planeMapper->SetInputConnection(plane->GetOutputPort());
#else
    planeMapper->SetInput(plane->GetOutput());
#endif

// Actor
vtkActor* planeActor = vtkActor::New();
planeActor->SetTexture(texture);
planeActor->SetMapper(planeMapper);
planeActor->PickableOn();

// Final rendering with simple interactor:
vtkRenderer *ren = vtkRenderer::New();
vtkRenderWindow *renwin = vtkRenderWindow::New();
renwin->AddRenderer(ren);
vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
iren->SetRenderWindow(renwin);
ren->AddActor(planeActor);
ren->SetBackground(0,0,0.5);

vtkAnnotatedCubeActor* cube = vtkAnnotatedCubeActor::New();
cube->SetXPlusFaceText( "L" );
cube->SetXMinusFaceText( "R" );
cube->SetYPlusFaceText( "A" );
cube->SetYMinusFaceText( "P" );
cube->SetZPlusFaceText( "H" );
cube->SetZMinusFaceText( "F" );

vtkAxesActor* axes2 = vtkAxesActor::New();
// simulate a left-handed coordinate system
//
vtkTransform *transform = vtkTransform::New();
transform->Identity();
//transform->RotateY(180);
reader->GetDirectionCosines()->Print(std::cout);
transform->Concatenate(reader->GetDirectionCosines());
//axes2->SetShaftTypeToCylinder();
axes2->SetUserTransform( transform );
//cube->SetUserTransform( transform ); // can't get it to work
cube->GetAssembly()->SetUserTransform( transform ); // can't get it to work

vtkPropAssembly* assembly = vtkPropAssembly::New();
assembly->AddPart( axes2 );
assembly->AddPart( cube );

vtkOrientationMarkerWidget* widget = vtkOrientationMarkerWidget::New();
//widget->SetOutlineColor( 0.9300, 0.5700, 0.1300 );
widget->SetOrientationMarker( assembly );

```

```

widget->SetInteractor( iren );
//widget->SetViewport( 0.0, 0.0, 0.4, 0.4 );
widget->SetEnabled( 1 );
widget->InteractiveOff();
widget->InteractiveOn();

renwin->Render();
iren->Start();

// Clean up:
reader->Delete();
table->Delete();
texture->Delete();
plane->Delete();
planeMapper->Delete();
planeActor->Delete();
ren->Delete();
renwin->Delete();
iren->Delete();

return 0;
}

```

## 12.162 gdcmvolume.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
#include "vtkVersion.h"
#include "vtkGDCMImageReader.h"
#include "vtkPiecewiseFunction.h"
#include "vtkColorTransferFunction.h"
#include "vtkVolume.h"
#include "vtkVolumeProperty.h"
#if VTK_MAJOR_VERSION < 7
#include "vtkVolumeTextureMapper3D.h"
#endif
#include "vtkFixedPointVolumeRayCastMapper.h"
#include "vtkInteractorStyleTrackballCamera.h"
#include "vtkRenderer.h"
#include "vtkRenderWindow.h"
#include "vtkImageClip.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkVersion.h"

// gdcmvolume gdcmlData/GE_DLX-8-MONO2-Multiframe-Jpeg_Lossless.dcm
int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
    reader->SetFileName( argv[1] );
    reader->Update();

    // Create the renderers, render window, and interactor
    vtkRenderWindow *renWin = vtkRenderWindow::New();
    vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
    iren->SetRenderWindow(renWin);
    vtkRenderer *ren = vtkRenderer::New();
    renWin->AddRenderer(ren);

    // Create a transfer function mapping scalar value to opacity
    vtkPiecewiseFunction *oTFun = vtkPiecewiseFunction::New();
    //oTFun->AddSegment(0, 1.0, 256, 0.1);
    oTFun->AddSegment(0, 1.0, 240, 0.1);

    vtkColorTransferFunction *cTFun = vtkColorTransferFunction::New();

```

```

cTFun->AddRGBPoint( 0, 1.0, 1.0, 1.0 );
//cTFun->AddRGBPoint( 255, 1.0, 1.0, 1.0 );
cTFun->AddRGBPoint( 240, 1.0, 1.0, 1.0 );

// Need to crop to actually see minimum intensity
vtkImageClip *clip = vtkImageClip::New();
clip->SetInputConnection( reader->GetOutputPort() );
clip->SetOutputWholeExtent(0,66,0,66,30,37);
clip->ClipDataOn();

vtkVolumeProperty *property = vtkVolumeProperty::New();
property->SetScalarOpacity(oTFun);
property->SetColor(cTFun);
property->SetInterpolationTypeToLinear();

vtkFixedPointVolumeRayCastMapper *mapper = vtkFixedPointVolumeRayCastMapper::New();
mapper->SetBlendModeToMinimumIntensity();
mapper->SetInputConnection( reader->GetOutputPort() );

vtkVolume *volume = vtkVolume::New();
volume->SetMapper(mapper);
volume->SetProperty(property);

ren->AddViewProp(volume);

renWin->Render();
{
    iren->Start();
}

volume->Delete();
mapper->Delete();
property->Delete();
clip->Delete();
cTFun->Delete();
oTFun->Delete();
reader->Delete();
renWin->Delete();
iren->Delete();
ren->Delete();

return 0;
}

```

## 12.163 offscreenimage.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"
#include "vtkRenderWindow.h"
#include "vtkRenderer.h"
#include "vtkImageMapToWindowLevelColors.h"
#include "vtkImageActor.h"
#include "vtkPNGWriter.h"
#include "vtkWindowToImageFilter.h"
#include "vtkMedicalImageProperties.h"
#include "vtkVersion.h"

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        return 1;
    }
}

```

```

const char *filename = argv[1];

vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
reader->SetFileName( filename );
reader->Update(); // important to read the window/level info

vtkMedicalImageProperties *prop = reader->GetMedicalImageProperties();

vtkRenderWindow *renWin = vtkRenderWindow::New();
renWin->OffScreenRenderingOn();

vtkRenderer *renderer = vtkRenderer::New();
renWin->AddRenderer(renderer);

vtkImageMapToWindowLevelColors *windowlevel = vtkImageMapToWindowLevelColors::New();
#if (VTK_MAJOR_VERSION >= 6)
    windowlevel->SetInputConnection( reader->GetOutputPort() );
#else
    windowlevel->SetInput( reader->GetOutput() );
#endif
unsigned int n = prop->GetNumberOfWindowLevelPresets();
if( n )
{
    // Take the first one by default:
    const double *wl = prop->GetNthWindowLevelPreset(0);
    windowlevel->SetWindow( wl[0] );
    windowlevel->SetLevel( wl[1] );
}

vtkImageActor *actor = vtkImageActor::New();
#if (VTK_MAJOR_VERSION >= 6)
    actor->SetInputData( windowlevel->GetOutput() );
#else
    actor->SetInput( windowlevel->GetOutput() );
#endif

renderer->AddActor( actor );

renWin->Render();

vtkWindowToImageFilter *w2if = vtkWindowToImageFilter::New();
w2if->SetInput( renWin );

vtkPNGWriter *wr = vtkPNGWriter::New();
#if (VTK_MAJOR_VERSION >= 6)
    wr->SetInputConnection( w2if->GetOutputPort() );
#else
    wr->SetInput( w2if->GetOutput() );
#endif
wr->SetFileName( "offscreenimage.png" );
wr->Write();

reader->Delete();
renWin->Delete();
renderer->Delete();
windowlevel->Delete();
actor->Delete();
w2if->Delete();
wr->Delete();

return 0;
}

```

## 12.164 reslicesphere.cxx

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====

```

```

=====*/
//
// Load a DICOM series.
// Position a sphere within the volume.
// Allow the user to change between Axial, Sagittal, Coronal, and
// Oblique view of the images and move through the slices.
// The display should show the resliced image and the cross section
// of the sphere intersecting that plane.
//

/*
from Scott Johnson /Scott Johnson neuwave com/
to VTK /vtkusers vtk.org/
date Tue, May 11, 2010 at 7:01 PM
*/
#include <string>

#include <vtkDICOMImageReader.h>
#include <vtkStringArray.h>
#include <vtkDirectory.h>
#include <vtkImageThreshold.h>
#include <vtkImageShiftScale.h>
#include <vtkImageReslice.h>
#include <vtkRenderWindowInteractor.h>
#include <vtkImageViewer2.h>
#include <vtkSphereSource.h>
#include <vtkPolyDataMapper.h>
#include <vtkPlane.h>
#include <vtkCutter.h>
#include <vtkActor.h>
#include <vtkCommand.h>
#include <vtkSmartPointer.h>
#include <vtkMatrix4x4.h>
#include <vtkInteractorObserver.h>
#include <vtkProperty.h>
#include <vtkRenderer.h>
#include <vtkImageData.h>
#include <vtkImageActor.h>
#include "vtkTransformPolyDataFilter.h"
#include <vtkCamera.h>
#include <vtkMath.h>
#include <vtkTransform.h>
#include <vtkTextActor.h>
#include <vtkActor2D.h>
#include <vtkPolyDataMapper2D.h>
#include <vtkProperty2D.h>
#include <vtkGDCMImageReader.h>
#include <vtkImageChangeInformation.h>
#include <vtkVersion.h>

#include "gdcmDirectory.h"
#include "gdcmTesting.h"
#include "gdcmIPPSorter.h"

// Change to match the path to find Raw_0.vti or provide
// the parameter when starting ResliceSphere.

const double sphereCenter[3]={74, 219, 70};

// Angles (0, 0, 0)
const double AxialMatrix[] = { 1.0, 0.0, 0.0, 0.0,
                               0.0, 1.0, 0.0, 0.0,
                               0.0, 0.0, 1.0, 0.0,
                               0.0, 0.0, 0.0, 1.0 };

// Angles (0, 90, 0)
const double SagittalMatrix[] = { 0.0, 0.0, 1.0, 0.0,
                                   0.0, 1.0, 0.0, 0.0,
                                   -1.0, 0.0, 0.0, 0.0,
                                   0.0, 0.0, 0.0, 1.0 };

// Angles (-90, 0, 0)
const double CoronalMatrix[] = { 1.0, 0.0, 0.0, 0.0,
                                  0.0, 0.0, 1.0, 0.0,
                                  0.0, -1.0, 0.0, 0.0,
                                  0.0, 0.0, 0.0, 1.0 };

// Angles (0, 90, 31)
const double ObliqueMatrix[] = { 0.0, -0.515038, 0.857167, 0.0,
                                  0.0, 0.857167, 0.515038, 0.0,
                                  -1.0, 0.0, 0.0, 0.0,

```

```

0.0, 0.0, 0.0, 1.0 };

class ResliceRender;

// Class to handle key press events.
class KeyCallback : public vtkCommand
{
public:
    static KeyCallback* New()
    {
        return new KeyCallback();
    }

    void Execute(vtkObject* caller, unsigned long eventId, void *calldata);
    void SetCallbackData(ResliceRender* reslice);

protected:
    ResliceRender* _reslice;
};

class ResliceRender
{
public:
    typedef enum _ORIENTATION
    {
        AXIAL = 0,
        SAGITTAL = 1,
        CORONAL = 2,
        OBLIQUE = 3
    } ORIENTATION;

    ResliceRender()
    {
        _orientation=AXIAL;
    }

    ~ResliceRender()
    {
        _transform->Delete();
        _reader->Delete();
        _reslice->Delete();
        _interactor->Delete();
        _imageViewer->Delete();

        _sphere->Delete();
        _sphereMapper->Delete();
        _sphereActor->Delete();

        _plane->Delete();
        _cutter->Delete();
        _polyTransform->Delete();
        _ROIMapper->Delete();
        _ROIActor->Delete();

        _annotation->Delete();
    }

    void CreatePipeline(const char* fileName)
    {
        vtkProperty2D* props;

        //_reader=vtkXMLImageDataReader::New();
        //_reader->SetFileName(fileName);
        //_reader->Update();

        //_reader=qzDICOMImageReader::New();
        _reader=vtkGDCMImageReader::New();

        //vtkDirectory *d = vtkDirectory::New();
        //d->Open(fileName);
        //d->Print( std::cout );
        gdcm::Directory d;
        d.Load(fileName);
        gdcm::Directory::FileNamesType const &files = d.GetFileNames();

        gdcm::IPPSorter s;
        s.SetComputeZSpacing( true );
        s.SetZSpacingTolerance( 1e-3 );
        bool b = s.Sort( files );
        if( !b )
        {

```



```

        std::cerr << "Failed to sort:" << fileName << std::endl;
        //return ;
    }
    //std::cout << "Sorting succeeded:" << std::endl;
    //s.Print( std::cout );

    //std::cout << "Found z-spacing:" << std::endl;
    //std::cout << s.GetZSpacing() << std::endl;
    double ippzspacing = s.GetZSpacing();

    const std::vector<std::string> & sorted = s.GetFilesNames();
    vtkStringArray *vtkfiles = vtkStringArray::New();
    std::vector< std::string >::const_iterator it = sorted.begin();
    for( ; it != sorted.end(); ++it)
    {
        const std::string &f = *it;
        vtkfiles->InsertNextValue( f.c_str() );
    }

    //_reader->SetDirectoryName(fileName);
    //_reader->SetFileNames( d->GetFiles() );
    _reader->SetFileNames( vtkfiles );
    _reader->Update();

#ifdef vtkFloatingPointType
#define vtkFloatingPointType double
#endif
    const vtkFloatingPointType *spacing = _reader->GetOutput()->GetSpacing();

    vtkImageChangeInformation *v16 = vtkImageChangeInformation::New();
    #if (VTK_MAJOR_VERSION >= 6)
        v16->SetInputConnection( _reader->GetOutputPort() );
    #else
        v16->SetInput( _reader->GetOutput() );
    #endif
    v16->SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
    v16->Update();

    _threshold=vtkImageThreshold::New();
    _threshold->ThresholdByUpper(-3024.0);
    _threshold->ReplaceOutOn();
    _threshold->SetOutValue(0.0);
    _threshold->SetInputConnection(v16->GetOutputPort());

    _shift=vtkImageShiftScale::New();
    _shift->SetShift(0);
    _shift->SetScale(1);
    _shift->SetInputConnection(_threshold->GetOutputPort());

    // Initialize the reslice with an axial orientation.
    vtkSmartPointer<vtkMatrix4x4> matrix =
        vtkSmartPointer<vtkMatrix4x4>::New();
    matrix->Identity();

    _transform = vtkTransform::New();
    _transform->SetMatrix(matrix);

    _reslice = vtkImageReslice::New();
    _reslice->SetOutputDimensionality(3);

    // PROBLEM:
    // The original intent was to connect the same transform
    // to the vtkImageReslice and vtkTransformPolyDataFilter,
    // but the resulting reslices appear different using the
    // vtkTransform as opposed to explicitly setting the
    // reslice axes via SetResliceAxes. Also, if the vtkTransform
    // is connected and orientated other than axial, the extents
    // don't seem to update resulting in VTK believing the slice
    // is out of range.

    //_reslice->SetResliceTransform(_transform);
    _reslice->SetResliceAxes(matrix);
    //_reslice->SetInputConnection(_reader->GetOutputPort());
    _reslice->SetInputConnection(_shift->GetOutputPort());

    // Create the sphere target shape.
    _sphere=vtkSphereSource::New();
    _sphere->SetRadius(7.0);
    _sphere->SetThetaResolution(16);
    _sphere->SetPhiResolution(16);

```

```

_sphere->SetCenter(sphereCenter[0], sphereCenter[1], sphereCenter[2]);

_sphereMapper=vtkPolyDataMapper::New();
_sphereMapper->SetInputConnection(_sphere->GetOutputPort());

_sphereActor=vtkActor::New();
_sphereActor->SetMapper(_sphereMapper);
_sphereActor->PickableOff();
_sphereActor->GetProperty()->SetColor(1.0, 0.0, 0.0);
_sphereActor->GetProperty()->SetEdgeColor(1.0, 0.0, 0.0);
_sphereActor->GetProperty()->SetDiffuseColor(1.0, 0.0, 0.0);
_sphereActor->SetVisibility(true);

// Create the cutting pipeline.
// This plane will be positioned in the original image coordinate system.
_plane = vtkPlane::New();
_plane->SetNormal(0.0, 0.0, 1.0);

_cutter = vtkCutter::New();
_cutter->SetInputConnection(_sphere->GetOutputPort());
_cutter->SetCutFunction(_plane);
_cutter->GenerateCutScalarsOn();
_cutter->SetValue(0, 0.5);

// The transform attached to _polyTransform should move the cut
// ROI into the resliced coordinate system, which should be the
// same as the coordinate system of the resliced images.
// PROBLEM: It doesn't.
_polyTransform = vtkTransformPolyDataFilter::New();
_polyTransform->SetTransform(_transform);
_polyTransform->SetInputConnection(_cutter->GetOutputPort());

_ROIMapper = vtkPolyDataMapper2D::New();
_ROIMapper->SetInputConnection(_polyTransform->GetOutputPort());

vtkCoordinate* coordinate = vtkCoordinate::New();
coordinate->SetCoordinateSystemToWorld();
_ROIMapper->SetTransformCoordinate(coordinate);

_ROIActor = vtkActor2D::New();
_ROIActor->SetMapper(_ROIMapper);

// Make sure the cut can be seen, especially the edges.
props=_ROIActor->GetProperty();
props->SetLineWidth(2);
props->SetOpacity(1.0);
// props->EdgeVisibilityOn();
// props->SetDiffuse(0.8);
// props->SetSpecular(0.3);
// props->SetSpecularPower(20);
// props->SetRepresentationToSurface();
// props->SetDiffuseColor(1.0, 0.0, 0.0);
// props->SetEdgeColor(1.0, 0.0, 0.0);
props->SetColor(1.0, 0.0, 0.0);

_interactor = vtkRenderWindowInteractor::New();

// Create the image viewer and add the actor with the cut ROI.
_imageViewer = vtkImageViewer2::New();
_imageViewer->SetupInteractor(_interactor);
_imageViewer->SetSize(400, 400);
_imageViewer->SetColorWindow(1024);
_imageViewer->SetColorLevel(800);
_imageViewer->SetInputConnection(_reslice->GetOutputPort());
_imageViewer->GetImageActor()->SetOpacity(0.5);

_annotation = vtkTextActor::New();
_annotation->SetTextScaleModeToViewport();
_imageViewer->GetRenderer()->AddActor(_annotation);

// Add the cut shape actor to the renderer.
_imageViewer->GetRenderer()->AddActor(_ROIActor);

// Set up the key handler.
vtkSmartPointer<KeyCallback> callback = vtkSmartPointer<KeyCallback>::New();
callback->SetCallbackData(this);
_interactor->AddObserver(vtkCommand::KeyPressEvent, callback);

_interactor->Initialize();

```

```

}

void Start()
{
    _interactor->Start();
}

void ResetOrientation()
{
    vtkSmartPointer<vtkMatrix4x4> matrix =
        vtkSmartPointer<vtkMatrix4x4>::New();
    matrix->Identity();

    SetOrientation(matrix);
}

// Make sure the orientation of the vtkImageReslice and
// vtkTransform are in sync.
void SetOrientation(vtkMatrix4x4* matrix)
{
    _reslice->SetResliceAxes(matrix);
    _reslice->Update();

    vtkMatrix4x4* inverse = vtkMatrix4x4::New();
    vtkMatrix4x4::Invert(matrix, inverse);

    _transform->SetMatrix(inverse);
    _transform->Update();
}

// Set the current slice of the current view.
void SetSlice(int slice)
{
    std::stringstream posString;

    double    center[3];
    double    spacing[3];
    double    origin[3];
    double    point[4];
    double    newPoint[4];

    vtkImageData* imageData;
    int newSlice;

    // Try to make sure the extents of the reslice are updated.
    // PROBLEM: It doesn't seem to work when changing the orientation.
    imageData=vtkImageData::SafeDownCast(_reslice->GetOutput());
    #if (VTK_MAJOR_VERSION >= 6)
        assert(0);
    #else
        imageData->UpdateInformation();
    #endif

    // Let vtkImageViewer2 handle the slice limits.
    _imageView->SetSlice(slice);
    newSlice=GetSlice();

    imageData->GetCenter(center);
    imageData->GetSpacing(spacing);
    imageData->GetOrigin(origin);

    // Compute the position of the center of the slice based on the
    // spacing of the slices. The resliced axis will always
    // be the "Z" axis.
    point[0]=center[0];
    point[1]=center[1];
    point[2]=(newSlice * spacing[2]) + origin[2];
    point[3]=1.0;

    // Convert the coordinate from the reslice coordinate system to the
    // original image coordinate system.
    // PROBLEM: Logically this seems like it should have been multiplied
    // by the inverse to translate from the resliced coordinate system to
    // the original coordinate system. However, multiplying by the inverse
    // sticks the plane in the wrong place completely. Using the original
    // matrix at least gets the Z coordinate right.
    vtkMatrix4x4* matrix=_reslice->GetResliceAxes();
    vtkSmartPointer<vtkMatrix4x4> inverse =
        vtkSmartPointer<vtkMatrix4x4>::New();

```

```

        vtkMatrix4x4::Invert(matrix, inverse);

        matrix->MultiplyPoint(point, newPoint);
        _plane->SetOrigin(newPoint[0], newPoint[1], newPoint[2]);

        // Annotate the image.
        posString << "Position: (" << newPoint[0] << ", " << newPoint[1]
            << ", " << newPoint[2] << ") Slice: " << newSlice;
        _annotation->SetInput(posString.str().c_str());

        _imageView->Render();
    }

    int GetSlice()
    {
        return _imageView->GetSlice();
    }

    // Set the orientation of the view.
    void SetOrientation(ResliceRender::ORIENTATION orientation)
    {
        vtkCamera* camera=_imageView->GetRenderer()->GetActiveCamera();

        double spacing[3];
        double origin[3];
        double point[4];
        double newPoint[4];
        double initialPosition;
        double xDirCosine[3];
        double yDirCosine[3];
        double zDirCosine[3];
        double normal[3];

        vtkImageData* imageData;

        vtkSmartPointer<vtkMatrix4x4> matrix =
            vtkSmartPointer<vtkMatrix4x4>::New();

        _orientation=orientation;

        // Reset ViewUp
        camera->SetViewUp(0.0, 1.0, 0.0);

        // Compute the cut plane position to the input coordinate system.
        imageData=vtkImageData::SafeDownCast(_reslice->GetInput());
    #if (VTK_MAJOR_VERSION >= 6)
        assert(0);
    #else
        imageData->UpdateInformation();
    #endif
        imageData->GetSpacing(spacing);
        imageData->GetOrigin(origin);

        point[0]=origin[0];
        point[1]=origin[1];
        point[2]=origin[2];
        point[3]=1.0;

        switch (_orientation)
        {
        case AXIAL:
            matrix->DeepCopy(AxialMatrix);
            initialPosition=sphereCenter[2];
            break;

        case CORONAL:
            matrix->DeepCopy(CoronalMatrix);
            initialPosition=sphereCenter[1];
            break;

        case SAGITTAL:
            matrix->DeepCopy(SagittalMatrix);
            initialPosition=sphereCenter[0];
            break;

        case OBLIQUE:
            matrix->DeepCopy(ObliqueMatrix);
            initialPosition=sphereCenter[2];
            break;
        }
    }

```

```

    }

    // Move the origin from the original image coordinate system to the
    // resliced image coordinate system.
    matrix->MultiplyPoint(point, newPoint);
    matrix->SetElement(0, 3, newPoint[0]);
    matrix->SetElement(1, 3, newPoint[1]);
    matrix->SetElement(2, 3, newPoint[2]);

    ResetOrientation();
    SetOrientation(matrix);

    // Compute the cutting plane normal and set it.
    // PROBLEM: If the transformation is connected rather than
    // using SetResliceAxes, the Direction Cosines do not reflect
    // the orientation of the vtkImageReslice.
    _reslice->GetResliceAxesDirectionCosines(xDirCosine, yDirCosine,
                                             zDirCosine);
    vtkMath::Cross(xDirCosine, yDirCosine, normal);
    _plane->SetNormal(normal);

    // Set the extents and spacing of the reslice to account for
    // all of the data.
    _reslice->SetOutputExtentToDefault();
    _reslice->SetOutputSpacing(spacing[0], spacing[0], spacing[0]);

    // Force the vtkImageViewer2 to update.
    // PROBLEM: The whole extent does not seem to be set in time
    // for the first render. This results in an error because the
    // slice is positioned outside the old bounds.
#if (VTK_MAJOR_VERSION >= 6)
    _imageView->SetInputData(NULL);
#else
    _imageView->SetInput(NULL);
#endif
    _imageView->SetInputConnection(_reslice->GetOutputPort());

    _imageView->GetRenderer()->ResetCameraClippingRange();
    _imageView->GetRenderer()->ResetCamera();

    // Set the initial slice to be at the center of the sphere.
    // Divide by the spacing because this will be undone in SetSlice.
    SetSlice( (int)(initialPosition / spacing[0]));
}

vtkRenderWindowInteractor* GetInteractor()
{
    return _interactor;
}

protected:
    ORIENTATION          _orientation;

    //qzDICOMImageReader*    _reader;
    vtkGDCMImageReader*    _reader;
    vtkImageThreshold*      _threshold;
    vtkImageShiftScale*     _shift;
    vtkImageReslice*        _reslice;
    vtkRenderWindowInteractor* _interactor;
    vtkImageViewer2*        _imageView;

    vtkSphereSource*        _sphere;
    vtkPolyDataMapper*      _sphereMapper;
    vtkActor*               _sphereActor;

    vtkPlane*               _plane;
    vtkCutter*              _cutter;
    vtkTransform*           _transform;
    vtkTransformPolyDataFilter* _polyTransform;
    vtkPolyDataMapper2D*     _ROIMapper;
    vtkActor2D*             _ROIActor;

    vtkTextActor*           _annotation;
};

// Catch KeyPress events.
// Up Arrow - increases the slice
// Down Arrow - decreases the slice
// 'A' - sets the view to Axial
// 'S' - sets the view to Sagittal

```

```

// 'C'      - sets the view to Coronal
// 'O'      - set the view to Oblique

void KeyCallback::Execute(vtkObject* caller, unsigned long eventId, void *calldata)
{
    (void)caller;
    (void)eventId;
    (void)calldata;
    std::string sym=_reslice->GetInteractor()->GetKeySym();

    if (!sym.compare("Up"))
    {
        _reslice->SetSlice(_reslice->GetSlice() + 1);
    }
    else if (!sym.compare("Down"))
    {
        _reslice->SetSlice(_reslice->GetSlice() - 1);
    }
    else if ((!sym.compare("A")) || (!sym.compare("a")))
    {
        _reslice->SetOrientation(ResliceRender::AXIAL);
    }
    else if ((!sym.compare("C")) || (!sym.compare("c")))
    {
        _reslice->SetOrientation(ResliceRender::CORONAL);
    }
    else if ((!sym.compare("S")) || (!sym.compare("s")))
    {
        _reslice->SetOrientation(ResliceRender::SAGITTAL);
    }
    else if ((!sym.compare("O")) || (!sym.compare("o")))
    {
        _reslice->SetOrientation(ResliceRender::OBLIQUE);
    }
}

void KeyCallback::SetCallbackData(ResliceRender* reslice)
{
    _reslice=reslice;
}

// Usage: ResliceSphere [fileName]
int main(int argc, char *argv[])
{
    ResliceRender render;

    if (argc == 1)
    {
        const char *root = gdcm::Testing::GetDataExtraRoot();
        std::string dir3 = root;
        dir3 += "/gdcmSampleData/ForSeriesTesting/Dentist/images/";
        render.CreatePipeline(dir3.c_str());
    }
    else
    {
        render.CreatePipeline(argv[1]);
    }

    render.SetOrientation(ResliceRender::AXIAL);
    render.Start();

    return EXIT_SUCCESS;
}

```

## 12.165 rtstructapp.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====

```

```

===== */
#include "vtkGDCMPolyDataReader.h"
#include "vtkGDCMPolyDataWriter.h"

#include "vtkPolyDataWriter.h"
#include "vtkPolyDataMapper.h"
#include "vtkPolyDataMapper2D.h"
#include "vtkActor2D.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkMedicalImageProperties.h"
#include "vtkRenderWindow.h"
#include "vtkRenderer.h"
#include "vtkCamera.h"
#include "vtkProperty.h"
#include "vtkProperty2D.h"
#include "vtkAppendPolyData.h"
#include "vtkImageData.h"
#include "vtkVersion.h"

/*
 * Small example to read in a RTSTRUCT and write it out (displays it too).
 */

// gdcmDataExtra/gdcmNonImageData/exRT_Structure_Set_Storage.dcm
// gdcmDataExtra/gdcmNonImageData/RTSTRUCT_1.3.6.1.4.1.22213.1.1396.2.dcm
// gdcmDataExtra/gdcmNonImageData/RT/RTStruct.dcm

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm\n";
        return 1;
    }
    const char * filename = argv[1];
    const char * outfilename = argv[2];
    vtkGDCMPolyDataReader * reader = vtkGDCMPolyDataReader::New();
    reader->SetFileName( filename );
    reader->Update();

    //std::cout << reader->GetMedicalImageProperties()->GetStudyDate() << std::endl;

    vtkGDCMPolyDataWriter * writer = vtkGDCMPolyDataWriter::New();
    writer->SetNumberOfInputPorts( reader->GetNumberOfOutputPorts() );
    writer->SetFileName( outfilename );
    for(int num = 0; num < reader->GetNumberOfOutputPorts(); ++num )
    #if (VTK_MAJOR_VERSION >= 6)
        writer->SetInputConnection( num, reader->GetOutputPort( num) );
    #else
        writer->SetInput( num, reader->GetOutput( num) );
    #endif

    //doesn't look like the medical properties are actually written out
    writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
    writer->SetRTStructSetProperties( reader->GetRTStructSetProperties() );
    writer->Write();

    // print reader output:
    reader->Print( std::cout );
    // print first output:
    reader->GetOutput()->Print( std::cout );

    vtkAppendPolyData *append = vtkAppendPolyData::New();

    int n = reader->GetNumberOfOutputPorts();
    for(int i = 0; i < n; ++i)
    {
    #if (VTK_MAJOR_VERSION >= 6)
        append->AddInputConnection( reader->GetOutputPort(i) );
    #else
        append->AddInput( reader->GetOutput(i) );
    #endif
    }

    // Now we'll look at it.
    vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
    #if (VTK_MAJOR_VERSION >= 6)
        cubeMapper->SetInputConnection( append->GetOutputPort() );
    #else
        cubeMapper->SetInput( append->GetOutput() );
    #endif
}

```

```

cubeMapper->SetScalarRange(0,7);
vtkActor *cubeActor = vtkActor::New();
cubeActor->SetMapper(cubeMapper);
vtkProperty * property = cubeActor->GetProperty();
property->SetRepresentationToWireframe();

vtkRenderer *renderer = vtkRenderer::New();
vtkRenderWindow *renWin = vtkRenderWindow::New();
renWin->AddRenderer(renderer);

vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
iren->SetRenderWindow(renWin);

renderer->AddActor(cubeActor);
renderer->ResetCamera();
renderer->SetBackground(1,1,1);

renWin->SetSize(300,300);

renWin->Render();
iren->Start();

reader->Delete();
append->Delete();
cubeMapper->Delete();
cubeActor->Delete();
renderer->Delete();
renWin->Delete();
iren->Delete();
writer->Delete();

return 0;
}

```

## 12.166 threadgdcm.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmDirectory.h"
#include "gdcmSystem.h"

#include "vtkImageData.h"
#include "vtkStructuredPointsWriter.h"
#include "vtkVersion.h"

#include <pthread.h>

struct threadparams
{
    const char **filenames;
    size_t nfiles;
    char *scalarpointer;
// TODO I should also pass in the dim of the reference image just in case
};

void *ReadFilesThread(void *voidparams)
{
    const threadparams *params = static_cast<const threadparams *> (voidparams);

    const size_t nfiles = params->nfiles;
    for(unsigned int file = 0; file < nfiles; ++file)
    {
        /*
        // TODO: update progress

```



```

pthread_mutex_lock(&params->lock);
//section critique
ReadingProgress+=params->stepProgress;
pthread_mutex_unlock(&params->lock);
*/
const char *filename = params->filenames[file];
//std::cerr << filename << std::endl;

gdcm::ImageReader reader;
reader.SetFileName( filename );
try
{
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        break;
    }
}
catch( ... )
{
    std::cerr << "Failed to read: " << filename << std::endl;
    break;
}

const gdcm::Image &image = reader.GetImage();
unsigned long len = image.GetBufferLength();
char * pointer = params->scalarpointer;
#if 0
char *tempimage = new char[len];
image.GetBuffer(tempimage);

memcpy(pointer + file*len, tempimage, len);
delete[] tempimage;
#else
char *tempimage = pointer + file * len;
image.GetBuffer(tempimage);
#endif
}

return voidparams;
}

void ShowFilenames(const threadparams &params)
{
    std::cout << "start" << std::endl;
    for(unsigned int i = 0; i < params.nfiles; ++i)
    {
        const char *filename = params.filenames[i];
        std::cout << filename << std::endl;
    }
    std::cout << "end" << std::endl;
}

void ReadFiles(size_t nfiles, const char *filenames[])
{
    // \precondition: nfiles > 0
    assert( nfiles > 0 );
    const char *reference= filenames[0]; // take the first image as reference

    gdcm::ImageReader reader;
    reader.SetFileName( reference );
    if( !reader.Read() )
    {
        // That would be very bad...
        assert(0);
    }

    const gdcm::Image &image = reader.GetImage();
    gdcm::PixelFormat pixeltype = image.GetPixelFormat();
    unsigned long len = image.GetBufferLength();
    const unsigned int *dims = image.GetDimensions();
    unsigned short pixelsize = pixeltype.GetPixelSize();
    (void)pixelsize;
    assert( image.GetNumberOfDimensions() == 2 );

    vtkImageData *output = vtkImageData::New();
    output->SetDimensions(dims[0], dims[1], (int)nfiles);

    #if (VTK_MAJOR_VERSION >= 6)
    int numscal = pixeltype.GetSamplesPerPixel();
    switch( pixeltype )

```

```

    {
    case gdcm::PixelFormat::INT8:
        output->AllocateScalars( VTK_SIGNED_CHAR, numscal );
        break;
    case gdcm::PixelFormat::UINT8:
        output->AllocateScalars( VTK_UNSIGNED_CHAR, numscal );
        break;
    case gdcm::PixelFormat::INT16:
        output->AllocateScalars( VTK_SHORT, numscal );
        break;
    case gdcm::PixelFormat::UINT16:
        output->AllocateScalars( VTK_UNSIGNED_SHORT, numscal );
        break;
    case gdcm::PixelFormat::INT32:
        output->AllocateScalars( VTK_INT, numscal );
        break;
    case gdcm::PixelFormat::UINT32:
        output->AllocateScalars( VTK_UNSIGNED_INT, numscal );
        break;
    default:
        assert(0);
    }
#else
    switch( pixeltype )
    {
    case gdcm::PixelFormat::INT8:
    #if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
        output->SetScalarType ( VTK_SIGNED_CHAR );
    #else
        output->SetScalarType ( VTK_CHAR );
    #endif
        break;
    case gdcm::PixelFormat::UINT8:
        output->SetScalarType ( VTK_UNSIGNED_CHAR );
        break;
    case gdcm::PixelFormat::INT16:
        output->SetScalarType ( VTK_SHORT );
        break;
    case gdcm::PixelFormat::UINT16:
        output->SetScalarType ( VTK_UNSIGNED_SHORT );
        break;
    case gdcm::PixelFormat::INT32:
        output->SetScalarType ( VTK_INT );
        break;
    case gdcm::PixelFormat::UINT32:
        output->SetScalarType ( VTK_UNSIGNED_INT );
        break;
    default:
        assert(0);
    }
    output->SetNumberOfScalarComponents ( pixeltype.GetSamplesPerPixel() );
    output->AllocateScalars();
#endif
    char * scalarpointer = static_cast<char*>(output->GetScalarPointer());

    const unsigned int nthreads = 4;
    threadparams params[nthreads];

    //pthread_mutex_t lock;
    //pthread_mutex_init(&lock, NULL);

    pthread_t *pthread = new pthread_t[nthreads];

    // There is nfiles, and nThreads
    assert( nfiles > nthreads );
    const size_t partition = nfiles / nthreads;
    for (unsigned int thread=0; thread < nthreads; ++thread)
    {
        params[thread].filenames = filenames + thread * partition;
        params[thread].nfiles = partition;
        if( thread == nthreads - 1 )
        {
            // There is slightly more files to process in this thread:
            params[thread].nfiles += nfiles % nthreads;
        }
        assert( thread * partition < nfiles );
        params[thread].scalarpointer = scalarpointer + thread * partition * len;
        //assert( params[thread].scalarpointer < scalarpointer + 2 * dims[0] * dims[1] * dims[2] );
        // start thread:
        int res = pthread_create( &pthread[thread], NULL, ReadFilesThread, &params[thread]);
        if( res )

```

```

        {
            std::cerr << "Unable to start a new thread, pthread returned: " << res << std::endl;
            assert(0);
        }
        //ShowFileNames(params[thread]);
    }
// DEBUG
    size_t total = 0;
    for (unsigned int thread=0; thread < nthreads; ++thread)
    {
        total += params[thread].nfiles;
    }
    assert( total == nfiles );
// END DEBUG

    for (unsigned int thread=0;thread<nthreads;thread++)
    {
        pthread_join( pthread[thread], NULL);
    }
    delete[] pthread;

    //pthread_mutex_destroy(&lock);

    // For some reason writing down the file is painfully slow...
    vtkStructuredPointsWriter *writer = vtkStructuredPointsWriter::New();
    #if (VTK_MAJOR_VERSION >= 6)
        writer->SetInputData( output );
    #else
        writer->SetInput( output );
    #endif
    writer->SetFileName( "/tmp/threadgdcmm.vtk" );
    writer->SetFileTypeToBinary();
    //writer->Write();
    writer->Delete();

    //output->Print( std::cout );
    output->Delete();
}

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " [directory|list of filenames]\n";
        return 1;
    }

    // Check if user pass in a single directory
    if( argc == 2 && gdcm::System::FileIsDirectory( argv[1] ) )
    {
        gdcm::Directory d;
        d.Load( argv[1] );
        gdcm::Directory::FileNamesType l = d.GetFilesNames();
        const size_t nfiles = l.size();
        const char **filenames = new const char* [ nfiles ];
        for(unsigned int i = 0; i < nfiles; ++i)
        {
            filenames[i] = l[i].c_str();
        }
        ReadFiles(nfiles, filenames);
        delete[] filenames;
    }
    else
    {
        // Simply copy all filenames into the vector:
        const char **filenames = const_cast<const char**>(argv+1);
        const size_t nfiles = argc - 1;
        ReadFiles(nfiles, filenames);
    }

    return 0;
}

```

## 12.167 AWTMedical3.java

```

/*=====

```

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre

All rights reserved.

See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even  
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR  
PURPOSE. See the above copyright notice for more information.

```
=====*/
package examples;

import vtk.*;
//import gdcm.*;

import vtk.util.VtkPanelContainer;
import vtk.util.VtkPanelUtil;
import vtk.util.VtkUtil;

import java.util.ArrayList;

import javax.swing.*;
import java.awt.*;
import java.io.File;

public class AWTMedical3 extends JComponent implements VtkPanelContainer {

    private vtkPanel renWin;

    vtkImageData ReadDataFile(File inSelectedFile){

        vtkImageData outImageData = null;
        Directory theDir = new Directory();

        String theInputDirectory = inSelectedFile.getPath();
        theDir.Load(theInputDirectory);

        Scanner theScanner = new Scanner();
        Tag theStudyTag = new Tag(0x0020,0x000d);
        Tag theSeriesTag = new Tag(0x0020,0x000e);
        theScanner.AddTag(theStudyTag); //get studies,
        theScanner.AddTag(theSeriesTag); //get studies,
        theScanner.Scan(theDir.GetFilenames());

        FilenamesType theStudyValues = theScanner.GetOrderedValues(theStudyTag);
        long theNumStudies = theStudyValues.size();
        //for now, take the first study, and nothing else.
        //and the return is actually not FilenamesType, just a
        //vector of strings
        if (theNumStudies != 1)
            return outImageData;
        String theStudyVal = theStudyValues.get(0);
        //now, get all the values from the scanner that are in that
        //study, then from that get their different series
        FilenamesType theFilenames =
            theScanner.GetAllFilenamesFromTagToValue(theStudyTag, theStudyVal);

        //from that set of filenames, isolate individual series
        //conclude that singleton series = RT struct (can do further
        //checking for things like MIPs and the like)
        //and multiple series entries = volumetric data
        theScanner.Scan(theFilenames);
        FilenamesType theSeriesValues = theScanner.GetOrderedValues(theSeriesTag);
        String studyUID = theScanner.GetValue(theScanner.GetFilenames().get(0), theStudyTag);
        long theNumSeries = theSeriesValues.size();
        for (int i = 0; i < theNumSeries; i++) {
            FilenamesType theSeriesFiles =
                theScanner.GetAllFilenamesFromTagToValue(theSeriesTag, theSeriesValues.get(i));
            long theNumFilesInSeries = theSeriesFiles.size();
            if (theNumFilesInSeries > 1) { //assume it's CT or volumetric data
                //for now, assume a single volume
                //could have multiples, like PET and CT

                IPPSorter sorter = new IPPSorter();
                sorter.SetComputeZSpacing(true);
                sorter.SetZSpacingTolerance(0.001);
                Boolean sorted = sorter.Sort(theSeriesFiles);
                if (!sorted){
```

```

        //need some better way to handle failures here
        return outImageData;
    }

    FileNamesType sortedFT = sorter.GetFileNames();
    long theSize = sortedFT.size();
    vtkStringArray sa = new vtkStringArray();
    ArrayList<String> theStrings = new ArrayList<String>();

    vtkGDCMImageReader gdcmReader = new vtkGDCMImageReader();
    for (int j = 0; j < theSize; j++) {
        String theFileName = sortedFT.get(j);
        if (gdcmReader.CanReadFile(theFileName) > 0) {
            theStrings.add(theFileName);
            sa.InsertNextValue(theFileName);
        } else {
            //this is a busted series
            //need some more appropriate error here
            return outImageData;
        }
    }

    gdcmReader.SetFileNames(sa);

    gdcmReader.Update();

    outImageData = gdcmReader.GetOutput(); //the zeroth output should be the image
}

String theImageInfo = "";
if (outImageData != null) {
    theImageInfo = outImageData.Print();
}
return outImageData;
}

//this function is a rewrite of Medical3 to see if data can
//be loaded via gdcm easily
public AWTMedical3(File inFile) {
    // Create the buttons.
    renWin = new vtkPanel();

    vtkImageData theImageData = ReadDataFile(inFile);

    // An isosurface, or contour value of 500 is known to correspond to the
    // skin of the patient. Once generated, a vtkPolyDataNormals filter is
    // is used to create normals for smooth surface shading during rendering.
    // The triangle stripper is used to create triangle strips from the
    // isosurface these render much faster on some systems.
    vtkContourFilter skinExtractor = new vtkContourFilter();
    skinExtractor.SetInput(theImageData);
    skinExtractor.SetValue(0, 500);
    vtkPolyDataNormals skinNormals = new vtkPolyDataNormals();
    skinNormals.SetInput(skinExtractor.GetOutput());
    skinNormals.SetFeatureAngle(60.0);
    //      vtkStripper skinStripper = new vtkStripper();
    //      skinStripper.SetInput(skinNormals.GetOutput());
    vtkPolyDataMapper skinMapper = new vtkPolyDataMapper();
    skinMapper.SetInput(skinNormals.GetOutput());
    skinMapper.ScalarVisibilityOff();
    vtkActor skin = new vtkActor();
    skin.SetMapper(skinMapper);
    skin.GetProperty().SetDiffuseColor(1, .49, .25);
    skin.GetProperty().SetSpecular(.3);
    skin.GetProperty().SetSpecularPower(20);

    // An isosurface, or contour value of 1150 is known to correspond to the
    // skin of the patient. Once generated, a vtkPolyDataNormals filter is
    // is used to create normals for smooth surface shading during rendering.
    // The triangle stripper is used to create triangle strips from the
    // isosurface these render much faster on some systems.
    vtkContourFilter boneExtractor = new vtkContourFilter();
    boneExtractor.SetInput(theImageData);
    boneExtractor.SetValue(0, 1150);
    vtkPolyDataNormals boneNormals = new vtkPolyDataNormals();
    boneNormals.SetInput(boneExtractor.GetOutput());
    boneNormals.SetFeatureAngle(60.0);
    vtkStripper boneStripper = new vtkStripper();
    boneStripper.SetInput(boneNormals.GetOutput());
    vtkPolyDataMapper boneMapper = new vtkPolyDataMapper();
    boneMapper.SetInput(boneStripper.GetOutput());

```

```

boneMapper.ScalarVisibilityOff();
vtkActor bone = new vtkActor();
bone.SetMapper(boneMapper);
bone.GetProperty().SetDiffuseColor(1, 1, .9412);

// An outline provides context around the data.
vtkOutlineFilter outlineData = new vtkOutlineFilter();
outlineData.SetInput(theImageData);
vtkPolyDataMapper mapOutline = new vtkPolyDataMapper();
mapOutline.SetInput(outlineData.GetOutput());
vtkActor outline = new vtkActor();
outline.SetMapper(mapOutline);
outline.GetProperty().SetColor(0, 0, 0);

// Now we are creating three orthogonal planes passing through the
// volume. Each plane uses a different texture map and therefore has
// different coloration.

// Start by creating a black/white lookup table.
vtkLookupTable bwLut = new vtkLookupTable();
bwLut.SetTableRange(0, 2000);
bwLut.SetSaturationRange(0, 0);
bwLut.SetHueRange(0, 0);
bwLut.SetValueRange(0, 1);
bwLut.Build();

// Now create a lookup table that consists of the full hue circle (from
// HSV);.
vtkLookupTable hueLut = new vtkLookupTable();
hueLut.SetTableRange(0, 2000);
hueLut.SetHueRange(0, 1);
hueLut.SetSaturationRange(1, 1);
hueLut.SetValueRange(1, 1);
hueLut.Build();

// Finally, create a lookup table with a single hue but having a range
// in the saturation of the hue.
vtkLookupTable satLut = new vtkLookupTable();
satLut.SetTableRange(0, 2000);
satLut.SetHueRange(.6, .6);
satLut.SetSaturationRange(0, 1);
satLut.SetValueRange(1, 1);
satLut.Build();

// Create the first of the three planes. The filter vtkImageMapToColors
// maps the data through the corresponding lookup table created above.
// The vtkImageActor is a type of vtkProp and conveniently displays an
// image on a single quadrilateral plane. It does this using texture
// mapping and as a result is quite fast. (Note: the input image has to
// be unsigned char values, which the vtkImageMapToColors produces.);
// Note also that by specifying the DisplayExtent, the pipeline
// requests data of this extent and the vtkImageMapToColors only
// processes a slice of data.
vtkImageMapToColors sagittalColors = new vtkImageMapToColors();
sagittalColors.SetInput(theImageData);
sagittalColors.SetLookupTable(bwLut);
vtkImageActor sagittal = new vtkImageActor();
sagittal.SetInput(sagittalColors.GetOutput());
sagittal.SetDisplayExtent(32, 32, 0, 63, 0, 92);

// Create the second (axial); plane of the three planes. We use the same
// approach as before except that the extent differs.
vtkImageMapToColors axialColors = new vtkImageMapToColors();
axialColors.SetInput(theImageData);
axialColors.SetLookupTable(hueLut);
vtkImageActor axial = new vtkImageActor();
axial.SetInput(axialColors.GetOutput());
axial.SetDisplayExtent(0, 63, 0, 63, 46, 46);

// Create the third (coronal); plane of the three planes. We use the same
// approach as before except that the extent differs.
vtkImageMapToColors coronalColors = new vtkImageMapToColors();
coronalColors.SetInput(theImageData);
coronalColors.SetLookupTable(satLut);
vtkImageActor coronal = new vtkImageActor();
coronal.SetInput(coronalColors.GetOutput());
coronal.SetDisplayExtent(0, 63, 32, 32, 0, 92);

// It is convenient to create an initial view of the data. The FocalPoint
// and Position form a vector direction. Later on (ResetCamera() method)
// this vector is used to position the camera to look at the data in

```

```

// this direction.
vtkCamera aCamera = new vtkCamera();
aCamera.SetViewUp(0, 0, -1);
aCamera.SetPosition(0, 1, 0);
aCamera.SetFocalPoint(0, 0, 0);
aCamera.ComputeViewPlaneNormal();

// Actors are added to the renderer. An initial camera view is created.
// The Dolly() method moves the camera towards the FocalPoint,
// thereby enlarging the image.
renWin.GetRenderer().AddActor(sagittal);
renWin.GetRenderer().AddActor(axial);
renWin.GetRenderer().AddActor(coronal);
renWin.GetRenderer().AddActor(outline);
renWin.GetRenderer().AddActor(skin);
renWin.GetRenderer().AddActor(bone);

// Turn off bone for this example.
bone.VisibilityOff();

// Set skin to semi-transparent.
skin.GetProperty().SetOpacity(0.5);

// An initial camera view is created. The Dolly() method moves
// the camera towards the FocalPoint, thereby enlarging the image.
renWin.GetRenderer().SetActiveCamera(aCamera);
renWin.GetRenderer().ResetCamera();
aCamera.Dolly(1.5);

// Set a background color for the renderer and set the size of the
// render window (expressed in pixels).
renWin.GetRenderer().SetBackground(1, 1, 1);
VtkPanelUtil.setSize(renWin, 640, 480);

// Note that when camera movement occurs (as it does in the Dolly()
// method), the clipping planes often need adjusting. Clipping planes
// consist of two planes: near and far along the view direction. The
// near plane clips out objects in front of the plane the far plane
// clips out objects behind the plane. This way only what is drawn
// between the planes is actually rendered.
renWin.GetRenderer().ResetCameraClippingRange();

// Setup panel
setLayout(new BorderLayout());
add(renWin, BorderLayout.CENTER);
}

public vtkPanel getRenWin() {
    return renWin;
}

public static void main(String s[]) {
    if (s.length == 0){
        return; //need a filename here
    }
    File theFile = new File(s[0]);
    //File theFile = new
    File("/Users/mmroden/Documents/MVSDownloadDirectory/Documents/1.2.840.113704.1.111.3384.1271766367.5/");
    AWTMedical3 panel = new AWTMedical3(theFile);

    JFrame frame = new JFrame("AWTMedical3");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().add("Center", panel);
    frame.pack();
    frame.setVisible(true);
}
}

```

## 12.168 HelloVTKWorld.java

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

```

Copyright (c) 2006-2011 Mathieu Malaterre  
 All rights reserved.  
 See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even  
 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR  
 PURPOSE. See the above copyright notice for more information.

```

=====*/
// We are required to call the package 'vtk' even though I (MM) would have preferred
// an import statement along the line of:
// import vtkgdc.*;
import vtk.*;

/*
 * Compilation:
 * CLASSPATH=vtkgdc.jar:/usr/share/java/vtk.jar javac HelloVTKWorld.java
 *
 * Usage:
 * LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/jni:.
 * CLASSPATH=/usr/share/java/vtk.jar:vtkgdc.jar:gdcm.jar:. java HelloVTKWorld gdcmData/012345.002.050.dcm
 * bla.dcm
 */
public class HelloVTKWorld
{
    static {
        System.loadLibrary("vtkCommonJava");
        System.loadLibrary("vtkFilteringJava");
        System.loadLibrary("vtkIOJava");
        System.loadLibrary("vtkImagingJava");
        System.loadLibrary("vtkGraphicsJava");
        System.loadLibrary("vtkgdcJava");
        try {
            System.loadLibrary("vtkRenderingJava");
        } catch (Throwable e) {
            System.out.println("cannot load vtkHybrid, skipping...");
        }
        try {
            System.loadLibrary("vtkHybridJava");
        } catch (Throwable e) {
            System.out.println("cannot load vtkHybrid, skipping...");
        }
        try {
            System.loadLibrary("vtkVolumeRenderingJava");
        } catch (Throwable e) {
            System.out.println("cannot load vtkVolumeRendering, skipping...");
        }
    }

    public static void main(String[] args)
    {
        String filename = args[0];
        vtkGDCMImageReader reader = new vtkGDCMImageReader();
        reader.SetFileName( filename );
        reader.Update();

        vtkMedicalImageProperties prop = reader.GetMedicalImageProperties();
        System.out.println( prop.GetPatientName() ); //

        // if( reader.GetImageFormat() == vtkgdc.vtkgdc.VTK_LUMINANCE ) // MONOCHROME2
        // {
        //     System.out.println( "Image is MONOCHROME2" ); //
        // }

        // Just for fun, invert the direction cosines, output should reflect that:
        vtkMatrix4x4 dircos = reader.GetDirectionCosines();
        dircos.Invert();

        // We need to maintain in sync information stored in vtkMedicalImageProperties:
        double[] cosines = new double[6];
        cosines[0] = dircos.GetElement(0,0);
        cosines[1] = dircos.GetElement(1,0);
        cosines[2] = dircos.GetElement(2,0);
        cosines[3] = dircos.GetElement(0,1);
        cosines[4] = dircos.GetElement(1,1);
        cosines[5] = dircos.GetElement(2,1);
        reader.GetMedicalImageProperties().SetDirectionCosine( cosines );

        String outfilename = args[1];

```



```

        vtkGDCMImageWriter writer = new vtkGDCMImageWriter();
        writer.SetMedicalImageProperties( reader.GetMedicalImageProperties() );
        writer.SetDirectionCosines( dirs );
        writer.SetShift( reader.GetShift() );
        writer.SetScale( reader.GetScale() );
        writer.SetImageFormat( reader.GetImageFormat() );
        writer.SetFileName( outfilename );
        writer.SetInputConnection( reader.GetOutputPort() ); // new
        //writer.SetInput( reader.GetOutput() ); // old
        writer.Write();

        System.out.println("Success reading: " + filename );
    }
}

```

## 12.169 MIPViewer.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
import vtk.*;
import gdcm.*;
import java.io.File;
import java.awt.Canvas;

/*
 * Compilation:
 * CLASSPATH=vtkgdcm.jar:/usr/share/java/vtk.jar javac MIPViewer.java
 *
 * Usage:
 * LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/jni:.
 * CLASSPATH=/usr/share/java/vtk.jar:vtkgdcm.jar:gdcm.jar:. java MIPViewer BRAINX
 *
 */
public class MIPViewer extends Canvas
{
    static {
        // VTK
        System.loadLibrary("vtkCommonJava");
        System.loadLibrary("vtkFilteringJava");
        System.loadLibrary("vtkIOJava");
        System.loadLibrary("vtkImagingJava");
        System.loadLibrary("vtkGraphicsJava");
        System.loadLibrary("vtkRenderingJava");
        System.loadLibrary("vtkVolumeRenderingJava"); // vtkSmartVolumeMapper
        System.loadLibrary("vtkWidgetsJava"); // vtkBoxWidget
        // VTK-GDCM
        System.loadLibrary("vtkgdcmJava");
    }

    static FilenamesType fns = new FilenamesType();

    protected native int Lock();

    protected native int UnLock();

    public static void process(String path)
    {
        fns.add( path );
    }

    // Process only files under dir
    public static void visitAllFiles(File dir)
    {
        if (dir.isDirectory())
        {
            String[] children = dir.list();

```

```

        for (int i=0; i<children.length; i++)
        {
            visitAllFiles(new File(dir, children[i]));
        }
    }
    else
    {
        process(dir.getPath());
    }
}

public static void main(String[] args) throws Exception
{
    String dirname = args[0];
    if( !PosixEmulation.FileIsDirectory( dirname ) )
    {
        return;
    }

    File dir = new File(dirname);
    visitAllFiles(dir);

    IPPSorter ipp = new IPPSorter();
    ipp.SetComputeZSpacing( true );
    ipp.SetZSpacingTolerance( 1e-3 );
    boolean b = ipp.Sort( fns );
    if(!b)
    {
        throw new Exception("Could not scan");
    }
    double ippzspacing = ipp.GetZSpacing();

    FilenamesType sorted = ipp.GetFilenames();
    vtkStringArray files = new vtkStringArray();
    long nfiles = sorted.size();
    //for( String f : sorted )
    for (int i = 0; i < nfiles; i++) {
        String f = sorted.get(i);
        files.InsertNextValue( f );
    }
    vtkGDCMImageReader reader = new vtkGDCMImageReader();
    reader.SetFileNames( files );
    reader.Update(); // get spacing value

    double[] spacing = reader.GetOutput().GetSpacing();

    vtkImageChangeInformation change = new vtkImageChangeInformation();
    change.SetInputConnection( reader.GetOutputPort() );
    change.SetOutputSpacing( spacing[0], spacing[1], ippzspacing );

    // Create our volume and mapper
    vtkVolume volume = new vtkVolume();
    vtkSmartVolumeMapper mapper = new vtkSmartVolumeMapper();

    vtkRenderWindowInteractor iren = new vtkRenderWindowInteractor();

    // Add a box widget if the clip option was selected
    vtkBoxWidget box = new vtkBoxWidget();
    box.SetInteractor(iren);
    box.SetPlaceFactor(1.01);
    box.SetInputConnection(change.GetOutputPort());

    //box.SetDefaultRenderer(renderer);
    box.InsideOutOn();
    box.PlaceWidget();
    //vtkBoxWidgetCallback callback = vtkBoxWidgetCallback::New();
    //callback.SetMapper(mapper);
    //box.AddObserver(vtkCommand::InteractionEvent, callback);
    //callback.Delete();
    // Lock();
    // box.EnabledOn();
    // Unlock();
    box.GetSelectedFaceProperty().SetOpacity(0.0);

    mapper.SetInputConnection( change.GetOutputPort() );

    // Create our transfer function
    vtkColorTransferFunction colorFun = new vtkColorTransferFunction();
    vtkPiecewiseFunction opacityFun = new vtkPiecewiseFunction();

    // Create the property and attach the transfer functions

```

```

vtkVolumeProperty property = new vtkVolumeProperty();
property.IndependentComponentsOn();
property.SetColor( colorFun );
property.SetScalarOpacity( opacityFun );
property.SetInterpolationTypeToLinear();

// connect up the volume to the property and the mapper
volume.SetProperty( property );
volume.SetMapper( mapper );

vtkMedicalImageProperties medprop = reader.GetMedicalImageProperties();
int n = medprop.GetNumberOfWindowLevelPresets();
double opacityWindow = 4096;
double opacityLevel = 2048;

// Override default with value from DICOM files:
for( int i = 0; i < n; ++i )
{
    double wl[] = medprop.GetNthWindowLevelPreset(i);
    //System.out.println( "W/L: " + wl[0] + " " + wl[1] );
    opacityWindow = wl[0];
    opacityLevel = wl[1];
}

colorFun.AddRGBSegment(0.0, 1.0, 1.0, 1.0, 255.0, 1.0, 1.0, 1.0 );
opacityFun.AddSegment( opacityLevel - 0.5*opacityWindow, 0.0,
    opacityLevel + 0.5*opacityWindow, 1.0 );
mapper.SetBlendModeToMaximumIntensity();

// Create the RenderWindow, Renderer
vtkRenderer ren1 = new vtkRenderer();
vtkRenderWindow renWin = new vtkRenderWindow();
renWin.AddRenderer(ren1);

// Set the default window size
renWin.SetSize(600,600);

// Add the volume to the scene
ren1.AddVolume( volume );
ren1.ResetCamera();

iren.SetRenderWindow( renWin );

// interact with data
renWin.Render();

iren.Start();
}
}

```

## 12.170 MPRViewer.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
import vtk.*;
import gdcml.*;
import java.io.File;

/*
 * Compilation:
 * CLASSPATH=vtkgdcml.jar:/usr/share/java/vtk.jar javac MPRViewer.java
 *
 * Usage:
 * LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/jni:.
 * CLASSPATH=/usr/share/java/vtk.jar:vtkgdcml.jar:gdcml.jar:. java MPRViewer BRAINX
 *
 */

```

```

*/
public class MPRViewer
{
    static {
        // VTK
        System.loadLibrary("vtkCommonJava");
        System.loadLibrary("vtkFilteringJava");
        System.loadLibrary("vtkIOJava");
        System.loadLibrary("vtkImagingJava");
        System.loadLibrary("vtkGraphicsJava");
        System.loadLibrary("vtkRenderingJava");
        // VTK-GDCM
        System.loadLibrary("vtkgdcmJava");
    }

    static FilenamesType fns = new FilenamesType();

    public static void process(String path)
    {
        fns.add( path );
    }

    // Process only files under dir
    public static void visitAllFiles(File dir)
    {
        if (dir.isDirectory())
        {
            String[] children = dir.list();
            for (int i=0; i<children.length; i++)
            {
                visitAllFiles(new File(dir, children[i]));
            }
        }
        else
        {
            process(dir.getPath());
        }
    }

    public static void main(String[] args) throws Exception
    {
        String dirname = args[0];
        if( !PosixEmulation.FileIsDirectory( dirname ) )
        {
            return;
        }

        File dir = new File(dirname);
        visitAllFiles(dir);

        IPPSorter ipp = new IPPSorter();
        ipp.SetComputeZSpacing( true );
        ipp.SetZSpacingTolerance( 1e-3 );
        boolean b = ipp.Sort( fns );
        if(!b)
        {
            throw new Exception("Could not scan");
        }
        double ippzspacing = ipp.GetZSpacing();

        FilenamesType sorted = ipp.GetFilenames();
        vtkStringArray files = new vtkStringArray();
        long nfiles = sorted.size();
        //for( String f : sorted )
        for (int i = 0; i < nfiles; i++) {
            String f = sorted.get(i);
            files.InsertNextValue( f );
        }
        vtkGDCMImageReader reader = new vtkGDCMImageReader();
        reader.SetFileNames( files );
        reader.Update(); // get spacing value

        double[] spacing = reader.GetOutput().GetSpacing();

        vtkImageChangeInformation change = new vtkImageChangeInformation();
        change.SetInputConnection( reader.GetOutputPort() );
        change.SetOutputSpacing( spacing[0], spacing[1], ippzspacing );

        // A simple vtkInteractorStyleImage example for
        // 3D image viewing with the vtkImageResliceMapper.
        //

```

```

// Drag Left mouse button to window/level
// Shift-Left drag to rotate (oblique slice)
// Shift-Middle drag to slice through image
// OR Ctrl-Right drag to slice through image

// Create the RenderWindow, Renderer
vtkRenderer ren1 = new vtkRenderer();
vtkRenderWindow renWin = new vtkRenderWindow();
renWin.AddRenderer(ren1);

vtkImageResliceMapper im = new vtkImageResliceMapper();
im.SetInputConnection(change.GetOutputPort());
im.SliceFacesCameraOn();
im.SliceAtFocalPointOn();
im.BorderOff();

vtkImageProperty ip = new vtkImageProperty();
ip.SetColorWindow(2000);
ip.SetColorLevel(1000);
ip.SetAmbient(0.0);
ip.SetDiffuse(1.0);
ip.SetOpacity(1.0);
ip.SetInterpolationTypeToLinear();

vtkImageSlice ia = new vtkImageSlice();
ia.SetMapper(im);
ia.SetProperty(ip);

ren1.AddViewProp(ia);
ren1.SetBackground(0.1,0.2,0.4);
renWin.SetSize(300,300);

vtkRenderWindowInteractor iren = new vtkRenderWindowInteractor();
vtkInteractorStyleImage style = new vtkInteractorStyleImage();
style.SetInteractionModeToImage3D();
iren.SetInteractorStyle(style);
renWin.SetInteractor(iren);

// render the image
renWin.Render();
vtkCamera cam1 = ren1.GetActiveCamera();
cam1.ParallelProjectionOn();
ren1.ResetCameraClippingRange();
renWin.Render();

iren.Start();
}
}

```

## 12.171 MPRViewer2.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
import vtk.*;
import gdcm.*;
import java.io.File;

/*
 * Compilation:
 * CLASSPATH=vtkgdcm.jar:/usr/share/java/vtk.jar javac MPRViewer2.java
 *
 * Usage:
 * LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/jni:.
 * CLASSPATH=/usr/share/java/vtk.jar:vtkgdcm.jar:gdcm.jar:. java MPRViewer2 BRAINX
 *
 */

```

```

public class MPRViewer2
{
    static {
        // VTK
        System.loadLibrary("vtkCommonJava");
        System.loadLibrary("vtkFilteringJava");
        System.loadLibrary("vtkIOJava");
        System.loadLibrary("vtkImagingJava");
        System.loadLibrary("vtkGraphicsJava");
        System.loadLibrary("vtkRenderingJava");
        System.loadLibrary("vtkHybridJava");
        System.loadLibrary("vtkWidgetsJava");
        // VTK-GDCM
        System.loadLibrary("vtkgdcmJava");
    }

    static FilenamesType fns = new FilenamesType();

    public static void process(String path)
    {
        fns.add( path );
    }

    // Process only files under dir
    public static void visitAllFiles(File dir)
    {
        if (dir.isDirectory())
        {
            String[] children = dir.list();
            for (int i=0; i<children.length; i++)
            {
                visitAllFiles(new File(dir, children[i]));
            }
        }
        else
        {
            process(dir.getPath());
        }
    }

    public void dointer(vtkImagePlaneWidget current_widget)
    {
        int cstat = current_widget.GetCursorDataStatus();
        double[] v = current_widget.GetCurrentCursorPosition();
        //System.out.println( cstat );
        //System.out.println( v[0] );
        //System.out.println( v[1] );
        //System.out.println( v[2] );
        planeWidgetX.SetSliceIndex( (int)v[0] );
        planeWidgetY.SetSliceIndex( (int)v[1] );
        planeWidgetZ.SetSliceIndex( (int)v[2] );
        planeWidgetX.GetCurrentRenderer().ResetCameraClippingRange();
        planeWidgetY.GetCurrentRenderer().ResetCameraClippingRange();
        planeWidgetZ.GetCurrentRenderer().ResetCameraClippingRange();
    }

    public void startinterX()
    {
        dointer( planeWidgetX );
    }

    public void interX()
    {
        dointer( planeWidgetX );
    }

    public void endinterX()
    {
    }

    public void startinterY()
    {
        dointer( planeWidgetY );
    }

    public void interY()
    {
        dointer( planeWidgetY );
    }

    public void endinterY()
    {
    }

    public void startinterZ()
    {
        dointer( planeWidgetZ );
    }
}

```

```

public void interZ()
{
    dointer( planeWidgetZ );
}
public void endinterZ()
{
    //System.out.println( "endinter" );
}

public static void AlignCamera(int slice_number, vtkImagePlaneWidget current_widget)
{
    vtkImageData image = (vtkImageData)current_widget.GetInput();
    vtkRenderer ren = current_widget.GetCurrentRenderer();
    double[] origin = image.GetOrigin();
    double ox = origin[0];
    double oy = origin[1];
    double oz = origin[2];

    int dims[] = image.GetDimensions();
    int xMin = 0;
    int xMax = 1;
    int yMin = 2;
    int yMax = dims[0]-1;
    int zMin = dims[1]-1;
    int zMax = dims[2]-1;

    double[] spacing = image.GetSpacing();
    double sx = spacing[0];
    double sy = spacing[1];
    double sz = spacing[2];

    double cx = ox+(0.5*(xMax-xMin))*sx;
    double cy = oy+(0.5*(yMax-yMin))*sy;
    double cz = oy+(0.5*(zMax-zMin))*sz;
    double vx = 0, vy = 0, vz = 0;
    double nx = 0, ny = 0, nz = 0;
    int iaxis = current_widget.GetPlaneOrientation();
    if ( iaxis == 0 ) {
        vz = -1;
        nx = ox + xMax*sx;
        cx = ox + slice_number*sx;
    }
    else if ( iaxis == 1 ) {
        vz = -1;
        ny = oy+yMax*sy;
        cy = oy+slice_number*sy;
    }
    else {
        vy = 1;
        nz = oz+zMax*sz;
        cz = oz+slice_number*sz;
    }
    double px = cx+nx*2;
    double py = cy+ny*2;
    double pz = cz+nz*3;

    vtkCamera camera = ren.GetActiveCamera();
    camera.SetViewUp(vx, vy, vz);
    camera.SetFocalPoint(cx, cy, cz);
    camera.SetPosition(px, py, pz);
    camera.OrthogonalizeViewUp();
    ren.ResetCameraClippingRange();
}

private vtkImagePlaneWidget planeWidgetX = new vtkImagePlaneWidget();
private vtkImagePlaneWidget planeWidgetY = new vtkImagePlaneWidget();
private vtkImagePlaneWidget planeWidgetZ = new vtkImagePlaneWidget();

public void config()
{
    //System.out.println( "config" );
    planeWidgetX.GetCurrentRenderer().ResetCamera();
    planeWidgetY.GetCurrentRenderer().ResetCamera();
    planeWidgetZ.GetCurrentRenderer().ResetCamera();
}

public void Run(String dirname)
{
    File dir = new File(dirname);

```

```

visitAllFiles(dir);

IPPSorter ipp = new IPPSorter();
ipp.SetComputeZSpacing( true );
ipp.SetZSpacingTolerance( 1e-3 );
boolean b = ipp.Sort( fns );
if(!b)
{
    //throw new Exception("Could not scan");
}
double ippzspacing = ipp.GetZSpacing();

FileNamesType sorted = ipp.GetFileNames();
vtkStringArray files = new vtkStringArray();
long nfiles = sorted.size();
//for( String f : sorted )
for (int i = 0; i < nfiles; i++) {
    String f = sorted.get(i);
    files.InsertNextValue( f );
}
vtkGDCMImageReader reader = new vtkGDCMImageReader();
reader.SetFileNames( files );
reader.Update(); // get spacing value

double[] spacing = reader.GetOutput().GetSpacing();

vtkImageChangeInformation change = new vtkImageChangeInformation();
change.SetInputConnection( reader.GetOutputPort() );
change.SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
change.Update();

System.out.println( change.GetOutput().toString() );

vtkRenderer ren1 = new vtkRenderer();
ren1.SetViewport(0., 0., 0.333, 1);
ren1.SetBackground(0.1,0.2,0.4);
vtkRenderer ren2 = new vtkRenderer();
ren2.SetViewport(0.333, 0., 0.667, 1);
ren2.SetBackground(0.1,0.2,0.4);
vtkRenderer ren3 = new vtkRenderer();
ren3.SetViewport(0.667, 0., 1., 1.);
ren3.SetBackground(0.1,0.2,0.4);

vtkRenderWindow renWin = new vtkRenderWindow();
renWin.AddRenderer(ren1);
renWin.AddRenderer(ren2);
renWin.AddRenderer(ren3);

vtkRenderWindowInteractor iren = new vtkRenderWindowInteractor();
iren.SetRenderWindow(renWin);

vtkInteractorStyleImage style = new vtkInteractorStyleImage();
iren.SetInteractorStyle( style );

vtkCellPicker picker = new vtkCellPicker();
picker.SetTolerance(0.005);

vtkProperty ipwProp = new vtkProperty();

//vtkImagePlaneWidget planeWidgetX = new vtkImagePlaneWidget();
planeWidgetX.SetInteractor(iren);
planeWidgetX.SetCurrentRenderer(ren1);
planeWidgetX.SetDefaultRenderer(ren1);
planeWidgetX.RestrictPlaneToVolumeOn();
planeWidgetX.SetTexturePlaneProperty(ipwProp);
//planeWidgetX.GetPlaneProperty().SetColor(1,0,0);
//planeWidgetX.TextureInterpolateOff();
//planeWidgetX.SetResliceInterpolateToNearestNeighbour();
planeWidgetX.SetInputConnection(change.GetOutputPort());
planeWidgetX.SetPlaneOrientationToXAxes();
planeWidgetX.SetSliceIndex(62);
planeWidgetX.SetPicker(picker);
planeWidgetX.SetKeyPressActivationValue('x');
planeWidgetX.On();
planeWidgetX.InteractionOn();

//vtkImagePlaneWidget planeWidgetY = new vtkImagePlaneWidget();
planeWidgetY.SetInteractor(iren);
planeWidgetY.SetCurrentRenderer(ren2);
planeWidgetY.SetDefaultRenderer(ren2);
planeWidgetY.RestrictPlaneToVolumeOn();

```



```

planeWidgetY.SetTexturePlaneProperty(ipwProp);
//planeWidgetY.GetPlaneProperty().SetColor(1,0,0);
//planeWidgetY.TextureInterpolateOff();
//planeWidgetY.SetResliceInterpolateToNearestNeighbour();
planeWidgetY.SetInputConnection(change.GetOutputPort());
planeWidgetY.SetLookupTable( planeWidgetX.GetLookupTable() );
planeWidgetY.SetPlaneOrientationToYAxes();
planeWidgetY.SetSliceIndex(32);
planeWidgetY.SetPicker(picker);
planeWidgetY.SetKeyPressActivationValue('y');
planeWidgetY.On();

//vtkImagePlaneWidget planeWidgetZ = new vtkImagePlaneWidget();
planeWidgetZ.SetInteractor(iren);
planeWidgetZ.SetCurrentRenderer(ren3);
planeWidgetZ.SetDefaultRenderer(ren3);
planeWidgetZ.RestrictPlaneToVolumeOn();
planeWidgetZ.SetTexturePlaneProperty(ipwProp);
//planeWidgetZ.GetPlaneProperty().SetColor(1,0,0);
//planeWidgetZ.TextureInterpolateOff();
//planeWidgetZ.SetResliceInterpolateToNearestNeighbour();
planeWidgetZ.SetInputConnection(change.GetOutputPort());
planeWidgetZ.SetLookupTable( planeWidgetX.GetLookupTable() );
planeWidgetZ.SetPlaneOrientationToZAxes();
planeWidgetZ.SetSliceIndex(32);
planeWidgetZ.SetPicker(picker);
planeWidgetZ.SetKeyPressActivationValue('z');
planeWidgetZ.On();

iren.Initialize();

renWin.Render();
AlignCamera(52, planeWidgetX);
AlignCamera(32, planeWidgetY);
AlignCamera(32, planeWidgetZ);

planeWidgetX.GetCurrentRenderer().ResetCamera();
planeWidgetY.GetCurrentRenderer().ResetCamera();
planeWidgetZ.GetCurrentRenderer().ResetCamera();

renWin.Render();

planeWidgetX.AddObserver("StartInteractionEvent", this,"startinterX");
planeWidgetX.AddObserver("InteractionEvent", this,"interX");
planeWidgetX.AddObserver("EndInteractionEvent", this,"endinterX");
planeWidgetY.AddObserver("StartInteractionEvent", this,"startinterY");
planeWidgetY.AddObserver("InteractionEvent", this,"interY");
planeWidgetY.AddObserver("EndInteractionEvent", this,"endinterY");
planeWidgetZ.AddObserver("StartInteractionEvent", this,"startinterZ");
planeWidgetZ.AddObserver("InteractionEvent", this,"interZ");
planeWidgetZ.AddObserver("EndInteractionEvent", this,"endinterZ");

iren.AddObserver("ConfigureEvent", this,"config");

iren.Start();
}

public static void main(String[] args) throws Exception
{
    String dirname = args[0];
    if( !PosixEmulation.FileIsDirectory( dirname ) )
    {
        return;
    }

    MPRViewer2 me = new MPRViewer2();
    me.Run( dirname );
}
}

```

## 12.172 ReadSeriesIntoVTK.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

```

Copyright (c) 2006-2011 Mathieu Malaterre  
 All rights reserved.  
 See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even  
 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR  
 PURPOSE. See the above copyright notice for more information.

```

=====*/
// We are required to call the package 'vtk' even though I (MM) would have preferred
// an import statement along the line of:
// import vtkgdc.*;
import vtk.*;

/*
 * Usage:
 * export LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:.
 * java -classpath `pwd`/vtkgdc.jar:/usr/share/java/vtk.jar:. ReadSeriesIntoVTK
 */
public class ReadSeriesIntoVTK
{
    static {
        System.loadLibrary("vtkCommonJava");
        System.loadLibrary("vtkFilteringJava");
        System.loadLibrary("vtkIOJava");
        System.loadLibrary("vtkImagingJava");
        System.loadLibrary("vtkGraphicsJava");
        System.loadLibrary("vtkgdcJava");
        try {
            System.loadLibrary("vtkRenderingJava");
        } catch (Throwable e) {
            System.out.println("cannot load vtkHybrid, skipping...");
        }
        try {
            System.loadLibrary("vtkHybridJava");
        } catch (Throwable e) {
            System.out.println("cannot load vtkHybrid, skipping...");
        }
        try {
            System.loadLibrary("vtkVolumeRenderingJava");
        } catch (Throwable e) {
            System.out.println("cannot load vtkVolumeRendering, skipping...");
        }
    }

    public static void main(String[] args)
    {
        vtkFileOutputWindow outWin = new vtkFileOutputWindow();
        outWin.SetInstance(outWin);
        outWin.SetFileName("MVSVTKViewer.log");

        // See: http://review.source.kitware.com/#change,888
        // vtkWrapJava does not handle static keyword
        // String directory = vtkGDCMTesting.GetGDCMDataRoot();
        vtkGDCMTesting t = new vtkGDCMTesting();
        String directory = t.GetGDCMDataRoot();
        String file0 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq0.dcm";
        String file1 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq1.dcm";
        String file2 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq2.dcm";
        String file3 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq3.dcm";

        vtkStringArray s = new vtkStringArray();
        System.out.println("adding : " + file0);
        s.InsertNextValue(file0);
        s.InsertNextValue(file1);
        s.InsertNextValue(file2);
        s.InsertNextValue(file3);

        vtkGDCMImageReader reader = new vtkGDCMImageReader();
        reader.SetFileNames(s);
        reader.Update();

        System.out.println("Success reading: " + file0);

        vtkMetaImageWriter writer = new vtkMetaImageWriter();
        writer.DebugOn();
        writer.SetCompression(false);
        writer.SetInputConnection(reader.GetOutputPort());
        writer.SetFileName("ReadSeriesIntoVTK.mhd");
    }
}

```

```

        writer.Write();

        System.out.println("Success writing: " + writer.GetFileName() );
    }
}

```

## 12.173 CastConvertPhilips.py

```

00001
00014
00015 """
00016 Usage:
00017
00018 python --public /path/to/directory/
00019 or
00020 python --private /path/to/directory/
00021
00022 python --public --extension bak /path/to/directory/
00023
00024 rename -f 's/\.bak$//' *.bak
00025
00026 TODO:
00027 http://docs.python.org/library/optparse.html#module-optparse
00028 """
00029
00030 import vtkgdc
00031 import vtk
00032 import sys
00033 import gdc
00034
00035 def ProcessOneFilePublic(filename, outfilename, tmpfile):
00036     gdc.ImageHelper.SetForceRescaleInterceptSlope(True)
00037     vtkreader = vtkgdc.vtkGDCMImageReader()
00038     vtkreader.SetFileName( filename )
00039     vtkreader.Update()
00040
00041     cast = vtk.vtkImageCast()
00042     cast.SetInput( vtkreader.GetOutput() )
00043     cast.SetOutputScalarTypeToUnsignedShort()
00044
00045     # vtkGDCMImageWriter does not support Sequence, so let's write a tmp file first:
00046     # Some operation will actually be discarded (we simply need a temp storage)
00047     vtkwriter = vtkgdc.vtkGDCMImageWriter()
00048     vtkwriter.SetFileName( tmpfile )
00049     vtkwriter.SetMedicalImageProperties( vtkreader.GetMedicalImageProperties() )
00050     vtkwriter.SetDirectionCosines( vtkreader.GetDirectionCosines() )
00051     print "Format:", vtkreader.GetImageFormat()
00052     vtkwriter.SetImageFormat( vtkreader.GetImageFormat() )
00053     vtkwriter.SetInput( cast.GetOutput() )
00054     #vtkwriter.Update()
00055     vtkwriter.Write()
00056
00057     # ok now rewrite the exact same file as the original (keep all info)
00058     # but use the Pixel Data Element from the written file
00059     tmpreader = gdc.ImageReader()
00060     tmpreader.SetFileName( tmpfile )
00061     if not tmpreader.Read():
00062         sys.exit(1)
00063
00064     reader = gdc.Reader()
00065     reader.SetFileName( filename )
00066     if not reader.Read():
00067         sys.exit(1)
00068
00069     # Make sure to remove Slope/Rescale to avoid re-execution
00070     ds = reader.GetFile().GetDataSet()
00071     tags = [
00072         gdc.Tag(0x0028,0x1052),
00073         gdc.Tag(0x0028,0x1053),
00074         gdc.Tag(0x0028,0x1053),
00075     ]
00076     for tag in tags:
00077         ds.Remove( tag )
00078
00079     writer = gdc.ImageWriter()
00080     writer.SetFileName( outfilename )
00081     # Pass image from vtk written file

```

```

00082     writer.SetImage( tmpreader.GetImage() )
00083     # pass dataset from initial 'reader'
00084     writer.SetFile( reader.GetFile() )
00085     if not writer.Write():
00086         sys.exit(1)
00087
00088 def ProcessOneFilePrivate(filename, outfilename, tmpfile):
00089     vtkreader = vtkgdcmm.vtkGDCMImageReader()
00090     vtkreader.SetFileName( filename )
00091     vtkreader.Update()
00092
00093
00094     # (2005,1409)      DS      4      0.0
00095     # (2005,140a)     DS      16     1.52283272283272
00096
00097     # (2005,0014)     LO      26     Philips MR Imaging DD 005
00098     tag1 = gdcmm.PrivateTag(0x2005,0x09,"Philips MR Imaging DD 005")
00099     tag2 = gdcmm.PrivateTag(0x2005,0x0a,"Philips MR Imaging DD 005")
00100
00101
00102
00103     # Need to access some private tags, reread the file (for now):
00104     reader = gdcmm.Reader()
00105     reader.SetFileName( filename )
00106     if not reader.Read():
00107         sys.exit(1)
00108
00109     ds = reader.GetFile().GetDataSet()
00110
00111     e11 = ds.GetDataElement( tag1 )
00112     e12 = ds.GetDataElement( tag2 )
00113
00114
00115     #pf = gdcmm.PythonFilter()
00116     #pf.SetFile( reader.GetFile() )
00117     #print e11.GetTag()
00118
00119     print e11.GetByteValue()
00120     v1 = eval(e11.GetByteValue().GetBuffer())
00121     print e12.GetByteValue()
00122     v2 = eval(e12.GetByteValue().GetBuffer())
00123
00124     print v1
00125     shift = v1
00126     print v2
00127     scale = v2
00128
00129     ss = vtk.vtkImageShiftScale()
00130     ss.SetInput( vtkreader.GetOutput() )
00131     # because VTK image shift / scale convention is inverted from DICOM make sure shift is 0
00132     assert shift == 0
00133     ss.SetShift( shift )
00134     ss.SetScale( scale )
00135     ss.SetOutputScalarTypeToUnsignedShort()
00136     ss.Update()
00137
00138     # vtkGDCMImageWriter does not support Sequence, so let's write a tmp file first:
00139     # Some operation will actually be discarded (we simply need a temp storage)
00140     vtkwriter = vtkgdcmm.vtkGDCMImageWriter()
00141     vtkwriter.SetFileName( tmpfile )
00142     vtkwriter.SetMedicalImageProperties( vtkreader.GetMedicalImageProperties() )
00143     vtkwriter.SetDirectionCosines( vtkreader.GetDirectionCosines() )
00144     vtkwriter.SetImageFormat( reader.GetImageFormat() )
00145     # do not pass shift/scale again
00146     vtkwriter.SetInput( ss.GetOutput() )
00147     #vtkwriter.Update()
00148     vtkwriter.Write()
00149
00150     # ok now rewrite the exact same file as the original (keep all info)
00151     # but use the Pixel Data Element from the written file
00152     tmpreader = gdcmm.ImageReader()
00153     tmpreader.SetFileName( tmpfile )
00154     if not tmpreader.Read():
00155         sys.exit(1)
00156
00157     writer = gdcmm.ImageWriter()
00158     writer.SetFileName( outfilename )
00159     # Pass image from vtk written file
00160     writer.SetImage( tmpreader.GetImage() )
00161     # pass dataset from initial 'reader'
00162     writer.SetFile( reader.GetFile() )

```

```

00163     if not writer.Write():
00164         sys.exit(1)
00165
00166 if __name__ == "__main__":
00167     gdcm.Trace.DebugOff()
00168     gdcm.Trace.WarningOff()
00169     #filename = sys.argv[1]
00170     #outfilename = sys.argv[2]
00171     tmpfile = "/tmp/philips_rescaled.dcm"
00172     #ProcessOneFile( filename, outfilename, tmpfile )
00173     rescaletype = sys.argv[1]
00174     assert rescaletype == "--public" or rescaletype == "--private"
00175     dirname = sys.argv[2]
00176     d = gdcm.Directory()
00177     d.Load( dirname )
00178
00179     for f in d.GetFileNames():
00180         #print f
00181         ProcessOneFilePublic( f, f + ".bak", tmpfile )
00182
00183
00184
00185 print "success"

```

## 12.174 headsq2dcm.py

```

00001
00014
00015 """
00016 Usage:
00017 python headsq2dcm.py -D /path/to/VTKData
00018 """
00019
00020 import vtk
00021 import vtkgdcm
00022 from vtk.util.misc import vtkGetDataRoot
00023 VTK_DATA_ROOT = vtkGetDataRoot()
00024
00025 reader = vtk.vtkVolume16Reader()
00026 reader.SetDataDimensions(64, 64)
00027 reader.SetDataByteOrderToLittleEndian()
00028 reader.SetFilePrefix(VTK_DATA_ROOT + "/Data/headsq/quarter")
00029 reader.SetImageRange(1, 93)
00030 reader.SetDataSpacing(3.2, 3.2, 1.5)
00031
00032 cast = vtk.vtkImageCast()
00033 cast.SetInput( reader.GetOutput() )
00034 cast.SetOutputScalarTypeToUnsignedChar()
00035
00036 # By default this is creating a Multiframe Grayscale Word Secondary Capture Image Storage
00037 writer = vtkgdcm.vtkGDCMImageWriter()
00038 writer.SetFileName( "headsq.dcm" )
00039 writer.SetInput( reader.GetOutput() )
00040 # cast -> Multiframe Grayscale Byte Secondary Capture Image Storage
00041 #writer.SetInput( cast.GetOutput() )
00042 writer.SetFileDimensionality( 3 )
00043 writer.Write()

```



# Index

- ~ASN1
  - gdcm::ASN1, [127](#)
- ~AnonymizeEvent
  - gdcm::AnonymizeEvent, [105](#)
- ~Anonymizer
  - gdcm::Anonymizer, [110](#)
- ~Attribute
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [149](#)
- ~AudioCodec
  - gdcm::AudioCodec, [165](#)
- ~BaseCompositeMessage
  - gdcm::network::BaseCompositeMessage, [170](#)
- ~BaseNormalizedMessage
  - gdcm::network::BaseNormalizedMessage, [172](#)
- ~BasePDU
  - gdcm::network::BasePDU, [174](#)
- ~BaseQuery
  - gdcm::BaseQuery, [177](#)
- ~BaseRootQuery
  - gdcm::BaseRootQuery, [182](#)
- ~Bitmap
  - gdcm::Bitmap, [195](#)
- ~BitmapToBitmapFilter
  - gdcm::BitmapToBitmapFilter, [207](#)
- ~BoxRegion
  - gdcm::BoxRegion, [210](#)
- ~ByteSwapFilter
  - gdcm::ByteSwapFilter, [217](#)
- ~ByteValue
  - gdcm::ByteValue, [220](#)
- ~CAPICryptographicMessageSyntax
  - gdcm::CAPICryptographicMessageSyntax, [229](#)
- ~CSAHeader
  - gdcm::CSAHeader, [290](#)
- ~Cleaner
  - gdcm::Cleaner, [241](#)
- ~Coder
  - gdcm::Coder, [252](#)
- ~Command
  - gdcm::Command, [259](#)
- ~CommandDataSet
  - gdcm::CommandDataSet, [263](#)
- ~CryptoFactory
  - gdcm::CryptoFactory, [277](#)
- ~CryptographicMessageSyntax
  - gdcm::CryptographicMessageSyntax, [279](#)
- ~Curve
  - gdcm::Curve, [305](#)
- ~DICOMDIRGenerator
  - gdcm::DICOMDIRGenerator, [354](#)
- ~DPath
  - gdcm::DPath, [386](#)
- ~DataEvent
  - gdcm::DataEvent, [323](#)
- ~DataSetEvent
  - gdcm::DataSetEvent, [339](#)
- ~Decoder
  - gdcm::Decoder, [342](#)
- ~Defs
  - gdcm::Defs, [345](#)
- ~DeltaEncodingCodec
  - gdcm::DeltaEncodingCodec, [351](#)
- ~DictConverter
  - gdcm::DictConverter, [361](#)
- ~DictPrinter
  - gdcm::DictPrinter, [370](#)
- ~Dicts
  - gdcm::Dicts, [373](#)
- ~DirectionCosines
  - gdcm::DirectionCosines, [378](#)
- ~Directory
  - gdcm::Directory, [381](#)
- ~Dumper
  - gdcm::Dumper, [391](#)
- ~Element
  - gdcm::Element< TVR, VM::VM1\_n >, [400](#)
- ~EmptyMaskGenerator
  - gdcm::EmptyMaskGenerator, [424](#)
- ~Event
  - gdcm::Event, [434](#)
- ~Exception
  - gdcm::Exception, [437](#)
- ~File
  - gdcm::File, [450](#)
- ~FileAnonymizer
  - gdcm::FileAnonymizer, [455](#)
- ~FileChangeTransferSyntax
  - gdcm::FileChangeTransferSyntax, [459](#)
- ~FileDecompressLookupTable

- gdcmm::FileDecompressLookupTable, 463
- ~FileDerivation
  - gdcmm::FileDerivation, 465
- ~FileExplicitFilter
  - gdcmm::FileExplicitFilter, 469
- ~FileMetaInformation
  - gdcmm::FileMetaInformation, 475
- ~FileNameEvent
  - gdcmm::FileNameEvent, 486
- ~FileStreamer
  - gdcmm::FileStreamer, 496
- ~FilenameGenerator
  - gdcmm::FilenameGenerator, 489
- ~Global
  - gdcmm::Global, 515
- ~GroupDict
  - gdcmm::GroupDict, 519
- ~IconImageFilter
  - gdcmm::IconImageFilter, 521
- ~IconImageGenerator
  - gdcmm::IconImageGenerator, 524
- ~Image
  - gdcmm::Image, 532
- ~ImageApplyLookupTable
  - gdcmm::ImageApplyLookupTable, 538
- ~ImageChangePhotometricInterpretation
  - gdcmm::ImageChangePhotometricInterpretation, 542
- ~ImageChangePlanarConfiguration
  - gdcmm::ImageChangePlanarConfiguration, 547
- ~ImageChangeTransferSyntax
  - gdcmm::ImageChangeTransferSyntax, 551
- ~ImageCodec
  - gdcmm::ImageCodec, 557
- ~ImageConverter
  - gdcmm::ImageConverter, 566
- ~ImageFragmentSplitter
  - gdcmm::ImageFragmentSplitter, 569
- ~ImageReader
  - gdcmm::ImageReader, 580
- ~ImageRegionReader
  - gdcmm::ImageRegionReader, 585
- ~ImageToImageFilter
  - gdcmm::ImageToImageFilter, 589
- ~ImageWriter
  - gdcmm::ImageWriter, 593
- ~JPEG12Codec
  - gdcmm::JPEG12Codec, 627
- ~JPEG16Codec
  - gdcmm::JPEG16Codec, 632
- ~JPEG2000Codec
  - gdcmm::JPEG2000Codec, 636
- ~JPEG8Codec
  - gdcmm::JPEG8Codec, 644
- ~JPEGCodec
  - gdcmm::JPEGCodec, 649
- ~JPEGLSCodec
  - gdcmm::JPEGLSCodec, 658
- ~JSON
  - gdcmm::JSON, 662
- ~KAKADUCodec
  - gdcmm::KAKADUCodec, 667
- ~LookupTable
  - gdcmm::LookupTable, 674
- ~MemberCommand
  - gdcmm::MemberCommand< T >, 702
- ~MeshPrimitive
  - gdcmm::MeshPrimitive, 708
- ~ModuleEntry
  - gdcmm::ModuleEntry, 723
- ~MrProtocol
  - gdcmm::MrProtocol, 735
- ~Object
  - gdcmm::Object, 764
- ~OpenSSLCryptographicMessageSyntax
  - gdcmm::OpenSSLCryptographicMessageSyntax, 769
- ~OpenSSL7CryptographicMessageSyntax
  - gdcmm::OpenSSL7CryptographicMessageSyntax, 774
- ~Orientation
  - gdcmm::Orientation, 777
- ~Overlay
  - gdcmm::Overlay, 782
- ~PDBHeader
  - gdcmm::PDBHeader, 802
- ~PDFCodec
  - gdcmm::PDFCodec, 805
- ~PGXCodec
  - gdcmm::PGXCodec, 815
- ~PNMCodec
  - gdcmm::PNMCodec, 850
- ~PVRGCodec
  - gdcmm::PVRGCodec, 889
- ~ParseException
  - gdcmm::ParseException, 789
- ~Parser
  - gdcmm::Parser, 792
- ~Pixmap
  - gdcmm::Pixmap, 832
- ~PixmapReader
  - gdcmm::PixmapReader, 838
- ~PixmapToPixmapFilter
  - gdcmm::PixmapToPixmapFilter, 841
- ~PixmapWriter
  - gdcmm::PixmapWriter, 845
- ~Preamble
  - gdcmm::Preamble, 853
- ~Printer
  - gdcmm::Printer, 873



- ~PrivateDict
  - gdcm::PrivateDict, [876](#)
- ~ProgressEvent
  - gdcm::ProgressEvent, [885](#)
- ~PythonFilter
  - gdcm::PythonFilter, [891](#)
- ~QueryBase
  - gdcm::QueryBase, [893](#)
- ~RAWCodec
  - gdcm::RAWCodec, [909](#)
- ~RLECodec
  - gdcm::RLECodec, [929](#)
- ~Reader
  - gdcm::Reader, [914](#)
- ~Region
  - gdcm::Region, [920](#)
- ~Rescaler
  - gdcm::Rescaler, [923](#)
- ~SHA1
  - gdcm::SHA1, [1006](#)
- ~Scanner
  - gdcm::Scanner, [938](#)
- ~Scanner2
  - gdcm::Scanner2, [948](#)
- ~Segment
  - gdcm::Segment, [957](#)
- ~SegmentReader
  - gdcm::SegmentReader, [969](#)
- ~SegmentWriter
  - gdcm::SegmentWriter, [972](#)
- ~SegmentedPaletteColorLookupTable
  - gdcm::SegmentedPaletteColorLookupTable, [965](#)
- ~SerieHelper
  - gdcm::SerieHelper, [992](#)
- ~ServiceClassUser
  - gdcm::ServiceClassUser, [1000](#)
- ~SimpleMemberCommand
  - gdcm::SimpleMemberCommand< T >, [1010](#)
- ~SimpleSubjectWatcher
  - gdcm::SimpleSubjectWatcher, [1013](#)
- ~SmartPointer
  - gdcm::SmartPointer< ObjectType >, [1019](#)
- ~Sorter
  - gdcm::Sorter, [1026](#)
- ~Spacing
  - gdcm::Spacing, [1030](#)
- ~SplitMosaicFilter
  - gdcm::SplitMosaicFilter, [1033](#)
- ~StreamImageReader
  - gdcm::StreamImageReader, [1038](#)
- ~StreamImageWriter
  - gdcm::StreamImageWriter, [1043](#)
- ~StrictScanner
  - gdcm::StrictScanner, [1052](#)
- ~StrictScanner2
  - gdcm::StrictScanner2, [1061](#)
- ~StringFilter
  - gdcm::StringFilter, [1073](#)
- ~Subject
  - gdcm::Subject, [1078](#)
- ~Surface
  - gdcm::Surface, [1084](#)
- ~SurfaceReader
  - gdcm::SurfaceReader, [1099](#)
- ~SurfaceWriter
  - gdcm::SurfaceWriter, [1103](#)
- ~Table
  - gdcm::Table, [1116](#)
- ~TableEntry
  - gdcm::TableEntry, [1118](#)
- ~TableReader
  - gdcm::TableReader, [1119](#)
- ~TableRow
  - gdcm::network::TableRow, [1122](#)
- ~TagPath
  - gdcm::TagPath, [1133](#)
- ~Testing
  - gdcm::Testing, [1136](#)
- ~Trace
  - gdcm::Trace, [1143](#)
- ~Transition
  - gdcm::network::Transition, [1154](#)
- ~ULAction
  - gdcm::network::ULAction, [1199](#)
- ~ULBasicCallback
  - gdcm::network::ULBasicCallback, [1239](#)
- ~ULConnection
  - gdcm::network::ULConnection, [1241](#)
- ~ULConnectionCallback
  - gdcm::network::ULConnectionCallback, [1246](#)
- ~ULConnectionManager
  - gdcm::network::ULConnectionManager, [1252](#)
- ~ULEvent
  - gdcm::network::ULEvent, [1257](#)
- ~ULWritingCallback
  - gdcm::network::ULWritingCallback, [1261](#)
- ~UserInformation
  - gdcm::network::UserInformation, [1275](#)
- ~Validate
  - gdcm::Validate, [1278](#)
- ~Value
  - gdcm::Value, [1280](#)
- ~Version
  - gdcm::Version, [1284](#)
- ~Writer
  - gdcm::Writer, [1414](#)
- ~XMLDictReader
  - gdcm::XMLDictReader, [1418](#)

- ~XMLPrinter
  - gdcm::XMLPrinter, [1421](#)
- ~XMLPrivateDictReader
  - gdcm::XMLPrivateDictReader, [1424](#)
- ~vtkGDCMImageReader
  - vtkGDCMImageReader, [1310](#)
- ~vtkGDCMImageReader2
  - vtkGDCMImageReader2, [1322](#)
- ~vtkGDCMImageWriter
  - vtkGDCMImageWriter, [1334](#)
- ~vtkGDCMMedicalImageProperties
  - vtkGDCMMedicalImageProperties, [1341](#)
- ~vtkGDCMPolyDataReader
  - vtkGDCMPolyDataReader, [1344](#)
- ~vtkGDCMPolyDataWriter
  - vtkGDCMPolyDataWriter, [1348](#)
- ~vtkGDCMTesting
  - vtkGDCMTesting, [1352](#)
- ~vtkGDCMThreadedImageReader
  - vtkGDCMThreadedImageReader, [1357](#)
- ~vtkGDCMThreadedImageReader2
  - vtkGDCMThreadedImageReader2, [1361](#)
- ~vtkImageColorViewer
  - vtkImageColorViewer, [1369](#)
- ~vtkImageMapToColors16
  - vtkImageMapToColors16, [1380](#)
- ~vtkImageMapToWindowLevelColors2
  - vtkImageMapToWindowLevelColors2, [1385](#)
- ~vtkImagePlanarComponentsToComponents
  - vtkImagePlanarComponentsToComponents, [1388](#)
- ~vtkImageRGBToYBR
  - vtkImageRGBToYBR, [1390](#)
- ~vtkImageYBRToRGB
  - vtkImageYBRToRGB, [1392](#)
- ~vtkLookupTable16
  - vtkLookupTable16, [1394](#)
- ~vtkRTStructSetProperties
  - vtkRTStructSetProperties, [1398](#)
- AAbortPDU
  - gdcm::network::AAbortPDU, [86](#)
- AAssociateACPDU
  - gdcm::network::AAssociateACPDU, [89](#)
  - gdcm::network::AAssociateRQPDU, [99](#)
- AAssociateRJPDU
  - gdcm::network::AAssociateRJPDU, [93](#)
- AAssociateRQPDU
  - gdcm::network::AAssociateACPDU, [91](#)
  - gdcm::network::AAssociateRQPDU, [96](#)
- AbstractMultiDimensionalImageModel
  - gdcm::UIDs, [1186](#)
- AbstractSyntax
  - gdcm::network::AbstractSyntax, [101](#)
  - gdcm::PresentationContext, [858](#)
- AcquisitionContextSRStorage
  - gdcm::UIDs, [1185](#)
- ActiveComponent
  - vtkImageMapToColors16, [1383](#)
- Add
  - gdcm::GroupDict, [519](#)
- add1
  - gdcm, [63](#)
- AddAcceptedPresentationContext
  - gdcm::network::ULConnection, [1242](#)
- AddContourReferencedFrameOfReference
  - vtkRTStructSetProperties, [1399](#)
- AddCSAHeaderDictEntry
  - gdcm::CSAHeaderDict, [294](#)
- AddDerivationDescription
  - gdcm::FileDerivation, [466](#)
- AddDictEntry
  - gdcm::Dict, [358](#)
  - gdcm::PrivateDict, [876](#)
- AddFile
  - gdcm::FileSet, [492](#)
  - gdcm::SerieHelper, [992](#)
- AddFileName
  - gdcm::SerieHelper, [992](#)
- AddFragment
  - gdcm::SequenceOfFragments, [978](#)
- AddFromFile
  - gdcm::PresentationContextGenerator, [863](#)
- AddGroupLength
  - gdcm::DictConverter, [362](#)
- AddImageDirectoryRecord
  - gdcm::DICOmdirGenerator, [354](#)
- AddInput
  - vtkImageColorViewer, [1370](#)
- AddInputConnection
  - vtkImageColorViewer, [1370](#)
- AddIOD
  - gdcm::IODs, [610](#)
- AddIODEntry
  - gdcm::IOD, [605](#)
- AddItem
  - gdcm::SequenceOfItems, [985](#)
- AddMacro
  - gdcm::Macros, [685](#)
  - gdcm::Module, [720](#)
- AddMacroEntry
  - gdcm::Macro, [682](#)
- AddModule
  - gdcm::Modules, [726](#)
- AddModuleEntry
  - gdcm::Module, [720](#)
  - gdcm::NestedModuleEntries, [748](#)
- AddNewUndefinedLengthItem
  - gdcm::SequenceOfItems, [985](#)

- AddObserver
  - gdcm::Subject, [1079](#)
- AddPatientDirectoryRecord
  - gdcm::DICOMDIRGenerator, [354](#)
- AddPresentationContext
  - gdcm::network::AAssociateRQPDU, [96](#)
  - gdcm::PresentationContextGenerator, [863](#)
- AddPresentationContextAC
  - gdcm::network::AAssociateACPDU, [90](#)
- AddPresentationDataValue
  - gdcm::network::PDataTFPDU, [796](#)
- AddPrimitiveData
  - gdcm::MeshPrimitive, [708](#)
- AddPrivateTag
  - gdcm::Scanner, [938](#)
  - gdcm::Scanner2, [948](#)
  - gdcm::StrictScanner, [1052](#)
  - gdcm::StrictScanner2, [1062](#)
- AddPublicTag
  - gdcm::Scanner2, [948](#)
  - gdcm::StrictScanner2, [1062](#)
- AddPurposeOfReferenceCodeSequence
  - gdcm::FileDerivation, [466](#)
- AddQueryDataSet
  - gdcm::BaseQuery, [177](#)
- AddReference
  - gdcm::FileDerivation, [466](#)
- AddReferencedFrameOfReference
  - vtkRTStructSetProperties, [1399](#)
- AddRestriction
  - gdcm::SerieHelper, [992](#), [993](#)
- AddRoleSelectionSub
  - gdcm::network::UserInformation, [1275](#)
- AddSegment
  - gdcm::SegmentWriter, [973](#)
- AddSelect
  - gdcm::Sorter, [1026](#)
- AddSeriesDirectoryRecord
  - gdcm::DICOMDIRGenerator, [354](#)
- AddSkipTag
  - gdcm::Scanner, [938](#)
  - gdcm::Scanner2, [948](#)
  - gdcm::StrictScanner, [1052](#)
  - gdcm::StrictScanner2, [1062](#)
- AddSOPClassExtendedNegotiationSub
  - gdcm::network::UserInformation, [1275](#)
- AddSourceImageSequence
  - gdcm::FileDerivation, [466](#)
- AddStructureSetROI
  - vtkRTStructSetProperties, [1399](#)
- AddStructureSetROIObservation
  - vtkRTStructSetProperties, [1399](#)
- AddStudyDirectoryRecord
  - gdcm::DICOMDIRGenerator, [354](#)
- AddSurface
  - gdcm::Segment, [957](#)
- AddTag
  - gdcm::Scanner, [939](#)
  - gdcm::StrictScanner, [1052](#)
- AddTransferSyntax
  - gdcm::network::PresentationContextRQ, [865](#)
  - gdcm::PresentationContext, [857](#)
- AdultMouseAnatomyOntology
  - gdcm::UIDs, [1183](#)
- AdvancedBlendingPresentationStateStorage
  - gdcm::UIDs, [1184](#)
- AE
  - gdcm::VR, [1296](#)
- AEComp
  - gdcm, [58](#)
- AES128\_CIPHER
  - gdcm::CryptographicMessageSyntax, [279](#)
- AES192\_CIPHER
  - gdcm::CryptographicMessageSyntax, [279](#)
- AES256\_CIPHER
  - gdcm::CryptographicMessageSyntax, [279](#)
- AffectedSOPClassUID
  - gdcm::network::CEchoRQ, [232](#)
- AGFA
  - gdcm::EquipmentManufacturer, [431](#)
- ALGOType
  - gdcm::Segment, [956](#)
- ALGOType\_END
  - gdcm::Segment, [957](#)
- Allocate
  - gdcm::LookupTable, [675](#)
- AmbulatoryECGWaveformStorage
  - gdcm::MediaStorage, [693](#)
  - gdcm::UIDs, [1180](#)
- AnatomicRegion
  - gdcm::Segment, [961](#)
- AnatomicRegionModifiers
  - gdcm::Segment, [961](#)
- AnonymizeEvent
  - gdcm::AnonymizeEvent, [105](#)
- Anonymizer
  - gdcm::Anonymizer, [110](#)
- Append
  - gdcm::ByteValue, [221](#)
  - gdcm::Global, [516](#)
- AppendFrameEncode
  - gdcm::ImageCodec, [557](#)
  - gdcm::JPEG2000Codec, [636](#)
  - gdcm::JPEGCodec, [650](#)
  - gdcm::JPEGLSCCodec, [658](#)
  - gdcm::RLECodec, [930](#)
- AppendImplementationClassUID
  - gdcm::FileMetaInformation, [475](#)

- AppendRowEncode
  - gdcm::ImageCodec, 557
  - gdcm::JPEG2000Codec, 636
  - gdcm::JPEGCodec, 650
  - gdcm::JPEGLSCCodec, 658
  - gdcm::RLECodec, 930
- AppendToDataElement
  - gdcm::FileStreamer, 496
- AppendToGroupDataElement
  - gdcm::FileStreamer, 496
- ApplicationContext
  - gdcm::network::ApplicationContext, 117
- Apply
  - gdcm::ImageApplyLookupTable, 539
- ApplyInverseVideo
  - vtkGDCMImageReader, 1317
  - vtkGDCMImageReader2, 1329
- ApplyLookupTable
  - vtkGDCMImageReader, 1317
  - vtkGDCMImageReader2, 1329
- ApplyPlanarConfiguration
  - vtkGDCMImageReader, 1317
  - vtkGDCMImageReader2, 1329
- ApplyShiftScale
  - vtkGDCMImageReader, 1317
  - vtkGDCMImageReader2, 1329
- ApplyYBRToRGB
  - vtkGDCMImageReader, 1317
  - vtkGDCMImageReader2, 1329
- Area
  - gdcm::BoxRegion, 210
  - gdcm::Region, 920
- AReleaseRPPDU
  - gdcm::network::AReleaseRPPDU, 121
- AReleaseRQPDU
  - gdcm::network::AReleaseRQPDU, 124
- AreOverlaysInPixelData
  - gdcm::Bitmap, 196
  - gdcm::Pixmap, 832
- ARGB
  - gdcm::PhotometricInterpretation, 818
- ArrayIncludeMacroType
  - gdcm::Macro, 682
  - gdcm::Module, 719
- ArrayType
  - gdcm::Attribute< Group, Element, TVR, TVM >, 132
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 139
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, 149
- ArterialPulseWaveformStorage
  - gdcm::UIDs, 1184
- ARTIMTimer
  - gdcm::network::ARTIMTimer, 125
- AS
  - gdcm::VR, 1296
- ASComp
  - gdcm, 58
- ASN1
  - gdcm::ASN1, 127
- AsynchronousOperationsWindowSub
  - gdcm::network::AsynchronousOperationsWindowSub, 129
- AT
  - gdcm::VR, 1296
- Attribute
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, 149
  - gdcm::terminal, 82
- Audio
  - gdcm::MediaStorage, 695
- AudioCodec
  - gdcm::AudioCodec, 165
- AudioSRStorageTrialRetired
  - gdcm::UIDs, 1181
- AUTOMATIC
  - gdcm::Segment, 957
- AutoPixelMinMax
  - gdcm::IconImageGenerator, 524
- AutorefractionMeasurementsStorage
  - gdcm::UIDs, 1184
- AXIAL
  - gdcm::Orientation, 777
- backslash
  - gdcm, 63
- BadBigEndian
  - gdcm::SwapCode, 1105
- BadLittleEndian
  - gdcm::SwapCode, 1105
- BALCPPProtect
  - gdcm::Anonymizer, 110
- Base64
  - gdcm::Base64, 167
- BaseQuery
  - gdcm::BaseQuery, 177
- BaseRootQuery
  - gdcm::BaseRootQuery, 182
- BasicAnnotationBoxSOPClass
  - gdcm::UIDs, 1179
- BasicApplicationLevelConfidentialityProfile
  - gdcm::Anonymizer, 110
- BasicCodedEntry
  - gdcm::SegmentHelper::BasicCodedEntry, 186
- BasicCodedEntryVector
  - gdcm::Segment, 956
- BasicColorImageBoxSOPClass
  - gdcm::UIDs, 1179

- BasicColorPrintManagementMetaSOPClass
  - gdcm::UIDs, [1179](#)
- BasicFilmBoxSOPClass
  - gdcm::UIDs, [1179](#)
- BasicFilmSessionSOPClass
  - gdcm::UIDs, [1179](#)
- BasicGrayscaleImageBoxSOPClass
  - gdcm::UIDs, [1179](#)
- BasicGrayscalePrintManagementMetaSOPClass
  - gdcm::UIDs, [1179](#)
- BasicOffsetTable
  - gdcm::BasicOffsetTable, [191](#)
- BasicPrintImageOverlayBoxSOPClassRetired
  - gdcm::UIDs, [1179](#)
- BasicStructuredDisplayStorage
  - gdcm::UIDs, [1185](#)
- BasicStudyContentNotificationSOPClassRetired
  - gdcm::UIDs, [1178](#)
- BasicTextSR
  - gdcm::MediaStorage, [694](#)
- BasicTextSRStorage
  - gdcm::UIDs, [1181](#)
- BasicVoiceAudioWaveformStorage
  - gdcm::MediaStorage, [693](#)
  - gdcm::UIDs, [1180](#)
- Begin
  - gdcm::CSAHeaderDict, [294](#)
  - gdcm::DataSet, [328](#)
  - gdcm::Dict, [358](#)
  - gdcm::IODs, [610](#)
  - gdcm::Scanner, [939](#)
  - gdcm::Scanner2, [948](#)
  - gdcm::SequenceOfFragments, [978](#)
  - gdcm::SequenceOfItems, [986](#)
  - gdcm::StrictScanner, [1052](#)
  - gdcm::StrictScanner2, [1062](#)
- BigEndian
  - gdcm::SwapCode, [1105](#)
- Bitmap
  - gdcm::Bitmap, [195](#)
  - gdcm::JPEG2000Codec, [640](#)
  - gdcm::PixelFormat, [827](#)
- BitmapToBitmapFilter
  - gdcm::BitmapToBitmapFilter, [207](#)
- BitSample
  - gdcm::JPEGCodec, [654](#)
  - gdcm::LookupTable, [679](#)
- black
  - gdcm::terminal, [82](#)
- BlendingSoftcopyPresentationStateStorageSOPClass
  - gdcm::UIDs, [1180](#)
- blink
  - gdcm::terminal, [82](#)
- BLUE
  - gdcm::LookupTable, [674](#)
- blue
  - gdcm::terminal, [82](#)
- BOOL\_FUNCTION\_PFILE\_PFILE\_POINTER
  - gdcm, [58](#)
- BoundingBox
  - gdcm::BoxRegion, [210](#)
- BoxRegion
  - gdcm::BoxRegion, [210](#)
- BreakConnection
  - gdcm::network::ULConnectionManager, [1252](#)
- BreakConnectionNow
  - gdcm::network::ULConnectionManager, [1252](#)
- BreastImagingRelevantPatientInformationQuery
  - gdcm::UIDs, [1182](#)
- BreastProjectionXRayImageStorageForPresentation
  - gdcm::MediaStorage, [695](#)
  - gdcm::UIDs, [1184](#)
- BreastProjectionXRayImageStorageForProcessing
  - gdcm::MediaStorage, [695](#)
  - gdcm::UIDs, [1184](#)
- BreastTomosynthesisImageStorage
  - gdcm::MediaStorage, [694](#)
  - gdcm::UIDs, [1183](#)
- bright
  - gdcm::terminal, [82](#)
- Bug List, [7](#)
- Build
  - vtkLookupTable16, [1395](#)
- ByteBuffer
  - gdcm::ByteBuffer, [214](#)
- bytes
  - gdcm::Tag, [1132](#)
- ByteSwap
  - gdcm::ByteSwapFilter, [217](#)
- ByteSwapFilter
  - gdcm::ByteSwapFilter, [217](#)
- ByteValue
  - gdcm::ByteValue, [220](#)
- C\_CANCEL\_RQ
  - gdcm::network::DIMSE, [376](#)
- C\_ECHO\_RQ
  - gdcm::network::DIMSE, [376](#)
- C\_ECHO\_RSP
  - gdcm::network::DIMSE, [376](#)
- C\_FIND\_RQ
  - gdcm::network::DIMSE, [376](#)
- C\_FIND\_RSP
  - gdcm::network::DIMSE, [376](#)
- C\_GET\_RQ
  - gdcm::network::DIMSE, [376](#)
- C\_GET\_RSP
  - gdcm::network::DIMSE, [376](#)

- C\_MOVE\_RQ
  - gdcm::network::DIMSE, 376
- C\_MOVE\_RSP
  - gdcm::network::DIMSE, 376
- C\_STORE\_RQ
  - gdcm::network::DIMSE, 376
- C\_STORE\_RSP
  - gdcm::network::DIMSE, 376
- CALIBRATED
  - gdcm::Spacing, 1030
- CanCode
  - gdcm::AudioCodec, 165
  - gdcm::Coder, 252
  - gdcm::ImageCodec, 558
  - gdcm::JPEG2000Codec, 637
  - gdcm::JPEGCodec, 650
  - gdcm::JPEGLSCodec, 659
  - gdcm::KAKADUCodec, 667
  - gdcm::PDFCodec, 805
  - gdcm::PGXCodec, 815
  - gdcm::PNMCodec, 850
  - gdcm::PVRGCodec, 889
  - gdcm::RAWCodec, 909
  - gdcm::RLECodec, 930
- CanDecode
  - gdcm::AudioCodec, 165
  - gdcm::Decoder, 342
  - gdcm::DeltaEncodingCodec, 351
  - gdcm::ImageCodec, 558
  - gdcm::JPEG2000Codec, 637
  - gdcm::JPEGCodec, 650
  - gdcm::JPEGLSCodec, 659
  - gdcm::KAKADUCodec, 667
  - gdcm::PDFCodec, 805
  - gdcm::PGXCodec, 815
  - gdcm::PNMCodec, 850
  - gdcm::PVRGCodec, 889
  - gdcm::RAWCodec, 909
  - gdcm::RLECodec, 930
- CanDisplay
  - gdcm::VR, 1297
- CanEmptyTag
  - gdcm::Anonymizer, 110
- CanRead
  - gdcm::Reader, 914
- CanReadFile
  - vtkGDCMImageReader, 1310
  - vtkGDCMImageReader2, 1322
- CanReadImage
  - gdcm::StreamImageReader, 1039
- CanStoreLossy
  - gdcm::TransferSyntax, 1150
- CanWriteFile
  - gdcm::StreamImageWriter, 1044
- CAPI
  - gdcm::CryptoFactory, 277
- CAPICryptoFactory
  - gdcm::CAPICryptoFactory, 227
- CAPICryptographicMessageSyntax
  - gdcm::CAPICryptographicMessageSyntax, 229
- CardiacElectrophysiologyWaveformStorage
  - gdcm::MediaStorage, 693
  - gdcm::UIDs, 1180
- CardiacRelevantPatientInformationQuery
  - gdcm::UIDs, 1182
- CEcho
  - gdcm::CompositeNetworkFunctions, 267
- CFind
  - gdcm::CompositeNetworkFunctions, 268
- Change
  - gdcm::FileChangeTransferSyntax, 460
  - gdcm::FileDecompressLookupTable, 463
  - gdcm::FileExplicitFilter, 469
  - gdcm::ImageChangePhotometricInterpretation, 542
  - gdcm::ImageChangePlanarConfiguration, 547
  - gdcm::ImageChangeTransferSyntax, 551
- ChangeFMI
  - gdcm::FileExplicitFilter, 469
- ChangeMonochrome
  - gdcm::ImageChangePhotometricInterpretation, 542
- ChangeRGB2YBR
  - gdcm::ImageChangePhotometricInterpretation, 542
- ChangeYBR2RGB
  - gdcm::ImageChangePhotometricInterpretation, 542
- CharacterDataHandler
  - gdcm::TableReader, 1119
  - gdcm::XMLDictReader, 1419
  - gdcm::XMLPrivateDictReader, 1425
- CheckDataElement
  - gdcm::FileStreamer, 496
- CheckEvent
  - gdcm::AnonymizeEvent, 105
  - gdcm::DataEvent, 324
  - gdcm::DataSetEvent, 339
  - gdcm::Event, 434
  - gdcm::FileNameEvent, 486
  - gdcm::ProgressEvent, 885
- CheckFileMetaInformationOff
  - gdcm::Writer, 1414
- CheckFileMetaInformationOn
  - gdcm::Writer, 1414
- CheckTemplateFileName
  - gdcm::FileStreamer, 496
- ChestCADSRStorage
  - gdcm::UIDs, 1181
- CipherTypes
  - gdcm::CryptographicMessageSyntax, 279
- Clamp

- gdcM, 63
- Clean
  - gdcM::Cleaner, 242
- clean
  - gdcM, 64
- Cleaner
  - gdcM::Cleaner, 241
- CleanupUnusedBits
  - gdcM::ImageCodec, 558
- Clear
  - gdcM::Anonymizer, 110
  - gdcM::Bitmap, 196
  - gdcM::ByteValue, 221
  - gdcM::DataElement, 312
  - gdcM::DataSet, 329
  - gdcM::IOD, 605
  - gdcM::IODs, 610
  - gdcM::Item, 620
  - gdcM::LookupTable, 675
  - gdcM::Macro, 682
  - gdcM::Macros, 685
  - gdcM::Module, 720
  - gdcM::Modules, 726
  - gdcM::Preamble, 853
  - gdcM::SequenceOfFragments, 978
  - gdcM::SequenceOfItems, 986
  - gdcM::SerieHelper, 993
  - gdcM::Value, 1281
  - vtkGDCMMedicalImageProperties, 1341
  - vtkRTStructSetProperties, 1399
- ClearInternalUIDs
  - gdcM::Anonymizer, 111
- ClearPrivateTags
  - gdcM::Scanner2, 948
  - gdcM::StrictScanner2, 1062
- ClearPublicTags
  - gdcM::Scanner2, 949
  - gdcM::StrictScanner2, 1062
- ClearSkipTags
  - gdcM::Scanner, 939
  - gdcM::Scanner2, 949
  - gdcM::StrictScanner, 1053
  - gdcM::StrictScanner2, 1062
- ClearTags
  - gdcM::Scanner, 939
  - gdcM::StrictScanner, 1053
- Clone
  - gdcM::BoxRegion, 211
  - gdcM::ImageCodec, 558
  - gdcM::JPEG2000Codec, 637
  - gdcM::JPEGCodec, 650
  - gdcM::JPEGLSCodec, 659
  - gdcM::KAKADUCodec, 667
  - gdcM::PGXCodec, 815
  - gdcM::PNMCodec, 850
  - gdcM::PVRGCodec, 889
  - gdcM::RAWCodec, 909
  - gdcM::Region, 920
  - gdcM::RLECodec, 930
- CM
  - gdcM::SegmentHelper::BasicCodedEntry, 187
- cMaxEventID
  - gdcM::network, 80
- cMaxStateID
  - gdcM::network, 80
- CMove
  - gdcM::CompositeNetworkFunctions, 268
- CMYK
  - gdcM::PhotometricInterpretation, 818
- Code
  - gdcM::Coder, 252
  - gdcM::JPEG2000Codec, 637
  - gdcM::JPEGCodec, 650
  - gdcM::JPEGLSCodec, 659
  - gdcM::JSON, 663
  - gdcM::KAKADUCodec, 667
  - gdcM::PVRGCodec, 890
  - gdcM::RAWCodec, 910
  - gdcM::RLECodec, 930
- CodeMeaning
  - gdcM::RealWorldValueMappingContent, 918
- CodeString
  - gdcM::CodeString, 255, 256
- CodeValue
  - gdcM::RealWorldValueMappingContent, 918
- ColonCADSRStorage
  - gdcM::UIDs, 1185
- Color
  - gdcM::terminal, 82
- ColorArray
  - gdcM::SurfaceHelper, 1094
- ColorPaletteQueryRetrieveInformationModelFIND
  - gdcM::UIDs, 1186
- ColorPaletteQueryRetrieveInformationModelGET
  - gdcM::UIDs, 1186
- ColorPaletteQueryRetrieveInformationModelMOVE
  - gdcM::UIDs, 1186
- ColorPaletteStorage
  - gdcM::UIDs, 1186
- ColorSoftcopyPresentationStateStorageSOPClass
  - gdcM::UIDs, 1180
- Command
  - gdcM::Command, 259
- CommandDataSet
  - gdcM::CommandDataSet, 263
- CommandTypes
  - gdcM::network::DIMSE, 376
- Compatible



- gdcmm::VM, [1292](#)
- gdcmm::VR, [1297](#)
- Component
  - gdcmm::PersonName, [811](#)
- CompOperators
  - gdcmm, [60](#)
- CompositeInstanceRetrieveWithoutBulkDataGET
  - gdcmm::UIDs, [1185](#)
- CompositeInstanceRootRetrieveGET
  - gdcmm::UIDs, [1185](#)
- CompositeInstanceRootRetrieveMOVE
  - gdcmm::UIDs, [1185](#)
- CompositingPlanarMPRVolumetricPresentationStateStorage
  - gdcmm::UIDs, [1184](#)
- Comprehensive3DSRStorage
  - gdcmm::UIDs, [1185](#)
- ComprehensiveSR
  - gdcmm::MediaStorage, [694](#)
- ComprehensiveSRStorage
  - gdcmm::UIDs, [1181](#)
- ComprehensiveSRStorageTrialRetired
  - gdcmm::UIDs, [1181](#)
- CompressionTypes
  - vtkGDCMImageWriter, [1334](#)
- Compute
  - gdcmm::EquipmentManufacturer, [432](#)
  - gdcmm::MD5, [688](#)
  - gdcmm::SHA1, [1006](#)
- ComputeBoundingBox
  - gdcmm::BoxRegion, [211](#)
  - gdcmm::Region, [920](#)
- ComputeBufferLength
  - gdcmm::ImageRegionReader, [586](#)
- ComputeByteLength
  - gdcmm::SequenceOfFragments, [978](#)
- ComputeDataElement
  - gdcmm::DataSet, [329](#)
- ComputeDataSetMediaStorageSOPClass
  - gdcmm::FileMetaInformation, [475](#)
- ComputeDataSetTransferSyntax
  - gdcmm::FileMetaInformation, [476](#)
- ComputeDistAlongNormal
  - gdcmm::DirectionCosines, [378](#)
- ComputedRadiographyImageStorage
  - gdcmm::MediaStorage, [693](#)
  - gdcmm::UIDs, [1179](#)
- ComputeFile
  - gdcmm::MD5, [688](#)
  - gdcmm::SHA1, [1006](#)
- ComputeFileMD5
  - gdcmm::Testing, [1136](#)
- ComputeGroupLength
  - gdcmm::DataSet, [329](#)
- ComputeInterceptSlopePixelType
  - gdcmm::Rescaler, [923](#)
- ComputeLength
  - gdcmm::ByteValue, [221](#)
  - gdcmm::Fragment, [513](#)
  - gdcmm::SequenceOfFragments, [978](#)
  - gdcmm::SequenceOfItems, [986](#)
- ComputeLossyFlag
  - gdcmm::Bitmap, [196](#)
- ComputeMD5
  - gdcmm::Testing, [1136](#)
- ComputeMediaStorageFromModality
  - gdcmm::ImageHelper, [572](#)
- ComputeMOSAICDimensions
  - gdcmm::SplitMosaicFilter, [1033](#)
- ComputeMOSAICImagePositionPatient
  - gdcmm::SplitMosaicFilter, [1033](#)
- ComputeMOSAICSliceNormal
  - gdcmm::SplitMosaicFilter, [1033](#)
- ComputeMOSAICSlicePosition
  - gdcmm::SplitMosaicFilter, [1033](#)
- ComputeNumberOfSurfaces
  - gdcmm::SurfaceWriter, [1103](#)
- ComputeOffsetTable
  - gdcmm::JPEGCodec, [651](#)
- ComputePixelAspectRatioFromPixelSpacing
  - gdcmm::Spacing, [1031](#)
- ComputePixelTypeFromMinMax
  - gdcmm::Rescaler, [923](#)
- ComputeSpacingFromImagePositionPatient
  - gdcmm::ImageHelper, [572](#)
- ComputeTargetMediaStorage
  - gdcmm::ImageWriter, [593](#)
- ComputeVR
  - gdcmm::DataSetHelper, [341](#)
- ComputeZSpacing
  - gdcmm::IPPSorter, [615](#)
- ConcatenatePDVBlobs
  - gdcmm::network::PresentationDataValue, [868](#)
- ConcatenatePDVBlobsAsExplicit
  - gdcmm::network::PresentationDataValue, [868](#)
- CONDENSED\_STYLE
  - gdcmm::Printer, [873](#)
- Conditional
  - gdcmm::Usage, [1271](#)
- CONSOLE
  - gdcmm::terminal, [82](#)
- const
  - gdcmm::SOPClassUIDToIOD, [1023](#)
- const\_iterator
  - gdcmm::CodeString, [254](#)
  - gdcmm::LO, [669](#)
  - gdcmm::String< TDelimiter, TMaxLength, TPadChar >, [1069](#)
- const\_reference



- gdcmm::CodeString, 254
- gdcmm::LO, 669
- gdcmm::String< TDelimiter, TMaxLength, TPadChar  
>, 1069
- const\_reverse\_iterator
  - gdcmm::CodeString, 254
  - gdcmm::LO, 669
  - gdcmm::String< TDelimiter, TMaxLength, TPadChar  
>, 1069
- ConstCharWrapper
  - gdcmm::ConstCharWrapper, 271
- ConstIterator
  - gdcmm::CSAHeaderDict, 294
  - gdcmm::DataSet, 328
  - gdcmm::Dict, 357
  - gdcmm::Scanner, 938
  - gdcmm::SequenceOfFragments, 977
  - gdcmm::SequenceOfItems, 985
  - gdcmm::StrictScanner, 1051
- Construct
  - gdcmm::BaseRootQuery, 183
- ConstructAbortPDU
  - gdcmm::network::PDUFactory, 807
- ConstructCEchoRQ
  - gdcmm::network::CompositeMessageFactory, 265
- ConstructCFindRQ
  - gdcmm::network::CompositeMessageFactory, 265
- ConstructCMoveRQ
  - gdcmm::network::CompositeMessageFactory, 265
- ConstructCStoreRQ
  - gdcmm::network::CompositeMessageFactory, 265
- ConstructCStoreRSP
  - gdcmm::network::CompositeMessageFactory, 266
- ConstructFromString
  - gdcmm::DPath, 387
  - gdcmm::TagPath, 1133
- ConstructFromTagList
  - gdcmm::TagPath, 1133
- ConstructNAction
  - gdcmm::network::NormalizedMessageFactory, 756
- ConstructNCreate
  - gdcmm::network::NormalizedMessageFactory, 756
- ConstructNDelete
  - gdcmm::network::NormalizedMessageFactory, 757
- ConstructNEventReport
  - gdcmm::network::NormalizedMessageFactory, 757
- ConstructNGet
  - gdcmm::network::NormalizedMessageFactory, 757
- ConstructNSet
  - gdcmm::network::NormalizedMessageFactory, 757
- ConstructorType
  - gdcmm::Dicts, 372
- ConstructPDU
  - gdcmm::network::PDUFactory, 807
- ConstructPDV
  - gdcmm::network::BaseCompositeMessage, 170
  - gdcmm::network::BaseNormalizedMessage, 172
  - gdcmm::network::CEchoRQ, 232
  - gdcmm::network::CFindRQ, 237
  - gdcmm::network::CMoveRQ, 248
  - gdcmm::network::CStoreRQ, 301
  - gdcmm::network::CStoreRSP, 302
  - gdcmm::network::NActionRQ, 738
  - gdcmm::network::NCreateRQ, 741
  - gdcmm::network::NDeleteRQ, 744
  - gdcmm::network::NEventReportRQ, 750
  - gdcmm::network::NGetRQ, 753
  - gdcmm::network::NSetRQ, 761
- ConstructPDVByDataSet
  - gdcmm::network::CEchoRSP, 234
  - gdcmm::network::CFindCancelRQ, 235
  - gdcmm::network::CFindRSP, 238
  - gdcmm::network::CMoveCancelRq, 247
  - gdcmm::network::CMoveRSP, 249
  - gdcmm::network::NActionRSP, 739
  - gdcmm::network::NCreateRSP, 742
  - gdcmm::network::NDeleteRSP, 745
  - gdcmm::network::NEventReportRSP, 752
  - gdcmm::network::NGetRSP, 755
  - gdcmm::network::NSetRSP, 762
- ConstructQuery
  - gdcmm::CompositeNetworkFunctions, 269
  - gdcmm::NormalizedNetworkFunctions, 758
- ConstructReleasePDU
  - gdcmm::network::PDUFactory, 807
- ContentAssessmentResultsStorage
  - gdcmm::UIDs, 1185
- Convert
  - gdcmm::DictConverter, 362
  - gdcmm::ImageConverter, 566
- ConvertRGBToPaletteColor
  - gdcmm::IconImageGenerator, 524
- ConvertToCXX
  - gdcmm::DictConverter, 362
- ConvertToUNC
  - gdcmm::System, 1109
- ConvertToXML
  - gdcmm::DictConverter, 362
- CornealTopographyMapStorage
  - gdcmm::UIDs, 1185
- CORONAL
  - gdcmm::Orientation, 777
- Create
  - gdcmm::Preamble, 853
- CreateCEchoPDU
  - gdcmm::network::PDUFactory, 807
- CreateCFindPDU
  - gdcmm::network::PDUFactory, 807

- CreateCMovePDU
  - gdcm::network::PDUFactory, [807](#)
- CreateCMSProvider
  - gdcm::CAPICryptoFactory, [227](#)
  - gdcm::CryptoFactory, [277](#)
  - gdcm::OpenSSLCryptoFactory, [767](#)
  - gdcm::OpenSSL7CryptoFactory, [772](#)
- CreateCStoreRQPDU
  - gdcm::network::PDUFactory, [807](#)
- CreateCStoreRSPPDU
  - gdcm::network::PDUFactory, [808](#)
- CreateDefaultUniqueSeriesIdentifier
  - gdcm::SerieHelper, [993](#)
- CreateNActionPDU
  - gdcm::network::PDUFactory, [808](#)
- CreateNCreatePDU
  - gdcm::network::PDUFactory, [808](#)
- CreateNDeletePDU
  - gdcm::network::PDUFactory, [808](#)
- CreateNEventReportPDU
  - gdcm::network::PDUFactory, [808](#)
- CreateNGetPDU
  - gdcm::network::PDUFactory, [808](#)
- CreateNSetPDU
  - gdcm::network::PDUFactory, [808](#)
- CreateUniqueSeriesIdentifier
  - gdcm::SerieHelper, [993](#)
- Cross
  - gdcm::DirectionCosines, [378](#)
- CrossDot
  - gdcm::DirectionCosines, [378](#)
- CryptoFactory
  - gdcm::CryptoFactory, [277](#)
- CryptographicMessageSyntax
  - gdcm::CryptographicMessageSyntax, [279](#)
- CryptoLib
  - gdcm::CryptoFactory, [276](#)
- CS
  - gdcm::VR, [1296](#)
- CSAElement
  - gdcm::CSAElement, [283](#)
- CSAHeader
  - gdcm::CSAHeader, [290](#)
  - gdcm::DataSet, [336](#)
- CSAHeaderDict
  - gdcm::CSAHeaderDict, [294](#)
- CSAHeaderDictEntry
  - gdcm::CSAHeaderDictEntry, [297](#)
- CSAHeaderType
  - gdcm::CSAHeader, [289](#)
- CSANonImageStorage
  - gdcm::MediaStorage, [694](#)
- CSComp
  - gdcm, [58](#)
- CSD
  - gdcm::SegmentHelper::BasicCodedEntry, [187](#)
- CStore
  - gdcm::CompositeNetworkFunctions, [270](#)
- CSV
  - gdcm::SegmentHelper::BasicCodedEntry, [187](#)
- CT\_private\_ELE
  - gdcm::TransferSyntax, [1149](#)
- CTDefinedProcedureProtocolStorage
  - gdcm::UIDs, [1185](#)
- CTImageStorage
  - gdcm::MediaStorage, [693](#)
  - gdcm::UIDs, [1179](#)
- CTPerformedProcedureProtocolStorage
  - gdcm::UIDs, [1185](#)
- Curve
  - gdcm::Curve, [305](#)
  - vtkGDCMImageReader, [1317](#)
  - vtkGDCMImageReader2, [1329](#)
- Curves
  - gdcm::Pixmap, [834](#)
- CV
  - gdcm::SegmentHelper::BasicCodedEntry, [187](#)
- CXX
  - gdcm::Printer, [873](#)
- cyan
  - gdcm::terminal, [82](#)
- DA
  - gdcm::VR, [1296](#)
- DAComp
  - gdcm, [58](#)
- DataElement
  - gdcm::DataElement, [311](#)
  - gdcm::Value, [1282](#)
- DataElementSet
  - gdcm::DataSet, [328](#)
- DataElementType
  - gdcm::ModuleEntry, [724](#)
- DataEvent
  - gdcm::DataEvent, [323](#), [324](#)
- DataField
  - gdcm::CSAElement, [287](#)
- DataPtr
  - gdcm::CSAElement, [283](#)
- DATASET\_FORMAT
  - gdcm::CSAHeader, [290](#)
- DataSetEvent
  - gdcm::DataSetEvent, [339](#)
- DataSetHandled
  - gdcm::network::ULConnectionCallback, [1246](#)
- DataSetHandles
  - gdcm::network::ULConnectionCallback, [1246](#)
- DataSetMS

- gdcM::FileMetaInformation, [480](#)
- DataSetTS
  - gdcM::FileMetaInformation, [480](#)
- DataWasPassed
  - vtkImageMapToColors16, [1383](#)
- dCor
  - gdcM::MrProtocol::Vector3, [1283](#)
- DebugOff
  - gdcM::Trace, [1143](#)
- DebugOn
  - gdcM::Trace, [1143](#)
- Decode
  - gdcM::AudioCodec, [166](#)
  - gdcM::Base64, [167](#)
  - gdcM::Curve, [305](#)
  - gdcM::Decoder, [342](#)
  - gdcM::DeltaEncodingCodec, [351](#)
  - gdcM::ImageCodec, [558](#)
  - gdcM::JPEG2000Codec, [637](#)
  - gdcM::JPEGCodec, [651](#)
  - gdcM::JPEGLSCodec, [659](#), [660](#)
  - gdcM::JSON, [663](#)
  - gdcM::KAKADUCodec, [667](#)
  - gdcM::LookupTable, [675](#)
  - gdcM::PDFCodec, [805](#)
  - gdcM::PVRGCodec, [890](#)
  - gdcM::RAWCodec, [910](#)
  - gdcM::RLECodec, [931](#)
- Decode8
  - gdcM::LookupTable, [675](#)
- DecodeByStreams
  - gdcM::Decoder, [342](#)
  - gdcM::ImageCodec, [559](#)
  - gdcM::JPEG12Codec, [627](#)
  - gdcM::JPEG16Codec, [632](#)
  - gdcM::JPEG2000Codec, [638](#)
  - gdcM::JPEG8Codec, [644](#)
  - gdcM::JPEGCodec, [651](#)
  - gdcM::RAWCodec, [910](#)
  - gdcM::RLECodec, [931](#)
- DecodeBytes
  - gdcM::RAWCodec, [910](#)
- DecodeExtent
  - gdcM::JPEG2000Codec, [638](#)
  - gdcM::JPEGCodec, [651](#)
  - gdcM::JPEGLSCodec, [660](#)
  - gdcM::RLECodec, [931](#)
- Decompress
  - gdcM::Overlay, [783](#)
- Decrypt
  - gdcM::CAPICryptographicMessageSyntax, [229](#)
  - gdcM::CryptographicMessageSyntax, [279](#)
  - gdcM::OpenSSLCryptographicMessageSyntax, [769](#)
  - gdcM::OpenSSLP7CryptographicMessageSyntax, [774](#)
- DeepCopy
  - vtkRTStructSetProperties, [1399](#)
- DEFAULT
  - gdcM::CryptoFactory, [277](#)
- Default
  - gdcM::FileMetaInformation, [476](#)
- DefinedProcedureProtocolInformationModelFIND
  - gdcM::UIDs, [1185](#)
- DefinedProcedureProtocolInformationModelGET
  - gdcM::UIDs, [1185](#)
- DefinedProcedureProtocolInformationModelMOVE
  - gdcM::UIDs, [1185](#)
- DefinedTerms
  - gdcM::DefinedTerms, [343](#)
- DefinePixelExtent
  - gdcM::StreamImageReader, [1039](#)
  - gdcM::StreamImageWriter, [1044](#)
- DefineProperBufferLength
  - gdcM::StreamImageReader, [1039](#)
  - gdcM::StreamImageWriter, [1044](#)
- DeflatedExplicitVRLittleEndian
  - gdcM::TransferSyntax, [1149](#)
  - gdcM::UIDs, [1177](#)
- DeformableSpatialRegistrationStorage
  - gdcM::UIDs, [1180](#)
- Defs
  - gdcM::Defs, [345](#)
- DeleteDirectory
  - gdcM::System, [1109](#)
- DeltaEncodingCodec
  - gdcM::DeltaEncodingCodec, [351](#)
- Deprecated List, [5](#)
- Derive
  - gdcM::FileDerivation, [466](#)
- DES3\_CIPHER
  - gdcM::CryptographicMessageSyntax, [279](#)
- Description
  - gdcM::ModuleEntry, [723](#)
- DescriptionField
  - gdcM::ModuleEntry, [724](#)
- DetachedInterpretationManagementSOPClassRetired
  - gdcM::UIDs, [1179](#)
- DetachedPatientManagementMetaSOPClassRetired
  - gdcM::UIDs, [1178](#)
- DetachedPatientManagementSOPClass
  - gdcM::MediaStorage, [694](#)
- DetachedPatientManagementSOPClassRetired
  - gdcM::UIDs, [1178](#)
- DetachedResultsManagementMetaSOPClassRetired
  - gdcM::UIDs, [1179](#)
- DetachedResultsManagementSOPClassRetired
  - gdcM::UIDs, [1179](#)

DetachedStudyManagementMetaSOPClassRetired  
     gdcM::UIDs, [1179](#)  
 DetachedStudyManagementSOPClass  
     gdcM::MediaStorage, [694](#)  
 DetachedStudyManagementSOPClassRetired  
     gdcM::UIDs, [1178](#)  
 DetachedVisitManagementSOPClass  
     gdcM::MediaStorage, [694](#)  
 DetachedVisitManagementSOPClassRetired  
     gdcM::UIDs, [1178](#)  
 DetailSRStorageTrialRetired  
     gdcM::UIDs, [1181](#)  
 DETECTOR  
     gdcM::Spacing, [1030](#)  
 DetermineEventByPDU  
     gdcM::network::PDUFactory, [809](#)  
 dicomAETitle  
     gdcM::UIDs, [1182](#)  
 dicomApplicationCluster  
     gdcM::UIDs, [1182](#)  
 DICOMApplicationContextName  
     gdcM::UIDs, [1178](#)  
 dicomAssociationAcceptor  
     gdcM::UIDs, [1182](#)  
 dicomAssociationInitiator  
     gdcM::UIDs, [1182](#)  
 dicomAuthorizedNodeCertificateReference  
     gdcM::UIDs, [1183](#)  
 dicomConfigurationRoot  
     gdcM::UIDs, [1183](#)  
 DICOMContentMappingResource  
     gdcM::UIDs, [1186](#)  
 DICOMControlledTerminology  
     gdcM::UIDs, [1178](#)  
 dicomDescription  
     gdcM::UIDs, [1182](#)  
 dicomDevice  
     gdcM::UIDs, [1183](#)  
 dicomDeviceName  
     gdcM::UIDs, [1182](#)  
 dicomDeviceSerialNumber  
     gdcM::UIDs, [1183](#)  
 dicomDevicesRoot  
     gdcM::UIDs, [1183](#)  
 DICOMDIR  
     gdcM::DICOMDIR, [352](#)  
 DICOMDIRGenerator  
     gdcM::DICOMDIRGenerator, [354](#)  
 dicomHostname  
     gdcM::UIDs, [1182](#)  
 dicomInstalled  
     gdcM::UIDs, [1183](#)  
 dicomInstitutionAddress  
     gdcM::UIDs, [1183](#)  
 dicomInstitutionDepartmentName  
     gdcM::UIDs, [1183](#)  
 dicomInstitutionName  
     gdcM::UIDs, [1183](#)  
 dicomIssuerOfPatientID  
     gdcM::UIDs, [1183](#)  
 dicomManufacturer  
     gdcM::UIDs, [1182](#)  
 dicomManufacturerModelName  
     gdcM::UIDs, [1182](#)  
 dicomNetworkAE  
     gdcM::UIDs, [1183](#)  
 dicomNetworkConnection  
     gdcM::UIDs, [1183](#)  
 dicomNetworkConnectionReference  
     gdcM::UIDs, [1182](#)  
 dicomPort  
     gdcM::UIDs, [1182](#)  
 dicomPreferredCalledAETitle  
     gdcM::UIDs, [1182](#)  
 dicomPreferredCallingAETitle  
     gdcM::UIDs, [1183](#)  
 dicomPrimaryDeviceType  
     gdcM::UIDs, [1182](#)  
 dicomRelatedDeviceReference  
     gdcM::UIDs, [1182](#)  
 dicomSoftwareVersion  
     gdcM::UIDs, [1182](#)  
 dicomSOPClass  
     gdcM::UIDs, [1182](#)  
 dicomStationName  
     gdcM::UIDs, [1183](#)  
 dicomSupportedCharacterSet  
     gdcM::UIDs, [1183](#)  
 dicomThisNodeCertificateReference  
     gdcM::UIDs, [1183](#)  
 dicomTLSCyphersuite  
     gdcM::UIDs, [1183](#)  
 dicomTransferCapability  
     gdcM::UIDs, [1183](#)  
 dicomTransferRole  
     gdcM::UIDs, [1182](#)  
 dicomTransferSyntax  
     gdcM::UIDs, [1182](#)  
 DICOMUIDRegistry  
     gdcM::UIDs, [1178](#)  
 dicomUniqueAETitle  
     gdcM::UIDs, [1183](#)  
 dicomUniqueAETitlesRegistryRoot  
     gdcM::UIDs, [1183](#)  
 dicomVendorData  
     gdcM::UIDs, [1182](#)  
 DICOS2DAITStorage  
     gdcM::UIDs, [1185](#)

- DICOS3DAITStorage
  - gdcm::UIDs, [1185](#)
- DICOSCTImageStorage
  - gdcm::UIDs, [1185](#)
- DICOSDigitalXRayImageStorageForPresentation
  - gdcm::UIDs, [1185](#)
- DICOSDigitalXRayImageStorageForProcessing
  - gdcm::UIDs, [1185](#)
- DICOSQuadrupoleResonanceQRStorage
  - gdcm::UIDs, [1185](#)
- DICOSThreatDetectionReportStorage
  - gdcm::UIDs, [1185](#)
- Dict
  - gdcm::Dict, [357](#)
  - gdcm::DictEntry, [368](#)
- DICT\_DEBUG
  - gdcm::DictConverter, [361](#)
- DICT\_DEFAULT
  - gdcm::DictConverter, [361](#)
- DICT\_XML
  - gdcm::DictConverter, [361](#)
- DictConverter
  - gdcm::DictConverter, [361](#)
- DictEntry
  - gdcm::DictEntry, [365](#)
- DictPrinter
  - gdcm::DictPrinter, [370](#)
- Dicts
  - gdcm::CSAHeaderDict, [295](#)
  - gdcm::Dict, [360](#)
  - gdcm::Dicts, [373](#)
  - gdcm::PrivateDict, [877](#)
- difference\_type
  - gdcm::CodeString, [254](#)
  - gdcm::LO, [669](#)
  - gdcm::String< TDelimiter, TMaxLength, TPadChar >, [1069](#)
- DigitalIntraoralXRayImageStorageForPresentation
  - gdcm::UIDs, [1179](#)
- DigitalIntraoralXrayImageStorageForPresentation
  - gdcm::MediaStorage, [693](#)
- DigitalIntraoralXRayImageStorageForProcessing
  - gdcm::MediaStorage, [693](#)
  - gdcm::UIDs, [1179](#)
- DigitalMammographyImageStorageForPresentation
  - gdcm::MediaStorage, [693](#)
- DigitalMammographyImageStorageForProcessing
  - gdcm::MediaStorage, [693](#)
- DigitalMammographyXRayImageStorageForPresentation
  - gdcm::UIDs, [1179](#)
- DigitalMammographyXRayImageStorageForProcessing
  - gdcm::UIDs, [1179](#)
- DigitalXRayImageStorageForPresentation
  - gdcm::MediaStorage, [693](#)
- gdcm::UIDs, [1179](#)
- DigitalXRayImageStorageForProcessing
  - gdcm::MediaStorage, [693](#)
  - gdcm::UIDs, [1179](#)
- dim
  - gdcm::terminal, [82](#)
- Dimensions
  - gdcm::Bitmap, [205](#)
  - gdcm::ImageCodec, [564](#)
- DirCosTolerance
  - gdcm::IPPSorter, [615](#)
- DirectionCosines
  - gdcm::DirectionCosines, [377](#)
  - vtkGDCMImageReader, [1317](#)
  - vtkGDCMImageReader2, [1330](#)
- Directory
  - gdcm::Directory, [381](#)
- DisplaySystemSOPClass
  - gdcm::UIDs, [1184](#)
- DisplaySystemSOPInstance
  - gdcm::UIDs, [1184](#)
- DoByteSwap
  - gdcm::ImageCodec, [559](#)
- DolconImage
  - gdcm::PixmapWriter, [845](#)
- DoInvertMonochrome
  - gdcm::ImageCodec, [559](#)
- DoOverlayCleanup
  - gdcm::ImageCodec, [559](#)
- DoPaddedCompositePixelCode
  - gdcm::ImageCodec, [559](#)
- DoPlanarConfiguration
  - gdcm::ImageCodec, [559](#)
- doround
  - gdcm, [64](#)
- DoSimpleCopy
  - gdcm::ImageCodec, [560](#)
- Dot
  - gdcm::DirectionCosines, [378](#)
- DoYBR
  - gdcm::ImageCodec, [560](#)
- DoYBRFull422
  - gdcm::ImageCodec, [560](#)
- DPath
  - gdcm::DPath, [386](#)
- DropDuplicatePositions
  - gdcm::IPPSorter, [615](#)
- DS
  - gdcm::VR, [1296](#)
- dSag
  - gdcm::MrProtocol::Vector3, [1283](#)
- DT
  - gdcm::VR, [1296](#)
- DTComp

- gdcM, [58](#)
- dTra
  - gdcM::MrProtocol::Vector3, [1283](#)
- Dumper
  - gdcM::Dumper, [391](#)
- DuplicateAttributeError
  - gdcM::Parser, [792](#)
- eAABORTPDURceivedOpen
  - gdcM::network, [79](#)
- eAABORTRequest
  - gdcM::network, [79](#)
- eAASSOCIATE\_RQPDURceived
  - gdcM::network, [79](#)
- eAASSOCIATERequestLocalUser
  - gdcM::network, [79](#)
- eAASSOCIATEResponseAccept
  - gdcM::network, [79](#)
- eAASSOCIATEResponseReject
  - gdcM::network, [79](#)
- eArabic
  - gdcM, [62](#)
- eARELEASE\_RPPDURceived
  - gdcM::network, [79](#)
- eARELEASE\_RQPDURceivedOpen
  - gdcM::network, [79](#)
- eARELEASERequest
  - gdcM::network, [79](#)
- eARELEASEResponse
  - gdcM::network, [79](#)
- eARTIMTimerExpired
  - gdcM::network, [79](#)
- eASSOCIATE\_ACPDURceived
  - gdcM::network, [79](#)
- eASSOCIATE\_RJPDURceived
  - gdcM::network, [79](#)
- ECG12leadWaveformStorage
  - gdcM::UIDs, [1180](#)
- ECharSet
  - gdcM, [60](#)
- eCreateMMPS
  - gdcM, [62](#)
- eCyrillic
  - gdcM, [62](#)
- EddyCurrentImageStorage
  - gdcM::UIDs, [1185](#)
- EddyCurrentMultiframeImageStorage
  - gdcM::UIDs, [1185](#)
- EDGE
  - gdcM::MeshPrimitive, [707](#)
- eEventDoesNotExist
  - gdcM::network, [79](#)
- EEventID
  - gdcM::network, [79](#)
- eFind
  - gdcM, [63](#)
- eGB18030
  - gdcM, [62](#)
- eGreek
  - gdcM, [62](#)
- eHebrew
  - gdcM, [62](#)
- eImage
  - gdcM, [62](#)
- eJapanese
  - gdcM, [62](#)
- eJapaneseKanjiMultibyte
  - gdcM, [62](#)
- eJapaneseSupplementaryKanjiMultibyte
  - gdcM, [62](#)
- eKoreanHangulHanjaMultibyte
  - gdcM, [62](#)
- eLatin1
  - gdcM, [62](#)
- eLatin2
  - gdcM, [62](#)
- eLatin3
  - gdcM, [62](#)
- eLatin4
  - gdcM, [62](#)
- eLatin5
  - gdcM, [62](#)
- elem
  - gdcM::SerieHelper, [994](#)
- Element
  - gdcM::Element< TVR, VM::VM1\_n >, [400](#)
- eMove
  - gdcM, [63](#)
- Empty
  - gdcM::Anonymizer, [111](#)
  - gdcM::BoxRegion, [211](#)
  - gdcM::Cleaner, [242](#)
  - gdcM::DataElement, [312](#)
  - gdcM::FileAnonymizer, [455](#)
  - gdcM::Region, [920](#)
- EmptyMaskGenerator
  - gdcM::EmptyMaskGenerator, [424](#)
- EmptyWhenScrubFails
  - gdcM::Cleaner, [242](#)
- EncapsulatedCDASStorage
  - gdcM::MediaStorage, [694](#)
  - gdcM::UIDs, [1181](#)
- EncapsulatedDocument
  - gdcM::EncapsulatedDocument, [425](#)
- EncapsulatedPDFStorage
  - gdcM::MediaStorage, [694](#)
  - gdcM::UIDs, [1181](#)
- EncapsulatedSTLStorage

- gdcmm::UIDs, 1185
- Encode
  - gdcmm::Base64, 167
- EncodeBuffer
  - gdcmm::JPEG12Codec, 627
  - gdcmm::JPEG16Codec, 632
  - gdcmm::JPEG8Codec, 644
  - gdcmm::JPEGCodec, 651
- EncodeBytes
  - gdcmm::System, 1109
- Encrypt
  - gdcmm::CAPICryptographicMessageSyntax, 229
  - gdcmm::CryptographicMessageSyntax, 279
  - gdcmm::OpenSSLCryptographicMessageSyntax, 769
  - gdcmm::OpenSSL7CryptographicMessageSyntax, 774
- End
  - gdcmm::CSAHeaderDict, 295
  - gdcmm::DataSet, 329
  - gdcmm::Dict, 358
  - gdcmm::IODs, 610
  - gdcmm::Scanner, 939
  - gdcmm::Scanner2, 949
  - gdcmm::SequenceOfFragments, 978
  - gdcmm::SequenceOfItems, 986
  - gdcmm::StrictScanner, 1053
  - gdcmm::StrictScanner2, 1062
- EndElement
  - gdcmm::TableReader, 1119
  - gdcmm::XMLDictReader, 1419
  - gdcmm::XMLPrivateDictReader, 1425
- EndElementHandler
  - gdcmm::Parser, 792
- EndFilter
  - gdcmm::SimpleSubjectWatcher, 1014
- EndWith
  - gdcmm::Filename, 482
- EnhancedCTImageStorage
  - gdcmm::MediaStorage, 693
  - gdcmm::UIDs, 1179
- EnhancedMRColorImageStorage
  - gdcmm::MediaStorage, 695
  - gdcmm::UIDs, 1186
- EnhancedMRImageStorage
  - gdcmm::MediaStorage, 693
  - gdcmm::UIDs, 1180
- EnhancedPETImageStorage
  - gdcmm::MediaStorage, 694
  - gdcmm::UIDs, 1185
- EnhancedSR
  - gdcmm::MediaStorage, 694
- EnhancedSRStorage
  - gdcmm::UIDs, 1181
- EnhancedUSVolumeStorage
  - gdcmm::MediaStorage, 694
  - gdcmm::UIDs, 1183
- EnhancedXAImageStorage
  - gdcmm::MediaStorage, 694
  - gdcmm::UIDs, 1180
- EnhancedXRImageStorage
  - gdcmm::UIDs, 1180
- ENQueryType
  - gdcmm, 62
- EnumeratedValues
  - gdcmm::EnumeratedValues, 430
- ePatient
  - gdcmm, 62
- ePatientRootType
  - gdcmm, 63
- ePDATArequest
  - gdcmm::network, 79
- ePDATATFPDU
  - gdcmm::network, 79
- EQueryLevel
  - gdcmm, 62
- EQueryType
  - gdcmm, 62
- ERootType
  - gdcmm, 63
- ErrorOff
  - gdcmm::Trace, 1143
- ErrorOn
  - gdcmm::Trace, 1143
- ErrorType
  - gdcmm::Parser, 792
- eSeries
  - gdcmm, 62
- eSetMMPS
  - gdcmm, 62
- eSta10ReleaseCollisionAc
  - gdcmm::network, 80
- eSta11ReleaseCollisionRq
  - gdcmm::network, 80
- eSta12ReleaseCollisionAcLocal
  - gdcmm::network, 80
- eSta13AwaitingClose
  - gdcmm::network, 80
- eSta1Idle
  - gdcmm::network, 80
- eSta2Open
  - gdcmm::network, 80
- eSta3WaitLocalAssoc
  - gdcmm::network, 80
- eSta4LocalAssocDone
  - gdcmm::network, 80
- eSta5WaitRemoteAssoc
  - gdcmm::network, 80
- eSta6TransferReady



- gdcmm::network, 80
- eSta7WaitRelease
  - gdcmm::network, 80
- eSta8WaitLocalRelease
  - gdcmm::network, 80
- eSta9ReleaseCollisionRqLocal
  - gdcmm::network, 80
- EstablishConnection
  - gdcmm::network::ULConnectionManager, 1252
- EstablishConnectionMove
  - gdcmm::network::ULConnectionManager, 1252
- eStaDoesNotExist
  - gdcmm::network, 80
- EStateID
  - gdcmm::network, 79
- eStudy
  - gdcmm, 62
- eStudyRootType
  - gdcmm, 63
- eThai
  - gdcmm, 62
- eTransportConnConfirmLocal
  - gdcmm::network, 79
- eTransportConnectionClosed
  - gdcmm::network, 79
- eTransportConnIndicLocal
  - gdcmm::network, 79
- eUnrecognizedPDURceived
  - gdcmm::network, 79
- eUTF8
  - gdcmm, 62
- Event
  - gdcmm::Event, 434
- eWLMFind
  - gdcmm, 63
- Exception
  - gdcmm::Exception, 436
- Execute
  - gdcmm::Command, 259
  - gdcmm::EmptyMaskGenerator, 424
  - gdcmm::MemberCommand< T >, 703
  - gdcmm::SimpleMemberCommand< T >, 1011
- ExecuteData
  - vtkGDCMImageReader, 1310
  - vtkGDCMThreadedImageReader, 1357
- ExecuteInformation
  - vtkGDCMImageReader, 1310
  - vtkGDCMThreadedImageReader, 1357
- ExecuteQuery
  - gdcmm::StringFilter, 1074
- Explicit
  - gdcmm::TransferSyntax, 1148
- ExplicitVRBigEndian
  - gdcmm::TransferSyntax, 1149
- gdcmm::UIDs, 1177
- ExplicitVRLittleEndian
  - gdcmm::TransferSyntax, 1149
  - gdcmm::UIDs, 1177
- Explore
  - gdcmm::Directory, 382
- ExtensibleSRStorage
  - gdcmm::UIDs, 1185
- Extract
  - gdcmm::IconImageFilter, 522
- ExtractIconImages
  - gdcmm::IconImageFilter, 522
- ExtractVeprolIconImages
  - gdcmm::IconImageFilter, 522
- F
  - gdcmm::Printer, 874
  - gdcmm::Reader, 917
  - gdcmm::Validate, 1279
  - gdcmm::XMLPrinter, 1422
- FACET
  - gdcmm::MeshPrimitive, 707
- FallColorPaletteSOPInstance
  - gdcmm::UIDs, 1183
- FD
  - gdcmm::VR, 1296
- Fiducials
  - gdcmm::Fiducials, 447
- File
  - gdcmm::File, 450
- FileAnonymizer
  - gdcmm::FileAnonymizer, 455
- FileChangeTransferSyntax
  - gdcmm::FileChangeTransferSyntax, 459
  - gdcmm::ImageCodec, 564
- FileDecompressLookupTable
  - gdcmm::FileDecompressLookupTable, 463
- FileDerivation
  - gdcmm::FileDerivation, 465
- FileExists
  - gdcmm::System, 1109
- FileExplicitFilter
  - gdcmm::FileExplicitFilter, 469
- FilesDirectory
  - gdcmm::System, 1109
- FilesSymlink
  - gdcmm::System, 1110
- FileList
  - gdcmm, 59
- FileMetaInformation
  - gdcmm::FileMetaInformation, 475
- FileName
  - vtkGDCMPolyDataReader, 1346
- Filename



- gdcm::Filename, [482](#)
- filename
  - gdcm::FileWithName, [501](#)
- FileNameEvent
  - gdcm::FileNameEvent, [486](#)
- FilenameGenerator
  - gdcm::FilenameGenerator, [489](#)
- FileNameOrdering
  - gdcm::SerieHelper, [993](#)
- FileNames
  - vtkGDCMImageReader, [1318](#)
- FileNames
  - gdcm::Sorter, [1028](#)
- FileNamesType
  - gdcm::DICOmdirGenerator, [354](#)
  - gdcm::Directory, [381](#)
  - gdcm::FilenameGenerator, [488](#)
- FilenameType
  - gdcm::DICOmdirGenerator, [354](#)
  - gdcm::Directory, [381](#)
  - gdcm::FilenameGenerator, [488](#)
- FileSet
  - gdcm::FileSet, [492](#)
- FileSize
  - gdcm::System, [1110](#)
- FileStreamer
  - gdcm::FileStreamer, [496](#)
- FileType
  - gdcm::FileSet, [491](#)
- FileTime
  - gdcm::System, [1110](#)
- FileType
  - gdcm::FileSet, [491](#)
- FileWithName
  - gdcm::FileWithName, [501](#)
- Fill
  - gdcm::ByteValue, [221](#)
- FillFromDataSet
  - gdcm::FileMetaInformation, [476](#)
- FillMedicalImageInformation
  - vtkGDCMImageReader, [1310](#)
  - vtkGDCMImageReader2, [1322](#)
  - vtkGDCMPolyDataReader, [1344](#)
- FindContext
  - gdcm::network::ULConnection, [1242](#)
- FindCSAElementByName
  - gdcm::CSAHeader, [290](#)
- FindDataElement
  - gdcm::DataSet, [329](#), [330](#)
  - gdcm::Item, [620](#)
  - gdcm::SequenceOfItems, [986](#)
- FindDictEntry
  - gdcm::PrivateDict, [876](#)
- FindMacroEntry
  - gdcm::Macro, [683](#)
- FindModuleEntryInMacros
  - gdcm::Module, [720](#)
- FindMrProtocolByName
  - gdcm::MrProtocol, [736](#)
- FindNextDataElement
  - gdcm::DataSet, [330](#)
- FindPatientRootQuery
  - gdcm::FindPatientRootQuery, [504](#)
- FindPDBElementByName
  - gdcm::PDBHeader, [802](#)
- FindStudyRootQuery
  - gdcm::FindStudyRootQuery, [508](#)
- FirstRender
  - vtkImageColorViewer, [1377](#)
- FL
  - gdcm::VR, [1296](#)
- FLOAT16
  - gdcm::PixelFormat, [822](#)
- FLOAT32
  - gdcm::PixelFormat, [822](#)
- FLOAT64
  - gdcm::PixelFormat, [822](#)
- ForceRescale
  - vtkGDCMImageReader, [1318](#)
  - vtkGDCMImageReader2, [1330](#)
- FormatDateTime
  - gdcm::System, [1110](#)
- Fragment
  - gdcm::Fragment, [513](#)
- FragmentVector
  - gdcm::SequenceOfFragments, [977](#)
- FromString
  - gdcm::StringFilter, [1074](#)
- FUJI
  - gdcm::EquipmentManufacturer, [431](#)
- FujiPrivateCRLImageStorage
  - gdcm::MediaStorage, [694](#)
- FujiPrivateMammoCRLImageStorage
  - gdcm::MediaStorage, [695](#)
- gdcm, [43](#)
  - add1, [63](#)
  - AEComp, [58](#)
  - ASComp, [58](#)
  - backslash, [63](#)
  - BOOL\_FUNCTION\_PFILE\_PFILE\_POINTER, [58](#)
  - Clamp, [63](#)
  - clean, [64](#)
  - CompOperators, [60](#)
  - CSComp, [58](#)
  - DAComp, [58](#)
  - doround, [64](#)
  - DTComp, [58](#)

- eArabic, 62
- ECharSet, 60
- eCreateMMPS, 62
- eCyrillic, 62
- eFind, 63
- eGB18030, 62
- eGreek, 62
- eHebrew, 62
- eImage, 62
- eJapanese, 62
- eJapaneseKanjiMultibyte, 62
- eJapaneseSupplementaryKanjiMultibyte, 62
- eKoreanHangulHanjaMultibyte, 62
- eLatin1, 62
- eLatin2, 62
- eLatin3, 62
- eLatin4, 62
- eLatin5, 62
- eMove, 63
- ENQueryType, 62
- ePatient, 62
- ePatientRootType, 63
- EQueryLevel, 62
- EQueryType, 62
- ERootType, 63
- eSeries, 62
- eSetMMPS, 62
- eStudy, 62
- eStudyRootType, 63
- eThai, 62
- eUTF8, 62
- eWLMFind, 63
- FileList, 59
- GDCM\_DIFFERENT, 60
- GDCM\_EQUAL, 60
- GDCM\_GREATER, 60
- GDCM\_GREATEROREQUAL, 60
- GDCM\_LESS, 60
- GDCM\_LESSEOREQUAL, 60
- GetVRFromTag, 64
- GlobalInstance, 74
- IconImage, 59
- LD\_ALL, 63
- LD\_NOSEQ, 63
- LD\_NOSHADOW, 63
- LD\_NOSHADOWSEQ, 63
- LOComp, 59
- LodModeType, 63
- LTComp, 59
- MacroEntry, 59
- NestedMacroEntries, 59
- operator!=, 64
- operator<<, 64–73
- operator>>, 73
- operator==, 73
- PNComp, 59
- Round, 74
- roundat, 74
- SHComp, 59
- STComp, 59
- TMComp, 59
- UCComp, 60
- UIComp, 60
- URComp, 60
- UTComp, 60
- x16printf, 74
- GDCM Documentation, 1
- gdcm::AbortEvent, 99
- gdcm::AnonymizeEvent, 103
  - ~AnonymizeEvent, 105
  - AnonymizeEvent, 105
  - CheckEvent, 105
  - GetEventName, 105
  - GetTag, 105
  - MakeObject, 105
  - operator=, 106
  - Self, 104
  - SetTag, 106
  - Superclass, 104
- gdcm::Anonymizer, 106
  - ~Anonymizer, 110
  - Anonymizer, 110
  - BALCPPProtect, 110
  - BasicApplicationLevelConfidentialityProfile, 110
  - CanEmptyTag, 110
  - Clear, 110
  - ClearInternalUIDs, 111
  - Empty, 111
  - GetBasicApplicationLevelConfidentialityProfileAttributes, 111
  - GetCryptographicMessageSyntax, 111
  - GetFile, 112
  - New, 112
  - RecurseDataSet, 112
  - Remove, 112
  - RemoveGroupLength, 112
  - RemovePrivateTags, 113
  - RemoveRetired, 113
  - Replace, 113, 114
  - SetCryptographicMessageSyntax, 114
  - SetFile, 114
- gdcm::AnyEvent, 115
- gdcm::ApplicationEntity, 118
  - Internal, 120
  - IsValid, 119
  - MaxLength, 120
  - MaxNumberOfComponents, 120
  - Padding, 120

- Print, [119](#)
- Separator, [120](#)
- SetBlob, [119](#)
- Squeeze, [119](#)
- gdcmm::ASN1, [126](#)
- ~ASN1, [127](#)
- ASN1, [127](#)
- operator=, [127](#)
- ParseDump, [127](#)
- ParseDumpFile, [128](#)
- TestPBKDF2, [128](#)
- gdcmm::Attribute< Group, Element, TVR, TVM >, [130](#)
- ArrayType, [132](#)
- GDCM\_STATIC\_ASSERT, [132](#)
- GetAsDataElement, [132](#)
- GetDictVM, [133](#)
- GetDictVR, [133](#)
- GetNumberOfValues, [133](#)
- GetTag, [133](#)
- GetValue, [133](#), [134](#)
- GetValues, [134](#)
- GetVM, [134](#)
- GetVR, [134](#)
- Internal, [137](#)
- operator!=, [134](#)
- operator<, [135](#)
- operator==, [135](#)
- operator[], [135](#)
- Print, [135](#)
- Set, [135](#)
- SetByteValue, [136](#)
- SetByteValueNoSwap, [136](#)
- SetFromDataElement, [136](#)
- SetFromDataSet, [136](#)
- SetValue, [137](#)
- SetValues, [137](#)
- VMType, [132](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [138](#)
- ArrayType, [139](#)
- GDCM\_STATIC\_ASSERT, [139](#), [140](#)
- GetAsDataElement, [140](#)
- GetDictVM, [140](#)
- GetDictVR, [140](#)
- GetNumberOfValues, [140](#)
- GetTag, [140](#)
- GetValue, [141](#)
- GetValues, [141](#)
- GetVM, [141](#)
- GetVR, [141](#)
- Internal, [143](#)
- operator!=, [141](#)
- operator<, [141](#)
- operator==, [142](#)
- Print, [142](#)
- Set, [142](#)
- SetByteValue, [142](#)
- SetByteValueNoSwap, [142](#)
- SetFromDataElement, [142](#)
- SetFromDataSet, [143](#)
- SetValue, [143](#)
- VMType, [139](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1\_3 >, [144](#)
- GetVM, [146](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1\_8 >, [146](#)
- GetVM, [148](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, [148](#)
- ~Attribute, [149](#)
- ArrayType, [149](#)
- Attribute, [149](#)
- GDCM\_STATIC\_ASSERT, [150](#)
- GetAsDataElement, [150](#)
- GetDictVM, [150](#)
- GetDictVR, [150](#)
- GetNumberOfValues, [150](#)
- GetTag, [150](#)
- GetValue, [151](#)
- GetValues, [151](#)
- GetVM, [151](#)
- GetVR, [151](#)
- operator[], [151](#)
- Print, [151](#)
- Set, [152](#)
- SetByteValue, [152](#)
- SetFromDataElement, [152](#)
- SetFromDataSet, [152](#)
- SetNumberOfValues, [152](#)
- SetValue, [152](#), [153](#)
- SetValues, [153](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM2\_2n >, [153](#)
- GetVM, [156](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM2\_n >, [156](#)
- GetVM, [158](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM3\_3n >, [158](#)
- GetVM, [161](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM3\_n >, [161](#)
- GetVM, [163](#)
- gdcmm::AudioCodec, [163](#)
- ~AudioCodec, [165](#)
- AudioCodec, [165](#)
- CanCode, [165](#)
- CanDecode, [165](#)

- Decode, 166
- gdcmm::Base64, 166
  - Base64, 167
  - Decode, 167
  - Encode, 167
  - GetDecodeLength, 168
  - GetEncodeLength, 168
  - operator=, 168
- gdcmm::BaseQuery, 175
  - ~BaseQuery, 177
  - AddQueryDataSet, 177
  - BaseQuery, 177
  - GetAbstractSyntaxUID, 177
  - GetQueryDataSet, 177, 178
  - GetSOPInstanceUID, 178
  - mDataSet, 179
  - mSopInstanceUID, 179
  - Print, 178
  - QueryFactory, 179
  - SetSearchParameter, 178
  - SetSOPInstanceUID, 178
  - ValidateQuery, 179
  - ValidDataSet, 179
  - WriteHelpFile, 179
  - WriteQuery, 179
- gdcmm::BaseRootQuery, 180
  - ~BaseRootQuery, 182
  - BaseRootQuery, 182
  - Construct, 183
  - GetQueryLevelFromQueryRoot, 183
  - GetQueryLevelFromString, 183
  - GetQueryLevelString, 183
  - GetTagListByLevel, 183
  - InitializeDataSet, 183
  - mHelpDescription, 184
  - mImage, 184
  - mPatient, 184
  - mRootType, 185
  - mSeries, 185
  - mStudy, 185
  - QueryFactory, 184
  - ValidateQuery, 184
- gdcmm::BasicOffsetTable, 188
  - BasicOffsetTable, 191
  - operator<=, 192
  - Read, 191
- gdcmm::Bitmap, 192
  - ~Bitmap, 195
  - AreOverlaysInPixelData, 196
  - Bitmap, 195
  - Clear, 196
  - ComputeLossyFlag, 196
  - Dimensions, 205
  - GetBuffer, 196
  - GetBuffer2, 196
  - GetBufferLength, 196
  - GetColumns, 197
  - GetDataElement, 197
  - GetDimension, 197
  - GetDimensions, 197
  - GetLUT, 198
  - GetNeedByteSwap, 198
  - GetNumberOfDimensions, 198
  - GetPhotometricInterpretation, 198
  - GetPixelFormat, 199
  - GetPlanarConfiguration, 199
  - GetRows, 199
  - GetTransferSyntax, 199
  - ImageChangeTransferSyntax, 204
  - IsEmpty, 199
  - IsLossy, 200
  - IsTransferSyntaxCompatible, 200
  - LossyFlag, 205
  - LUT, 205
  - LUTPtr, 195
  - NeedByteSwap, 205
  - NumberOfDimensions, 205
  - PF, 205
  - PI, 205
  - PixelData, 205
  - PixmapReader, 204
  - PlanarConfiguration, 205
  - Print, 200
  - SetColumns, 200
  - SetDataElement, 200
  - SetDimension, 200
  - SetDimensions, 201
  - SetLossyFlag, 201
  - SetLUT, 201
  - SetNeedByteSwap, 201
  - SetNumberOfDimensions, 201
  - SetPhotometricInterpretation, 202
  - SetPixelFormat, 202
  - SetPlanarConfiguration, 202
  - SetRows, 202
  - SetTransferSyntax, 203
  - TryJPEG2000Codec, 203
  - TryJPEG2000Codec2, 203
  - TryJPEGCodec, 203
  - TryJPEGCodec2, 203
  - TryJPEGLSCodec, 203
  - TryKAKADUCoec, 203
  - TryPVRGCodec, 204
  - TryRAWCodec, 204
  - TryRLECodec, 204
  - TS, 205
  - UnusedBitsPresentInPixelData, 204
- gdcmm::BitmapToBitmapFilter, 206

- ~BitmapToBitmapFilter, [207](#)
- BitmapToBitmapFilter, [207](#)
- GetOutput, [207](#)
- GetOutputAsBitmap, [207](#)
- Input, [208](#)
- Output, [208](#)
- SetInput, [207](#)
- gdcmm::BoxRegion, [208](#)
  - ~BoxRegion, [210](#)
  - Area, [210](#)
  - BoundingBox, [210](#)
  - BoxRegion, [210](#)
  - Clone, [211](#)
  - ComputeBoundingBox, [211](#)
  - Empty, [211](#)
  - GetXMax, [211](#)
  - GetXMin, [211](#)
  - GetYMax, [211](#)
  - GetYMin, [211](#)
  - GetZMax, [212](#)
  - GetZMin, [212](#)
  - IsValid, [212](#)
  - operator=, [212](#)
  - Print, [212](#)
  - SetDomain, [212](#)
- gdcmm::ByteBuffer, [213](#)
  - ByteBuffer, [214](#)
  - Get, [214](#)
  - GetStart, [214](#)
  - ShiftEnd, [214](#)
  - UpdatePosition, [214](#)
- gdcmm::ByteSwap< T >, [214](#)
  - Swap, [215](#)
  - SwapFromSwapCodeIntoSystem, [215](#)
  - SwapRange, [215](#)
  - SwapRangeFromSwapCodeIntoSystem, [215](#)
  - SystemIsBigEndian, [216](#)
  - SystemIsLittleEndian, [216](#)
- gdcmm::ByteSwapFilter, [216](#)
  - ~ByteSwapFilter, [217](#)
  - ByteSwap, [217](#)
  - ByteSwapFilter, [217](#)
  - operator=, [217](#)
  - SetByteSwapTag, [217](#)
- gdcmm::ByteValue, [218](#)
  - ~ByteValue, [220](#)
  - Append, [221](#)
  - ByteValue, [220](#)
  - Clear, [221](#)
  - ComputeLength, [221](#)
  - Fill, [221](#)
  - GetBuffer, [221](#)
  - GetLength, [221](#)
  - GetPointer, [222](#)
  - GetVoidPointer, [222](#)
  - IsEmpty, [222](#)
  - IsPrintable, [223](#)
  - operator const std::vector< char > &, [223](#)
  - operator=, [223](#)
  - operator==, [223](#)
  - Print, [223](#)
  - PrintASCII, [223](#)
  - PrintASCIIXML, [224](#)
  - PrintGroupLength, [224](#)
  - PrintHex, [224](#)
  - PrintHexXML, [224](#)
  - PrintPNXML, [224](#)
  - Read, [224](#)
  - SetLength, [225](#)
  - SetLengthOnly, [225](#)
  - Write, [225](#)
  - WriteBuffer, [225](#)
- gdcmm::CAPICryptoFactory, [226](#)
  - CAPICryptoFactory, [227](#)
  - CreateCMSProvider, [227](#)
- gdcmm::CAPICryptographicMessageSyntax, [227](#)
  - ~CAPICryptographicMessageSyntax, [229](#)
  - CAPICryptographicMessageSyntax, [229](#)
  - Decrypt, [229](#)
  - Encrypt, [229](#)
  - GetCipherType, [229](#)
  - GetInitialized, [229](#)
  - ParseCertificateFile, [229](#)
  - ParseKeyFile, [230](#)
  - SetCipherType, [230](#)
  - SetPassword, [230](#)
- gdcmm::Cleaner, [239](#)
  - ~Cleaner, [241](#)
  - Clean, [242](#)
  - Cleaner, [241](#)
  - Empty, [242](#)
  - EmptyWhenScrubFails, [242](#)
  - GetFile, [243](#)
  - New, [243](#)
  - Preserve, [243](#)
  - Remove, [243, 244](#)
  - RemoveAllGroupLength, [244](#)
  - RemoveAllIllegal, [244](#)
  - RemoveAllMissingPrivateCreator, [244](#)
  - RemoveMissingPrivateCreator, [244](#)
  - Scrub, [244, 245](#)
  - SetFile, [245](#)
- gdcmm::Codec, [250](#)
- gdcmm::Coder, [251](#)
  - ~Coder, [252](#)
  - CanCode, [252](#)
  - Code, [252](#)
  - InternalCode, [252](#)

- gdcmm::CodeString, 253
  - CodeString, 255, 256
  - const\_iterator, 254
  - const\_reference, 254
  - const\_reverse\_iterator, 254
  - difference\_type, 254
  - GetAsString, 256
  - IsValid, 256
  - iterator, 255
  - operator!=, 256
  - operator<<, 256
  - operator==, 257
  - pointer, 255
  - reference, 255
  - reverse\_iterator, 255
  - Size, 256
  - size\_type, 255
  - TrimInternal, 256
  - value\_type, 255
- gdcmm::Command, 257
  - ~Command, 259
  - Command, 259
  - Execute, 259
  - operator=, 260
- gdcmm::CommandDataSet, 260
  - ~CommandDataSet, 263
  - CommandDataSet, 263
  - Insert, 263
  - operator<<, 264
  - Read, 263
  - Replace, 264
  - Write, 264
- gdcmm::CompositeNetworkFunctions, 266
  - CEcho, 267
  - CFind, 268
  - CMove, 268
  - ConstructQuery, 269
  - CStore, 270
  - KeyValuePairArrayType, 267
  - KeyValuePairType, 267
- gdcmm::ConstCharWrapper, 271
  - ConstCharWrapper, 271
  - operator const char \*, 271
- gdcmm::CP246ExplicitDataElement, 272
  - GetLength, 274
  - Read, 274
  - ReadPreValue, 275
  - ReadValue, 275
  - ReadWithLength, 275
- gdcmm::CryptoFactory, 275
  - ~CryptoFactory, 277
  - CAPI, 277
  - CreateCMSProvider, 277
  - CryptoFactory, 277
  - CryptoLib, 276
  - DEFAULT, 277
  - GetFactoryInstance, 277
  - OPENSSL, 277
  - OPENSSL7, 277
- gdcmm::CryptographicMessageSyntax, 278
  - ~CryptographicMessageSyntax, 279
  - AES128\_CIPHER, 279
  - AES192\_CIPHER, 279
  - AES256\_CIPHER, 279
  - CipherTypes, 279
  - CryptographicMessageSyntax, 279
  - Decrypt, 279
  - DES3\_CIPHER, 279
  - Encrypt, 279
  - GetCipherType, 280
  - operator=, 280
  - ParseCertificateFile, 280
  - ParseKeyFile, 280
  - SetCipherType, 280
  - SetPassword, 281
- gdcmm::CSAElement, 281
  - CSAElement, 283
  - DataField, 287
  - DataPtr, 283
  - GetByteValue, 283
  - GetKey, 283
  - GetName, 284
  - GetNoOfItems, 284
  - GetSyngoDT, 284
  - GetValue, 284
  - GetVM, 284
  - GetVR, 285
  - IsEmpty, 285
  - KeyField, 287
  - NameField, 287
  - NoOfItemsField, 287
  - operator<, 285
  - operator<<, 287
  - operator=, 285
  - operator==, 285
  - SetByteValue, 285
  - SetKey, 286
  - SetName, 286
  - SetNoOfItems, 286
  - SetSyngoDT, 286
  - SetValue, 286
  - SetVM, 286
  - SetVR, 286
  - SyngoDTField, 287
  - ValueMultiplicityField, 287
  - VRField, 287
- gdcmm::CSAHeader, 288
  - ~CSAHeader, 290

- CSAHeader, [290](#)
- CSAHeaderType, [289](#)
- DATASET\_FORMAT, [290](#)
- FindCSAElementByName, [290](#)
- GetCSADataInfo, [290](#)
- GetCSAEEnd, [290](#)
- GetCSAElementByName, [291](#)
- GetCSAImageHeaderInfoTag, [291](#)
- GetCSASeriesHeaderInfoTag, [291](#)
- GetDataSet, [291](#)
- GetFormat, [291](#)
- GetInterfile, [292](#)
- GetMrProtocol, [292](#)
- INTERFILE, [290](#)
- LoadFromDataElement, [292](#)
- NOMAGIC, [290](#)
- operator<<, [293](#)
- Print, [292](#)
- SV10, [290](#)
- UNKNOWN, [290](#)
- ZEROED\_OUT, [290](#)
- gdcmm::CSAHeaderDict, [293](#)
  - AddCSAHeaderDictEntry, [294](#)
  - Begin, [294](#)
  - ConstIterator, [294](#)
  - CSAHeaderDict, [294](#)
  - Dicts, [295](#)
  - End, [295](#)
  - GetCSAHeaderDictEntry, [295](#)
  - IsEmpty, [295](#)
  - Iterator, [294](#)
  - LoadDefault, [295](#)
  - MapCSAHeaderDictEntry, [294](#)
  - operator<<, [295](#)
  - operator=, [295](#)
- gdcmm::CSAHeaderDictEntry, [296](#)
  - CSAHeaderDictEntry, [297](#)
  - GetDescription, [297](#)
  - GetName, [297](#)
  - GetVM, [297](#)
  - GetVR, [298](#)
  - operator<, [298](#)
  - operator<<, [299](#)
  - SetDescription, [298](#)
  - SetName, [298](#)
  - SetVM, [298](#)
  - SetVR, [298](#)
- gdcmm::CSAHeaderDictException, [299](#)
- gdcmm::Curve, [303](#)
  - ~Curve, [305](#)
  - Curve, [305](#)
  - Decode, [305](#)
  - GetAsPoints, [305](#)
  - GetCurveDataDescriptor, [305](#)
  - GetDataValueRepresentation, [305](#)
  - GetDimensions, [305](#)
  - GetGroup, [305](#)
  - GetNumberOfCurves, [306](#)
  - GetNumberOfPoints, [306](#)
  - GetTypeOfData, [306](#)
  - GetTypeOfDataDescription, [306](#)
  - IsEmpty, [306](#)
  - Print, [306](#)
  - SetCoordinateStartValue, [306](#)
  - SetCoordinateStepValue, [306](#)
  - SetCurve, [306](#)
  - SetCurveDataDescriptor, [307](#)
  - SetCurveDescription, [307](#)
  - SetDataValueRepresentation, [307](#)
  - SetDimensions, [307](#)
  - SetGroup, [307](#)
  - SetNumberOfPoints, [307](#)
  - SetTypeOfData, [307](#)
  - Update, [307](#)
- gdcmm::DataElement, [308](#)
  - Clear, [312](#)
  - DataElement, [311](#)
  - Empty, [312](#)
  - GetByteValue, [312](#)
  - GetLength, [312](#)
  - GetSequenceOfFragments, [312, 313](#)
  - GetTag, [313](#)
  - GetValue, [313](#)
  - GetValueAsSQ, [314](#)
  - GetVL, [314](#)
  - GetVR, [314](#)
  - IsEmpty, [315](#)
  - IsUndefinedLength, [315](#)
  - operator<, [315](#)
  - operator<<, [319](#)
  - operator=, [315](#)
  - operator==, [316](#)
  - Read, [316](#)
  - ReadOrSkip, [316](#)
  - ReadPreValue, [316](#)
  - ReadValue, [316](#)
  - ReadValueWithLength, [316](#)
  - ReadWithLength, [317](#)
  - SetByteValue, [317](#)
  - SetTag, [317](#)
  - SetValue, [318](#)
  - SetValueFieldLength, [318](#)
  - SetVL, [318](#)
  - SetVLToUndefined, [318](#)
  - SetVR, [319](#)
  - TagField, [320](#)
  - ValueField, [320](#)
  - ValueLengthField, [320](#)

- ValuePtr, 311
- VRField, 320
- Write, 319
- gdcm::DataElementException, 321
- gdcm::DataEvent, 321
  - ~DataEvent, 323
  - CheckEvent, 324
  - DataEvent, 323, 324
  - GetData, 324
  - GetDataLength, 324
  - GetEventName, 324
  - MakeObject, 324
  - operator=, 324
  - Self, 323
  - SetData, 325
  - Superclass, 323
- gdcm::DataSet, 325
  - Begin, 328
  - Clear, 329
  - ComputeDataElement, 329
  - ComputeGroupLength, 329
  - ConstIterator, 328
  - CSAHeader, 336
  - DataElementSet, 328
  - End, 329
  - FindDataElement, 329, 330
  - FindNextDataElement, 330
  - GetDataElement, 330
  - GetDEEnd, 331
  - GetDES, 331
  - GetLength, 331
  - GetMediaStorage, 331
  - GetPrivateCreator, 332
  - GetPrivateTag, 332
  - Insert, 332
  - InsertDataElement, 332
  - IsEmpty, 333
  - Iterator, 328
  - operator<=, 336
  - operator(), 333
  - operator=, 333
  - operator[], 333
  - Print, 333
  - Read, 333
  - ReadNested, 333
  - ReadSelectedPrivateTags, 334
  - ReadSelectedPrivateTagsWithLength, 334
  - ReadSelectedTags, 334
  - ReadSelectedTagsWithLength, 334
  - ReadUpToTag, 334
  - ReadUpToTagWithLength, 334
  - ReadWithLength, 335
  - Remove, 335
  - Replace, 335
  - ReplaceEmpty, 335
  - Size, 336
  - SizeType, 328
  - Write, 336
- gdcm::DataSetEvent, 337
  - ~DataSetEvent, 339
  - CheckEvent, 339
  - DataSetEvent, 339
  - GetDataSet, 339
  - GetEventName, 339
  - m\_DataSet, 340
  - MakeObject, 339
  - operator=, 339
  - Self, 338
  - Superclass, 338
- gdcm::DataSetHelper, 340
  - ComputeVR, 341
- gdcm::Decoder, 341
  - ~Decoder, 342
  - CanDecode, 342
  - Decode, 342
  - DecodeByStreams, 342
- gdcm::DefinedTerms, 343
  - DefinedTerms, 343
- gdcm::Defs, 344
  - ~Defs, 345
  - Defs, 345
  - GetIODFromFile, 345
  - GetIODNameFromMediaStorage, 345
  - GetIODs, 345
  - GetMacros, 345, 346
  - GetModules, 346
  - GetTypeFromTag, 346
  - Global, 347
  - IsEmpty, 346
  - LoadDefaults, 346
  - LoadFromFile, 346
  - operator=, 347
  - Verify, 347
- gdcm::DeltaEncodingCodec, 348
  - ~DeltaEncodingCodec, 351
  - CanDecode, 351
  - Decode, 351
  - DeltaEncodingCodec, 351
- gdcm::DICOMDIR, 351
  - DICOMDIR, 352
- gdcm::DICOMDIRGenerator, 352
  - ~DICOMDIRGenerator, 354
  - AddImageDirectoryRecord, 354
  - AddPatientDirectoryRecord, 354
  - AddSeriesDirectoryRecord, 354
  - AddStudyDirectoryRecord, 354
  - DICOMDIRGenerator, 354
  - FilenamesType, 354



- FilenameType, [354](#)
- Generate, [354](#)
- GetFile, [355](#)
- GetScanner, [355](#)
- SetDescriptor, [355](#)
- SetFile, [355](#)
- SetFileNames, [355](#)
- SetRootDirectory, [356](#)
- gdcM::Dict, [356](#)
  - AddDictEntry, [358](#)
  - Begin, [358](#)
  - ConstIterator, [357](#)
  - Dict, [357](#)
  - Dicts, [360](#)
  - End, [358](#)
  - GetDictEntry, [358](#)
  - GetDictEntryByKeyword, [358](#)
  - GetDictEntryByName, [359](#)
  - GetKeywordFromTag, [359](#)
  - IsEmpty, [359](#)
  - Iterator, [357](#)
  - LoadDefault, [359](#)
  - MapDictEntry, [357](#)
  - operator<<, [360](#)
  - operator=, [359](#)
- gdcM::DictConverter, [360](#)
  - ~DictConverter, [361](#)
  - AddGroupLength, [362](#)
  - Convert, [362](#)
  - ConvertToCXX, [362](#)
  - ConvertToXML, [362](#)
  - DICT\_DEBUG, [361](#)
  - DICT\_DEFAULT, [361](#)
  - DICT\_XML, [361](#)
  - DictConverter, [361](#)
  - GetDictName, [362](#)
  - GetInputFilename, [362](#)
  - GetOutputFilename, [362](#)
  - GetOutputType, [362](#)
  - OutputTypes, [361](#)
  - Readuint16, [363](#)
  - ReadVM, [363](#)
  - ReadVR, [363](#)
  - SetDictName, [363](#)
  - SetInputFileName, [363](#)
  - SetOutputFileName, [363](#)
  - SetOutputType, [363](#)
  - WriteFooter, [363](#)
  - WriteHeader, [364](#)
- gdcM::DictEntry, [364](#)
  - Dict, [368](#)
  - DictEntry, [365](#)
  - GetKeyword, [365](#)
  - GetName, [365](#)
  - GetRetired, [366](#)
  - GetVM, [366](#)
  - GetVR, [366](#)
  - IsUnique, [366](#)
  - operator<<, [368](#)
  - SetElementXX, [366](#)
  - SetGroupXX, [367](#)
  - SetKeyword, [367](#)
  - SetName, [367](#)
  - SetRetired, [367](#)
  - SetVM, [367](#)
  - SetVR, [367](#)
- gdcM::DictPrinter, [368](#)
  - ~DictPrinter, [370](#)
  - DictPrinter, [370](#)
  - Print, [371](#)
  - PrintDataElement2, [371](#)
  - PrintDataSet2, [371](#)
- gdcM::Dicts, [371](#)
  - ~Dicts, [373](#)
  - ConstructorType, [372](#)
  - Dicts, [373](#)
  - GEMS, [372](#)
  - GetConstructorString, [373](#)
  - GetCSAHeaderDict, [373](#)
  - GetDictEntry, [373](#)
  - GetPrivateDict, [374](#)
  - GetPublicDict, [374](#)
  - Global, [375](#)
  - IsEmpty, [374](#)
  - LoadDefaults, [374](#)
  - operator<<, [375](#)
  - operator=, [374](#)
  - PHILIPS, [372](#)
  - SIEMENS, [372](#)
- gdcM::DirectionCosines, [376](#)
  - ~DirectionCosines, [378](#)
  - ComputeDistAlongNormal, [378](#)
  - Cross, [378](#)
  - CrossDot, [378](#)
  - DirectionCosines, [377](#)
  - Dot, [378](#)
  - IsValid, [379](#)
  - Norm, [379](#)
  - Normalize, [379](#)
  - operator const double \*, [379](#)
  - Print, [379](#)
  - SetFromString, [379](#)
- gdcM::Directory, [380](#)
  - ~Directory, [381](#)
  - Directory, [381](#)
  - Explore, [382](#)
  - FileNamesType, [381](#)
  - FilenameType, [381](#)

- GetDirectories, [382](#)
- GetFileNames, [382](#)
- GetToplevel, [382](#)
- Load, [382](#)
- operator<=, [383](#)
- Print, [382](#)
- gdcmm::DirectoryHelper, [383](#)
  - GetCTImageSeriesUIDs, [384](#)
  - GetFileNamesFromSeriesUIDs, [384](#)
  - GetFrameOfReference, [384](#)
  - GetMRImageSeriesUIDs, [384](#)
  - GetRTStructSeriesUIDs, [384](#)
  - GetSeriesUIDsBySOPClassUID, [384](#)
  - GetSOPClassUID, [384](#)
  - GetStringValueFromTag, [385](#)
  - LoadImageFromFiles, [385](#)
  - RetrieveSOPInstanceUIDFromIndex, [385](#)
  - RetrieveSOPInstanceUIDFromZPosition, [385](#)
- gdcmm::DPath, [385](#)
  - ~DPath, [386](#)
  - ConstructFromString, [387](#)
  - DPath, [386](#)
  - IsValid, [387](#)
  - Match, [387](#)
  - operator<, [387](#)
  - operator<=, [387](#)
  - Print, [387](#)
- gdcmm::DummyValueGenerator, [388](#)
  - Generate, [388](#)
- gdcmm::Dumper, [389](#)
  - ~Dumper, [391](#)
  - Dumper, [391](#)
- gdcmm::Element< TVR, TVM >, [391](#)
  - GetAsDataElement, [393](#)
  - GetLength, [393](#)
  - GetValue, [394](#)
  - GetValues, [394](#)
  - GetVM, [394](#)
  - GetVR, [394](#)
  - Internal, [396](#)
  - operator[], [394](#)
  - Print, [395](#)
  - Read, [395](#)
  - Set, [395](#)
  - SetFromDataElement, [395](#)
  - SetNoSwap, [395](#)
  - SetValue, [396](#)
  - Type, [393](#)
  - Write, [396](#)
- gdcmm::Element< TVR, VM::VM1\_2 >, [397](#)
  - Parent, [398](#)
  - SetLength, [399](#)
- gdcmm::Element< TVR, VM::VM1\_n >, [399](#)
  - ~Element, [400](#)
  - Element, [400](#)
  - GetAsDataElement, [401](#)
  - GetLength, [401](#)
  - GetValue, [401](#)
  - GetVM, [401](#)
  - GetVR, [401](#)
  - operator=, [401](#)
  - operator[], [402](#)
  - Print, [402](#)
  - Read, [402](#)
  - Set, [402](#)
  - SetArray, [402](#)
  - SetFromDataElement, [402](#)
  - SetLength, [403](#)
  - SetNoSwap, [403](#)
  - SetValue, [403](#)
  - Type, [400](#)
  - Write, [403](#)
  - WriteASCII, [403](#)
- gdcmm::Element< TVR, VM::VM2\_2n >, [404](#)
  - Parent, [406](#)
  - SetLength, [406](#)
- gdcmm::Element< TVR, VM::VM2\_n >, [406](#)
  - Parent, [408](#)
  - SetLength, [408](#)
- gdcmm::Element< TVR, VM::VM3\_3n >, [408](#)
  - Parent, [411](#)
  - SetLength, [411](#)
- gdcmm::Element< TVR, VM::VM3\_4 >, [411](#)
  - Parent, [413](#)
  - SetLength, [413](#)
- gdcmm::Element< TVR, VM::VM3\_n >, [413](#)
  - Parent, [415](#)
  - SetLength, [416](#)
- gdcmm::Element< VR::AS, VM::VM5 >, [416](#)
  - GetLength, [416](#)
  - Internal, [417](#)
  - Print, [416](#)
- gdcmm::Element< VR::OB, VM::VM1 >, [417](#)
- gdcmm::Element< VR::OW, VM::VM1 >, [419](#)
- gdcmm::ElementDisableCombinations< TVR, TVM >, [421](#)
- gdcmm::ElementDisableCombinations< VR::OB, VM::VM1\_n >, [422](#)
- gdcmm::ElementDisableCombinations< VR::OW, VM::VM1\_n >, [422](#)
- gdcmm::EmptyMaskGenerator, [422](#)
  - ~EmptyMaskGenerator, [424](#)
  - EmptyMaskGenerator, [424](#)
  - Execute, [424](#)
  - SetInputDirectory, [424](#)
  - SetOutputDirectory, [424](#)
  - SetSOPClassUIDMode, [424](#)
  - SOPClassUIDMode, [423](#)
  - UseGrayscaleSecondaryImageStorage, [423](#)

- UseOriginalSOPClassUID, [423](#)
- gdcm::EncapsulatedDocument, [425](#)
  - EncapsulatedDocument, [425](#)
- gdcm::EncodingImplementation< T >, [425](#)
- gdcm::EncodingImplementation< VR::VRASCII >, [426](#)
  - Read, [426](#)
  - ReadComputeLength, [426](#)
  - ReadNoSwap, [427](#)
  - Write, [427](#)
- gdcm::EncodingImplementation< VR::VRBINARY >, [427](#)
  - Read, [428](#)
  - ReadComputeLength, [428](#)
  - ReadNoSwap, [428](#)
  - Write, [428](#)
- gdcm::EndEvent, [429](#)
- gdcm::EnumeratedValues, [430](#)
  - EnumeratedValues, [430](#)
- gdcm::EquipmentManufacturer, [431](#)
  - AGFA, [431](#)
  - Compute, [432](#)
  - FUJI, [431](#)
  - GEMS, [431](#)
  - HITACHI, [431](#)
  - KODAK, [431](#)
  - MARCONI, [431](#)
  - PMS, [431](#)
  - SAMSUNG, [431](#)
  - SIEMENS, [431](#)
  - TOSHIBA, [431](#)
  - Type, [431](#)
  - TypeToString, [432](#)
  - UIH, [431](#)
  - UNKNOWN, [431](#)
- gdcm::Event, [432](#)
  - ~Event, [434](#)
  - CheckEvent, [434](#)
  - Event, [434](#)
  - GetEventName, [434](#)
  - MakeObject, [434](#)
  - operator=, [435](#)
  - Print, [435](#)
- gdcm::Exception, [435](#)
  - ~Exception, [437](#)
  - Exception, [436](#)
  - GetDescription, [437](#)
  - what, [437](#)
- gdcm::ExitEvent, [438](#)
- gdcm::ExplicitDataElement, [439](#)
  - GetLength, [442](#)
  - Read, [442](#)
  - ReadPreValue, [442](#)
  - ReadValue, [442](#)
  - ReadWithLength, [442](#)
  - Write, [443](#)
- gdcm::ExplicitImplicitDataElement, [443](#)
  - GetLength, [446](#)
  - Read, [446](#)
  - ReadPreValue, [446](#)
  - ReadValue, [446](#)
  - ReadWithLength, [446](#)
- gdcm::Fiducials, [447](#)
  - Fiducials, [447](#)
- gdcm::File, [448](#)
  - ~File, [450](#)
  - File, [450](#)
  - GetDataSet, [450](#)
  - GetHeader, [451](#)
  - operator<<, [452](#)
  - Read, [451](#)
  - SetDataSet, [451](#)
  - SetHeader, [452](#)
  - Write, [452](#)
- gdcm::FileAnonymizer, [453](#)
  - ~FileAnonymizer, [455](#)
  - Empty, [455](#)
  - FileAnonymizer, [455](#)
  - Remove, [455](#)
  - Replace, [456](#)
  - SetInputFileName, [456](#)
  - SetOutputFileName, [456](#)
  - Write, [457](#)
- gdcm::FileChangeTransferSyntax, [457](#)
  - ~FileChangeTransferSyntax, [459](#)
  - Change, [460](#)
  - FileChangeTransferSyntax, [459](#)
  - GetCodec, [460](#)
  - New, [460](#)
  - SetInputFileName, [460](#)
  - SetOutputFileName, [460](#)
  - SetTransferSyntax, [461](#)
- gdcm::FileDecompressLookupTable, [461](#)
  - ~FileDecompressLookupTable, [463](#)
  - Change, [463](#)
  - FileDecompressLookupTable, [463](#)
  - GetFile, [463](#)
  - GetPixmap, [464](#)
  - SetFile, [464](#)
  - SetPixmap, [464](#)
- gdcm::FileDerivation, [464](#)
  - ~FileDerivation, [465](#)
  - AddDerivationDescription, [466](#)
  - AddPurposeOfReferenceCodeSequence, [466](#)
  - AddReference, [466](#)
  - AddSourceImageSequence, [466](#)
  - Derive, [466](#)
  - FileDerivation, [465](#)
  - GetFile, [466](#), [467](#)
  - SetAppendDerivationHistory, [467](#)

- SetDerivationCodeSequenceCodeValue, 467
- SetDerivationDescription, 467
- SetFile, 467
- SetPurposeOfReferenceCodeSequenceCodeValue, 468
- gdcm::FileExplicitFilter, 468
  - ~FileExplicitFilter, 469
  - Change, 469
  - ChangeFMI, 469
  - FileExplicitFilter, 469
  - GetFile, 470
  - ProcessDataSet, 470
  - SetChangePrivateTags, 470
  - SetFile, 470
  - SetRecomputeItemLength, 470
  - SetRecomputeSequenceLength, 470
  - SetUseVRUN, 470
- gdcm::FileMetaInformation, 471
  - ~FileMetaInformation, 475
  - AppendImplementationClassUID, 475
  - ComputeDataSetMediaStorageSOPClass, 475
  - ComputeDataSetTransferSyntax, 476
  - DataSetMS, 480
  - DataSetTS, 480
  - Default, 476
  - FileMetaInformation, 475
  - FillFromDataSet, 476
  - GetDataSetTransferSyntax, 476
  - GetFileMetaInformationVersion, 476
  - GetFullLength, 476
  - GetGDCMImplementationClassUID, 476
  - GetGDCMImplementationVersionName, 476
  - GetGDCMSourceApplicationEntityTitle, 477
  - GetImplementationClassUID, 477
  - GetImplementationVersionName, 477
  - GetMediaStorage, 477
  - GetMediaStorageAsString, 477
  - GetMetaInformationTS, 477
  - GetPreamble, 477
  - GetSourceApplicationEntityTitle, 477
  - Insert, 478
  - IsValid, 478
  - MetaInformationTS, 480
  - operator<<, 480
  - operator=, 478
  - Read, 478
  - ReadCompat, 478
  - ReadCompatInternal, 478
  - Replace, 478
  - SetDataSetTransferSyntax, 479
  - SetImplementationClassUID, 479
  - SetImplementationVersionName, 479
  - SetPreamble, 479
  - SetSourceApplicationEntityTitle, 479
  - Write, 480
- gdcm::Filename, 481
  - EndWith, 482
  - Filename, 482
  - GetExtension, 482
  - GetFileName, 482
  - GetName, 482
  - GetPath, 482
  - IsEmpty, 482
  - IsIdentical, 483
  - Join, 483
  - operator const char \*, 483
  - ToUnixSlashes, 483
  - ToWindowsSlashes, 483
- gdcm::FileNameEvent, 484
  - ~FileNameEvent, 486
  - CheckEvent, 486
  - FileNameEvent, 486
  - GetEventName, 486
  - GetFileName, 486
  - MakeObject, 486
  - operator=, 487
  - Self, 485
  - SetFileName, 487
  - Superclass, 485
- gdcm::FilenameGenerator, 487
  - ~FilenameGenerator, 489
  - FilenameGenerator, 489
  - FilenamesType, 488
  - FilenameType, 488
  - Generate, 489
  - GetFilename, 489
  - GetFilenames, 489
  - GetNumberOfFilenames, 489
  - GetPattern, 490
  - GetPrefix, 490
  - SetNumberOfFilenames, 490
  - SetPattern, 490
  - SetPrefix, 490
  - SizeType, 488
- gdcm::FileSet, 491
  - AddFile, 492
  - FileSet, 492
  - FilesType, 491
  - FileType, 491
  - GetFiles, 492
  - operator<<, 493
  - SetFiles, 492
- gdcm::FileStreamer, 493
  - ~FileStreamer, 496
  - AppendToDataElement, 496
  - AppendToGroupDataElement, 496
  - CheckDataElement, 496
  - CheckTemplateFileName, 496

- FileStreamer, [496](#)
- New, [497](#)
- ReserveDataElement, [497](#)
- ReserveGroupDataElement, [497](#)
- SetOutputFileName, [497](#)
- SetTemplateFileName, [497](#)
- StartDataElement, [498](#)
- StartGroupDataElement, [498](#)
- StopDataElement, [498](#)
- StopGroupDataElement, [498](#)
- gdcmm::FileWithName, [499](#)
  - filename, [501](#)
  - FileWithName, [501](#)
- gdcmm::FindPatientRootQuery, [502](#)
  - FindPatientRootQuery, [504](#)
  - GetAbstractSyntaxUID, [504](#)
  - GetTagListByLevel, [504](#)
  - InitializeDataSet, [505](#)
  - QueryFactory, [505](#)
  - ValidateQuery, [505](#)
- gdcmm::FindStudyRootQuery, [506](#)
  - FindStudyRootQuery, [508](#)
  - GetAbstractSyntaxUID, [508](#)
  - GetTagListByLevel, [508](#)
  - InitializeDataSet, [509](#)
  - QueryFactory, [509](#)
  - ValidateQuery, [509](#)
- gdcmm::Fragment, [510](#)
  - ComputeLength, [513](#)
  - Fragment, [513](#)
  - GetLength, [513](#)
  - operator<<, [514](#)
  - Read, [513](#)
  - ReadBacktrack, [513](#)
  - ReadPreValue, [513](#)
  - ReadValue, [513](#)
  - Write, [514](#)
- gdcmm::Global, [514](#)
  - ~Global, [515](#)
  - Append, [516](#)
  - GetDefs, [516](#)
  - GetDicts, [516](#)
  - GetInstance, [516](#)
  - Global, [515](#)
  - LoadResourcesFiles, [516](#)
  - Locate, [517](#)
  - operator<<, [517](#)
  - operator=, [517](#)
  - Prepend, [517](#)
- gdcmm::GroupDict, [518](#)
  - ~GroupDict, [519](#)
  - Add, [519](#)
  - GetAbbreviation, [519](#)
  - GetName, [519](#)
  - GroupDict, [519](#)
  - GroupStringVector, [518](#)
  - Insert, [519](#)
  - operator<<, [520](#)
  - Size, [519](#)
- gdcmm::IconImageFilter, [520](#)
  - ~IconImageFilter, [521](#)
  - Extract, [522](#)
  - ExtractIconImages, [522](#)
  - ExtractVeprolIconImages, [522](#)
  - GetFile, [522](#)
  - GetIconImage, [522](#)
  - GetNumberOfIconImages, [522](#)
  - IconImageFilter, [521](#)
  - SetFile, [523](#)
- gdcmm::IconImageGenerator, [523](#)
  - ~IconImageGenerator, [524](#)
  - AutoPixelMinMax, [524](#)
  - ConvertRGBToPaletteColor, [524](#)
  - Generate, [525](#)
  - GetIconImage, [525](#)
  - GetPixmap, [525](#)
  - IconImageGenerator, [524](#)
  - SetOutputDimensions, [525](#)
  - SetOutsideValuePixel, [525](#)
  - SetPixelMinMax, [526](#)
  - SetPixmap, [526](#)
- gdcmm::ignore\_char, [526](#)
  - ignore\_char, [527](#)
  - m\_char, [527](#)
- gdcmm::Image, [527](#)
  - ~Image, [532](#)
  - GetDirectionCosines, [533](#)
  - GetIntercept, [533](#)
  - GetOrigin, [533](#)
  - GetSlope, [533](#)
  - GetSpacing, [533](#)
  - Image, [532](#)
  - Print, [534](#)
  - SetDirectionCosines, [534](#)
  - SetIntercept, [534](#)
  - SetOrigin, [534](#), [535](#)
  - SetSlope, [535](#)
  - SetSpacing, [535](#)
- gdcmm::ImageApplyLookupTable, [536](#)
  - ~ImageApplyLookupTable, [538](#)
  - Apply, [539](#)
  - ImageApplyLookupTable, [538](#)
  - SetRGB8, [539](#)
- gdcmm::ImageChangePhotometricInterpretation, [539](#)
  - ~ImageChangePhotometricInterpretation, [542](#)
  - Change, [542](#)
  - ChangeMonochrome, [542](#)
  - ChangeRGB2YBR, [542](#)

- ChangeYBR2RGB, 542
- GetPhotometricInterpretation, 542
- ImageChangePhotometricInterpretation, 542
- RGB2YBR, 542
- SetPhotometricInterpretation, 543
- YBR2RGB, 543
- gdcmm::ImageChangePlanarConfiguration, 543
  - ~ImageChangePlanarConfiguration, 547
  - Change, 547
  - GetPlanarConfiguration, 547
  - ImageChangePlanarConfiguration, 547
  - RGBPixelsToRGBPlanes, 547
  - RGBPlanesToRGBPixels, 547
  - SetPlanarConfiguration, 548
- gdcmm::ImageChangeTransferSyntax, 548
  - ~ImageChangeTransferSyntax, 551
  - Change, 551
  - GetTransferSyntax, 551
  - ImageChangeTransferSyntax, 551
  - SetCompressIconImage, 552
  - SetForce, 552
  - SetTransferSyntax, 552
  - SetUserCodec, 552
  - TryJPEG2000Codec, 553
  - TryJPEGCodec, 553
  - TryJPEGLSCodec, 553
  - TryRAWCodec, 553
  - TryRLECodec, 553
- gdcmm::ImageCodec, 554
  - ~ImageCodec, 557
  - AppendFrameEncode, 557
  - AppendRowEncode, 557
  - CanCode, 558
  - CanDecode, 558
  - CleanupUnusedBits, 558
  - Clone, 558
  - Decode, 558
  - DecodeByStreams, 559
  - Dimensions, 564
  - DoByteSwap, 559
  - DoInvertMonochrome, 559
  - DoOverlayCleanup, 559
  - DoPaddedCompositePixelCode, 559
  - DoPlanarConfiguration, 559
  - DoSimpleCopy, 560
  - DoYBR, 560
  - DoYBRFull422, 560
  - FileChangeTransferSyntax, 564
  - GetDimensions, 560
  - GetHeaderInfo, 560
  - GetLossyFlag, 560
  - GetLUT, 560
  - GetNeedByteSwap, 561
  - GetNumberOfDimensions, 561
  - GetPhotometricInterpretation, 561
  - GetPixelFormat, 561
  - GetPlanarConfiguration, 561
  - ImageChangePhotometricInterpretation, 564
  - ImageCodec, 557
  - IsFrameEncoder, 561
  - IsLossy, 561
  - IsRowEncoder, 562
  - IsValid, 562
  - LossyFlag, 564
  - LUT, 564
  - LUTPtr, 557
  - NeedByteSwap, 565
  - NeedOverlayCleanup, 565
  - NumberOfDimensions, 565
  - PF, 565
  - PI, 565
  - PlanarConfiguration, 565
  - RequestPaddedCompositePixelCode, 565
  - RequestPlanarConfiguration, 565
  - SetDimensions, 562
  - SetLossyFlag, 562
  - SetLUT, 562
  - SetNeedByteSwap, 562
  - SetNeedOverlayCleanup, 563
  - SetNumberOfDimensions, 563
  - SetPhotometricInterpretation, 563
  - SetPixelFormat, 563
  - SetPlanarConfiguration, 563
  - StartEncode, 563
  - StopEncode, 564
- gdcmm::ImageConverter, 566
  - ~ImageConverter, 566
  - Convert, 566
  - GetOutput, 566
  - ImageConverter, 566
  - SetInput, 567
- gdcmm::ImageFragmentSplitter, 567
  - ~ImageFragmentSplitter, 569
  - GetFragmentSizeMax, 570
  - ImageFragmentSplitter, 569
  - SetForce, 570
  - SetFragmentSizeMax, 570
  - Split, 570
- gdcmm::ImageHelper, 570
  - ComputeMediaStorageFromModality, 572
  - ComputeSpacingFromImagePositionPatient, 572
  - GetDimensionsValue, 572
  - GetDirectionCosinesFromDataSet, 572
  - GetDirectionCosinesValue, 573
  - GetForcePixelSpacing, 573
  - GetForceRescaleInterceptSlope, 573
  - GetLUT, 573
  - GetOriginValue, 573

- GetPhotometricInterpretationValue, 573
- GetPixelFormatValue, 573
- GetPlanarConfigurationValue, 574
- GetPMSRescaleInterceptSlope, 574
- GetPointerFromElement, 574
- GetRealWorldValueMappingContent, 574
- GetRescaleInterceptSlopeValue, 574
- GetSecondaryCaptureImagePlaneModule, 574
- GetSpacingTagFromMediaStorage, 575
- GetSpacingValue, 575
- GetZSpacingTagFromMediaStorage, 575
- SetDimensionsValue, 575
- SetDirectionCosinesValue, 575
- SetForcePixelSpacing, 575
- SetForceRescaleInterceptSlope, 575
- SetOriginValue, 576
- SetPMSRescaleInterceptSlope, 576
- SetRescaleInterceptSlopeValue, 576
- SetSecondaryCaptureImagePlaneModule, 576
- SetSpacingValue, 576
- gdcm::ImageReader, 577
  - ~ImageReader, 580
  - GetImage, 580
  - ImageReader, 580
  - Read, 581
  - ReadACRNEMAImage, 581
  - ReadImage, 581
- gdcm::ImageRegionReader, 582
  - ~ImageRegionReader, 585
  - ComputeBufferLength, 586
  - GetRegion, 586
  - ImageRegionReader, 585
  - Read, 586
  - ReadInformation, 586
  - ReadIntoBuffer, 586
  - SetRegion, 587
- gdcm::ImageToImageFilter, 587
  - ~ImageToImageFilter, 589
  - GetInput, 589
  - GetOutput, 589
  - ImageToImageFilter, 589
- gdcm::ImageWriter, 590
  - ~ImageWriter, 593
  - ComputeTargetMediaStorage, 593
  - GetImage, 593, 594
  - ImageWriter, 593
  - Write, 594
- gdcm::ImplicitDataElement, 598
  - GetLength, 601
  - Read, 601
  - ReadPreValue, 601
  - ReadValue, 601
  - ReadValueWithLength, 601
  - ReadWithLength, 601
  - Write, 601
- gdcm::InitializeEvent, 602
- gdcm::IOD, 603
  - AddIODEntry, 605
  - Clear, 605
  - GetIODEntry, 605
  - GetNumberOfIODs, 605
  - GetTypeFromTag, 605
  - IOD, 604
  - MapIODEntry, 604
  - operator<<, 605
  - SizeType, 604
- gdcm::IODEntry, 606
  - GetIE, 607
  - GetName, 607
  - GetRef, 607
  - GetUsage, 607
  - GetUsageType, 607
  - IODEntry, 607
  - operator<<, 608
  - SetIE, 607
  - SetName, 607
  - SetRef, 608
  - SetUsage, 608
- gdcm::IODs, 608
  - AddIOD, 610
  - Begin, 610
  - Clear, 610
  - End, 610
  - GetIOD, 610
  - IODMapType, 609
  - IODMapTypeConstIterator, 609
  - IODName, 609
  - IODs, 610
  - operator<<, 611
- gdcm::IPPSorter, 611
  - ComputeZSpacing, 615
  - DirCosTolerance, 615
  - DropDuplicatePositions, 615
  - GetDirectionCosinesTolerance, 613
  - GetZSpacing, 613
  - GetZSpacingTolerance, 614
  - IPPSorter, 613
  - SetComputeZSpacing, 614
  - SetDirectionCosinesTolerance, 614
  - SetDropDuplicatePositions, 614
  - SetZSpacingTolerance, 615
  - Sort, 615
  - ZSpacing, 616
  - ZTolerance, 616
- gdcm::Item, 616
  - Clear, 620
  - FindDataElement, 620
  - GetDataElement, 620



- GetLength, [620](#)
- GetNestedDataSet, [620](#)
- InsertDataElement, [620](#)
- Item, [619](#)
- operator<<, [621](#)
- Read, [621](#)
- SetNestedDataSet, [621](#)
- Write, [621](#)
- gdcmm::IterationEvent, [622](#)
- gdcmm::JPEG12Codec, [623](#)
  - ~JPEG12Codec, [627](#)
  - DecodeByStreams, [627](#)
  - EncodeBuffer, [627](#)
  - GetHeaderInfo, [627](#)
  - InternalCode, [627](#)
  - IsStateSuspension, [627](#)
  - JPEG12Codec, [627](#)
- gdcmm::JPEG16Codec, [628](#)
  - ~JPEG16Codec, [632](#)
  - DecodeByStreams, [632](#)
  - EncodeBuffer, [632](#)
  - GetHeaderInfo, [632](#)
  - InternalCode, [632](#)
  - IsStateSuspension, [632](#)
  - JPEG16Codec, [632](#)
- gdcmm::JPEG2000Codec, [633](#)
  - ~JPEG2000Codec, [636](#)
  - AppendFrameEncode, [636](#)
  - AppendRowEncode, [636](#)
  - Bitmap, [640](#)
  - CanCode, [637](#)
  - CanDecode, [637](#)
  - Clone, [637](#)
  - Code, [637](#)
  - Decode, [637](#)
  - DecodeByStreams, [638](#)
  - DecodeExtent, [638](#)
  - GetHeaderInfo, [638](#)
  - GetQuality, [638](#)
  - GetRate, [638](#)
  - ImageRegionReader, [640](#)
  - IsFrameEncoder, [638](#)
  - IsRowEncoder, [639](#)
  - JPEG2000Codec, [636](#)
  - SetMCT, [639](#)
  - SetNumberOfResolutions, [639](#)
  - SetNumberOfThreadsForDecompression, [639](#)
  - SetQuality, [639](#)
  - SetRate, [639](#)
  - SetReversible, [640](#)
  - SetTileSize, [640](#)
  - StartEncode, [640](#)
  - StopEncode, [640](#)
- gdcmm::JPEG8Codec, [641](#)
  - ~JPEG8Codec, [644](#)
  - DecodeByStreams, [644](#)
  - EncodeBuffer, [644](#)
  - GetHeaderInfo, [645](#)
  - InternalCode, [645](#)
  - IsStateSuspension, [645](#)
  - JPEG8Codec, [644](#)
- gdcmm::JPEGCodec, [646](#)
  - ~JPEGCodec, [649](#)
  - AppendFrameEncode, [650](#)
  - AppendRowEncode, [650](#)
  - BitSample, [654](#)
  - CanCode, [650](#)
  - CanDecode, [650](#)
  - Clone, [650](#)
  - Code, [650](#)
  - ComputeOffsetTable, [651](#)
  - Decode, [651](#)
  - DecodeByStreams, [651](#)
  - DecodeExtent, [651](#)
  - EncodeBuffer, [651](#)
  - GetHeaderInfo, [652](#)
  - GetLossless, [652](#)
  - GetQuality, [652](#)
  - ImageRegionReader, [654](#)
  - IsFrameEncoder, [652](#)
  - IsRowEncoder, [652](#)
  - IsStateSuspension, [652](#)
  - IsValid, [653](#)
  - JPEGCodec, [649](#)
  - Quality, [654](#)
  - SetBitSample, [653](#)
  - SetLossless, [653](#)
  - SetPixelFormat, [653](#)
  - SetQuality, [653](#)
  - StartEncode, [653](#)
  - StopEncode, [654](#)
- gdcmm::JPEGLSCCodec, [655](#)
  - ~JPEGLSCCodec, [658](#)
  - AppendFrameEncode, [658](#)
  - AppendRowEncode, [658](#)
  - CanCode, [659](#)
  - CanDecode, [659](#)
  - Clone, [659](#)
  - Code, [659](#)
  - Decode, [659](#), [660](#)
  - DecodeExtent, [660](#)
  - GetBufferLength, [660](#)
  - GetHeaderInfo, [660](#)
  - GetLossless, [660](#)
  - ImageRegionReader, [662](#)
  - IsFrameEncoder, [661](#)
  - IsRowEncoder, [661](#)
  - JPEGLSCCodec, [658](#)



- SetBufferLength, [661](#)
- SetLossless, [661](#)
- SetLossyError, [661](#)
- StartEncode, [661](#)
- StopEncode, [661](#)
- gdcmm::JSON, [662](#)
  - ~JSON, [662](#)
  - Code, [663](#)
  - Decode, [663](#)
  - GetPrettyPrint, [663](#)
  - JSON, [662](#)
  - PrettyPrintOff, [663](#)
  - PrettyPrintOn, [663](#)
  - SetPrettyPrint, [663](#)
- gdcmm::KAKADUCodec, [664](#)
  - ~KAKADUCodec, [667](#)
  - CanCode, [667](#)
  - CanDecode, [667](#)
  - Clone, [667](#)
  - Code, [667](#)
  - Decode, [667](#)
  - KAKADUCodec, [667](#)
- gdcmm::LO, [668](#)
  - const\_iterator, [669](#)
  - const\_reference, [669](#)
  - const\_reverse\_iterator, [669](#)
  - difference\_type, [669](#)
  - IsValid, [671](#)
  - iterator, [670](#)
  - LO, [670](#), [671](#)
  - pointer, [670](#)
  - reference, [670](#)
  - reverse\_iterator, [670](#)
  - size\_type, [670](#)
  - Superclass, [670](#)
  - value\_type, [670](#)
- gdcmm::LookupTable, [671](#)
  - ~LookupTable, [674](#)
  - Allocate, [675](#)
  - BitSample, [679](#)
  - BLUE, [674](#)
  - Clear, [675](#)
  - Decode, [675](#)
  - Decode8, [675](#)
  - GetBitSample, [676](#)
  - GetBufferAsRGBA, [676](#)
  - GetLUT, [676](#)
  - GetLUTDescriptor, [676](#)
  - GetLUTLength, [676](#)
  - GetPointer, [676](#)
  - GRAY, [674](#)
  - GREEN, [674](#)
  - IncompleteLUT, [679](#)
  - InitializeBlueLUT, [679](#)
  - Initialize, [677](#)
  - InitializeGreenLUT, [677](#)
  - InitializeLUT, [677](#)
  - InitializeRedLUT, [677](#)
  - Internal, [679](#)
  - IsRGB8, [677](#)
  - LookupTable, [674](#), [675](#)
  - LookupTableType, [674](#)
  - Print, [678](#)
  - RED, [674](#)
  - SetBlueLUT, [678](#)
  - SetGreenLUT, [678](#)
  - SetLUT, [678](#)
  - SetRedLUT, [678](#)
  - UNKNOWN, [674](#)
  - WriteBufferAsRGBA, [678](#)
- gdcmm::Macro, [681](#)
  - AddMacroEntry, [682](#)
  - ArrayIncludeMacrosType, [682](#)
  - Clear, [682](#)
  - FindMacroEntry, [683](#)
  - GetMacroEntry, [683](#)
  - GetName, [683](#)
  - Macro, [682](#)
  - MapModuleEntry, [682](#)
  - operator<<, [683](#)
  - SetName, [683](#)
  - Verify, [683](#)
- gdcmm::Macros, [684](#)
  - AddMacro, [685](#)
  - Clear, [685](#)
  - GetMacro, [685](#)
  - IsEmpty, [685](#)
  - Macros, [685](#)
  - ModuleMapType, [684](#)
  - operator<<, [685](#)
- gdcmm::MD5, [687](#)
  - Compute, [688](#)
  - ComputeFile, [688](#)
- gdcmm::MEC\_MR3, [688](#)
  - GetCanonMECMR3Tag, [688](#)
  - GetPMTFInformationDataTag, [688](#)
  - GetToshibaMECMR3Tag, [689](#)
  - Print, [689](#)
- gdcmm::MediaStorage, [689](#)
  - AmbulatoryECGWaveformStorage, [693](#)
  - Audio, [695](#)
  - BasicTextSR, [694](#)
  - BasicVoiceAudioWaveformStorage, [693](#)
  - BreastProjectionXRayImageStorageForPresentation, [695](#)
  - BreastProjectionXRayImageStorageForProcessing, [695](#)
  - BreastTomosynthesisImageStorage, [694](#)

- CardiacElectrophysiologyWaveformStorage, [693](#)
- ComprehensiveSR, [694](#)
- ComputedRadiographylmageStorage, [693](#)
- CSANonImageStorage, [694](#)
- CTImageStorage, [693](#)
- DetachedPatientManagementSOPClass, [694](#)
- DetachedStudyManagementSOPClass, [694](#)
- DetachedVisitManagementSOPClass, [694](#)
- DigitalIntraoralXRayImageStorageForPresentation, [693](#)
- DigitalIntraoralXRayImageStorageForProcessing, [693](#)
- DigitalMammographyImageStorageForPresentation, [693](#)
- DigitalMammographyImageStorageForProcessing, [693](#)
- DigitalXRayImageStorageForPresentation, [693](#)
- DigitalXRayImageStorageForProcessing, [693](#)
- EncapsulatedCDASStorage, [694](#)
- EncapsulatedPDFStorage, [694](#)
- EnhancedCTImageStorage, [693](#)
- EnhancedMRColorImageStorage, [695](#)
- EnhancedMRIImageStorage, [693](#)
- EnhancedPETImageStorage, [694](#)
- EnhancedSR, [694](#)
- EnhancedUSVolumeStorage, [694](#)
- EnhancedXAImageStorage, [694](#)
- FujiPrivateCRLImageStorage, [694](#)
- FujiPrivateMammoCRLImageStorage, [695](#)
- GeneralECGWaveformStorage, [693](#)
- GeneralElectricMagneticResonanceImageStorage, [694](#)
- GEPrivate3DModelStorage, [694](#)
- GetModality, [696](#)
- GetModalityDimension, [696](#)
- GetMSString, [696](#)
- GetMSType, [696](#)
- GetNumberOfModality, [696](#)
- GetNumberOfMSString, [696](#)
- GetNumberOfMSType, [696](#)
- GetString, [696](#)
- GrayscaleSoftcopyPresentationStateStorageSOP-Class, [693](#)
- GuessFromModality, [697](#)
- HangingProtocolStorage, [694](#)
- HardcopyColorImageStorage, [695](#)
- HardcopyGrayscaleImageStorage, [694](#)
- HemodynamicWaveformStorage, [693](#)
- IsImage, [697](#)
- IsUndefined, [697](#)
- IVOCTForPresentation, [695](#)
- IVOCTForProcessing, [695](#)
- KeyObjectSelectionDocument, [694](#)
- LeadECGWaveformStorage, [693](#)
- LegacyConvertedEnhancedCTImageStorage, [695](#)
- LegacyConvertedEnhancedMRIImageStorage, [695](#)
- LegacyConvertedEnhancedPETImageStorage, [695](#)
- MammographyCADSR, [694](#)
- MediaStorage, [695](#)
- MediaStorageDirectoryStorage, [693](#)
- ModalityPerformedProcedureStepSOPClass, [694](#)
- MRIImageStorage, [693](#)
- MRSpectroscopyStorage, [693](#)
- MS\_END, [695](#)
- MSType, [692](#)
- MultiframeGrayscaleByteSecondaryCaptureImageStorage, [693](#)
- MultiframeGrayscaleWordSecondaryCaptureImageStorage, [693](#)
- MultiframeSingleBitSecondaryCaptureImageStorage, [693](#)
- MultiframeTrueColorSecondaryCaptureImageStorage, [693](#)
- NoObject, [695](#)
- NuclearMedicineImageStorage, [693](#)
- NuclearMedicineImageStorageRetired, [693](#)
- ObjectEnd, [695](#)
- ObjectType, [695](#)
- operator MSType, [697](#)
- operator< <, [699](#)
- OphthalmicPhotography16BitImageStorage, [695](#)
- OphthalmicPhotography8BitImageStorage, [694](#)
- OphthalmicTomographyImageStorage, [694](#)
- PDF, [695](#)
- PETImageStorage, [693](#)
- Philips3D, [694](#)
- PhilipsPrivateMRSyntheticImageStorage, [694](#)
- RawDataStorage, [693](#)
- RTDoseStorage, [694](#)
- RTImageStorage, [693](#)
- RTIonBeamsTreatmentRecordStorage, [694](#)
- RTIonPlanStorage, [694](#)
- RTPlanStorage, [694](#)
- RTStructureSetStorage, [694](#)
- RTTreatmentSummaryRecordStorage, [694](#)
- SecondaryCaptureImageStorage, [693](#)
- Segmentation, [695](#)
- SegmentationStorage, [694](#)
- SetFromDataSet, [698](#)
- SetFromFile, [698](#)
- SetFromHeader, [698](#)
- SetFromModality, [698](#)
- SetFromSourceImageSequence, [698](#)
- SpacialFiducialsStorage, [693](#)
- SpacialRegistrationStorage, [693](#)
- StandaloneCurveStorage, [693](#)
- StandaloneModalityLUTStorage, [693](#)
- StandaloneOverlayStorage, [693](#)

- StandaloneVOILUTStorage, [693](#)
- StudyComponentManagementSOPClass, [694](#)
- SurfaceSegmentationStorage, [694](#)
- ToshibaPrivateDataStorage, [694](#)
- UltrasoundImageStorage, [693](#)
- UltrasoundImageStorageRetired, [693](#)
- UltrasoundMultiFrameImageStorage, [693](#)
- UltrasoundMultiFrameImageStorageRetired, [693](#)
- URI, [695](#)
- Video, [695](#)
- VideoEndoscopicImageStorage, [694](#)
- VideoMicroscopicImageStorage, [695](#)
- VideoPhotographicImageStorage, [694](#)
- VLEndoscopicImageStorage, [694](#)
- VLMicroscopicImageStorage, [694](#)
- VLPotographicImageStorage, [694](#)
- VLWholeSlideMicroscopyImageStorage, [694](#)
- Waveform, [695](#)
- XRay3DAngiographicImageStorage, [694](#)
- XRay3DCraniofacialImageStorage, [695](#)
- XRayAngiographicBiPlaneImageStorageRetired, [693](#)
- XRayAngiographicImageStorage, [693](#)
- XRayRadiationDoseSR, [694](#)
- XRayRadiofluoroscopicImageStorage, [693](#)
- gdcM::MemberCommand< T >, [699](#)
  - ~MemberCommand, [702](#)
  - Execute, [703](#)
  - m\_ConstMemberFunction, [704](#)
  - m\_MemberFunction, [704](#)
  - m\_This, [704](#)
  - MemberCommand, [702](#)
  - New, [703](#)
  - operator=, [703](#)
  - Self, [702](#)
  - SetCallbackFunction, [703](#), [704](#)
  - TConstMemberFunctionPointer, [702](#)
  - TMemberFunctionPointer, [702](#)
- gdcM::MeshPrimitive, [705](#)
  - ~MeshPrimitive, [708](#)
  - AddPrimitiveData, [708](#)
  - EDGE, [707](#)
  - FACET, [707](#)
  - GetMPType, [708](#)
  - GetMPTypeString, [708](#)
  - GetNumberOfPrimitivesData, [708](#)
  - GetPrimitiveData, [708](#), [709](#)
  - GetPrimitivesData, [709](#)
  - GetPrimitiveType, [709](#)
  - LINE, [707](#)
  - MeshPrimitive, [708](#)
  - MPTType, [707](#)
  - MPTType\_END, [707](#)
  - PrimitiveData, [710](#)
  - PrimitivesData, [707](#)
  - PrimitiveType, [710](#)
  - SetPrimitiveData, [709](#)
  - SetPrimitivesData, [709](#)
  - SetPrimitiveType, [709](#)
  - TRIANGLE, [707](#)
  - TRIANGLE\_FAN, [707](#)
  - TRIANGLE\_STRIP, [707](#)
  - VERTEX, [707](#)
- gdcM::ModalityPerformedProcedureStepCreateQuery, [710](#)
  - GetAbstractSyntaxUID, [713](#)
  - GetRequiredDataSet, [713](#)
  - ModalityPerformedProcedureStepCreateQuery, [712](#)
  - QueryFactory, [713](#)
  - ValidateQuery, [713](#)
- gdcM::ModalityPerformedProcedureStepSetQuery, [713](#)
  - GetAbstractSyntaxUID, [716](#)
  - GetRequiredDataSet, [716](#)
  - ModalityPerformedProcedureStepSetQuery, [716](#)
  - QueryFactory, [717](#)
  - ValidateQuery, [716](#)
- gdcM::ModifiedEvent, [717](#)
- gdcM::Module, [718](#)
  - AddMacro, [720](#)
  - AddModuleEntry, [720](#)
  - ArrayIncludeMacrosType, [719](#)
  - Clear, [720](#)
  - FindModuleEntryInMacros, [720](#)
  - GetModuleEntryInMacros, [720](#)
  - GetName, [720](#)
  - MapModuleEntry, [719](#)
  - Module, [720](#)
  - operator<<, [721](#)
  - SetName, [721](#)
  - Verify, [721](#)
- gdcM::ModuleEntry, [721](#)
  - ~ModuleEntry, [723](#)
  - DataElementType, [724](#)
  - Description, [723](#)
  - DescriptionField, [724](#)
  - GetDescription, [723](#)
  - GetName, [723](#)
  - GetType, [724](#)
  - ModuleEntry, [723](#)
  - Name, [725](#)
  - operator<<, [724](#)
  - SetDescription, [724](#)
  - SetName, [724](#)
  - SetType, [724](#)
- gdcM::Modules, [725](#)
  - AddModule, [726](#)
  - Clear, [726](#)
  - GetModule, [726](#)
  - IsEmpty, [726](#)

- ModuleMapType, 726
- Modules, 726
- operator<<, 726
- gdcmm::MovePatientRootQuery, 727
  - GetAbstractSyntaxUID, 729
  - GetTagListByLevel, 729
  - InitializeDataSet, 730
  - MovePatientRootQuery, 729
  - QueryFactory, 730
  - ValidateQuery, 730
- gdcmm::MoveStudyRootQuery, 731
  - GetAbstractSyntaxUID, 733
  - GetTagListByLevel, 733
  - InitializeDataSet, 734
  - MoveStudyRootQuery, 733
  - QueryFactory, 734
  - ValidateQuery, 734
- gdcmm::MrProtocol, 735
  - ~MrProtocol, 735
  - FindMrProtocolByName, 736
  - GetMrProtocolByName, 736
  - GetSliceArray, 736
  - GetVersion, 736
  - Load, 736
  - MrProtocol, 735
  - operator<<, 736
  - Print, 736
- gdcmm::MrProtocol::Slice, 1015
  - Normal, 1016
  - Position, 1016
- gdcmm::MrProtocol::SliceArray, 1016
  - Slices, 1017
- gdcmm::MrProtocol::Vector3, 1283
  - dCor, 1283
  - dSag, 1283
  - dTra, 1283
- gdcmm::NestedModuleEntries, 746
  - AddModuleEntry, 748
  - GetModuleEntry, 748
  - GetNumberOfModuleEntries, 748
  - NestedModuleEntries, 748
  - operator<<, 749
  - SizeType, 748
- gdcmm::network, 75
  - cMaxEventID, 80
  - cMaxStateID, 80
  - eAABORTPDUReturnedOpen, 79
  - eAABORTRequest, 79
  - eAASSOCIATE\_RQPDUReturned, 79
  - eAASSOCIATERequestLocalUser, 79
  - eAASSOCIATEResponseAccept, 79
  - eAASSOCIATEResponseReject, 79
  - eARELEASE\_RPPDUReturned, 79
  - eARELEASE\_RQPDUReturnedOpen, 79
  - eARELEASERequest, 79
  - eARELEASEResponse, 79
  - eARTIMTimerExpired, 79
  - eASSOCIATE\_ACPDUReturned, 79
  - eASSOCIATE\_RJPDUReturned, 79
  - eEventDoesNotExist, 79
  - EEventID, 79
  - ePDATArequest, 79
  - ePDATATFPDU, 79
  - eSta10ReleaseCollisionAc, 80
  - eSta11ReleaseCollisionRq, 80
  - eSta12ReleaseCollisionAcLocal, 80
  - eSta13AwaitingClose, 80
  - eSta1Idle, 80
  - eSta2Open, 80
  - eSta3WaitLocalAssoc, 80
  - eSta4LocalAssocDone, 80
  - eSta5WaitRemoteAssoc, 80
  - eSta6TransferReady, 80
  - eSta7WaitRelease, 80
  - eSta8WaitLocalRelease, 80
  - eSta9ReleaseCollisionRqLocal, 80
  - eStaDoesNotExist, 80
  - EStateID, 79
  - eTransportConnConfirmLocal, 79
  - eTransportConnectionClosed, 79
  - eTransportConnIndicLocal, 79
  - eUnrecognizedPDUReturned, 79
  - GetStateIndex, 80
- gdcmm::network::AAabortPDU, 85
  - AAabortPDU, 86
  - IsLastFragment, 86
  - Print, 86
  - Read, 86
  - SetReason, 87
  - SetSource, 87
  - Size, 87
  - Write, 87
- gdcmm::network::AAssociateACPDU, 88
  - AAssociateACPDU, 89
  - AAssociateRQPDUReturned, 91
  - AddPresentationContextAC, 90
  - GetNumberOfPresentationContextAC, 90
  - GetPresentationContextAC, 90
  - GetUserInformation, 90
  - InitFromRQ, 90
  - IsLastFragment, 90
  - Print, 90
  - Read, 90
  - SetCalledAETitle, 91
  - SetCallingAETitle, 91
  - Size, 91
  - SizeType, 89
  - Write, 91

- gdcmm::network::AAssociateRJPDU, 92
  - AAssociateRJPDU, 93
  - IsLastFragment, 93
  - Print, 93
  - Read, 93
  - Size, 93
  - Write, 93
- gdcmm::network::AAssociateRQPDU, 94
  - AAssociateACPDU, 99
  - AAssociateRQPDU, 96
  - AddPresentationContext, 96
  - GetCalledAETitle, 96
  - GetCallingAETitle, 96
  - GetNumberOfPresentationContext, 97
  - GetPresentationContext, 97
  - GetPresentationContextByAbstractSyntax, 97
  - GetPresentationContextById, 97
  - GetPresentationContexts, 97
  - GetReserved43\_74, 97
  - GetUserInformation, 97
  - IsAETitleValid, 97
  - IsLastFragment, 98
  - PresentationContextArrayType, 96
  - Print, 98
  - Read, 98
  - SetCalledAETitle, 98
  - SetCallingAETitle, 98
  - SetUserInformation, 98
  - Size, 98
  - SizeType, 96
  - Write, 99
- gdcmm::network::AbstractSyntax, 100
  - AbstractSyntax, 101
  - GetAsDataElement, 101
  - GetName, 101
  - operator==, 101
  - Print, 101
  - Read, 102
  - SetName, 102
  - SetNameFromUID, 102
  - Size, 102
  - Write, 102
- gdcmm::network::ApplicationContext, 116
  - ApplicationContext, 117
  - GetName, 117
  - Print, 117
  - Read, 117
  - SetName, 117
  - Size, 117
  - Write, 117
- gdcmm::network::AReleaseRPPDU, 120
  - AReleaseRPPDU, 121
  - IsLastFragment, 122
  - Print, 122
  - Read, 122
  - Size, 122
  - Write, 122
- gdcmm::network::AReleaseRQPDU, 123
  - AReleaseRQPDU, 124
  - IsLastFragment, 124
  - Print, 124
  - Read, 124
  - Size, 124
  - Write, 124
- gdcmm::network::ARTIMTimer, 125
  - ARTIMTimer, 125
  - GetElapsedTime, 126
  - GetHasExpired, 126
  - GetTimeout, 126
  - SetTimeout, 126
  - Start, 126
  - Stop, 126
- gdcmm::network::AsynchronousOperationsWindowSub, 128
  - AsynchronousOperationsWindowSub, 129
  - Print, 129
  - Read, 129
  - Size, 129
  - Write, 129
- gdcmm::network::BaseCompositeMessage, 169
  - ~BaseCompositeMessage, 170
  - ConstructPDV, 170
- gdcmm::network::BaseNormalizedMessage, 170
  - ~BaseNormalizedMessage, 172
  - ConstructPDV, 172
- gdcmm::network::BasePDU, 173
  - ~BasePDU, 174
  - IsLastFragment, 174
  - Print, 174
  - Read, 174
  - Size, 174
  - Write, 175
- gdcmm::network::CEchoRQ, 231
  - AffectedSOPClassUID, 232
  - ConstructPDV, 232
  - MessageID, 232
- gdcmm::network::CEchoRSP, 233
  - ConstructPDVByDataSet, 234
- gdcmm::network::CFind, 234
- gdcmm::network::CFindCancelRQ, 234
  - ConstructPDVByDataSet, 235
- gdcmm::network::CFindRQ, 236
  - ConstructPDV, 237
- gdcmm::network::CFindRSP, 237
  - ConstructPDVByDataSet, 238
- gdcmm::network::CMoveCancelRq, 246
  - ConstructPDVByDataSet, 247
- gdcmm::network::CMoveRQ, 247

- ConstructPDV, 248
- gdcmm::network::CMoveRSP, 248
  - ConstructPDVByDataSet, 249
- gdcmm::network::CompositeMessageFactory, 264
  - ConstructCEchoRQ, 265
  - ConstructCFindRQ, 265
  - ConstructCMoveRQ, 265
  - ConstructCStoreRQ, 265
  - ConstructCStoreRSP, 266
- gdcmm::network::CStoreRQ, 300
  - ConstructPDV, 301
- gdcmm::network::CStoreRSP, 301
  - ConstructPDV, 302
- gdcmm::network::DIMSE, 375
  - C\_CANCEL\_RQ, 376
  - C\_ECHO\_RQ, 376
  - C\_ECHO\_RSP, 376
  - C\_FIND\_RQ, 376
  - C\_FIND\_RSP, 376
  - C\_GET\_RQ, 376
  - C\_GET\_RSP, 376
  - C\_MOVE\_RQ, 376
  - C\_MOVE\_RSP, 376
  - C\_STORE\_RQ, 376
  - C\_STORE\_RSP, 376
  - CommandTypes, 376
  - N\_ACTION\_RQ, 376
  - N\_ACTION\_RSP, 376
  - N\_CREATE\_RQ, 376
  - N\_CREATE\_RSP, 376
  - N\_DELETE\_RQ, 376
  - N\_DELETE\_RSP, 376
  - N\_EVENT\_REPORT\_RQ, 376
  - N\_EVENT\_REPORT\_RSP, 376
  - N\_GET\_RQ, 376
  - N\_GET\_RSP, 376
  - N\_SET\_RQ, 376
  - N\_SET\_RSP, 376
- gdcmm::network::ImplementationClassUIDSub, 594
  - ImplementationClassUIDSub, 595
  - Print, 595
  - Read, 595
  - Size, 595
  - Write, 595
- gdcmm::network::ImplementationUIDSub, 596
  - ImplementationUIDSub, 596
  - Write, 596
- gdcmm::network::ImplementationVersionNameSub, 596
  - ImplementationVersionNameSub, 597
  - Print, 597
  - Read, 597
  - Size, 597
  - Write, 597
- gdcmm::network::MaximumLengthSub, 686
  - GetMaximumLength, 686
  - MaximumLengthSub, 686
  - Print, 686
  - Read, 686
  - SetMaximumLength, 687
  - Size, 687
  - Write, 687
- gdcmm::network::NActionRQ, 737
  - ConstructPDV, 738
- gdcmm::network::NActionRSP, 738
  - ConstructPDVByDataSet, 739
- gdcmm::network::NCreateRQ, 740
  - ConstructPDV, 741
- gdcmm::network::NCreateRSP, 741
  - ConstructPDVByDataSet, 742
- gdcmm::network::NDeleteRQ, 743
  - ConstructPDV, 744
- gdcmm::network::NDeleteRSP, 744
  - ConstructPDVByDataSet, 745
- gdcmm::network::NEventReportRQ, 749
  - ConstructPDV, 750
- gdcmm::network::NEventReportRSP, 751
  - ConstructPDVByDataSet, 752
- gdcmm::network::NGetRQ, 752
  - ConstructPDV, 753
- gdcmm::network::NGetRSP, 754
  - ConstructPDVByDataSet, 755
- gdcmm::network::NormalizedMessageFactory, 756
  - ConstructNAction, 756
  - ConstructNCreate, 756
  - ConstructNDelete, 757
  - ConstructNEventReport, 757
  - ConstructNGet, 757
  - ConstructNSet, 757
- gdcmm::network::NSetRQ, 760
  - ConstructPDV, 761
- gdcmm::network::NSetRSP, 761
  - ConstructPDVByDataSet, 762
- gdcmm::network::PDataTFPDU, 795
  - AddPresentationDataValue, 796
  - GetNumberOfPresentationDataValues, 796
  - GetPresentationDataValue, 796
  - IsLastFragment, 796
  - PDataTFPDU, 796
  - Print, 797
  - Read, 797
  - ReadInto, 797
  - Size, 797
  - SizeType, 796
  - Write, 797
- gdcmm::network::PDUFactory, 806
  - ConstructAbortPDU, 807
  - ConstructPDU, 807
  - ConstructReleasePDU, 807

- CreateCEchoPDU, 807
- CreateCFindPDU, 807
- CreateCMovePDU, 807
- CreateCStoreRQPDU, 807
- CreateCStoreRSPPDU, 808
- CreateNActionPDU, 808
- CreateNCreatePDU, 808
- CreateNDeletePDU, 808
- CreateNEventReportPDU, 808
- CreateNGetPDU, 808
- CreateNSetPDU, 808
- DetermineEventByPDU, 809
- GetPDVs, 809
- gdcmm::network::PresentationContextAC, 859
  - GetPresentationContextID, 859
  - GetReason, 859
  - GetTransferSyntax, 860
  - PresentationContextAC, 859
  - Print, 860
  - Read, 860
  - SetPresentationContextID, 860
  - SetReason, 860
  - SetTransferSyntax, 860
  - Size, 860
  - Write, 860
- gdcmm::network::PresentationContextRQ, 864
  - AddTransferSyntax, 865
  - GetAbstractSyntax, 865, 866
  - GetNumberOfTransferSyntaxes, 866
  - GetPresentationContextID, 866
  - GetTransferSyntax, 866
  - GetTransferSyntaxes, 866
  - operator==, 866
  - PresentationContextRQ, 865
  - Print, 866
  - Read, 866
  - SetAbstractSyntax, 867
  - SetPresentationContextID, 867
  - Size, 867
  - SizeType, 865
  - Write, 867
- gdcmm::network::PresentationDataValue, 867
  - ConcatenatePDVBlobs, 868
  - ConcatenatePDVBlobsAsExplicit, 868
  - GetBlob, 869
  - GetIsCommand, 869
  - GetIsLastFragment, 869
  - GetMessageHeader, 869
  - GetPresentationContextID, 869
  - PresentationDataValue, 868
  - Print, 869
  - Read, 869
  - ReadInto, 869
  - SetBlob, 869
  - SetCommand, 870
  - SetDataSet, 870
  - SetLastFragment, 870
  - SetMessageHeader, 870
  - SetPresentationContextID, 870
  - Size, 870
  - Write, 870
- gdcmm::network::RoleSelectionSub, 933
  - Print, 933
  - Read, 933
  - RoleSelectionSub, 933
  - SetTuple, 934
  - Size, 934
  - Write, 934
- gdcmm::network::ServiceClassApplicationInformation, 996
  - Print, 996
  - Read, 996
  - ServiceClassApplicationInformation, 996
  - SetTuple, 996
  - Size, 996
  - Write, 997
- gdcmm::network::SOPClassExtendedNegotiationSub, 1021
  - Print, 1021
  - Read, 1021
  - SetTuple, 1021
  - Size, 1022
  - SOPClassExtendedNegotiationSub, 1021
  - Write, 1022
- gdcmm::network::TableRow, 1122
  - ~TableRow, 1122
  - TableRow, 1122
  - transitions, 1123
- gdcmm::network::TransferSyntaxSub, 1152
  - GetName, 1153
  - operator==, 1153
  - Print, 1153
  - Read, 1153
  - SetName, 1153
  - SetNameFromUID, 1153
  - Size, 1153
  - TransferSyntaxSub, 1152
  - Write, 1153
- gdcmm::network::Transition, 1154
  - ~Transition, 1154
  - mAction, 1155
  - MakeNew, 1155
  - mEnd, 1155
  - Transition, 1154, 1155
- gdcmm::network::ULAction, 1197
  - ~ULAction, 1199
  - operator=, 1200
  - PerformAction, 1200
  - ULAction, 1199



- gdcmm::network::ULActionAA1, 1200
  - PerformAction, 1201
- gdcmm::network::ULActionAA2, 1202
  - PerformAction, 1203
- gdcmm::network::ULActionAA3, 1203
  - PerformAction, 1204
- gdcmm::network::ULActionAA4, 1204
  - PerformAction, 1205
- gdcmm::network::ULActionAA5, 1206
  - PerformAction, 1207
- gdcmm::network::ULActionAA6, 1207
  - PerformAction, 1208
- gdcmm::network::ULActionAA7, 1208
  - PerformAction, 1209
- gdcmm::network::ULActionAA8, 1210
  - PerformAction, 1211
- gdcmm::network::ULActionAE1, 1211
  - PerformAction, 1212
- gdcmm::network::ULActionAE2, 1212
  - PerformAction, 1213
- gdcmm::network::ULActionAE3, 1214
  - PerformAction, 1215
- gdcmm::network::ULActionAE4, 1215
  - PerformAction, 1216
- gdcmm::network::ULActionAE5, 1216
  - PerformAction, 1217
- gdcmm::network::ULActionAE6, 1218
  - PerformAction, 1219
- gdcmm::network::ULActionAE7, 1219
  - PerformAction, 1220
- gdcmm::network::ULActionAE8, 1220
  - PerformAction, 1221
- gdcmm::network::ULActionAR1, 1222
  - PerformAction, 1223
- gdcmm::network::ULActionAR10, 1223
  - PerformAction, 1224
- gdcmm::network::ULActionAR2, 1224
  - PerformAction, 1225
- gdcmm::network::ULActionAR3, 1226
  - PerformAction, 1227
- gdcmm::network::ULActionAR4, 1227
  - PerformAction, 1228
- gdcmm::network::ULActionAR5, 1228
  - PerformAction, 1229
- gdcmm::network::ULActionAR6, 1230
  - PerformAction, 1231
- gdcmm::network::ULActionAR7, 1231
  - PerformAction, 1232
- gdcmm::network::ULActionAR8, 1232
  - PerformAction, 1233
- gdcmm::network::ULActionAR9, 1234
  - PerformAction, 1235
- gdcmm::network::ULActionDT1, 1235
  - PerformAction, 1236
- gdcmm::network::ULActionDT2, 1236
  - PerformAction, 1237
- gdcmm::network::ULBasicCallback, 1238
  - ~ULBasicCallback, 1239
  - GetDataSets, 1239
  - GetResponses, 1239
  - HandleDataSet, 1240
  - HandleResponse, 1240
  - ULBasicCallback, 1239
- gdcmm::network::ULConnection, 1240
  - ~ULConnection, 1241
  - AddAcceptedPresentationContext, 1242
  - FindContext, 1242
  - GetAcceptedPresentationContexts, 1242
  - GetConnectionInfo, 1242
  - GetMaxPDUSize, 1242
  - GetPresentationContextACByID, 1242
  - GetPresentationContextIDFromPresentationContext, 1243
  - GetPresentationContextRQByID, 1243
  - GetPresentationContexts, 1243
  - GetProtocol, 1243
  - GetState, 1243
  - GetTimer, 1243
  - InitializeConnection, 1243
  - InitializeIncomingConnection, 1243
  - operator=, 1244
  - SetMaxPDUSize, 1244
  - SetPresentationContexts, 1244
  - SetState, 1244
  - StopProtocol, 1244
  - ULActionAE6, 1244
  - ULConnection, 1241, 1242
  - ULConnectionManager, 1244
- gdcmm::network::ULConnectionCallback, 1245
  - ~ULConnectionCallback, 1246
  - DataSetHandled, 1246
  - DataSetHandles, 1246
  - HandleDataSet, 1246
  - HandleResponse, 1246
  - mImplicit, 1247
  - ResetHandledDataSet, 1246
  - SetImplicitFlag, 1247
  - ULConnectionCallback, 1246
- gdcmm::network::ULConnectionInfo, 1247
  - GetCalledAETitle, 1248
  - GetCalledComputerName, 1248
  - GetCalledIPAddress, 1248
  - GetCalledIPPort, 1248
  - GetCallingAETitle, 1248
  - GetMaxPDULength, 1248
  - Initialize, 1248
  - SetMaxPDULength, 1248
  - ULConnectionInfo, 1248



- gdcmm::network::ULConnectionManager, 1249
  - ~ULConnectionManager, 1252
  - BreakConnection, 1252
  - BreakConnectionNow, 1252
  - EstablishConnection, 1252
  - EstablishConnectionMove, 1252
  - mConnection, 1255
  - mSecondaryConnection, 1255
  - mTransitions, 1256
  - RunEventLoop, 1252
  - RunMoveEventLoop, 1253
  - SendEcho, 1253
  - SendFind, 1253
  - SendMove, 1253
  - SendNAction, 1253, 1254
  - SendNCreate, 1254
  - SendNDelete, 1254
  - SendNEventReport, 1254
  - SendNGet, 1254, 1255
  - SendNSet, 1255
  - SendStore, 1255
  - ULConnectionManager, 1251
- gdcmm::network::ULEvent, 1256
  - ~ULEvent, 1257
  - GetDataSetPos, 1257
  - GetEvent, 1257
  - GetIStream, 1257
  - GetPDUs, 1257
  - SetEvent, 1257
  - SetPDU, 1258
  - ULEvent, 1257
- gdcmm::network::ULTransitionTable, 1258
  - HandleEvent, 1259
  - PrintTable, 1259
  - ULTransitionTable, 1258
- gdcmm::network::ULWritingCallback, 1259
  - ~ULWritingCallback, 1261
  - HandleDataSet, 1261
  - HandleResponse, 1261
  - SetDirectory, 1261
  - ULWritingCallback, 1261
- gdcmm::network::UserInformation, 1274
  - ~UserInformation, 1275
  - AddRoleSelectionSub, 1275
  - AddSOPClassExtendedNegociationSub, 1275
  - GetMaximumLengthSub, 1275
  - operator=, 1275
  - Print, 1275
  - Read, 1276
  - Size, 1276
  - UserInformation, 1275
  - Write, 1276
- gdcmm::NoEvent, 755
- gdcmm::NormalizedNetworkFunctions, 757
  - ConstructQuery, 758
  - NAction, 758
  - NCreate, 759
  - NDelete, 759
  - NEventReport, 759
  - NGet, 759
  - NSet, 759
- gdcmm::Object, 763
  - ~Object, 764
  - Object, 764
  - operator<<, 765
  - operator=, 765
  - Print, 765
  - Register, 765
  - SmartPointer, 765
  - UnRegister, 765
- gdcmm::OpenSSLCryptoFactory, 766
  - CreateCMSProvider, 767
  - InitOpenSSL, 767
  - OpenSSLCryptoFactory, 767
- gdcmm::OpenSSLCryptographicMessageSyntax, 768
  - ~OpenSSLCryptographicMessageSyntax, 769
  - Decrypt, 769
  - Encrypt, 769
  - GetCipherType, 770
  - OpenSSLCryptographicMessageSyntax, 769
  - ParseCertificateFile, 770
  - ParseKeyFile, 770
  - SetCipherType, 770
  - SetPassword, 770
- gdcmm::OpenSSL7CryptoFactory, 771
  - CreateCMSProvider, 772
  - OpenSSL7CryptoFactory, 772
- gdcmm::OpenSSL7CryptographicMessageSyntax, 773
  - ~OpenSSL7CryptographicMessageSyntax, 774
  - Decrypt, 774
  - Encrypt, 774
  - GetCipherType, 775
  - OpenSSL7CryptographicMessageSyntax, 774
  - ParseCertificateFile, 775
  - ParseKeyFile, 775
  - SetCipherType, 775
  - SetPassword, 775
- gdcmm::Orientation, 776
  - ~Orientation, 777
  - AXIAL, 777
  - CORONAL, 777
  - GetLabel, 778
  - GetMajorAxisFromPatientRelativeDirectionCosine, 778
  - GetObliquityThresholdCosineValue, 778
  - GetType, 778
  - OBLIQUE, 777
  - operator<<, 779

- Orientation, [777](#)
- OrientationType, [777](#)
- Print, [778](#)
- SAGITTAL, [777](#)
- SetObliquityThresholdCosineValue, [778](#)
- UNKNOWN, [777](#)
- gdcmm::Overlay, [779](#)
  - ~Overlay, [782](#)
  - Decompress, [783](#)
  - GetBitPosition, [783](#)
  - GetBitsAllocated, [783](#)
  - GetColumns, [783](#)
  - GetDescription, [783](#)
  - GetGroup, [783](#)
  - GetOrigin, [784](#)
  - GetOverlayData, [784](#)
  - GetOverlayTypeAsString, [784](#)
  - GetOverlayTypeFromString, [784](#)
  - GetRows, [784](#)
  - GetType, [784](#)
  - GetTypeAsEnum, [784](#)
  - GetUnpackBuffer, [785](#)
  - GetUnpackBufferLength, [785](#)
  - GrabOverlayFromPixelData, [785](#)
  - Graphics, [782](#)
  - Invalid, [782](#)
  - IsEmpty, [785](#)
  - IsInPixelData, [785](#)
  - IsZero, [785](#)
  - operator=, [786](#)
  - Overlay, [782](#), [783](#)
  - OverlayType, [782](#)
  - Print, [786](#)
  - ROI, [782](#)
  - SetBitPosition, [786](#)
  - SetBitsAllocated, [786](#)
  - SetColumns, [786](#)
  - SetDescription, [786](#)
  - SetFrameOrigin, [787](#)
  - SetGroup, [787](#)
  - SetNumberOfFrames, [787](#)
  - SetOrigin, [787](#)
  - SetOverlay, [787](#)
  - SetRows, [787](#)
  - SetType, [788](#)
  - Update, [788](#)
- gdcmm::ParseException, [788](#)
  - ~ParseException, [789](#)
  - GetLastElement, [790](#)
  - operator=, [790](#)
  - ParseException, [789](#), [790](#)
  - SetLastElement, [790](#)
- gdcmm::Parser, [790](#)
  - ~Parser, [792](#)
- DuplicateAttributeError, [792](#)
- EndElementHandler, [792](#)
- ErrorType, [792](#)
- GetBuffer, [793](#)
- GetCurrentByteIndex, [793](#)
- GetErrorCode, [793](#)
- GetErrorString, [793](#)
- GetUserData, [793](#)
- JunkAfterDocElementError, [792](#)
- NoElementsError, [792](#)
- NoError, [792](#)
- NoMemoryError, [792](#)
- Parse, [793](#)
- ParseBuffer, [793](#)
- Parser, [792](#)
- Process, [793](#)
- SetElementHandler, [793](#)
- SetUserData, [794](#)
- StartElementHandler, [792](#)
- SyntaxError, [792](#)
- TagMismatchError, [792](#)
- UndefinedEntityError, [792](#)
- UnexpectedStateError, [792](#)
- gdcmm::Patient, [794](#)
  - Patient, [794](#)
- gdcmm::PDBelement, [798](#)
  - GetName, [799](#)
  - GetValue, [799](#)
  - NameField, [800](#)
  - operator<<, [800](#)
  - operator==, [799](#)
  - PDBelement, [799](#)
  - SetName, [799](#)
  - SetValue, [799](#)
  - ValueField, [800](#)
- gdcmm::PDBHeader, [800](#)
  - ~PDBHeader, [802](#)
  - FindPDBelementByName, [802](#)
  - GetPDBelementEnd, [802](#)
  - GetPDBelementByName, [802](#)
  - GetPDBInfoTag, [802](#)
  - LoadFromDataElement, [802](#)
  - operator<<, [803](#)
  - PDBHeader, [802](#)
  - Print, [803](#)
- gdcmm::PDFCodec, [803](#)
  - ~PDFCodec, [805](#)
  - CanCode, [805](#)
  - CanDecode, [805](#)
  - Decode, [805](#)
  - PDFCodec, [805](#)
- gdcmm::PersonName, [809](#)
  - Component, [811](#)
  - GetMaxLength, [810](#)

- GetNumberOfComponents, [810](#)
- MaxLength, [811](#)
- MaxNumberOfComponents, [811](#)
- Padding, [811](#)
- Print, [810](#)
- Separator, [811](#)
- SetBlob, [810](#)
- SetComponents, [810](#)
- gdcmm::PGXCodec, [812](#)
  - ~PGXCodec, [815](#)
  - CanCode, [815](#)
  - CanDecode, [815](#)
  - Clone, [815](#)
  - GetHeaderInfo, [815](#)
  - PGXCodec, [815](#)
  - Read, [815](#)
  - Write, [816](#)
- gdcmm::PhotometricInterpretation, [816](#)
  - ARGB, [818](#)
  - CMYK, [818](#)
  - GetPIString, [818](#)
  - GetPIType, [818](#)
  - GetSamplesPerPixel, [818](#)
  - GetString, [818](#)
  - GetType, [819](#)
  - HSV, [817](#)
  - IsLossless, [819](#)
  - IsLossy, [819](#)
  - IsRetired, [819](#)
  - IsSameColorSpace, [819](#)
  - MONOCHROME1, [817](#)
  - MONOCHROME2, [817](#)
  - operator PType, [819](#)
  - operator <, [819](#)
  - PALETTE\_COLOR, [817](#)
  - PhotometricInterpretation, [818](#)
  - PI\_END, [818](#)
  - PType, [817](#)
  - RGB, [817](#)
  - UNKNOWN, [817](#)
  - YBR\_FULL, [818](#)
  - YBR\_FULL\_422, [818](#)
  - YBR\_ICT, [818](#)
  - YBR\_PARTIAL\_420, [818](#)
  - YBR\_PARTIAL\_422, [818](#)
  - YBR\_RCT, [818](#)
- gdcmm::PixelFormat, [820](#)
  - Bitmap, [827](#)
  - FLOAT16, [822](#)
  - FLOAT32, [822](#)
  - FLOAT64, [822](#)
  - GetBitsAllocated, [823](#)
  - GetBitsStored, [823](#)
  - GetHighBit, [823](#)
  - GetMax, [823](#)
  - GetMin, [823](#)
  - GetPixelRepresentation, [823](#)
  - GetPixelSize, [824](#)
  - GetSamplesPerPixel, [824](#)
  - GetScalarType, [824](#)
  - GetScalarTypeAsString, [824](#)
  - INT12, [822](#)
  - INT16, [822](#)
  - INT32, [822](#)
  - INT64, [822](#)
  - INT8, [822](#)
  - IsCompatible, [825](#)
  - IsValid, [825](#)
  - operator ScalarType, [825](#)
  - operator !=, [825](#)
  - operator < <, [827](#)
  - operator ==, [825](#)
  - PixelFormat, [822](#)
  - Print, [825](#)
  - ScalarType, [822](#)
  - SetBitsAllocated, [826](#)
  - SetBitsStored, [826](#)
  - SetHighBit, [826](#)
  - SetPixelRepresentation, [826](#)
  - SetSamplesPerPixel, [826](#)
  - SetScalarType, [826](#)
  - SINGLEBIT, [822](#)
  - UINT12, [822](#)
  - UINT16, [822](#)
  - UINT32, [822](#)
  - UINT64, [822](#)
  - UINT8, [822](#)
  - UNKNOWN, [822](#)
  - Validate, [827](#)
- gdcmm::Pixmap, [828](#)
  - ~Pixmap, [832](#)
  - AreOverlaysInPixelData, [832](#)
  - Curves, [834](#)
  - GetCurve, [832](#)
  - GetIconImage, [832](#)
  - GetNumberOfCurves, [832](#)
  - GetNumberOfOverlays, [833](#)
  - GetOverlay, [833](#)
  - Icon, [834](#)
  - Overlays, [834](#)
  - Pixmap, [832](#)
  - Print, [833](#)
  - RemoveOverlay, [833](#)
  - SetIconImage, [833](#)
  - SetNumberOfCurves, [833](#)
  - SetNumberOfOverlays, [833](#)
  - UnusedBitsPresentInPixelData, [834](#)
- gdcmm::PixmapReader, [835](#)

- ~PixmapReader, 838
- GetPixmap, 838
- ImageData, 839
- PixmapReader, 838
- Read, 838
- ReadACRNEMAIImage, 838
- ReadImage, 839
- ReadImageInternal, 839
- gdcmm::PixmapToPixmapFilter, 839
  - ~PixmapToPixmapFilter, 841
  - GetInput, 841
  - GetOutput, 841
  - GetOutputAsPixmap, 841
  - PixmapToPixmapFilter, 841
- gdcmm::PixmapWriter, 842
  - ~PixmapWriter, 845
  - DolconImage, 845
  - GetImage, 845
  - GetPixmap, 845
  - ImageData, 846
  - PixmapWriter, 845
  - PrepareWrite, 845
  - SetImage, 846
  - SetPixmap, 846
  - Write, 846
- gdcmm::PNMCodec, 847
  - ~PNMCodec, 850
  - CanCode, 850
  - CanDecode, 850
  - Clone, 850
  - GetBufferLength, 851
  - GetHeaderInfo, 851
  - PNMCodec, 850
  - Read, 851
  - SetBufferLength, 851
  - Write, 851
- gdcmm::Preamble, 852
  - ~Preamble, 853
  - Clear, 853
  - Create, 853
  - GetInternal, 853
  - GetLength, 853
  - IsEmpty, 853
  - IsValid, 854
  - operator<=, 855
  - operator=, 854
  - Preamble, 853
  - Print, 854
  - Read, 854
  - Remove, 854
  - Valid, 854
  - Write, 854
- gdcmm::PresentationContext, 855
  - AbstractSyntax, 858
  - AddTransferSyntax, 857
  - GetAbstractSyntax, 857
  - GetNumberOfTransferSyntaxes, 857
  - GetPresentationContextID, 857
  - GetTransferSyntax, 857
  - ID, 858
  - operator==, 857
  - PresentationContext, 857
  - Print, 858
  - SetAbstractSyntax, 858
  - SetPresentationContextID, 858
  - SizeType, 856
  - TransferSyntaxArrayType, 856
  - TransferSyntaxes, 858
- gdcmm::PresentationContextGenerator, 861
  - AddFromFile, 863
  - AddPresentationContext, 863
  - GenerateFromFilenames, 863
  - GenerateFromUID, 863
  - GetDefaultTransferSyntax, 863
  - GetPresentationContexts, 863
  - PresentationContextArrayType, 862
  - PresentationContextGenerator, 862
  - SetDefaultTransferSyntax, 863
  - SetMergeModeToAbstractSyntax, 864
  - SetMergeModeToTransferSyntax, 864
  - SizeType, 862
- gdcmm::Printer, 871
  - ~Printer, 873
  - CONDENSED\_STYLE, 873
  - CXX, 873
  - F, 874
  - GetPrintStyle, 873
  - MaxPrintLength, 874
  - Print, 873
  - PrintDataElement, 873
  - PrintDataSet, 873
  - Printer, 873
  - PrintSQ, 874
  - PrintStyle, 875
  - PrintStyles, 872
  - SetColor, 874
  - SetFile, 874
  - SetStyle, 874
  - VERBOSE\_STYLE, 873
  - XML, 873
- gdcmm::PrivateDict, 875
  - ~PrivateDict, 876
  - AddDictEntry, 876
  - Dicts, 877
  - FindDictEntry, 876
  - GetDictEntry, 876
  - IsEmpty, 876
  - LoadDefault, 876

- operator<<, [877](#)
- PrintXML, [876](#)
- PrivateDict, [876](#)
- RemoveDictEntry, [877](#)
- gdcmm::PrivateTag, [877](#)
  - GetAsDataElement, [881](#)
  - GetOwner, [881](#)
  - operator!=, [881](#)
  - operator<, [881](#)
  - operator<<, [882](#)
  - operator=, [881](#)
  - operator==, [881](#), [882](#)
  - PrivateTag, [880](#)
  - ReadFromCommaSeparatedString, [882](#)
  - SetOwner, [882](#)
- gdcmm::ProgressEvent, [883](#)
  - ~ProgressEvent, [885](#)
  - CheckEvent, [885](#)
  - GetEventName, [885](#)
  - GetProgress, [885](#)
  - MakeObject, [885](#)
  - operator=, [886](#)
  - ProgressEvent, [885](#)
  - Self, [884](#)
  - SetProgress, [886](#)
  - Superclass, [884](#)
- gdcmm::PVRGCodec, [886](#)
  - ~PVRGCodec, [889](#)
  - CanCode, [889](#)
  - CanDecode, [889](#)
  - Clone, [889](#)
  - Code, [890](#)
  - Decode, [890](#)
  - PVRGCodec, [889](#)
  - SetLossyFlag, [890](#)
- gdcmm::PythonFilter, [890](#)
  - ~PythonFilter, [891](#)
  - GetFile, [891](#)
  - PythonFilter, [891](#)
  - SetDicts, [891](#)
  - SetFile, [891](#)
  - ToPyObject, [892](#)
  - UseDictAlways, [892](#)
- gdcmm::QueryBase, [892](#)
  - ~QueryBase, [893](#)
  - GetAllRequiredTags, [893](#)
  - GetAllTags, [893](#)
  - GetHierarchicalSearchTags, [893](#)
  - GetName, [894](#)
  - GetOptionalTags, [894](#)
  - GetQueryLevel, [894](#)
  - GetRequiredTags, [894](#)
  - GetUniqueTags, [894](#)
- gdcmm::QueryFactory, [895](#)
  - GetCharacterFromCurrentLocale, [895](#)
  - ListCharSets, [895](#)
  - ProduceCharacterSetDataElement, [895](#)
  - ProduceQuery, [896](#)
- gdcmm::QueryImage, [896](#)
  - GetHierarchicalSearchTags, [897](#)
  - GetName, [897](#)
  - GetOptionalTags, [898](#)
  - GetQueryLevel, [898](#)
  - GetRequiredTags, [898](#)
  - GetUniqueTags, [898](#)
- gdcmm::QueryPatient, [899](#)
  - GetHierarchicalSearchTags, [900](#)
  - GetName, [900](#)
  - GetOptionalTags, [900](#)
  - GetQueryLevel, [900](#)
  - GetRequiredTags, [900](#)
  - GetUniqueTags, [901](#)
- gdcmm::QuerySeries, [901](#)
  - GetHierarchicalSearchTags, [902](#)
  - GetName, [902](#)
  - GetOptionalTags, [903](#)
  - GetQueryLevel, [903](#)
  - GetRequiredTags, [903](#)
  - GetUniqueTags, [903](#)
- gdcmm::QueryStudy, [904](#)
  - GetHierarchicalSearchTags, [905](#)
  - GetName, [905](#)
  - GetOptionalTags, [905](#)
  - GetQueryLevel, [905](#)
  - GetRequiredTags, [905](#)
  - GetUniqueTags, [906](#)
- gdcmm::RAWCodec, [906](#)
  - ~RAWCodec, [909](#)
  - CanCode, [909](#)
  - CanDecode, [909](#)
  - Clone, [909](#)
  - Code, [910](#)
  - Decode, [910](#)
  - DecodeByStreams, [910](#)
  - DecodeBytes, [910](#)
  - GetHeaderInfo, [910](#)
  - RAWCodec, [909](#)
- gdcmm::Reader, [911](#)
  - ~Reader, [914](#)
  - CanRead, [914](#)
  - F, [917](#)
  - GetFile, [914](#)
  - GetStreamCurrentPosition, [914](#)
  - GetStreamPtr, [915](#)
  - Read, [915](#)
  - ReadDataSet, [915](#)
  - Reader, [914](#)
  - ReadMetaInformation, [915](#)

- ReadPreamble, 915
- ReadSelectedPrivateTags, 916
- ReadSelectedTags, 916
- ReadUpToTag, 916
- SetFile, 916
- SetFileName, 916
- SetStream, 917
- StreamImageReader, 917
- gdcm::RealWorldValueMappingContent, 918
  - CodeMeaning, 918
  - CodeValue, 918
  - RealWorldValueIntercept, 918
  - RealWorldValueSlope, 919
- gdcm::Region, 919
  - ~Region, 920
  - Area, 920
  - Clone, 920
  - ComputeBoundingBox, 920
  - Empty, 920
  - IsValid, 921
  - Print, 921
  - Region, 920
- gdcm::Rescaler, 921
  - ~Rescaler, 923
  - ComputeInterceptSlopePixelType, 923
  - ComputePixelTypeFromMinMax, 923
  - GetIntercept, 924
  - GetSlope, 924
  - InverseRescale, 924
  - InverseRescaleFunctionIntoBestFit, 924
  - Rescale, 924
  - RescaleFunctionIntoBestFit, 924
  - Rescaler, 923
  - SetIntercept, 925
  - SetMinMaxForPixelType, 925
  - SetPixelFormat, 925
  - SetSlope, 925
  - SetTargetPixelType, 925
  - SetUseTargetPixelType, 926
- gdcm::RLECodec, 926
  - ~RLECodec, 929
  - AppendFrameEncode, 930
  - AppendRowEncode, 930
  - CanCode, 930
  - CanDecode, 930
  - Clone, 930
  - Code, 930
  - Decode, 931
  - DecodeByStreams, 931
  - DecodeExtent, 931
  - GetBufferLength, 931
  - GetHeaderInfo, 931
  - ImageRegionReader, 933
  - IsFrameEncoder, 932
  - IsRowEncoder, 932
  - RLECodec, 929
  - SetBufferLength, 932
  - SetLength, 932
  - StartEncode, 932
  - StopEncode, 932
- gdcm::Scanner, 934
  - ~Scanner, 938
  - AddPrivateTag, 938
  - AddSkipTag, 938
  - AddTag, 939
  - Begin, 939
  - ClearSkipTags, 939
  - ClearTags, 939
  - ConstIterator, 938
  - End, 939
  - GetAllFilenamesFromTagToValue, 939
  - GetFilenameFromTagToValue, 939
  - GetFilenames, 940
  - GetKeys, 940
  - GetMapping, 940
  - GetMappingFromTagToValue, 940
  - GetMappings, 940
  - GetOrderedValues, 941
  - GetValue, 941
  - GetValues, 941
  - IsKey, 941
  - MappingType, 938
  - New, 942
  - operator<<, 943
  - Print, 942
  - PrintTable, 942
  - ProcessPublicTag, 942
  - Scan, 942
  - Scanner, 938
  - TagToValue, 938
  - TagToValueValueType, 938
  - ValuesType, 938
- gdcm::Scanner2, 943
  - ~Scanner2, 948
  - AddPrivateTag, 948
  - AddPublicTag, 948
  - AddSkipTag, 948
  - Begin, 948
  - ClearPrivateTags, 948
  - ClearPublicTags, 949
  - ClearSkipTags, 949
  - End, 949
  - GetAllFilenamesFromPrivateTagToValue, 949
  - GetAllFilenamesFromPublicTagToValue, 949
  - GetFilenameFromPrivateTagToValue, 949
  - GetFilenameFromPublicTagToValue, 949
  - GetFilenames, 949
  - GetKeys, 950

- GetMappingFromPrivateTagToValue, 950
- GetMappingFromPublicTagToValue, 950
- GetPrivateMapping, 950
- GetPrivateMappings, 950
- GetPrivateOrderedValues, 950
- GetPrivateValue, 950
- GetPrivateValues, 951
- GetPublicMapping, 951
- GetPublicMappings, 951
- GetPublicOrderedValues, 951
- GetPublicValue, 951
- GetPublicValues, 951
- GetValues, 952
- IsKey, 952
- New, 952
- operator<=, 953
- Print, 952
- PrintTable, 952
- PrivateBegin, 952
- PrivateConstIterator, 947
- PrivateEnd, 953
- PrivateMappingType, 947
- PrivateTagToValue, 947
- PrivateTagToValueValueType, 947
- ProcessPrivateTag, 953
- ProcessPublicTag, 953
- PublicConstIterator, 947
- PublicMappingType, 947
- PublicTagToValue, 947
- PublicTagToValueValueType, 947
- Scan, 953
- Scanner2, 948
- ValuesType, 947
- gdcm::Scanner2::Itstr, 679
  - operator(), 679
- gdcm::Scanner::Itstr, 680
  - operator(), 680
- gdcm::Segment, 954
  - ~Segment, 957
  - AddSurface, 957
  - ALGOType, 956
  - ALGOType\_END, 957
  - AnatomicRegion, 961
  - AnatomicRegionModifiers, 961
  - AUTOMATIC, 957
  - BasicCodedEntryVector, 956
  - GetALGOType, 957
  - GetALGOTypeString, 957
  - GetAnatomicRegion, 957
  - GetAnatomicRegionModifiers, 958
  - GetPropertyCategory, 958
  - GetPropertyType, 958
  - GetPropertyTypeModifiers, 958
  - GetSegmentAlgorithmName, 958
  - GetSegmentAlgorithmType, 958
  - GetSegmentDescription, 959
  - GetSegmentLabel, 959
  - GetSegmentNumber, 959
  - GetSurface, 959
  - GetSurfaceCount, 959
  - GetSurfaces, 959
  - MANUAL, 957
  - PropertyCategory, 961
  - PropertyType, 961
  - PropertyTypeModifiers, 961
  - Segment, 957
  - SegmentAlgorithmName, 961
  - SegmentAlgorithmType, 961
  - SegmentDescription, 962
  - SegmentLabel, 962
  - SegmentNumber, 962
  - SEMAUTOMATIC, 957
  - SetAnatomicRegion, 959
  - SetAnatomicRegionModifiers, 959
  - SetPropertyCategory, 960
  - SetPropertyType, 960
  - SetPropertyTypeModifiers, 960
  - SetSegmentAlgorithmName, 960
  - SetSegmentAlgorithmType, 960
  - SetSegmentDescription, 960
  - SetSegmentLabel, 960
  - SetSegmentNumber, 961
  - SetSurfaceCount, 961
  - SurfaceCount, 962
  - Surfaces, 962
  - SurfaceVector, 956
- gdcm::SegmentedPaletteColorLookupTable, 963
  - ~SegmentedPaletteColorLookupTable, 965
  - Print, 966
  - SegmentedPaletteColorLookupTable, 965
  - SetLUT, 966
- gdcm::SegmentHelper, 81
- gdcm::SegmentHelper::BasicCodedEntry, 185
  - BasicCodedEntry, 186
  - CM, 187
  - CSD, 187
  - CSV, 187
  - CV, 187
  - IsEmpty, 187
- gdcm::SegmentReader, 966
  - ~SegmentReader, 969
  - GetSegments, 969
  - Read, 969
  - ReadSegment, 969
  - ReadSegments, 970
  - SegmentMap, 969
  - SegmentReader, 969
  - Segments, 970



- SegmentVector, 969
- gdcmm::SegmentWriter, 970
  - ~SegmentWriter, 972
  - AddSegment, 973
  - GetNumberOfSegments, 973
  - GetSegment, 973
  - GetSegments, 973
  - PrepareWrite, 973
  - Segments, 974
  - SegmentVector, 972
  - SegmentWriter, 972
  - SetNumberOfSegments, 973
  - SetSegments, 973
  - Write, 973
- gdcmm::SequenceOfFragments, 974
  - AddFragment, 978
  - Begin, 978
  - Clear, 978
  - ComputeByteLength, 978
  - ComputeLength, 978
  - ConstIterator, 977
  - End, 978
  - FragmentVector, 977
  - GetBuffer, 978
  - GetFragBuffer, 979
  - GetFragment, 979
  - GetLength, 979
  - GetNumberOfFragments, 979
  - GetTable, 979
  - Iterator, 977
  - New, 980
  - operator==, 980
  - Print, 980
  - Read, 980
  - ReadPreValue, 980
  - ReadValue, 980
  - SequenceOfFragments, 977
  - SetLength, 981
  - SizeType, 977
  - Write, 981
  - WriteBuffer, 981
- gdcmm::SequenceOfItems, 982
  - AddItem, 985
  - AddNewUndefinedLengthItem, 985
  - Begin, 986
  - Clear, 986
  - ComputeLength, 986
  - ConstIterator, 985
  - End, 986
  - FindDataElement, 986
  - GetItem, 986, 987
  - GetLength, 987
  - GetNumberOfItems, 987
  - IsEmpty, 987
  - IsUndefinedLength, 987
  - Items, 989
  - ItemVector, 985
  - Iterator, 985
  - New, 987
  - operator=, 988
  - operator==, 988
  - Print, 988
  - Read, 988
  - RemoveItemByIndex, 988
  - SequenceLengthField, 989
  - SequenceOfItems, 985
  - SetLength, 989
  - SetLengthToUndefined, 989
  - SetNumberOfItems, 989
  - SizeType, 985
  - Write, 989
- gdcmm::SerieHelper, 990
  - ~SerieHelper, 992
  - AddFile, 992
  - AddFileName, 992
  - AddRestriction, 992, 993
  - Clear, 993
  - CreateDefaultUniqueSeriesIdentifier, 993
  - CreateUniqueSeriesIdentifier, 993
  - elem, 994
  - FileNameOrdering, 993
  - GetFirstSingleSerieUIDFileSet, 993
  - GetNextSingleSerieUIDFileSet, 993
  - ImageNumberOrdering, 993
  - ImagePositionPatientOrdering, 994
  - ItFileSetHt, 994
  - op, 995
  - OrderFileList, 994
  - Rule, 992
  - SerieHelper, 992
  - SerieRestrictions, 992
  - SetDirectory, 994
  - SetLoadMode, 994
  - SetUseSeriesDetails, 994
  - SingleSerieUIDFileSetHT, 995
  - SingleSerieUIDFileSetmap, 992
  - UserOrdering, 994
  - value, 995
- gdcmm::Series, 995
  - Series, 995
- gdcmm::ServiceClassUser, 997
  - ~ServiceClassUser, 1000
  - GetAETitle, 1000
  - GetCalledAETitle, 1000
  - GetTimeout, 1001
  - InitializeConnection, 1001
  - IsPresentationContextAccepted, 1001
  - New, 1001



- operator=, [1001](#)
- SendEcho, [1001](#)
- SendFind, [1001](#)
- SendMove, [1002](#)
- SendStore, [1002](#), [1003](#)
- ServiceClassUser, [1000](#)
- SetAETitle, [1003](#)
- SetCalledAETitle, [1003](#)
- SetHostname, [1003](#)
- SetPort, [1003](#)
- SetPortSCP, [1004](#)
- SetPresentationContexts, [1004](#)
- SetTimeout, [1004](#)
- StartAssociation, [1004](#)
- StopAssociation, [1005](#)
- gdcm::SHA1, [1005](#)
  - ~SHA1, [1006](#)
  - Compute, [1006](#)
  - ComputeFile, [1006](#)
  - operator=, [1006](#)
  - SHA1, [1006](#)
- gdcm::SimpleMemberCommand< T >, [1007](#)
  - ~SimpleMemberCommand, [1010](#)
  - Execute, [1011](#)
  - m\_MemberFunction, [1012](#)
  - m\_This, [1012](#)
  - New, [1011](#)
  - operator=, [1011](#)
  - Self, [1010](#)
  - SetCallbackFunction, [1011](#)
  - SimpleMemberCommand, [1010](#)
  - TMemberFunctionPointer, [1010](#)
- gdcm::SimpleSubjectWatcher, [1012](#)
  - ~SimpleSubjectWatcher, [1013](#)
  - EndFilter, [1014](#)
  - operator=, [1014](#)
  - ShowAbort, [1014](#)
  - ShowAnonymization, [1014](#)
  - ShowData, [1014](#)
  - ShowDataSet, [1014](#)
  - ShowFileName, [1014](#)
  - ShowIteration, [1014](#)
  - ShowProgress, [1015](#)
  - SimpleSubjectWatcher, [1013](#)
  - StartFilter, [1015](#)
  - TestAbortOff, [1015](#)
  - TestAbortOn, [1015](#)
- gdcm::SmartPointer< ObjectType >, [1017](#)
  - ~SmartPointer, [1019](#)
  - GetPointer, [1019](#)
  - operator ObjectType \*, [1019](#)
  - operator->, [1020](#)
  - operator=, [1020](#)
  - operator\*, [1020](#)
  - SmartPointer, [1019](#)
- gdcm::SOPClassUIDToIOD, [1022](#)
  - const, [1023](#)
  - GetIOD, [1023](#)
  - GetIODFromSOPClassUID, [1023](#)
  - GetNumberOfSOPClassToIOD, [1023](#)
  - GetSOPClassUIDFromIOD, [1023](#)
  - GetSOPClassUIDToIOD, [1023](#)
  - GetSOPClassUIDToIODs, [1024](#)
- gdcm::Sorter, [1024](#)
  - ~Sorter, [1026](#)
  - AddSelect, [1026](#)
  - FileNames, [1028](#)
  - GetFileNames, [1026](#)
  - operator<<, [1028](#)
  - Print, [1026](#)
  - Selection, [1028](#)
  - SelectionMap, [1026](#)
  - SetSortFunction, [1027](#)
  - SetTagsToRead, [1027](#)
  - Sort, [1027](#)
  - Sorter, [1026](#)
  - SortFunc, [1028](#)
  - SortFunction, [1026](#)
  - StableSort, [1027](#)
  - TagsToRead, [1028](#)
- gdcm::Spacing, [1029](#)
  - ~Spacing, [1030](#)
  - CALIBRATED, [1030](#)
  - ComputePixelAspectRatioFromPixelSpacing, [1031](#)
  - DETECTOR, [1030](#)
  - MAGNIFIED, [1030](#)
  - Spacing, [1030](#)
  - SpacingType, [1030](#)
  - UNKNOWN, [1030](#)
- gdcm::Spectroscopy, [1031](#)
  - Spectroscopy, [1031](#)
- gdcm::SplitMosaicFilter, [1032](#)
  - ~SplitMosaicFilter, [1033](#)
  - ComputeMOSAICDimensions, [1033](#)
  - ComputeMOSAICImagePositionPatient, [1033](#)
  - ComputeMOSAICSliceNormal, [1033](#)
  - ComputeMOSAICSlicePosition, [1033](#)
  - GetAcquisitionSize, [1033](#)
  - GetFile, [1034](#)
  - GetImage, [1034](#)
  - GetNumberOfImagesInMosaic, [1034](#)
  - SetFile, [1034](#)
  - SetImage, [1034](#)
  - Split, [1034](#)
  - SplitMosaicFilter, [1033](#)
- gdcm::StartEvent, [1035](#)
- gdcm::static\_assert\_test< x >, [1036](#)
- gdcm::STATIC\_ASSERTION\_FAILURE< true >, [1037](#)

- value, [1037](#)
- gdcm::STATIC\_ASSERTION\_FAILURE< x >, [1037](#)
- gdcm::StreamImageReader, [1037](#)
  - ~StreamImageReader, [1038](#)
  - CanReadImage, [1039](#)
  - DefinePixelExtent, [1039](#)
  - DefineProperBufferLength, [1039](#)
  - GetDimensionsValueForResolution, [1039](#)
  - GetFile, [1040](#)
  - Read, [1040](#)
  - ReadImageInformation, [1040](#)
  - SetFileName, [1040](#)
  - SetStream, [1041](#)
  - StreamImageReader, [1038](#)
- gdcm::StreamImageWriter, [1041](#)
  - ~StreamImageWriter, [1043](#)
  - CanWriteFile, [1044](#)
  - DefinePixelExtent, [1044](#)
  - DefineProperBufferLength, [1044](#)
  - mElementOffsets, [1046](#)
  - mElementOffsets1, [1046](#)
  - mSPFile, [1046](#)
  - mWriter, [1047](#)
  - mXMax, [1047](#)
  - mXMin, [1047](#)
  - mYMax, [1047](#)
  - mYMin, [1047](#)
  - mZMax, [1047](#)
  - mZMin, [1047](#)
  - SetFile, [1044](#)
  - SetFileName, [1045](#)
  - SetStream, [1045](#)
  - StreamImageWriter, [1043](#)
  - Write, [1045](#)
  - WriteImageInformation, [1045](#)
  - WriteImageSubregionRAW, [1046](#)
  - WriteRawHeader, [1046](#)
- gdcm::StrictScanner, [1048](#)
  - ~StrictScanner, [1052](#)
  - AddPrivateTag, [1052](#)
  - AddSkipTag, [1052](#)
  - AddTag, [1052](#)
  - Begin, [1052](#)
  - ClearSkipTags, [1053](#)
  - ClearTags, [1053](#)
  - ConstIterator, [1051](#)
  - End, [1053](#)
  - GetAllFileNamesFromTagToValue, [1053](#)
  - GetFilenameFromTagToValue, [1053](#)
  - GetFileNames, [1053](#)
  - GetKeys, [1053](#)
  - GetMapping, [1053](#)
  - GetMappingFromTagToValue, [1054](#)
  - GetMappings, [1054](#)
  - GetOrderedValues, [1054](#)
  - GetValue, [1054](#)
  - GetValues, [1054, 1055](#)
  - IsKey, [1055](#)
  - MappingType, [1051](#)
  - New, [1055](#)
  - operator<<, [1056](#)
  - Print, [1055](#)
  - PrintTable, [1055](#)
  - ProcessPublicTag, [1056](#)
  - Scan, [1056](#)
  - StrictScanner, [1052](#)
  - TagToValue, [1051](#)
  - TagToValueValueType, [1051](#)
  - ValuesType, [1052](#)
- gdcm::StrictScanner2, [1057](#)
  - ~StrictScanner2, [1061](#)
  - AddPrivateTag, [1062](#)
  - AddPublicTag, [1062](#)
  - AddSkipTag, [1062](#)
  - Begin, [1062](#)
  - ClearPrivateTags, [1062](#)
  - ClearPublicTags, [1062](#)
  - ClearSkipTags, [1062](#)
  - End, [1062](#)
  - GetAllFileNamesFromPrivateTagToValue, [1062](#)
  - GetAllFileNamesFromPublicTagToValue, [1063](#)
  - GetFilenameFromPrivateTagToValue, [1063](#)
  - GetFilenameFromPublicTagToValue, [1063](#)
  - GetFileNames, [1063](#)
  - GetKeys, [1063](#)
  - GetMappingFromPrivateTagToValue, [1063](#)
  - GetMappingFromPublicTagToValue, [1063](#)
  - GetPrivateMapping, [1064](#)
  - GetPrivateMappings, [1064](#)
  - GetPrivateOrderedValues, [1064](#)
  - GetPrivateValue, [1064](#)
  - GetPrivateValues, [1064](#)
  - GetPublicMapping, [1064](#)
  - GetPublicMappings, [1064](#)
  - GetPublicOrderedValues, [1065](#)
  - GetPublicValue, [1065](#)
  - GetPublicValues, [1065](#)
  - GetValues, [1065](#)
  - IsKey, [1065](#)
  - New, [1065](#)
  - operator<<, [1067](#)
  - Print, [1066](#)
  - PrintTable, [1066](#)
  - PrivateBegin, [1066](#)
  - PrivateConstIterator, [1060](#)
  - PrivateEnd, [1066](#)
  - PrivateMappingType, [1060](#)
  - PrivateTagToValue, [1060](#)

- PrivateTagToValueValueType, 1060
- ProcessPrivateTag, 1066
- ProcessPublicTag, 1066
- PublicConstIterator, 1061
- PublicMappingType, 1061
- PublicTagToValue, 1061
- PublicTagToValueValueType, 1061
- Scan, 1066
- StrictScanner2, 1061
- ValuesType, 1061
- gdcmm::StrictScanner2::ltstr, 680
  - operator(), 680
- gdcmm::StrictScanner::ltstr, 681
  - operator(), 681
- gdcmm::String< TDelimiter, TMaxLength, TPadChar >, 1067
  - const\_iterator, 1069
  - const\_reference, 1069
  - const\_reverse\_iterator, 1069
  - difference\_type, 1069
  - IsValid, 1071
  - iterator, 1069
  - operator const char \*, 1071
  - pointer, 1070
  - reference, 1070
  - reverse\_iterator, 1070
  - size\_type, 1070
  - String, 1070, 1071
  - Trim, 1071, 1072
  - Truncate, 1072
  - value\_type, 1070
- gdcmm::StringFilter, 1072
  - ~StringFilter, 1073
  - ExecuteQuery, 1074
  - FromString, 1074
  - GetFile, 1074
  - SetDicts, 1074
  - SetFile, 1074
  - StringFilter, 1073
  - ToString, 1075
  - ToStringPair, 1075, 1076
  - UseDictAlways, 1076
- gdcmm::Study, 1076
  - Study, 1076
- gdcmm::Subject, 1077
  - ~Subject, 1078
  - AddObserver, 1079
  - GetCommand, 1079
  - HasObserver, 1079
  - InvokeEvent, 1079, 1080
  - RemoveAllObservers, 1080
  - RemoveObserver, 1080
  - Subject, 1078
- gdcmm::Surface, 1081
  - ~Surface, 1084
  - GetAlgorithmFamily, 1085
  - GetAlgorithmName, 1085
  - GetAlgorithmVersion, 1085
  - GetAxisOfRotation, 1085
  - GetCenterOfRotation, 1085
  - GetFiniteVolume, 1085
  - GetManifold, 1085
  - GetMaximumPointDistance, 1086
  - GetMeanPointDistance, 1086
  - GetMeshPrimitive, 1086
  - GetNumberOfSurfacePoints, 1086
  - GetNumberOfVectors, 1086
  - GetPointCoordinatesData, 1086
  - GetPointPositionAccuracy, 1086
  - GetPointsBoundingBoxCoordinates, 1087
  - GetProcessingAlgorithm, 1087
  - GetRecommendedDisplayCIELabValue, 1087
  - GetRecommendedDisplayGrayscaleValue, 1087
  - GetRecommendedPresentationOpacity, 1087
  - GetRecommendedPresentationType, 1087
  - GetSTATES, 1088
  - GetSTATESString, 1088
  - GetSurfaceComments, 1088
  - GetSurfaceNumber, 1088
  - GetSurfaceProcessing, 1088
  - GetSurfaceProcessingDescription, 1088
  - GetSurfaceProcessingRatio, 1088
  - GetVectorAccuracy, 1088
  - GetVectorCoordinateData, 1088, 1089
  - GetVectorDimensionality, 1089
  - GetVIEWType, 1089
  - GetVIEWTypeString, 1089
  - NO, 1084
  - POINTS, 1084
  - SetAlgorithmFamily, 1089
  - SetAlgorithmName, 1089
  - SetAlgorithmVersion, 1089
  - SetAxisOfRotation, 1089
  - SetCenterOfRotation, 1089
  - SetFiniteVolume, 1090
  - SetManifold, 1090
  - SetMaximumPointDistance, 1090
  - SetMeanPointDistance, 1090
  - SetMeshPrimitive, 1090
  - SetNumberOfSurfacePoints, 1090
  - SetNumberOfVectors, 1090
  - SetPointCoordinatesData, 1090
  - SetPointPositionAccuracy, 1091
  - SetPointsBoundingBoxCoordinates, 1091
  - SetProcessingAlgorithm, 1091
  - SetRecommendedDisplayCIELabValue, 1091
  - SetRecommendedDisplayGrayscaleValue, 1091
  - SetRecommendedPresentationOpacity, 1091

- SetRecommendedPresentationType, [1092](#)
- SetSurfaceComments, [1092](#)
- SetSurfaceNumber, [1092](#)
- SetSurfaceProcessing, [1092](#)
- SetSurfaceProcessingDescription, [1092](#)
- SetSurfaceProcessingRatio, [1092](#)
- SetVectorAccuracy, [1092](#)
- SetVectorCoordinateData, [1092](#)
- SetVectorDimensionality, [1093](#)
- STATES, [1084](#)
- STATES\_END, [1084](#)
- SURFACE, [1084](#)
- Surface, [1084](#)
- UNKNOWN, [1084](#)
- VIEWType, [1084](#)
- VIEWType\_END, [1084](#)
- WIREFRAME, [1084](#)
- YES, [1084](#)
- gdcmm::SurfaceHelper, [1093](#)
  - ColorArray, [1094](#)
  - RecommendedDisplayCIELabToRGB, [1094](#)
  - RGBToRecommendedDisplayCIELab, [1095](#)
  - RGBToRecommendedDisplayGrayscale, [1095](#)
- gdcmm::SurfaceReader, [1096](#)
  - ~SurfaceReader, [1099](#)
  - GetNumberOfSurfaces, [1099](#)
  - Read, [1099](#)
  - ReadPointMacro, [1099](#)
  - ReadSurface, [1100](#)
  - ReadSurfaces, [1100](#)
  - SurfaceReader, [1099](#)
- gdcmm::SurfaceWriter, [1100](#)
  - ~SurfaceWriter, [1103](#)
  - ComputeNumberOfSurfaces, [1103](#)
  - GetNumberOfSurfaces, [1103](#)
  - NumberOfSurfaces, [1104](#)
  - PrepareWrite, [1103](#)
  - PrepareWritePointMacro, [1103](#)
  - SetNumberOfSurfaces, [1103](#)
  - SurfaceWriter, [1103](#)
  - Write, [1103](#)
- gdcmm::SwapCode, [1104](#)
  - BadBigEndian, [1105](#)
  - BadLittleEndian, [1105](#)
  - BigEndian, [1105](#)
  - GetIndex, [1105](#)
  - GetSwapCodeString, [1105](#)
  - LittleEndian, [1105](#)
  - operator SwapCode::SwapCodeType, [1105](#)
  - operator<<, [1106](#)
  - SwapCode, [1105](#)
  - SwapCodeType, [1105](#)
  - Unknown, [1105](#)
- gdcmm::SwapperDoOp, [1106](#)
- Swap, [1106](#)
- SwapArray, [1106](#)
- gdcmm::SwapperNoOp, [1107](#)
  - Swap, [1107](#)
  - SwapArray, [1107](#)
- gdcmm::System, [1107](#)
  - ConvertToUNC, [1109](#)
  - DeleteDirectory, [1109](#)
  - EncodeBytes, [1109](#)
  - FileExists, [1109](#)
  - FileIsDirectory, [1109](#)
  - FileIsSymlink, [1110](#)
  - FileSize, [1110](#)
  - FileTime, [1110](#)
  - FormatDateTime, [1110](#)
  - GetCurrentDateTime, [1111](#)
  - GetCurrentModuleFileName, [1111](#)
  - GetCurrentProcessFileName, [1111](#)
  - GetCurrentResourcesDirectory, [1111](#)
  - GetCWD, [1111](#)
  - GetHostName, [1111](#)
  - GetLastSystemError, [1112](#)
  - GetLocaleCharset, [1112](#)
  - GetPermissions, [1112](#)
  - GetTimezoneOffsetFromUTC, [1112](#)
  - MakeDirectory, [1112](#)
  - ParseDateTime, [1112](#), [1113](#)
  - RemoveFile, [1113](#)
  - SetPermissions, [1113](#)
  - StrCaseCmp, [1113](#)
  - StrNCaseCmp, [1113](#)
  - StrSep, [1114](#)
  - StrTokR, [1114](#)
- gdcmm::Table, [1114](#)
  - ~Table, [1116](#)
  - GetTableEntry, [1116](#)
  - InsertEntry, [1116](#)
  - MapTableEntry, [1116](#)
  - operator<<, [1117](#)
  - operator=, [1116](#)
  - Table, [1116](#)
  - TableInternal, [1117](#)
- gdcmm::TableEntry, [1117](#)
  - ~TableEntry, [1118](#)
  - TableEntry, [1118](#)
- gdcmm::TableReader, [1118](#)
  - ~TableReader, [1119](#)
  - CharacterDataHandler, [1119](#)
  - EndElement, [1119](#)
  - GetDefs, [1120](#)
  - GetFilename, [1120](#)
  - HandleIOD, [1120](#)
  - HandleIODEntry, [1120](#)
  - HandleMacro, [1120](#)

- HandleMacroEntry, [1120](#)
- HandleMacroEntryDescription, [1120](#)
- HandleModule, [1120](#)
- HandleModuleEntry, [1121](#)
- HandleModuleEntryDescription, [1121](#)
- HandleModuleInclude, [1121](#)
- Read, [1121](#)
- SetFilename, [1121](#)
- StartElement, [1121](#)
- TableReader, [1119](#)
- gdcmm::Tag, [1123](#)
  - bytes, [1132](#)
  - GetElement, [1126](#)
  - GetElementTag, [1126](#)
  - GetGroup, [1126](#)
  - GetLength, [1126](#)
  - GetPrivateCreator, [1127](#)
  - IsGroupLength, [1127](#)
  - IsGroupXX, [1127](#)
  - IsIllegal, [1127](#)
  - IsPrivate, [1127](#)
  - IsPrivateCreator, [1127](#)
  - IsPublic, [1128](#)
  - operator!=, [1128](#)
  - operator<, [1128](#)
  - operator<<, [1132](#)
  - operator<=, [1128](#)
  - operator>>, [1132](#)
  - operator=, [1128](#)
  - operator==, [1128](#)
  - operator[], [1129](#)
  - PrintAsContinuousString, [1129](#)
  - PrintAsContinuousUpperCaseString, [1129](#)
  - PrintAsPipeSeparatedString, [1129](#)
  - Read, [1129](#)
  - ReadFromCommaSeparatedString, [1130](#)
  - ReadFromContinuousString, [1130](#)
  - ReadFromPipeSeparatedString, [1130](#)
  - SetElement, [1130](#)
  - SetElementTag, [1130](#), [1131](#)
  - SetGroup, [1131](#)
  - SetPrivateCreator, [1131](#)
  - Tag, [1125](#)
  - tag, [1132](#)
  - tags, [1132](#)
  - Write, [1131](#)
- gdcmm::TagPath, [1132](#)
  - ~TagPath, [1133](#)
  - ConstructFromString, [1133](#)
  - ConstructFromTagList, [1133](#)
  - IsValid, [1134](#)
  - Print, [1134](#)
  - Push, [1134](#)
  - TagPath, [1133](#)
- gdcmm::terminal, [81](#)
  - Attribute, [82](#)
  - black, [82](#)
  - blink, [82](#)
  - blue, [82](#)
  - bright, [82](#)
  - Color, [82](#)
  - CONSOLE, [82](#)
  - cyan, [82](#)
  - dim, [82](#)
  - green, [82](#)
  - hidden, [82](#)
  - magenta, [82](#)
  - Mode, [82](#)
  - red, [82](#)
  - reset, [82](#)
  - reverse, [82](#)
  - setattribute, [83](#)
  - setbgcolor, [83](#)
  - setfgcolor, [83](#)
  - setmode, [83](#)
  - underline, [82](#)
  - VT100, [82](#)
  - white, [82](#)
  - yellow, [82](#)
- gdcmm::Testing, [1134](#)
  - ~Testing, [1136](#)
  - ComputeFileMD5, [1136](#)
  - ComputeMD5, [1136](#)
  - GetDataExtraRoot, [1137](#)
  - GetDataRoot, [1137](#)
  - GetFileName, [1137](#)
  - GetFileNames, [1137](#)
  - GetLossyFlagFromFile, [1138](#)
  - GetMD5DataImage, [1138](#)
  - GetMD5DataImages, [1138](#)
  - GetMD5FromBrokenFile, [1138](#)
  - GetMD5FromFile, [1138](#)
  - GetMediaStorageDataFile, [1138](#)
  - GetMediaStorageDataFiles, [1138](#)
  - GetMediaStorageFromFile, [1139](#)
  - GetNumberOfFileNames, [1139](#)
  - GetNumberOfMD5DataImages, [1139](#)
  - GetNumberOfMediaStorageDataFiles, [1139](#)
  - GetPixelSpacingDataRoot, [1139](#)
  - GetSelectedPrivateGroupOffsetFromFile, [1139](#)
  - GetSelectedTagsOffsetFromFile, [1139](#)
  - GetSourceDirectory, [1140](#)
  - GetStreamOffsetFromFile, [1140](#)
  - GetTempDirectory, [1140](#)
  - GetTempDirectoryW, [1140](#)
  - GetTempFilename, [1140](#)
  - GetTempFilenameW, [1141](#)
  - MD5DataImagesType, [1136](#)

- MediaStorageDataFileType, [1136](#)
- Print, [1141](#)
- Testing, [1136](#)
- gdcm::Trace, [1141](#)
  - ~Trace, [1143](#)
  - DebugOff, [1143](#)
  - DebugOn, [1143](#)
  - ErrorOff, [1143](#)
  - ErrorOn, [1143](#)
  - GetDebugFlag, [1143](#)
  - GetDebugStream, [1144](#)
  - GetErrorFlag, [1144](#)
  - GetErrorStream, [1144](#)
  - GetStream, [1144](#)
  - GetWarningFlag, [1144](#)
  - GetWarningStream, [1144](#)
  - SetDebug, [1144](#)
  - SetDebugStream, [1144](#)
  - SetError, [1145](#)
  - SetErrorStream, [1145](#)
  - SetStream, [1145](#)
  - SetStreamToFile, [1145](#)
  - SetWarning, [1145](#)
  - SetWarningStream, [1146](#)
  - Trace, [1143](#)
  - WarningOff, [1146](#)
  - WarningOn, [1146](#)
- gdcm::TransferSyntax, [1146](#)
  - CanStoreLossy, [1150](#)
  - CT\_private\_ELE, [1149](#)
  - DeflatedExplicitVRLittleEndian, [1149](#)
  - Explicit, [1148](#)
  - ExplicitVRBigEndian, [1149](#)
  - ExplicitVRLittleEndian, [1149](#)
  - GetNegociatedType, [1150](#)
  - GetString, [1150](#)
  - GetSwapCode, [1150](#)
  - GetTSString, [1150](#)
  - GetTSType, [1150](#)
  - Implicit, [1148](#)
  - ImplicitVRBigEndianACRNEMA, [1149](#)
  - ImplicitVRBigEndianPrivateGE, [1149](#)
  - ImplicitVRLittleEndian, [1149](#)
  - IsEncapsulated, [1150](#)
  - IsEncoded, [1151](#)
  - IsExplicit, [1151](#)
  - IsImplicit, [1151](#)
  - IsLossless, [1151](#)
  - IsLossy, [1151](#)
  - IsValid, [1151](#)
  - JPEG2000, [1149](#)
  - JPEG2000Lossless, [1149](#)
  - JPEG2000Part2, [1149](#)
  - JPEG2000Part2Lossless, [1149](#)
  - JPEGBaselineProcess1, [1149](#)
  - JPEGExtendedProcess2\_4, [1149](#)
  - JPEGExtendedProcess3\_5, [1149](#)
  - JPEGFullProgressionProcess10\_12, [1149](#)
  - JPEGLosslessProcess14, [1149](#)
  - JPEGLosslessProcess14\_1, [1149](#)
  - JPEGLSLossless, [1149](#)
  - JPEGLSNearLossless, [1149](#)
  - JPEGSpectralSelectionProcess6\_8, [1149](#)
  - JPIPRreferenced, [1149](#)
  - MPEG2MainProfile, [1149](#)
  - MPEG2MainProfileHighLevel, [1149](#)
  - MPEG4AVCH264BDcompatibleHighProfileLevel4\_1, [1149](#)
  - MPEG4AVCH264HighProfileLevel4\_1, [1149](#)
  - NegociatedType, [1148](#)
  - operator TSType, [1151](#)
  - operator <<, [1152](#)
  - RLELossless, [1149](#)
  - TransferSyntax, [1149](#)
  - TS\_END, [1149](#)
  - TSType, [1149](#)
  - Unknown, [1148](#)
  - WeirdPapryus, [1149](#)
- gdcm::Type, [1156](#)
  - GetTypeString, [1157](#)
  - GetTypeType, [1157](#)
  - operator TypeType, [1157](#)
  - operator <<, [1157](#)
  - T1, [1157](#)
  - T1C, [1157](#)
  - T2, [1157](#)
  - T2C, [1157](#)
  - T3, [1157](#)
  - Type, [1157](#)
  - TypeType, [1156](#)
  - UNKNOWN, [1157](#)
- gdcm::UI, [1158](#)
  - Internal, [1158](#)
  - operator <<, [1158](#)
- gdcm::UIDGenerator, [1158](#)
  - Generate, [1159](#)
  - GenerateUUID, [1159](#)
  - GetGDCMUID, [1160](#)
  - GetRoot, [1160](#)
  - IsValid, [1160](#)
  - SetRoot, [1160](#)
  - UIDGenerator, [1159](#)
- gdcm::UIDs, [1161](#)
  - AbstractMultiDimensionalImageModel, [1186](#)
  - AcquisitionContextSRStorage, [1185](#)
  - AdultMouseAnatomyOntology, [1183](#)
  - AdvancedBlendingPresentationStateStorage, [1184](#)
  - AmbulatoryECGWaveformStorage, [1180](#)

- ArterialPulseWaveformStorage, [1184](#)
- AudioSRStorageTrialRetired, [1181](#)
- AutorefractionMeasurementsStorage, [1184](#)
- BasicAnnotationBoxSOPClass, [1179](#)
- BasicColorImageBoxSOPClass, [1179](#)
- BasicColorPrintManagementMetaSOPClass, [1179](#)
- BasicFilmBoxSOPClass, [1179](#)
- BasicFilmSessionSOPClass, [1179](#)
- BasicGrayscaleImageBoxSOPClass, [1179](#)
- BasicGrayscalePrintManagementMetaSOPClass, [1179](#)
- BasicPrintImageOverlayBoxSOPClassRetired, [1179](#)
- BasicStructuredDisplayStorage, [1185](#)
- BasicStudyContentNotificationSOPClassRetired, [1178](#)
- BasicTextSRStorage, [1181](#)
- BasicVoiceAudioWaveformStorage, [1180](#)
- BlendingSoftcopyPresentationStateStorageSOPClass, [1180](#)
- BreastImagingRelevantPatientInformationQuery, [1182](#)
- BreastProjectionXRayImageStorageForPresentation, [1184](#)
- BreastProjectionXRayImageStorageForProcessing, [1184](#)
- BreastTomosynthesisImageStorage, [1183](#)
- CardiacElectrophysiologyWaveformStorage, [1180](#)
- CardiacRelevantPatientInformationQuery, [1182](#)
- ChestCADSRStorage, [1181](#)
- ColonCADSRStorage, [1185](#)
- ColorPaletteQueryRetrieveInformationModelFIND, [1186](#)
- ColorPaletteQueryRetrieveInformationModelGET, [1186](#)
- ColorPaletteQueryRetrieveInformationModelMOVE, [1186](#)
- ColorPaletteStorage, [1186](#)
- ColorSoftcopyPresentationStateStorageSOPClass, [1180](#)
- CompositeInstanceRetrieveWithoutBulkDataGET, [1185](#)
- CompositeInstanceRootRetrieveGET, [1185](#)
- CompositeInstanceRootRetrieveMOVE, [1185](#)
- CompositingPlanarMPRVolumetricPresentationStateStorage, [1184](#)
- Comprehensive3DSRStorage, [1185](#)
- ComprehensiveSRStorage, [1181](#)
- ComprehensiveSRStorageTrialRetired, [1181](#)
- ComputedRadiographyImageStorage, [1179](#)
- ContentAssessmentResultsStorage, [1185](#)
- CornealTopographyMapStorage, [1185](#)
- CTDefinedProcedureProtocolStorage, [1185](#)
- CTImageStorage, [1179](#)
- CTPerformedProcedureProtocolStorage, [1185](#)
- DefinedProcedureProtocolInformationModelFIND, [1185](#)
- DefinedProcedureProtocolInformationModelGET, [1185](#)
- DefinedProcedureProtocolInformationModelMOVE, [1185](#)
- DeflatedExplicitVRLittleEndian, [1177](#)
- DeformableSpatialRegistrationStorage, [1180](#)
- DetachedInterpretationManagementSOPClassRetired, [1179](#)
- DetachedPatientManagementMetaSOPClassRetired, [1178](#)
- DetachedPatientManagementSOPClassRetired, [1178](#)
- DetachedResultsManagementMetaSOPClassRetired, [1179](#)
- DetachedResultsManagementSOPClassRetired, [1179](#)
- DetachedStudyManagementMetaSOPClassRetired, [1179](#)
- DetachedStudyManagementSOPClassRetired, [1178](#)
- DetachedVisitManagementSOPClassRetired, [1178](#)
- DetailSRStorageTrialRetired, [1181](#)
- dicomAETitle, [1182](#)
- dicomApplicationCluster, [1182](#)
- DICOMApplicationContextName, [1178](#)
- dicomAssociationAcceptor, [1182](#)
- dicomAssociationInitiator, [1182](#)
- dicomAuthorizedNodeCertificateReference, [1183](#)
- dicomConfigurationRoot, [1183](#)
- DICOMContentMappingResource, [1186](#)
- DICOMControlledTerminology, [1178](#)
- dicomDescription, [1182](#)
- dicomDevice, [1183](#)
- dicomDeviceName, [1182](#)
- dicomDeviceSerialNumber, [1183](#)
- dicomDevicesRoot, [1183](#)
- dicomHostname, [1182](#)
- dicomInstalled, [1183](#)
- dicomInstitutionAddress, [1183](#)
- dicomInstitutionDepartmentName, [1183](#)
- dicomInstitutionName, [1183](#)
- dicomIssuerOfPatientID, [1183](#)
- dicomManufacturer, [1182](#)
- dicomManufacturerModelName, [1182](#)
- dicomNetworkAE, [1183](#)
- dicomNetworkConnection, [1183](#)
- dicomNetworkConnectionReference, [1182](#)
- dicomPort, [1182](#)
- dicomPreferredCalledAETitle, [1182](#)
- dicomPreferredCallingAETitle, [1183](#)
- dicomPrimaryDeviceType, [1182](#)
- dicomRelatedDeviceReference, [1182](#)
- dicomSoftwareVersion, [1182](#)



- dicomSOPClass, 1182
- dicomStationName, 1183
- dicomSupportedCharacterSet, 1183
- dicomThisNodeCertificateReference, 1183
- dicomTLSCyphersuite, 1183
- dicomTransferCapability, 1183
- dicomTransferRole, 1182
- dicomTransferSyntax, 1182
- DICOMUIDRegistry, 1178
- dicomUniqueAETitle, 1183
- dicomUniqueAETitlesRegistryRoot, 1183
- dicomVendorData, 1182
- DICOS2DAITStorage, 1185
- DICOS3DAITStorage, 1185
- DICOSCTImageStorage, 1185
- DICOSDigitalXRayImageStorageForPresentation, 1185
- DICOSDigitalXRayImageStorageForProcessing, 1185
- DICOSQuadrupoleResonanceQRStorage, 1185
- DICOSThreatDetectionReportStorage, 1185
- DigitalIntraoralXRayImageStorageForPresentation, 1179
- DigitalIntraoralXRayImageStorageForProcessing, 1179
- DigitalMammographyXRayImageStorageForPresentation, 1179
- DigitalMammographyXRayImageStorageForProcessing, 1179
- DigitalXRayImageStorageForPresentation, 1179
- DigitalXRayImageStorageForProcessing, 1179
- DisplaySystemSOPClass, 1184
- DisplaySystemSOPInstance, 1184
- ECG12leadWaveformStorage, 1180
- EddyCurrentImageStorage, 1185
- EddyCurrentMultiframeImageStorage, 1185
- EncapsulatedCDASStorage, 1181
- EncapsulatedPDFStorage, 1181
- EncapsulatedSTLStorage, 1185
- EnhancedCTImageStorage, 1179
- EnhancedMRColorImageStorage, 1186
- EnhancedMRIImageStorage, 1180
- EnhancedPETImageStorage, 1185
- EnhancedSRStorage, 1181
- EnhancedUSVolumeStorage, 1183
- EnhancedXAImageStorage, 1180
- EnhancedXRFImageStorage, 1180
- ExplicitVRBigEndian, 1177
- ExplicitVRLittleEndian, 1177
- ExtensibleSRStorage, 1185
- FallColorPaletteSOPInstance, 1183
- GeneralAudioWaveformStorage, 1184
- GeneralECGWaveformStorage, 1180
- GeneralPurposePerformedProcedureStepSOP-Class, 1182
- GeneralPurposeScheduledProcedureStepSOP-Class, 1182
- GeneralPurposeWorklistInformationModelFIND, 1182
- GeneralPurposeWorklistManagementMetaSOP-Class, 1182
- GeneralRelevantPatientInformationQuery, 1182
- GenericImplantTemplateInformationModelFIND, 1186
- GenericImplantTemplateInformationModelGET, 1186
- GenericImplantTemplateInformationModelMOVE, 1186
- GenericImplantTemplateStorage, 1186
- GetName, 1196
- GetNumberOfTransferSyntaxStrings, 1196
- GetString, 1196
- GetTransferSyntaxString, 1196
- GetTransferSyntaxStrings, 1196
- GetUIDName, 1196
- GetUIDString, 1196
- GrayscalePlanarMPRVolumetricPresentationStateStorage, 1184
- GrayscaleSoftcopyPresentationStateStorageSOP-Class, 1180
- HangingProtocolInformationModelFIND, 1182
- HangingProtocolInformationModelGET, 1186
- HangingProtocolInformationModelMOVE, 1182
- HangingProtocolStorage, 1182
- HardcopyColorImageStorageSOPClassRetired, 1179
- HardcopyGrayscaleImageStorageSOPClassRetired, 1179
- HemodynamicWaveformStorage, 1180
- HEVCH\_265Main10ProfileLevel5\_1, 1184
- HEVCH\_265MainProfileLevel5\_1, 1184
- HotIronColorPaletteSOPInstance, 1184
- HotMetalBlueColorPaletteSOPInstance, 1183
- ICBM452T1FrameofReference, 1178
- ICBMSingleSubjectMRIFrameofReference, 1178
- ICD11, 1183
- ImageBiomarkerStandardisationInitiative, 1184
- ImageOverlayBoxSOPClassRetired, 1179
- ImplantAssemblyTemplateInformationModelFIND, 1186
- ImplantAssemblyTemplateInformationModelGET, 1186
- ImplantAssemblyTemplateInformationModelMOVE, 1186
- ImplantAssemblyTemplateStorage, 1186
- ImplantationPlanSRStorage, 1185
- ImplantTemplateGroupInformationModelFIND, 1186
- ImplantTemplateGroupInformationModelGET, 1186



- ImplantTemplateGroupInformationModelMOVE, [1186](#)
- ImplantTemplateGroupStorage, [1186](#)
- ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM, [1177](#)
- InstanceAvailabilityNotificationSOPClass, [1182](#)
- IntegratedTaxonomicInformationSystemITISTaxonomicSerialNumberTSN, [1183](#)
- IntraocularLensCalculationsStorage, [1185](#)
- IntravascularOpticalCoherenceTomographyImageStorageForPresentation, [1184](#)
- IntravascularOpticalCoherenceTomographyImageStorageForProcessing, [1184](#)
- JPEG2000ImageCompression, [1177](#)
- JPEG2000ImageCompressionLosslessOnly, [1177](#)
- JPEG2000Part2MulticomponentImageCompression, [1177](#)
- JPEG2000Part2MulticomponentImageCompressionLosslessOnly,Class, [1179](#)
- JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImages, [1177](#)
- JPEGExtendedHierarchicalProcess1618Retired, [1177](#)
- JPEGExtendedHierarchicalProcess1719Retired, [1177](#)
- JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG8BitImages, [1177](#)
- JPEGExtendedProcess35Retired, [1177](#)
- JPEGFullProgressionHierarchicalProcess2426Retired, [1177](#)
- JPEGFullProgressionHierarchicalProcess2527Retired, [1177](#)
- JPEGFullProgressionNonHierarchicalProcess1012Retired, [1177](#)
- JPEGFullProgressionNonHierarchicalProcess1113Retired, [1177](#)
- JPEGLosslessHierarchicalProcess28Retired, [1177](#)
- JPEGLosslessHierarchicalProcess29Retired, [1177](#)
- JPEGLosslessNonHierarchicalFirstOrderPrediction-Process14SelectionValue1DefaultTransferSyntaxforLossyJPEG8BitImages, [1177](#)
- JPEGLosslessNonHierarchicalProcess14, [1177](#)
- JPEGLosslessNonHierarchicalProcess15Retired, [1177](#)
- JPEGLSLosslessImageCompression, [1177](#)
- JPEGLSLossyNearLosslessImageCompression, [1177](#)
- JPEGSpectralSelectionHierarchicalProcess2022Retired, [1177](#)
- JPEGSpectralSelectionHierarchicalProcess2123Retired, [1177](#)
- JPEGSpectralSelectionNonHierarchicalProcess68Retired, [1177](#)
- JPEGSpectralSelectionNonHierarchicalProcess79Retired, [1177](#)
- JPIPPreferenced, [1177](#)
- JPIPPreferencedDeflate, [1178](#)
- KeratometryMeasurementsStorage, [1184](#)
- KeyObjectSelectionDocumentStorage, [1181](#)
- LegacyConvertedEnhancedCTImageStorage, [1183](#)
- LegacyConvertedEnhancedMRIImageStorage, [1183](#)
- LegacyConvertedEnhancedPETImageStorage, [1183](#)
- LensometryMeasurementsStorage, [1184](#)
- MacularGridThicknessandVolumeReportStorage, [1185](#)
- MammographyCADSRStorage, [1181](#)
- MayoClinicNonradiologicalImagesSBSAnatomical-SurfaceRegionGuide, [1184](#)
- MediaCreationManagementSOPClassUID, [1179](#)
- MediaStorageDirectoryStorage, [1178](#)
- ModalityPerformedProcedureStepNotificationSOP-ModalityPerformedProcedureStepRetrieveSOP-Class, [1178](#)
- ModalityPerformedProcedureStepSOPClass, [1178](#)
- ModalityWorklistInformationModelFIND, [1182](#)
- MouseGenomeInitiativeMGI, [1183](#)
- MPEG2MainProfileHighLevel, [1183](#)
- MPEG2MainProfileMainLevel, [1178](#)
- MPEG4AVCH\_264HighProfileLevel4\_1, [1183](#)
- MPEG4AVCH\_264HighProfileLevel4\_1, [1183](#)
- MPEG4AVCH\_264HighProfileLevel4\_2For2DVideo, [1184](#)
- MPEG4AVCH\_264HighProfileLevel4\_2For3DVideo, [1184](#)
- MPEG4AVCH\_264StereoHighProfileLevel4\_2, [1184](#)
- MRIImageStorage, [1180](#)
- MRSpectroscopyStorage, [1180](#)
- MultiframeGrayscaleByteSecondaryCaptureImageStorage, [1180](#)
- MultiframeGrayscaleWordSecondaryCaptureImageStorage, [1180](#)
- MultiframeTrueColorSecondaryCaptureImageStorage, [1180](#)
- MultipleVolumeRenderingVolumetricPresentation-StateStorage, [1184](#)
- NativeDICOMModel, [1186](#)
- NewYorkUniversityMelanomaClinicalCooperative-Group, [1184](#)
- NuclearMedicineImageStorage, [1180](#)
- NuclearMedicineImageStorageRetired, [1180](#)
- Null0, [1184](#)
- Null1, [1184](#)
- operator TSType, [1197](#)
- OphthalmicAxialMeasurementsStorage, [1184](#)

- OphthalmicOpticalCoherenceTomographyBscanVolumeAnalysisStorage, [1184](#)
- OphthalmicOpticalCoherenceTomographyEnFacImageStorage, [1184](#)
- OphthalmicPhotography16BitImageStorage, [1181](#)
- OphthalmicPhotography8BitImageStorage, [1181](#)
- OphthalmicThicknessMapStorage, [1185](#)
- OphthalmicTomographyImageStorage, [1181](#)
- OphthalmicVisualFieldStaticPerimetryMeasurementsStorage, [1185](#)
- Papyrus3ImplicitVRLittleEndian, [1183](#)
- ParametricMapStorage, [1184](#)
- PatientRadiationDoseSRStorage, [1185](#)
- PatientRootQueryRetrieveInformationModelFIND, [1181](#)
- PatientRootQueryRetrieveInformationModelGET, [1181](#)
- PatientRootQueryRetrieveInformationModelMOVE, [1181](#)
- PatientStudyOnlyQueryRetrieveInformationModelFINDRetired, [1181](#)
- PatientStudyOnlyQueryRetrieveInformationModelGETRetired, [1182](#)
- PatientStudyOnlyQueryRetrieveInformationModelMOVERetired, [1182](#)
- PerformedImagingAgentAdministrationSRStorage, [1185](#)
- PET20StepColorPaletteSOPInstance, [1183](#)
- PETColorPaletteSOPInstance, [1183](#)
- PlannedImagingAgentAdministrationSRStorage, [1185](#)
- PositronEmissionTomographyImageStorage, [1181](#)
- PresentationLUTSOPClass, [1179](#)
- PrinterConfigurationRetrievalSOPClass, [1179](#)
- PrinterConfigurationRetrievalSOPInstance, [1179](#)
- PrinterSOPClass, [1179](#)
- PrinterSOPInstance, [1179](#)
- PrintJobSOPClass, [1179](#)
- PrintQueueManagementSOPClassRetired, [1179](#)
- PrintQueueSOPInstanceRetired, [1179](#)
- ProceduralEventLoggingSOPClass, [1178](#)
- ProceduralEventLoggingSOPInstance, [1178](#)
- ProcedureLogStorage, [1181](#)
- ProductCharacteristicsQuerySOPClass, [1182](#)
- ProtocolApprovalInformationModelFIND, [1185](#)
- ProtocolApprovalInformationModelGET, [1185](#)
- ProtocolApprovalInformationModelMOVE, [1185](#)
- ProtocolApprovalStorage, [1185](#)
- PseudoColorSoftcopyPresentationStateStorageSOPClass, [1180](#)
- PubChemCompoundCID, [1183](#)
- PullPrintRequestSOPClassRetired, [1179](#)
- PullStoredPrintManagementMetaSOPClassRetired, [1179](#)
- RadiomicsOntology, [1184](#)
- RadiopharmaceuticalRadiationDoseSRStorage, [1185](#)
- RawDataStorage, [1180](#)
- RealWorldValueMappingStorage, [1180](#)
- ReferencedColorPrintManagementMetaSOPClassRetired, [1179](#)
- ReferencedGrayscalePrintManagementMetaSOPClassRetired, [1179](#)
- ReferencedImageBoxSOPClassRetired, [1179](#)
- RespiratoryWaveformStorage, [1184](#)
- RFC2557MIMEencapsulation, [1178](#)
- RLELossless, [1178](#)
- RTBeamsDeliveryInstructionStorage, [1186](#)
- RTBeamsDeliveryInstructionStorageSupplement74FrozenDraft, [1182](#)
- RTBeamsTreatmentRecordStorage, [1181](#)
- RTBrachyApplicationSetupDeliveryInstructionStorage, [1186](#)
- RTBrachyTreatmentRecordStorage, [1181](#)
- RTConventionalMachineVerification, [1186](#)
- RTConventionalMachineVerificationSupplement74FrozenDraft, [1182](#)
- RTDoseStorage, [1181](#)
- RTImageStorage, [1181](#)
- RTIonBeamsTreatmentRecordStorage, [1181](#)
- RTIonMachineVerification, [1186](#)
- RTIonMachineVerificationSupplement74FrozenDraft, [1182](#)
- RTIonPlanStorage, [1181](#)
- RTPhysicianIntentStorage, [1185](#)
- RTPlanStorage, [1181](#)
- RTSegmentAnnotationStorage, [1185](#)
- RTStructureSetStorage, [1181](#)
- RTTreatmentSummaryRecordStorage, [1181](#)
- SecondaryCaptureImageStorage, [1180](#)
- SegmentationStorage, [1180](#)
- SegmentedVolumeRenderingVolumetricPresentationStateStorage, [1184](#)
- SetFromUID, [1197](#)
- SimplifiedAdultEchoSRStorage, [1185](#)
- SpatialFiducialsStorage, [1180](#)
- SpatialRegistrationStorage, [1180](#)
- SpectaclePrescriptionReportStorage, [1184](#)
- SPM2AVG152PDFrameofReference, [1178](#)
- SPM2AVG152T1FrameofReference, [1178](#)
- SPM2AVG152T2FrameofReference, [1178](#)
- SPM2AVG305T1FrameofReference, [1178](#)
- SPM2BRAINMASKFrameofReference, [1178](#)
- SPM2CSFFFrameofReference, [1178](#)
- SPM2EPIFrameofReference, [1178](#)
- SPM2FILT1FrameofReference, [1178](#)
- SPM2GRAYFrameofReference, [1178](#)
- SPM2PDFFrameofReference, [1178](#)

SPM2PETFrameofReference, [1178](#)  
SPM2SINGLESUBJT1FrameofReference, [1178](#)  
SPM2SPECTFrameofReference, [1178](#)  
SPM2T1FrameofReference, [1178](#)  
SPM2T2FrameofReference, [1178](#)  
SPM2TRANSMFrameofReference, [1178](#)  
SPM2WHITEFrameofReference, [1178](#)  
SpringColorPaletteSOPInstance, [1183](#)  
StandaloneCurveStorageRetired, [1180](#)  
StandaloneModalityLUTStorageRetired, [1180](#)  
StandaloneOverlayStorageRetired, [1180](#)  
StandalonePETCurveStorageRetired, [1181](#)  
StandaloneVOILUTStorageRetired, [1180](#)  
StereometricRelationshipStorage, [1181](#)  
StorageCommitmentPullModelSOPClassRetired, [1178](#)  
StorageCommitmentPullModelSOPInstanceRetired, [1178](#)  
StorageCommitmentPushModelSOPClass, [1178](#)  
StorageCommitmentPushModelSOPInstance, [1178](#)  
StorageServiceClass, [1179](#)  
StoredPrintStorageSOPClassRetired, [1179](#)  
StudyComponentManagementSOPClassRetired, [1178](#)  
StudyRootQueryRetrieveInformationModelIFIND, [1181](#)  
StudyRootQueryRetrieveInformationModelIGET, [1181](#)  
StudyRootQueryRetrieveInformationModelIMOVE, [1181](#)  
SubjectiveRefractionMeasurementsStorage, [1184](#)  
SubstanceAdministrationLoggingSOPClass, [1178](#)  
SubstanceAdministrationLoggingSOPInstance, [1178](#)  
SubstanceApprovalQuerySOPClass, [1182](#)  
SummerColorPaletteSOPInstance, [1183](#)  
SurfaceScanMeshStorage, [1184](#)  
SurfaceScanPointCloudStorage, [1184](#)  
SurfaceSegmentationStorage, [1183](#)  
TalairachBrainAtlasFrameofReference, [1178](#)  
TextSRStorageTrialRetired, [1181](#)  
TractographyResultsStorage, [1184](#)  
TransferSyntaxStringsType, [1177](#)  
TSName, [1177](#)  
TSType, [1186](#)  
UberonOntology, [1183](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_1, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_10, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_11, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_12, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_13, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_14, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_15, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_16, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_17, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_18, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_19, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_2, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_20, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_21, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_22, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_23, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_24, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_25, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_26, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_27, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_28, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_29, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_3, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_30, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_31, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_4, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_5, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_6, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_7, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_8, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_9, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_4\_1, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_4\_2, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_4\_3, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_4\_4, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_4\_5, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_4\_6, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_4\_7, [1192](#)  
uid\_1\_2\_840\_10008\_15\_0\_4\_8, [1192](#)  
uid\_1\_2\_840\_10008\_15\_1\_1, [1195](#)  
uid\_1\_2\_840\_10008\_1\_1, [1186](#)  
uid\_1\_2\_840\_10008\_1\_2, [1186](#)  
uid\_1\_2\_840\_10008\_1\_20, [1193](#)  
uid\_1\_2\_840\_10008\_1\_20\_1, [1188](#)  
uid\_1\_2\_840\_10008\_1\_20\_1\_1, [1188](#)  
uid\_1\_2\_840\_10008\_1\_20\_2, [1188](#)  
uid\_1\_2\_840\_10008\_1\_20\_2\_1, [1188](#)  
uid\_1\_2\_840\_10008\_1\_2\_1, [1186](#)  
uid\_1\_2\_840\_10008\_1\_2\_1\_99, [1186](#)  
uid\_1\_2\_840\_10008\_1\_2\_2, [1186](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_100, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_101, [1193](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_102, [1193](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_103, [1193](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_104, [1193](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_105, [1193](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_106, [1193](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_107, [1193](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_108, [1193](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_50, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_51, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_52, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_53, [1187](#)

uid\_1\_2\_840\_10008\_1\_2\_4\_54, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_55, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_56, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_57, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_58, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_59, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_60, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_61, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_62, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_63, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_64, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_65, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_66, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_70, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_80, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_81, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_90, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_91, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_92, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_93, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_94, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_95, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_5, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_6\_1, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_6\_2, [1187](#)  
uid\_1\_2\_840\_10008\_1\_3\_10, [1187](#)  
uid\_1\_2\_840\_10008\_1\_40, [1188](#)  
uid\_1\_2\_840\_10008\_1\_40\_1, [1188](#)  
uid\_1\_2\_840\_10008\_1\_42, [1188](#)  
uid\_1\_2\_840\_10008\_1\_42\_1, [1188](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_1, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_10, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_11, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_12, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_13, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_14, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_15, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_16, [1188](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_17, [1188](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_18, [1188](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_2, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_3, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_4, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_5, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_6, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_7, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_8, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_9, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_2\_1, [1188](#)  
uid\_1\_2\_840\_10008\_1\_4\_2\_2, [1188](#)  
uid\_1\_2\_840\_10008\_1\_5\_1, [1193](#)  
uid\_1\_2\_840\_10008\_1\_5\_2, [1193](#)  
uid\_1\_2\_840\_10008\_1\_5\_3, [1193](#)  
uid\_1\_2\_840\_10008\_1\_5\_4, [1193](#)  
uid\_1\_2\_840\_10008\_1\_5\_5, [1193](#)  
uid\_1\_2\_840\_10008\_1\_5\_6, [1193](#)  
uid\_1\_2\_840\_10008\_1\_5\_7, [1193](#)  
uid\_1\_2\_840\_10008\_1\_5\_8, [1193](#)  
uid\_1\_2\_840\_10008\_1\_9, [1188](#)  
uid\_1\_2\_840\_10008\_2\_16\_10, [1193](#)  
uid\_1\_2\_840\_10008\_2\_16\_11, [1193](#)  
uid\_1\_2\_840\_10008\_2\_16\_12, [1193](#)  
uid\_1\_2\_840\_10008\_2\_16\_13, [1193](#)  
uid\_1\_2\_840\_10008\_2\_16\_14, [1193](#)  
uid\_1\_2\_840\_10008\_2\_16\_4, [1188](#)  
uid\_1\_2\_840\_10008\_2\_16\_5, [1193](#)  
uid\_1\_2\_840\_10008\_2\_16\_6, [1193](#)  
uid\_1\_2\_840\_10008\_2\_16\_7, [1193](#)  
uid\_1\_2\_840\_10008\_2\_16\_8, [1193](#)  
uid\_1\_2\_840\_10008\_2\_16\_9, [1193](#)  
uid\_1\_2\_840\_10008\_2\_6\_1, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_1\_1, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_1\_1, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_1\_4, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_2\_1, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_3\_1, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_3\_2, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_3\_3, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_3\_4, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_3\_5, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_5\_1, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_5\_4, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_5\_5, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_6\_1, [1188](#)  
uid\_1\_2\_840\_10008\_4\_2, [1188](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_1, [1188](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_14, [1188](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_15, [1188](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_16, [1188](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_16\_376, [1188](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_17, [1188](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_17\_376, [1188](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_18, [1188](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_18\_1, [1188](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_2, [1188](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_22, [1188](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_23, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_24, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_24\_1, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_25, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_26, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_27, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_29, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_30, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_31, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_32, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_33, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_4, [1188](#)

uid\_1\_2\_840\_10008\_5\_1\_1\_40, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_40\_1, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_4\_1, [1188](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_4\_2, [1188](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_9, [1188](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_9\_1, [1188](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_10, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_104\_1, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_104\_2, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_104\_3, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_1, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_10, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_11, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_2, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_3, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_4, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_5, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_6, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_7, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_8, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_9, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_128, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_128\_1, [1192](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_129, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_1, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_1\_1, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_2, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_2\_1, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_3, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_77, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_130, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_131, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_1, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_2, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_3, [1192](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_4, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_5, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_14\_1, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_14\_2, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_1, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_1\_1, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_1\_2, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_2\_1, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_3, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_3\_1, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_2, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_20, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_1, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_2, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_3, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_4, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_5, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_6, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_2\_1, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_2\_2, [1192](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_3, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_30, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_3\_1, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_40, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_1, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_10, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_11, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_2, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_3, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_4, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_5, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_6, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_7, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_8, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_9, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4\_1, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4\_2, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4\_3, [1195](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4\_4, [1192](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_5, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_1, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_2\_1, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_2\_2, [1195](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_3, [1195](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_4, [1195](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_5, [1195](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_6, [1195](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_6, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_601\_1, [1195](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_601\_2, [1195](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_1, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_2, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_3, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_4, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_5, [1192](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_6, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_67, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_68\_1, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_68\_2, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_6\_1, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_6\_2, [1192](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_7, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_1, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_1\_1, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_2, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_2\_1, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_3, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_4, [1190](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_1\_2, 1189  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_1\_3, 1189  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_2\_1, 1189  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_3\_1, 1189  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_4\_1, 1189  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_4\_2, 1193  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_5\_1, 1193  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_6\_1, 1193  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_1\_1, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_1\_2, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_1\_3, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_2\_1, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_2\_2, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_2\_3, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_3\_1, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_3\_2, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_3\_3, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_4\_2, 1195  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_4\_3, 1195  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_5\_3, 1195  
uid\_1\_2\_840\_10008\_5\_1\_4\_20\_1, 1195  
uid\_1\_2\_840\_10008\_5\_1\_4\_20\_2, 1195  
uid\_1\_2\_840\_10008\_5\_1\_4\_20\_3, 1195  
uid\_1\_2\_840\_10008\_5\_1\_4\_31, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_32, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_32\_1, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_32\_2, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_32\_3, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_33, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_1, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_10, 1195  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_2, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_3, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_1, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_2, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_3, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_4, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_5, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_5\_1, 1195  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_6, 1195  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_6\_1, 1195  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_6\_2, 1195  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_6\_3, 1195  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_6\_4, 1195  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_7, 1195  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_8, 1195  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_9, 1195  
uid\_1\_2\_840\_10008\_5\_1\_4\_37\_1, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_37\_2, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_37\_3, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_38\_1, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_38\_2, 1191  
uid\_1\_2\_840\_10008\_5\_1\_4\_38\_3, 1191



- [uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_38\\_4](#), [1195](#)
- [uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_39\\_1](#), [1195](#)
- [uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_39\\_2](#), [1195](#)
- [uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_39\\_3](#), [1195](#)
- [uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_39\\_4](#), [1195](#)
- [uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_41](#), [1191](#)
- [uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_42](#), [1191](#)
- [uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_43\\_1](#), [1195](#)
- [uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_43\\_2](#), [1195](#)
- [uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_43\\_3](#), [1195](#)
- [uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_43\\_4](#), [1195](#)
- [uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_44\\_1](#), [1195](#)
- [uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_44\\_2](#), [1195](#)
- [uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_44\\_3](#), [1195](#)
- [uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_44\\_4](#), [1195](#)
- [uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_45\\_1](#), [1195](#)
- [uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_45\\_2](#), [1195](#)
- [uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_45\\_3](#), [1195](#)
- [uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_45\\_4](#), [1195](#)
- [uid\\_1\\_2\\_840\\_10008\\_7\\_1\\_1](#), [1195](#)
- [uid\\_1\\_2\\_840\\_10008\\_7\\_1\\_2](#), [1195](#)
- [uid\\_1\\_2\\_840\\_10008\\_8\\_1\\_1](#), [1195](#)
- [UltrasoundImageStorage](#), [1180](#)
- [UltrasoundImageStorageRetired](#), [1180](#)
- [UltrasoundMultiframeImageStorage](#), [1180](#)
- [UltrasoundMultiframeImageStorageRetired](#), [1179](#)
- [UnifiedProcedureStepEventSOPClass](#), [1182](#)
- [UnifiedProcedureStepEventSOPClass1](#), [1186](#)
- [UnifiedProcedureStepPullSOPClass](#), [1182](#)
- [UnifiedProcedureStepPullSOPClass1](#), [1186](#)
- [UnifiedProcedureStepPushSOPClass](#), [1182](#)
- [UnifiedProcedureStepPushSOPClass1](#), [1186](#)
- [UnifiedProcedureStepWatchSOPClass](#), [1182](#)
- [UnifiedProcedureStepWatchSOPClass1](#), [1186](#)
- [UnifiedWorklistandProcedureStepServiceClass](#), [1182](#)
- [UnifiedWorklistandProcedureStepServiceClass1](#), [1186](#)
- [UnifiedWorklistandProcedureStepSOPInstance](#), [1182](#)
- [UniversalCoordinatedTime](#), [1186](#)
- [UPSFilteredGlobalSubscriptionSOPInstance](#), [1185](#)
- [VerificationSOPClass](#), [1177](#)
- [VideoEndoscopicImageStorage](#), [1181](#)
- [VideoMicroscopicImageStorage](#), [1181](#)
- [VideoPhotographicImageStorage](#), [1181](#)
- [VisualAcuityMeasurementsStorage](#), [1184](#)
- [VLEndoscopicImageStorage](#), [1181](#)
- [VLImageStorageTrialRetired](#), [1180](#)
- [VLMicroscopicImageStorage](#), [1181](#)
- [VLMultiframeImageStorageTrialRetired](#), [1180](#)
- [VLPhotographicImageStorage](#), [1181](#)
- [VLSlideCoordinatesMicroscopicImageStorage](#), [1181](#)
- [VLWholeSlideMicroscopyImageStorage](#), [1183](#)
- [VOILUTBoxSOPClass](#), [1179](#)
- [VolumeRenderingVolumetricPresentationStateStorage](#), [1184](#)
- [WaveformStorageTrialRetired](#), [1180](#)
- [WideFieldOphthalmicPhotography3DCoordinatesImageStorage](#), [1184](#)
- [WideFieldOphthalmicPhotographyStereographicProjectionImageStorage](#), [1184](#)
- [WinterColorPaletteSOPInstance](#), [1183](#)
- [XAXRFGrayscaleSoftcopyPresentationStateStorage](#), [1184](#)
- [XMLEncoding](#), [1178](#)
- [XRay3DAngiographicImageStorage](#), [1180](#)
- [XRay3DCraniofacialImageStorage](#), [1180](#)
- [XRayAngiographicBiPlaneImageStorageRetired](#), [1180](#)
- [XRayAngiographicImageStorage](#), [1180](#)
- [XRayRadiationDoseSRStorage](#), [1181](#)
- [XRayRadiofluoroscopicImageStorage](#), [1180](#)
- [gdcmm::UNExplicitDataElement](#), [1262](#)
  - [GetLength](#), [1264](#)
  - [Read](#), [1264](#)
  - [ReadPreValue](#), [1265](#)
  - [ReadValue](#), [1265](#)
  - [ReadWithLength](#), [1265](#)
- [gdcmm::UNExplicitImplicitDataElement](#), [1265](#)
  - [GetLength](#), [1268](#)
  - [Read](#), [1268](#)
  - [ReadPreValue](#), [1268](#)
  - [ReadValue](#), [1269](#)
- [gdcmm::Unpacker12Bits](#), [1269](#)
  - [Pack](#), [1270](#)
  - [Unpack](#), [1270](#)
- [gdcmm::Usage](#), [1270](#)
  - [Conditional](#), [1271](#)
  - [GetUsageString](#), [1272](#)
  - [GetUsageType](#), [1272](#)
  - [Invalid](#), [1271](#)
  - [Mandatory](#), [1271](#)
  - [operator UsageType](#), [1272](#)
  - [operator <=](#), [1272](#)
  - [Usage](#), [1272](#)
  - [UsageType](#), [1271](#)
  - [UserOption](#), [1271](#)
- [gdcmm::UserEvent](#), [1273](#)
- [gdcmm::UUIDGenerator](#), [1276](#)
  - [Generate](#), [1277](#)
  - [IsValid](#), [1277](#)
- [gdcmm::Validate](#), [1277](#)
  - [~Validate](#), [1278](#)
  - [F](#), [1279](#)
  - [GetValidatedFile](#), [1278](#)
  - [SetFile](#), [1278](#)
  - [V](#), [1279](#)

- Validate, [1278](#)
- Validation, [1278](#)
- gdcM::Value, [1279](#)
  - ~Value, [1280](#)
  - Clear, [1281](#)
  - DataElement, [1282](#)
  - GetLength, [1281](#)
  - operator==, [1281](#)
  - SetLength, [1281](#)
  - SetLengthOnly, [1281](#)
  - Value, [1280](#)
- gdcM::ValueIO< TDE, TSwap, TType >, [1282](#)
  - Read, [1282](#)
  - Write, [1282](#)
- gdcM::Version, [1284](#)
  - ~Version, [1284](#)
  - GetBuildVersion, [1284](#)
  - GetMajorVersion, [1284](#)
  - GetMinorVersion, [1285](#)
  - GetVersion, [1285](#)
  - operator<<, [1285](#)
  - Print, [1285](#)
  - Version, [1284](#)
- gdcM::VL, [1285](#)
  - GetLength, [1287](#)
  - GetVL16Max, [1287](#)
  - GetVL32Max, [1287](#)
  - IsOdd, [1287](#)
  - IsUndefined, [1287](#)
  - operator uint32\_t, [1287](#)
  - operator<<, [1289](#)
  - operator++, [1287](#), [1288](#)
  - operator+=, [1288](#)
  - Read, [1288](#)
  - Read16, [1288](#)
  - SetToUndefined, [1288](#)
  - Type, [1286](#)
  - VL, [1287](#)
  - Write, [1288](#)
  - Write16, [1288](#)
- gdcM::VM, [1289](#)
  - Compatible, [1292](#)
  - GetIndex, [1292](#)
  - GetLength, [1292](#)
  - GetNumberOfElementsFromArray, [1292](#)
  - GetVMString, [1293](#)
  - GetVMType, [1293](#)
  - GetVMTypeFromLength, [1293](#)
  - IsValid, [1293](#)
  - operator VMType, [1293](#)
  - operator<<, [1293](#)
  - VM, [1292](#)
  - VM0, [1291](#)
  - VM1, [1291](#)
  - VM10, [1291](#)
  - VM12, [1291](#)
  - VM16, [1291](#)
  - VM18, [1291](#)
  - VM1\_2, [1291](#)
  - VM1\_3, [1291](#)
  - VM1\_32, [1291](#)
  - VM1\_4, [1291](#)
  - VM1\_5, [1291](#)
  - VM1\_8, [1291](#)
  - VM1\_99, [1291](#)
  - VM1\_n, [1291](#)
  - VM2, [1291](#)
  - VM24, [1291](#)
  - VM256, [1291](#)
  - VM28, [1291](#)
  - VM2\_2n, [1291](#)
  - VM2\_n, [1292](#)
  - VM3, [1291](#)
  - VM30\_30n, [1292](#)
  - VM32, [1291](#)
  - VM35, [1291](#)
  - VM3\_3n, [1292](#)
  - VM3\_4, [1292](#)
  - VM3\_n, [1292](#)
  - VM4, [1291](#)
  - VM47\_47n, [1292](#)
  - VM4\_4n, [1292](#)
  - VM5, [1291](#)
  - VM6, [1291](#)
  - VM6\_6n, [1292](#)
  - VM6\_n, [1292](#)
  - VM7\_7n, [1292](#)
  - VM8, [1291](#)
  - VM9, [1291](#)
  - VM99, [1291](#)
  - VM\_END, [1292](#)
  - VMType, [1291](#)
- gdcM::VMToLength< T >, [1294](#)
- gdcM::VR, [1294](#)
  - AE, [1296](#)
  - AS, [1296](#)
  - AT, [1296](#)
  - CanDisplay, [1297](#)
  - Compatible, [1297](#)
  - CS, [1296](#)
  - DA, [1296](#)
  - DS, [1296](#)
  - DT, [1296](#)
  - FD, [1296](#)
  - FL, [1296](#)
  - GetLength, [1298](#)
  - GetSize, [1298](#)
  - GetSizeof, [1298](#)



GetVRString, [1298](#)  
GetVRStringFromFile, [1298](#)  
GetVRType, [1298](#)  
GetVRTypeFromFile, [1299](#)  
INVALID, [1296](#)  
IS, [1296](#)  
IsASCII, [1299](#)  
IsASCII2, [1299](#)  
IsBinary, [1299](#)  
IsBinary2, [1299](#)  
IsDual, [1299](#)  
IsSwap, [1299](#)  
IsValid, [1299](#), [1300](#)  
IsVRFile, [1300](#)  
LO, [1296](#)  
LT, [1296](#)  
OB, [1296](#)  
OB\_OW, [1297](#)  
OD, [1296](#)  
OF, [1296](#)  
OL, [1296](#)  
operator VRType, [1300](#)  
operator<<, [1300](#)  
OV, [1296](#)  
OW, [1296](#)  
PN, [1296](#)  
Read, [1300](#)  
SH, [1296](#)  
SL, [1296](#)  
SQ, [1296](#)  
SS, [1296](#)  
ST, [1297](#)  
SV, [1297](#)  
TM, [1297](#)  
UC, [1297](#)  
UI, [1297](#)  
UL, [1297](#)  
UN, [1297](#)  
UR, [1297](#)  
US, [1297](#)  
US\_OW, [1297](#)  
US\_SS, [1297](#)  
US\_SS\_OW, [1297](#)  
UT, [1297](#)  
UV, [1297](#)  
VL16, [1297](#)  
VL32, [1297](#)  
VR, [1297](#)  
VR\_END, [1297](#)  
VR\_VM1, [1297](#)  
VRALL, [1297](#)  
VRASCII, [1297](#)  
VRBINARY, [1297](#)  
VRType, [1296](#)  
Write, [1300](#)  
gdcm::VR16ExplicitDataElement, [1301](#)  
    GetLength, [1303](#)  
    Read, [1303](#)  
    ReadPreValue, [1304](#)  
    ReadValue, [1304](#)  
    ReadWithLength, [1304](#)  
gdcm::VRToEncoding< T >, [1304](#)  
gdcm::VRToType< T >, [1305](#)  
gdcm::VRVLSIZE< 0 >, [1305](#)  
    Read, [1306](#)  
    Write, [1306](#)  
gdcm::VRVLSIZE< 1 >, [1306](#)  
    Read, [1306](#)  
    Write, [1306](#)  
gdcm::VRVLSIZE< T >, [1305](#)  
gdcm::Waveform, [1406](#)  
    Waveform, [1406](#)  
gdcm::WLMFindQuery, [1406](#)  
    GetAbstractSyntaxUID, [1409](#)  
    GetTagListByLevel, [1409](#)  
    GetValidDataSet, [1410](#)  
    InitializeDataSet, [1410](#)  
    QueryFactory, [1410](#)  
    ValidateQuery, [1410](#)  
    WLMFindQuery, [1409](#)  
gdcm::Writer, [1411](#)  
    ~Writer, [1414](#)  
    CheckFileMetaInformationOff, [1414](#)  
    CheckFileMetaInformationOn, [1414](#)  
    GetCheckFileMetaInformation, [1414](#)  
    GetFile, [1414](#)  
    GetStreamPtr, [1414](#)  
    Ofstream, [1416](#)  
    SetCheckFileMetaInformation, [1414](#)  
    SetFile, [1415](#)  
    SetFileName, [1415](#)  
    SetStream, [1415](#)  
    SetWriteDataSetOnly, [1416](#)  
    Stream, [1416](#)  
    StreamImageWriter, [1416](#)  
    Write, [1416](#)  
    Writer, [1414](#)  
gdcm::XMLDictReader, [1417](#)  
    ~XMLDictReader, [1418](#)  
    CharacterDataHandler, [1419](#)  
    EndElement, [1419](#)  
    GetDict, [1419](#)  
    HandleDescription, [1419](#)  
    HandleEntry, [1419](#)  
    StartElement, [1419](#)  
    XMLDictReader, [1418](#)  
gdcm::XMLPrinter, [1420](#)  
    ~XMLPrinter, [1421](#)

- F, 1422
- GetPrintStyle, 1421
- HandleBulkData, 1421
- LOADBULKDATA, 1421
- OnlyUUID, 1421
- Print, 1421
- PrintDataElement, 1422
- PrintDataSet, 1422
- PrintSQ, 1422
- PrintStyle, 1422
- PrintStyles, 1421
- SetFile, 1422
- SetStyle, 1422
- XMLPrinter, 1421
- gdcmm::XMLPrivateDictReader, 1423
  - ~XMLPrivateDictReader, 1424
  - CharacterDataHandler, 1425
  - EndElement, 1425
  - GetPrivateDict, 1425
  - HandleDescription, 1425
  - HandleEntry, 1425
  - StartElement, 1425
  - XMLPrivateDictReader, 1424
- GDCM\_DIFFERENT
  - gdcmm, 60
- GDCM\_DO\_JOIN
  - gdcmmStaticAssert.h, 1479
- GDCM\_DO\_JOIN2
  - gdcmmStaticAssert.h, 1479
- GDCM\_EQUAL
  - gdcmm, 60
- GDCM\_EXPORT
  - gdcmmWin32.h, 1505
- GDCM\_FUNCTION
  - gdcmmTrace.h, 1497
- GDCM\_GREATER
  - gdcmm, 60
- GDCM\_GREATEROREQUAL
  - gdcmm, 60
- GDCM\_JOIN
  - gdcmmStaticAssert.h, 1479
- GDCM\_LEGACY
  - gdcmmLegacyMacro.h, 1459
- GDCM\_LEGACY\_BODY
  - gdcmmLegacyMacro.h, 1459
- GDCM\_LEGACY\_REPLACED\_BODY
  - gdcmmLegacyMacro.h, 1459
- GDCM\_LESS
  - gdcmm, 60
- GDCM\_LESOREQUAL
  - gdcmm, 60
- GDCM\_NOOP\_STATEMENT
  - gdcmmLegacyMacro.h, 1460
- GDCM\_STATIC\_ASSERT
  - gdcmm::Attribute< Group, Element, TVR, TVM >, 132
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 139, 140
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, 150
  - gdcmmStaticAssert.h, 1479
- gdcmmAAAbortPDU.h, 1920
- gdcmmAAssociateACPDU.h, 1921, 1922
- gdcmmAAssociateRJPDU.h, 1923, 1924
- gdcmmAAssociateRQPDU.h, 1925
- gdcmmAbstractSyntax.h, 1927, 1928
- gdcmmAnonymizeEvent.h, 1729, 1731
- gdcmmAnonymizer.h, 1731, 1732
- gdcmmApplicationContext.h, 1929, 1930
- gdcmmApplicationEntity.h, 1733, 1734
- gdcmmAReleaseRPPDU.h, 1931, 1932
- gdcmmAReleaseRQPDU.h, 1932, 1933
- gdcmmARTIMTimer.h, 1934, 1935
- gdcmmASN1.h, 1427, 1428
- gdcmmAssertAlwaysMacro
  - gdcmmTrace.h, 1497
- gdcmmAssertMacro
  - gdcmmTrace.h, 1497
- gdcmmAsynchronousOperationsWindowSub.h, 1936
- gdcmmAttribute.h, 1543, 1544
- gdcmmAudioCodec.h, 1735, 1736
- gdcmmBase64.h, 1429
- gdcmmBaseCompositeMessage.h, 1937, 1938
- gdcmmBaseNormalizedMessage.h, 1939
- gdcmmBasePDU.h, 1940, 1941
- gdcmmBaseQuery.h, 1942, 1943
- gdcmmBaseRootQuery.h, 1944, 1945
- gdcmmBasicOffsetTable.h, 1557, 1558
- gdcmmBitmap.h, 1736, 1737
- gdcmmBitmapToBitmapFilter.h, 1740
- gdcmmBoxRegion.h, 1430, 1431
- gdcmmByteBuffer.h, 1560, 1561
- gdcmmByteSwap.h, 1431, 1432
- gdcmmByteSwapFilter.h, 1563
- gdcmmByteValue.h, 1564, 1565
- gdcmmCAPICryptoFactory.h, 1433, 1434
- gdcmmCAPICryptographicMessageSyntax.h, 1434, 1435
- gdcmmCEchoMessages.h, 1946, 1947
- gdcmmCFindMessages.h, 1947, 1948
- gdcmmCleaner.h, 1741, 1742
- gdcmmCMoveMessages.h, 1949, 1950
- gdcmmCodec.h, 1743, 1744
- gdcmmCoder.h, 1744, 1745
- gdcmmCodeString.h, 1568, 1569
- gdcmmCommand.h, 1436, 1437
- gdcmmCommandDataSet.h, 1950, 1951
- gdcmmCompositeMessageFactory.h, 1952, 1953
- gdcmmCompositeNetworkFunctions.h, 1953, 1954
- gdcmmConstCharWrapper.h, 1746

- gdcmCP246ExplicitDataElement.h, 1570
- gdcmCryptoFactory.h, 1439, 1440
- gdcmCryptographicMessageSyntax.h, 1441
- gdcmCSAElement.h, 1571, 1572
- gdcmCSAHeader.h, 1574, 1575
- gdcmCSAHeaderDict.h, 1506, 1507
- gdcmCSAHeaderDictEntry.h, 1509, 1510
- gdcmCStoreMessages.h, 1955, 1956
- gdcmCurve.h, 1747, 1748
- gdcmDataElement.h, 1577, 1578
- gdcmDataEvent.h, 1443, 1444
- gdcmDataSet.h, 1580, 1581
- gdcmDataSetEvent.h, 1585, 1586
- gdcmDataSetHelper.h, 1749
- gdcmDebugMacro
  - gdcmTrace.h, 1497
- gdcmDecoder.h, 1750, 1751
- gdcmDefinedTerms.h, 1682, 1683
- gdcmDeflateStream.h, 1445
- gdcmDefs.h, 1683, 1684
- gdcmDeltaEncodingCodec.h, 1752
- gdcmDICOMDIR.h, 1753, 1754
- gdcmDICOMDIRGenerator.h, 1754, 1755
- gdcmDict.h, 1512, 1513
- gdcmDictConverter.h, 1517, 1518
- gdcmDictEntry.h, 1519, 1520
- gdcmDictPrinter.h, 1756, 1757
- gdcmDicts.h, 1521, 1523
- gdcmDIMSE.h, 1957
- gdcmDirectionCosines.h, 1757, 1758
- gdcmDirectory.h, 1445, 1446
- gdcmDirectoryHelper.h, 1759
- gdcmDPath.h, 1760, 1761
- gdcmDummyValueGenerator.h, 1448
- gdcmDumper.h, 1762, 1763
- gdcmElement.h, 1587, 1588
- gdcmEmptyMaskGenerator.h, 1763, 1764
- gdcmEncapsulatedDocument.h, 1765
- gdcmEnumeratedValues.h, 1685, 1686
- gdcmEquipmentManufacturer.h, 1766
- gdcmErrorMacro
  - gdcmTrace.h, 1498
- gdcmEvent.h, 1449, 1451
  - gdcmEventMacro, 1450
- gdcmEventMacro
  - gdcmEvent.h, 1450
- gdcmException.h, 1452, 1453
- gdcmExplicitDataElement.h, 1599, 1600
- gdcmExplicitImplicitDataElement.h, 1601, 1602
- gdcmFiducials.h, 1767, 1768
- gdcmFile.h, 1602, 1603
- gdcmFileAnonymizer.h, 1768, 1769
- gdcmFileChangeTransferSyntax.h, 1770, 1771
- gdcmFileDecompressLookupTable.h, 1772
- gdcmFileDerivation.h, 1773, 1774
- gdcmFileExplicitFilter.h, 1775
- gdcmFileMetaInformation.h, 1604, 1605
- gdcmFilename.h, 1454, 1455
- gdcmFileNameEvent.h, 1455, 1456
- gdcmFilenameGenerator.h, 1457, 1458
- gdcmFileSet.h, 1607, 1608
- gdcmFileStreamer.h, 1776, 1777
- gdcmFindPatientRootQuery.h, 1959, 1960
- gdcmFindStudyRootQuery.h, 1961
- gdcmFragment.h, 1609, 1610
- gdcmGlobal.h, 1524, 1525
- gdcmGroupDict.h, 1526, 1527
- gdcmIconImage.h, 1778, 1779
- gdcmIconImageFilter.h, 1780
- gdcmIconImageGenerator.h, 1781, 1782
- gdcmImage.h, 1783, 1784
- gdcmImageApplyLookupTable.h, 1785
- gdcmImageChangePhotometricInterpretation.h, 1786, 1787
- gdcmImageChangePlanarConfiguration.h, 1788, 1789
- gdcmImageChangeTransferSyntax.h, 1790, 1791
- gdcmImageCodec.h, 1791, 1792
- gdcmImageConverter.h, 1795
- gdcmImageFragmentSplitter.h, 1796, 1797
- gdcmImageHelper.h, 1797, 1798
- gdcmImageReader.h, 1800, 1801
- gdcmImageRegionReader.h, 1801, 1802
- gdcmImageToImageFilter.h, 1803, 1804
- gdcmImageWriter.h, 1804, 1805
- gdcmImplementationClassUIDSub.h, 1962, 1963
- gdcmImplementationUIDSub.h, 1964
- gdcmImplementationVersionNameSub.h, 1965, 1966
- gdcmImplicitDataElement.h, 1613, 1614
- gdcmIOD.h, 1686, 1688
- gdcmIODEntry.h, 1689, 1691
- gdcmIODs.h, 1691, 1693
- gdcmIPPSorter.h, 1806, 1807
- gdcmItem.h, 1615, 1616
- gdcmJPEG12Codec.h, 1808
- gdcmJPEG16Codec.h, 1809, 1810
- gdcmJPEG2000Codec.h, 1811
- gdcmJPEG8Codec.h, 1813
- gdcmJPEGCodec.h, 1814, 1815
- gdcmJPEGLSCodec.h, 1817
- gdcmJSON.h, 1818, 1819
- gdcmKAKADUCodec.h, 1820, 1821
- gdcmLegacyMacro.h, 1458, 1460
  - GDCM\_LEGACY, 1459
  - GDCM\_LEGACY\_BODY, 1459
  - GDCM\_LEGACY\_REPLACED\_BODY, 1459
  - GDCM\_NOOP\_STATEMENT, 1460
- gdcmLO.h, 1621
- gdcmLookupTable.h, 1821, 1822

gdcMacro.h, 1694, 1695  
 gdcMacroEntry.h, 1697, 1698  
     GDCMMACROENTRY\_H, 1698  
 GDCMMACROENTRY\_H  
     gdcMacroEntry.h, 1698  
 gdcMacros.h, 1700, 1701  
 gdcMaximumLengthSub.h, 1967, 1968  
 gdcMD5.h, 1461, 1462  
 gdcMEC\_MR3.h, 1824  
 gdcMediaStorage.h, 1622, 1623  
 gdcMeshPrimitive.h, 1825, 1826  
 gdcModalityPerformedProcedureStepCreateQuery.h, 1969  
 gdcModalityPerformedProcedureStepSetQuery.h, 1970, 1971  
 gdcModule.h, 1702, 1703  
 gdcModuleEntry.h, 1705, 1707  
 gdcModules.h, 1708, 1709  
 gdcMovePatientRootQuery.h, 1971, 1972  
 gdcMoveStudyRootQuery.h, 1973  
 gdcMrProtocol.h, 1626, 1627  
 gdcNActionMessages.h, 1974, 1975  
 gdcNCreateMessages.h, 1975, 1976  
 gdcNDeleteMessages.h, 1977  
 gdcNestedModuleEntries.h, 1710, 1711  
 gdcNetworkEvents.h, 1978, 1979  
 gdcNetworkStateID.h, 1980, 1981  
 gdcNEventReportMessages.h, 1982, 1983  
 gdcNGetMessages.h, 1983, 1984  
 gdcNormalizedMessageFactory.h, 1984, 1985  
 gdcNormalizedNetworkFunctions.h, 1986, 1987  
 gdcNSetMessages.h, 1988  
 gdcObject.h, 1462, 1463  
 gdcOpenSSLCryptoFactory.h, 1464, 1465  
 gdcOpenSSLCryptographicMessageSyntax.h, 1466, 1467  
 gdcOpenSSLPT7CryptoFactory.h, 1468, 1469  
 gdcOpenSSLPT7CryptographicMessageSyntax.h, 1469, 1471  
 gdcOrientation.h, 1828  
 gdcOverlay.h, 1829, 1830  
 gdcParseException.h, 1628, 1629  
 gdcParser.h, 1630, 1631  
 gdcPatient.h, 1712  
 gdcPDDataTFPDU.h, 1989, 1990  
 gdcPDBelement.h, 1633, 1634  
 gdcPDHeader.h, 1635  
 gdcPDFCodec.h, 1832  
 gdcPDUFactory.h, 1991  
 gdcPersonName.h, 1833, 1834  
 gdcPGXCodec.h, 1835  
 gdcPhotometricInterpretation.h, 1836, 1837  
 gdcPixelFormat.h, 1838, 1840  
 gdcPixmap.h, 1842, 1843  
 gdcPixmapReader.h, 1844, 1846  
 gdcPixmapToPixmapFilter.h, 1847  
 gdcPixmapWriter.h, 1848, 1849  
 gdcPNMCodec.h, 1850, 1851  
 gdcPreamble.h, 1636, 1638  
 gdcPresentationContext.h, 1992, 1993  
 gdcPresentationContextAC.h, 1994, 1996  
 gdcPresentationContextGenerator.h, 1996, 1997  
 gdcPresentationContextRQ.h, 1998, 1999  
 gdcPresentationDataValue.h, 2000, 2001  
 gdcPrinter.h, 1851, 1853  
 gdcPrivateTag.h, 1639, 1640  
 gdcProgressEvent.h, 1471, 1472  
 gdcPVRGCodec.h, 1854, 1855  
 gdcPythonFilter.h, 2094, 2095  
 gdcQueryBase.h, 2002, 2004  
 gdcQueryFactory.h, 2005, 2006  
 gdcQueryImage.h, 2006, 2007  
 gdcQueryPatient.h, 2008, 2009  
 gdcQuerySeries.h, 2010  
 gdcQueryStudy.h, 2011, 2012  
 gdcRAWCodec.h, 1855, 1856  
 gdcReader.h, 1641, 1642  
 gdcRegion.h, 1473, 1474  
 gdcRescaler.h, 1857  
 gdcRLECodec.h, 1859  
 gdcRoleSelectionSub.h, 2013  
 gdcScanner.h, 1860, 1861  
 gdcScanner2.h, 1863, 1864  
 gdcSegment.h, 1866, 1868  
 gdcSegmentedPaletteColorLookupTable.h, 1870  
 gdcSegmentHelper.h, 1871, 1872  
 gdcSegmentReader.h, 1873, 1875  
 gdcSegmentWriter.h, 1875, 1877  
 gdcSequenceOfFragments.h, 1643, 1644  
 gdcSequenceOfItems.h, 1648, 1649  
 gdcSerieHelper.h, 1877, 1879  
 gdcSeries.h, 1713, 1714  
 gdcServiceClassApplicationInformation.h, 2014, 2015  
 gdcServiceClassUser.h, 2016, 2017  
 gdcSHA1.h, 1475, 1476  
 gdcSimpleSubjectWatcher.h, 1880, 1881  
 gdcSmartPointer.h, 1476, 1477  
 gdcSOPClassExtendedNegotiationSub.h, 2018, 2019  
 gdcSOPClassUIDToIOD.h, 1528  
 gdcSorter.h, 1882, 1884  
 gdcSpacing.h, 1885  
 gdcSpectroscopy.h, 1886, 1887  
 gdcSplitMosaicFilter.h, 1887, 1888  
 gdcStaticAssert.h, 1478, 1480  
     GDCM\_DO\_JOIN, 1479  
     GDCM\_DO\_JOIN2, 1479  
     GDCM\_JOIN, 1479  
     GDCM\_STATIC\_ASSERT, 1479

gdcmStreamImageReader.h, 1889, 1890  
gdcmStreamImageWriter.h, 1891, 1892  
gdcmStrictScanner.h, 1893, 1894  
gdcmStrictScanner2.h, 1895, 1896  
gdcmString.h, 1480, 1481  
gdcmStringFilter.h, 1898, 1899  
gdcmStudy.h, 1715, 1716  
gdcmSubject.h, 1483, 1484  
gdcmSurface.h, 1900, 1901  
gdcmSurfaceHelper.h, 1904, 1905  
gdcmSurfaceReader.h, 1907, 1908  
gdcmSurfaceWriter.h, 1909, 1910  
gdcmSwapCode.h, 1485, 1486  
gdcmSwapper.h, 1486, 1487  
gdcmSystem.h, 1489  
gdcmTable.h, 1716, 1717  
gdcmTableEntry.h, 1718, 1719  
gdcmTableReader.h, 1720, 1721  
gdcmTag.h, 1652, 1653  
gdcmTagPath.h, 1910, 1911  
gdcmTagToVR.h, 1657  
gdcmTerminal.h, 1491, 1492  
gdcmTestDriver.h, 1493  
gdcmTesting.h, 1494  
gdcmTrace.h, 1495, 1499  
    GDCM\_FUNCTION, 1497  
    gdcmAssertAlwaysMacro, 1497  
    gdcmAssertMacro, 1497  
    gdcmDebugMacro, 1497  
    gdcmErrorMacro, 1498  
    gdcmWarningMacro, 1498  
gdcmTransferSyntax.h, 1658, 1659  
gdcmTransferSyntaxSub.h, 2019, 2021  
gdcmType.h, 1722, 1723  
gdcmTypes.h, 1501, 1502  
gdcmUIDGenerator.h, 1912, 1913  
gdcmUIDs.h, 1529, 1530  
gdcmULAction.h, 2021, 2022  
gdcmULActionAA.h, 2023, 2024  
gdcmULActionAE.h, 2025, 2026  
gdcmULActionAR.h, 2027, 2028  
gdcmULActionDT.h, 2030  
gdcmULBasicCallback.h, 2031, 2032  
gdcmULConnection.h, 2032, 2033  
gdcmULConnectionCallback.h, 2035, 2036  
gdcmULConnectionInfo.h, 2036, 2038  
gdcmULConnectionManager.h, 2038, 2039  
gdcmULEvent.h, 2041, 2042  
gdcmULTransitionTable.h, 2043, 2044  
gdcmULWritingCallback.h, 2046  
gdcmUNExplicitDataElement.h, 1660, 1661  
gdcmUNExplicitImplicitDataElement.h, 1662, 1663  
gdcmUnpacker12Bits.h, 1502, 1503  
gdcmUsage.h, 1724, 1726  
gdcmUserInformation.h, 2047, 2048  
gdcmUUIDGenerator.h, 1914  
gdcmValidate.h, 1915  
gdcmValue.h, 1663, 1664  
gdcmValueIO.h, 1665, 1666  
gdcmVersion.h, 1503, 1504  
gdcmVL.h, 1666, 1667  
gdcmVM.h, 1669, 1670  
    TYPETOLENGTH, 1670  
gdcmVR.h, 1672, 1674  
    TYPETOENCODING, 1673  
    VRTypeTemplateCase, 1673  
gdcmVR16ExplicitDataElement.h, 1678, 1679  
gdcmWarningMacro  
    gdcmTrace.h, 1498  
gdcmWaveform.h, 1916, 1917  
gdcmWin32.h, 1504, 1505  
    GDCM\_EXPORT, 1505  
gdcmWLMFindQuery.h, 2049, 2050  
gdcmWriter.h, 1680, 1681  
gdcmXMLDictReader.h, 1727  
gdcmXMLPrinter.h, 1917, 1918  
gdcmXMLPrivateDictReader.h, 1728, 1729  
GEMS  
    gdcm::Dicts, 372  
    gdcm::EquipmentManufacturer, 431  
GeneralAudioWaveformStorage  
    gdcm::UIDs, 1184  
GeneralECGWaveformStorage  
    gdcm::MediaStorage, 693  
    gdcm::UIDs, 1180  
GeneralElectricMagneticResonanceImageStorage  
    gdcm::MediaStorage, 694  
GeneralPurposePerformedProcedureStepSOPClass  
    gdcm::UIDs, 1182  
GeneralPurposeScheduledProcedureStepSOPClass  
    gdcm::UIDs, 1182  
GeneralPurposeWorklistInformationModelFIND  
    gdcm::UIDs, 1182  
GeneralPurposeWorklistManagementMetaSOPClass  
    gdcm::UIDs, 1182  
GeneralRelevantPatientInformationQuery  
    gdcm::UIDs, 1182  
Generate  
    gdcm::DICOMDIRGenerator, 354  
    gdcm::DummyValueGenerator, 388  
    gdcm::FilenameGenerator, 489  
    gdcm::IconImageGenerator, 525  
    gdcm::UIDGenerator, 1159  
    gdcm::UUIDGenerator, 1277  
GenerateFromFilenames  
    gdcm::PresentationContextGenerator, 863  
GenerateFromUID  
    gdcm::PresentationContextGenerator, 863

- GenerateUUID
  - gdcm::UIDGenerator, [1159](#)
- GenericImplantTemplateInformationModelFIND
  - gdcm::UIDs, [1186](#)
- GenericImplantTemplateInformationModelGET
  - gdcm::UIDs, [1186](#)
- GenericImplantTemplateInformationModelMOVE
  - gdcm::UIDs, [1186](#)
- GenericImplantTemplateStorage
  - gdcm::UIDs, [1186](#)
- GEPrivate3DModelStorage
  - gdcm::MediaStorage, [694](#)
- Get
  - gdcm::ByteBuffer, [214](#)
- GetAbbreviation
  - gdcm::GroupDict, [519](#)
- GetAbstractSyntax
  - gdcm::network::PresentationContextRQ, [865](#), [866](#)
  - gdcm::PresentationContext, [857](#)
- GetAbstractSyntaxUID
  - gdcm::BaseQuery, [177](#)
  - gdcm::FindPatientRootQuery, [504](#)
  - gdcm::FindStudyRootQuery, [508](#)
  - gdcm::ModalityPerformedProcedureStepCreateQuery, [713](#)
  - gdcm::ModalityPerformedProcedureStepSetQuery, [716](#)
  - gdcm::MovePatientRootQuery, [729](#)
  - gdcm::MoveStudyRootQuery, [733](#)
  - gdcm::WLMFindQuery, [1409](#)
- GetAcceptedPresentationContexts
  - gdcm::network::ULConnection, [1242](#)
- GetAcquisitionSize
  - gdcm::SplitMosaicFilter, [1033](#)
- GetAETitle
  - gdcm::ServiceClassUser, [1000](#)
- GetAlgorithmFamily
  - gdcm::Surface, [1085](#)
- GetAlgorithmName
  - gdcm::Surface, [1085](#)
- GetAlgorithmVersion
  - gdcm::Surface, [1085](#)
- GetALGOType
  - gdcm::Segment, [957](#)
- GetALGOTypeString
  - gdcm::Segment, [957](#)
- GetAllFilenamesFromPrivateTagToValue
  - gdcm::Scanner2, [949](#)
  - gdcm::StrictScanner2, [1062](#)
- GetAllFilenamesFromPublicTagToValue
  - gdcm::Scanner2, [949](#)
  - gdcm::StrictScanner2, [1063](#)
- GetAllFilenamesFromTagToValue
  - gdcm::Scanner, [939](#)
  - gdcm::StrictScanner, [1053](#)
- GetAllRequiredTags
  - gdcm::QueryBase, [893](#)
- GetAllTags
  - gdcm::QueryBase, [893](#)
- GetAnatomicRegion
  - gdcm::Segment, [957](#)
- GetAnatomicRegionModifiers
  - gdcm::Segment, [958](#)
- GetAsDataElement
  - gdcm::Attribute< Group, Element, TVR, TVM >, [132](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [140](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [150](#)
  - gdcm::Element< TVR, TVM >, [393](#)
  - gdcm::Element< TVR, VM::VM1\_n >, [401](#)
  - gdcm::network::AbstractSyntax, [101](#)
  - gdcm::PrivateTag, [881](#)
- GetAsPoints
  - gdcm::Curve, [305](#)
- GetAsString
  - gdcm::CodeString, [256](#)
- GetAxisOfRotation
  - gdcm::Surface, [1085](#)
- GetBasicApplicationLevelConfidentialityProfileAttributes
  - gdcm::Anonymizer, [111](#)
- GetBitPosition
  - gdcm::Overlay, [783](#)
- GetBitsAllocated
  - gdcm::Overlay, [783](#)
  - gdcm::PixelFormat, [823](#)
- GetBitSample
  - gdcm::LookupTable, [676](#)
- GetBitsStored
  - gdcm::PixelFormat, [823](#)
- GetBlob
  - gdcm::network::PresentationDataValue, [869](#)
- GetBuffer
  - gdcm::Bitmap, [196](#)
  - gdcm::ByteValue, [221](#)
  - gdcm::Parser, [793](#)
  - gdcm::SequenceOfFragments, [978](#)
- GetBuffer2
  - gdcm::Bitmap, [196](#)
- GetBufferAsRGBA
  - gdcm::LookupTable, [676](#)
- GetBufferLength
  - gdcm::Bitmap, [196](#)
  - gdcm::JPEGLSCodec, [660](#)
  - gdcm::PNMCodec, [851](#)
  - gdcm::RLECodec, [931](#)
- GetBuildVersion
  - gdcm::Version, [1284](#)



- GetByteValue
  - gdcm::CSAElement, [283](#)
  - gdcm::DataElement, [312](#)
- GetCalledAETitle
  - gdcm::network::AAssociateRQPDU, [96](#)
  - gdcm::network::ULConnectionInfo, [1248](#)
  - gdcm::ServiceClassUser, [1000](#)
- GetCalledComputerName
  - gdcm::network::ULConnectionInfo, [1248](#)
- GetCalledIPAddress
  - gdcm::network::ULConnectionInfo, [1248](#)
- GetCalledIPPort
  - gdcm::network::ULConnectionInfo, [1248](#)
- GetCallingAETitle
  - gdcm::network::AAssociateRQPDU, [96](#)
  - gdcm::network::ULConnectionInfo, [1248](#)
- GetCanonMECMR3Tag
  - gdcm::MEC\_MR3, [688](#)
- GetCenterOfRotation
  - gdcm::Surface, [1085](#)
- GetCharacterFromCurrentLocale
  - gdcm::QueryFactory, [895](#)
- GetCheckFileMetaInformation
  - gdcm::Writer, [1414](#)
- GetCipherType
  - gdcm::CAPICryptographicMessageSyntax, [229](#)
  - gdcm::CryptographicMessageSyntax, [280](#)
  - gdcm::OpenSSLCryptographicMessageSyntax, [770](#)
  - gdcm::OpenSSL7CryptographicMessageSyntax, [775](#)
- GetCodec
  - gdcm::FileChangeTransferSyntax, [460](#)
- GetColorLevel
  - vtkImageColorViewer, [1370](#)
- GetColorWindow
  - vtkImageColorViewer, [1370](#)
- GetColumns
  - gdcm::Bitmap, [197](#)
  - gdcm::Overlay, [783](#)
- GetCommand
  - gdcm::Subject, [1079](#)
- GetConnectionInfo
  - gdcm::network::ULConnection, [1242](#)
- GetConstructorString
  - gdcm::Dicts, [373](#)
- GetContourReferencedFrameOfReferenceClassUID
  - vtkRTStructSetProperties, [1400](#)
- GetContourReferencedFrameOfReferenceInstanceUID
  - vtkRTStructSetProperties, [1400](#)
- GetCryptographicMessageSyntax
  - gdcm::Anonymizer, [111](#)
- GetCSADataInfo
  - gdcm::CSAHeader, [290](#)
- GetCSAEnd
  - gdcm::CSAHeader, [290](#)
- GetCSAElementByName
  - gdcm::CSAHeader, [291](#)
- GetCSAHeaderDict
  - gdcm::Dicts, [373](#)
- GetCSAHeaderDictEntry
  - gdcm::CSAHeaderDict, [295](#)
- GetCSAImageHeaderInfoTag
  - gdcm::CSAHeader, [291](#)
- GetCSASeriesHeaderInfoTag
  - gdcm::CSAHeader, [291](#)
- GetCTImageSeriesUIDs
  - gdcm::DirectoryHelper, [384](#)
- GetCurrentByteIndex
  - gdcm::Parser, [793](#)
- GetCurrentDateTime
  - gdcm::System, [1111](#)
- GetCurrentModuleFileName
  - gdcm::System, [1111](#)
- GetCurrentProcessFileName
  - gdcm::System, [1111](#)
- GetCurrentResourcesDirectory
  - gdcm::System, [1111](#)
- GetCurve
  - gdcm::Pixmap, [832](#)
- GetCurveDataDescriptor
  - gdcm::Curve, [305](#)
- GetCWD
  - gdcm::System, [1111](#)
- GetData
  - gdcm::DataEvent, [324](#)
- GetDataElement
  - gdcm::Bitmap, [197](#)
  - gdcm::DataSet, [330](#)
  - gdcm::Item, [620](#)
- GetDataExtraRoot
  - gdcm::Testing, [1137](#)
- GetDataLength
  - gdcm::DataEvent, [324](#)
- GetDataRoot
  - gdcm::Testing, [1137](#)
- GetDataSet
  - gdcm::CSAHeader, [291](#)
  - gdcm::DataSetEvent, [339](#)
  - gdcm::File, [450](#)
- GetDataSetPos
  - gdcm::network::ULEvent, [1257](#)
- GetDataSets
  - gdcm::network::ULBasicCallback, [1239](#)
- GetDataSetTransferSyntax
  - gdcm::FileMetaInformation, [476](#)
- GetDataValueRepresentation
  - gdcm::Curve, [305](#)
- GetDebugFlag

- gdcmm::Trace, 1143
- GetDebugStream
  - gdcmm::Trace, 1144
- GetDecodeLength
  - gdcmm::Base64, 168
- GetDEEnd
  - gdcmm::DataSet, 331
- GetDefaultTransferSyntax
  - gdcmm::PresentationContextGenerator, 863
- GetDefs
  - gdcmm::Global, 516
  - gdcmm::TableReader, 1120
- GetDES
  - gdcmm::DataSet, 331
- GetDescription
  - gdcmm::CSAHeaderDictEntry, 297
  - gdcmm::Exception, 437
  - gdcmm::ModuleEntry, 723
  - gdcmm::Overlay, 783
- GetDescriptiveName
  - vtkGDCMImageReader, 1310
  - vtkGDCMImageReader2, 1323
  - vtkGDCMImageWriter, 1334
- GetDict
  - gdcmm::XMLDictReader, 1419
- GetDictEntry
  - gdcmm::Dict, 358
  - gdcmm::Dicts, 373
  - gdcmm::PrivateDict, 876
- GetDictEntryByKeyword
  - gdcmm::Dict, 358
- GetDictEntryByName
  - gdcmm::Dict, 359
- GetDictName
  - gdcmm::DictConverter, 362
- GetDicts
  - gdcmm::Global, 516
- GetDictVM
  - gdcmm::Attribute< Group, Element, TVR, TVM >, 133
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 140
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, 150
- GetDictVR
  - gdcmm::Attribute< Group, Element, TVR, TVM >, 133
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 140
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, 150
- GetDimension
  - gdcmm::Bitmap, 197
- GetDimensions
  - gdcmm::Bitmap, 197
  - gdcmm::Curve, 305
  - gdcmm::ImageCodec, 560
- GetDimensionsValue
  - gdcmm::ImageHelper, 572
- GetDimensionsValueForResolution
  - gdcmm::StreamImageReader, 1039
- GetDirectionCosines
  - gdcmm::Image, 533
- GetDirectionCosinesFromDataSet
  - gdcmm::ImageHelper, 572
- GetDirectionCosinesTolerance
  - gdcmm::IPPSorter, 613
- GetDirectionCosinesValue
  - gdcmm::ImageHelper, 573
- GetDirectories
  - gdcmm::Directory, 382
- GetElapsedTime
  - gdcmm::network::ARTIMTimer, 126
- GetElement
  - gdcmm::Tag, 1126
- GetElementTag
  - gdcmm::Tag, 1126
- GetEncodeLength
  - gdcmm::Base64, 168
- GetErrorCode
  - gdcmm::Parser, 793
- GetErrorFlag
  - gdcmm::Trace, 1144
- GetErrorStream
  - gdcmm::Trace, 1144
- GetErrorString
  - gdcmm::Parser, 793
- GetEvent
  - gdcmm::network::ULEvent, 1257
- GetEventName
  - gdcmm::AnonymizeEvent, 105
  - gdcmm::DataEvent, 324
  - gdcmm::DataSetEvent, 339
  - gdcmm::Event, 434
  - gdcmm::FileNameEvent, 486
  - gdcmm::ProgressEvent, 885
- GetExtension
  - gdcmm::Filename, 482
- GetFactoryInstance
  - gdcmm::CryptoFactory, 277
- GetFile
  - gdcmm::Anonymizer, 112
  - gdcmm::Cleaner, 243
  - gdcmm::DICOMDIRGenerator, 355
  - gdcmm::FileDecompressLookupTable, 463
  - gdcmm::FileDerivation, 466, 467
  - gdcmm::FileExplicitFilter, 470
  - gdcmm::IconImageFilter, 522
  - gdcmm::PythonFilter, 891
  - gdcmm::Reader, 914



- gdcmm::SplitMosaicFilter, 1034
- gdcmm::StreamImageReader, 1040
- gdcmm::StringFilter, 1074
- gdcmm::Writer, 1414
- vtkGDCMMedicalImageProperties, 1341
- GetFileExtensions
  - vtkGDCMImageReader, 1311
  - vtkGDCMImageReader2, 1323
  - vtkGDCMImageWriter, 1334
- GetFileMetaInformationVersion
  - gdcmm::FileMetaInformation, 476
- GetFileName
  - gdcmm::Filename, 482
  - gdcmm::FileNameEvent, 486
  - gdcmm::Testing, 1137
  - vtkGDCMImageWriter, 1334
  - vtkGDCMThreadedImageReader2, 1361
- GetFilename
  - gdcmm::FilenameGenerator, 489
  - gdcmm::TableReader, 1120
- GetFilenameFromPrivateTagToValue
  - gdcmm::Scanner2, 949
  - gdcmm::StrictScanner2, 1063
- GetFilenameFromPublicTagToValue
  - gdcmm::Scanner2, 949
  - gdcmm::StrictScanner2, 1063
- GetFilenameFromTagToValue
  - gdcmm::Scanner, 939
  - gdcmm::StrictScanner, 1053
- GetFileNames
  - gdcmm::Testing, 1137
- GetFilenames
  - gdcmm::Directory, 382
  - gdcmm::FilenameGenerator, 489
  - gdcmm::Scanner, 940
  - gdcmm::Scanner2, 949
  - gdcmm::Sorter, 1026
  - gdcmm::StrictScanner, 1053
  - gdcmm::StrictScanner2, 1063
- GetFilenamesFromSeriesUIDs
  - gdcmm::DirectoryHelper, 384
- GetFiles
  - gdcmm::FileSet, 492
- GetFiniteVolume
  - gdcmm::Surface, 1085
- GetFirstSingleSeriesUIDFileSet
  - gdcmm::SerieHelper, 993
- GetForcePixelSpacing
  - gdcmm::ImageHelper, 573
- GetForceRescaleInterceptSlope
  - gdcmm::ImageHelper, 573
- GetFormat
  - gdcmm::CSAHeader, 291
- GetFragBuffer
  - gdcmm::SequenceOfFragments, 979
- GetFragment
  - gdcmm::SequenceOfFragments, 979
- GetFragmentSizeMax
  - gdcmm::ImageFragmentSplitter, 570
- GetFrameOfReference
  - gdcmm::DirectoryHelper, 384
- GetFullLength
  - gdcmm::FileMetaInformation, 476
- GetGDCMDataRoot
  - vtkGDCMTesting, 1352
- GetGDCMImplementationClassUID
  - gdcmm::FileMetaInformation, 476
- GetGDCMImplementationVersionName
  - gdcmm::FileMetaInformation, 476
- GetGDCMSourceApplicationEntityTitle
  - gdcmm::FileMetaInformation, 477
- GetGDCMUID
  - gdcmm::UIDGenerator, 1160
- GetGroup
  - gdcmm::Curve, 305
  - gdcmm::Overlay, 783
  - gdcmm::Tag, 1126
- GetHasExpired
  - gdcmm::network::ARTIMTimer, 126
- GetHeader
  - gdcmm::File, 451
- GetHeaderInfo
  - gdcmm::ImageCodec, 560
  - gdcmm::JPEG12Codec, 627
  - gdcmm::JPEG16Codec, 632
  - gdcmm::JPEG2000Codec, 638
  - gdcmm::JPEG8Codec, 645
  - gdcmm::JPEGCodec, 652
  - gdcmm::JPEGLSCodec, 660
  - gdcmm::PGXCodec, 815
  - gdcmm::PNMCodec, 851
  - gdcmm::RAWCodec, 910
  - gdcmm::RLECodec, 931
- GetHierarchicalSearchTags
  - gdcmm::QueryBase, 893
  - gdcmm::QueryImage, 897
  - gdcmm::QueryPatient, 900
  - gdcmm::QuerySeries, 902
  - gdcmm::QueryStudy, 905
- GetHighBit
  - gdcmm::PixelFormat, 823
- GetHostName
  - gdcmm::System, 1111
- GetIconImage
  - gdcmm::IconImageFilter, 522
  - gdcmm::IconImageGenerator, 525
  - gdcmm::Pixmap, 832
  - vtkGDCMImageReader, 1311

- vtkGDCMImageReader2, [1323](#)
- GetIconImagePort
  - vtkGDCMImageReader2, [1323](#)
- GetIE
  - gdcm::IODEntry, [607](#)
- GetImage
  - gdcm::ImageReader, [580](#)
  - gdcm::ImageWriter, [593](#), [594](#)
  - gdcm::PixmapWriter, [845](#)
  - gdcm::SplitMosaicFilter, [1034](#)
- GetImplementationClassUID
  - gdcm::FileMetaInformation, [477](#)
- GetImplementationVersionName
  - gdcm::FileMetaInformation, [477](#)
- GetIndex
  - gdcm::SwapCode, [1105](#)
  - gdcm::VM, [1292](#)
- GetInitialized
  - gdcm::CAPICryptographicMessageSyntax, [229](#)
- GetInput
  - gdcm::ImageToImageFilter, [589](#)
  - gdcm::PixmapToPixmapFilter, [841](#)
  - vtkImageColorViewer, [1370](#)
- GetInputFilename
  - gdcm::DictConverter, [362](#)
- GetInstance
  - gdcm::Global, [516](#)
- GetIntercept
  - gdcm::Image, [533](#)
  - gdcm::Rescaler, [924](#)
- GetInterfile
  - gdcm::CSAHeader, [292](#)
- GetInternal
  - gdcm::Preamble, [853](#)
- GetIOD
  - gdcm::IODs, [610](#)
  - gdcm::SOPClassUIDToIOD, [1023](#)
- GetIODEntry
  - gdcm::IOD, [605](#)
- GetIODFromFile
  - gdcm::Defs, [345](#)
- GetIODFromSOPClassUID
  - gdcm::SOPClassUIDToIOD, [1023](#)
- GetIODNameFromMediaStorage
  - gdcm::Defs, [345](#)
- GetIODs
  - gdcm::Defs, [345](#)
- GetIsCommand
  - gdcm::network::PresentationDataValue, [869](#)
- GetIsLastFragment
  - gdcm::network::PresentationDataValue, [869](#)
- GetIStream
  - gdcm::network::ULEvent, [1257](#)
- GetItem
  - gdcm::SequenceOfItems, [986](#), [987](#)
- GetKey
  - gdcm::CSAElement, [283](#)
- GetKeys
  - gdcm::Scanner, [940](#)
  - gdcm::Scanner2, [950](#)
  - gdcm::StrictScanner, [1053](#)
  - gdcm::StrictScanner2, [1063](#)
- GetKeyword
  - gdcm::DictEntry, [365](#)
- GetKeywordFromTag
  - gdcm::Dict, [359](#)
- GetLabel
  - gdcm::Orientation, [778](#)
- GetLastElement
  - gdcm::ParseException, [790](#)
- GetLastSystemError
  - gdcm::System, [1112](#)
- GetLength
  - gdcm::ByteValue, [221](#)
  - gdcm::CP246ExplicitDataElement, [274](#)
  - gdcm::DataElement, [312](#)
  - gdcm::DataSet, [331](#)
  - gdcm::Element< TVR, TVM >, [393](#)
  - gdcm::Element< TVR, VM::VM1\_n >, [401](#)
  - gdcm::Element< VR::AS, VM::VM5 >, [416](#)
  - gdcm::ExplicitDataElement, [442](#)
  - gdcm::ExplicitImplicitDataElement, [446](#)
  - gdcm::Fragment, [513](#)
  - gdcm::ImplicitDataElement, [601](#)
  - gdcm::Item, [620](#)
  - gdcm::Preamble, [853](#)
  - gdcm::SequenceOfFragments, [979](#)
  - gdcm::SequenceOfItems, [987](#)
  - gdcm::Tag, [1126](#)
  - gdcm::UNExplicitDataElement, [1264](#)
  - gdcm::UNExplicitImplicitDataElement, [1268](#)
  - gdcm::Value, [1281](#)
  - gdcm::VL, [1287](#)
  - gdcm::VM, [1292](#)
  - gdcm::VR, [1298](#)
  - gdcm::VR16ExplicitDataElement, [1303](#)
- GetLocaleCharset
  - gdcm::System, [1112](#)
- GetLossless
  - gdcm::JPEGCodec, [652](#)
  - gdcm::JPEGLSCodec, [660](#)
- GetLossyFlag
  - gdcm::ImageCodec, [560](#)
- GetLossyFlagFromFile
  - gdcm::Testing, [1138](#)
- GetLUT
  - gdcm::Bitmap, [198](#)
  - gdcm::ImageCodec, [560](#)

- gdcm::ImageHelper, [573](#)
- gdcm::LookupTable, [676](#)
- GetLUTDescriptor
  - gdcm::LookupTable, [676](#)
- GetLUTLength
  - gdcm::LookupTable, [676](#)
- GetMacro
  - gdcm::Macros, [685](#)
- GetMacroEntry
  - gdcm::Macro, [683](#)
- GetMacros
  - gdcm::Defs, [345](#), [346](#)
- GetMajorAxisFromPatientRelativeDirectionCosine
  - gdcm::Orientation, [778](#)
- GetMajorVersion
  - gdcm::Version, [1284](#)
- GetManifold
  - gdcm::Surface, [1085](#)
- GetMapping
  - gdcm::Scanner, [940](#)
  - gdcm::StrictScanner, [1053](#)
- GetMappingFromPrivateTagToValue
  - gdcm::Scanner2, [950](#)
  - gdcm::StrictScanner2, [1063](#)
- GetMappingFromPublicTagToValue
  - gdcm::Scanner2, [950](#)
  - gdcm::StrictScanner2, [1063](#)
- GetMappingFromTagToValue
  - gdcm::Scanner, [940](#)
  - gdcm::StrictScanner, [1054](#)
- GetMappings
  - gdcm::Scanner, [940](#)
  - gdcm::StrictScanner, [1054](#)
- GetMax
  - gdcm::PixelFormat, [823](#)
- GetMaximumLength
  - gdcm::network::MaximumLengthSub, [686](#)
- GetMaximumLengthSub
  - gdcm::network::UserInformation, [1275](#)
- GetMaximumPointDistance
  - gdcm::Surface, [1086](#)
- GetMaxLength
  - gdcm::PersonName, [810](#)
- GetMaxPDULength
  - gdcm::network::ULConnectionInfo, [1248](#)
- GetMaxPDUSize
  - gdcm::network::ULConnection, [1242](#)
- GetMD5DataImage
  - gdcm::Testing, [1138](#)
- GetMD5DataImages
  - gdcm::Testing, [1138](#)
- GetMD5FromBrokenFile
  - gdcm::Testing, [1138](#)
- GetMD5FromFile
  - gdcm::Testing, [1138](#)
- GetMD5MetaImage
  - gdcm::Testing, [1138](#)
- GetMD5MetaImage
  - gdcm::Testing, [1138](#)
- GetMeanPointDistance
  - gdcm::Surface, [1086](#)
- GetMediaStorage
  - gdcm::DataSet, [331](#)
  - gdcm::FileMetaInformation, [477](#)
- GetMediaStorageAsString
  - gdcm::FileMetaInformation, [477](#)
- GetMediaStorageDataFile
  - gdcm::Testing, [1138](#)
- GetMediaStorageDataFiles
  - gdcm::Testing, [1138](#)
- GetMediaStorageFromFile
  - gdcm::Testing, [1139](#)
- GetMeshPrimitive
  - gdcm::Surface, [1086](#)
- GetMessageHeader
  - gdcm::network::PresentationDataValue, [869](#)
- GetMetaInformationTS
  - gdcm::FileMetaInformation, [477](#)
- GetMHDMD5FromFile
  - gdcm::Testing, [1138](#)
- GetMin
  - gdcm::PixelFormat, [823](#)
- GetMinorVersion
  - gdcm::Version, [1285](#)
- GetModality
  - gdcm::MediaStorage, [696](#)
- GetModalityDimension
  - gdcm::MediaStorage, [696](#)
- GetModule
  - gdcm::Modules, [726](#)
- GetModuleEntry
  - gdcm::NestedModuleEntries, [748](#)
- GetModuleEntryInMacros
  - gdcm::Module, [720](#)
- GetModules
  - gdcm::Defs, [346](#)
- GetMPTType
  - gdcm::MeshPrimitive, [708](#)
- GetMPTTypeString
  - gdcm::MeshPrimitive, [708](#)
- GetMRImageSeriesUIDs
  - gdcm::DirectoryHelper, [384](#)
- GetMrProtocol
  - gdcm::CSAHeader, [292](#)
- GetMrProtocolByName
  - gdcm::MrProtocol, [736](#)
- GetMSString
  - gdcm::MediaStorage, [696](#)
- GetMSType
  - gdcm::MediaStorage, [696](#)

- GetMTime
  - vtkImageMapToColors16, 1380
- GetName
  - gdcm::CSAElement, 284
  - gdcm::CSAHeaderDictEntry, 297
  - gdcm::DictEntry, 365
  - gdcm::Filename, 482
  - gdcm::GroupDict, 519
  - gdcm::IODEntry, 607
  - gdcm::Macro, 683
  - gdcm::Module, 720
  - gdcm::ModuleEntry, 723
  - gdcm::network::AbstractSyntax, 101
  - gdcm::network::ApplicationContext, 117
  - gdcm::network::TransferSyntaxSub, 1153
  - gdcm::PDBElement, 799
  - gdcm::QueryBase, 894
  - gdcm::QueryImage, 897
  - gdcm::QueryPatient, 900
  - gdcm::QuerySeries, 902
  - gdcm::QueryStudy, 905
  - gdcm::UIDs, 1196
- GetNeedByteSwap
  - gdcm::Bitmap, 198
  - gdcm::ImageCodec, 561
- GetNegotiatedType
  - gdcm::TransferSyntax, 1150
- GetNestedDataSet
  - gdcm::Item, 620
- GetNextSingleSerieUIDFileSet
  - gdcm::SerieHelper, 993
- GetNoOfItems
  - gdcm::CSAElement, 284
- GetNumberOfComponents
  - gdcm::PersonName, 810
- GetNumberOfContourReferencedFrameOfReferences
  - vtkRTStructSetProperties, 1400
- GetNumberOfCurves
  - gdcm::Curve, 306
  - gdcm::Pixmap, 832
- GetNumberOfDimensions
  - gdcm::Bitmap, 198
  - gdcm::ImageCodec, 561
- GetNumberOfElementsFromArray
  - gdcm::VM, 1292
- GetNumberOfFileNames
  - gdcm::Testing, 1139
- GetNumberOfFilenames
  - gdcm::FilenameGenerator, 489
- GetNumberOfFragments
  - gdcm::SequenceOfFragments, 979
- GetNumberOfIconImages
  - gdcm::IconImageFilter, 522
- GetNumberOfImagesInMosaic
  - gdcm::SplitMosaicFilter, 1034
- GetNumberOfIODs
  - gdcm::IOD, 605
- GetNumberOfItems
  - gdcm::SequenceOfItems, 987
- GetNumberOfMD5DataImages
  - gdcm::Testing, 1139
- GetNumberOfMD5MetaImages
  - vtkGDCMTesting, 1353
- GetNumberOfMediaStorageDataFiles
  - gdcm::Testing, 1139
- GetNumberOfModality
  - gdcm::MediaStorage, 696
- GetNumberOfModuleEntries
  - gdcm::NestedModuleEntries, 748
- GetNumberOfMSString
  - gdcm::MediaStorage, 696
- GetNumberOfMSType
  - gdcm::MediaStorage, 696
- GetNumberOfOverlays
  - gdcm::Pixmap, 833
- GetNumberOfPoints
  - gdcm::Curve, 306
- GetNumberOfPresentationContext
  - gdcm::network::AAssociateRQPDU, 97
- GetNumberOfPresentationContextAC
  - gdcm::network::AAssociateACPDU, 90
- GetNumberOfPresentationDataValues
  - gdcm::network::PDataTFPDU, 796
- GetNumberOfPrimitivesData
  - gdcm::MeshPrimitive, 708
- GetNumberOfReferencedFrameOfReferences
  - vtkRTStructSetProperties, 1400
- GetNumberOfSegments
  - gdcm::SegmentWriter, 973
- GetNumberOfSOPClassToIOD
  - gdcm::SOPClassUIDToIOD, 1023
- GetNumberOfStructureSetROIs
  - vtkRTStructSetProperties, 1400
- GetNumberOfSurfacePoints
  - gdcm::Surface, 1086
- GetNumberOfSurfaces
  - gdcm::SurfaceReader, 1099
  - gdcm::SurfaceWriter, 1103
- GetNumberOfTransferSyntaxes
  - gdcm::network::PresentationContextRQ, 866
  - gdcm::PresentationContext, 857
- GetNumberOfTransferSyntaxStrings
  - gdcm::UIDs, 1196
- GetNumberOfValues
  - gdcm::Attribute< Group, Element, TVR, TVM >, 133
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 140

- gdcM::Attribute< Group, Element, TVR, VM::VM1\_n  
>, 150
- GetNumberOfVectors
  - gdcM::Surface, 1086
- GetObliquityThresholdCosineValue
  - gdcM::Orientation, 778
- GetOffScreenRendering
  - vtkImageColorViewer, 1370
- GetOptionalTags
  - gdcM::QueryBase, 894
  - gdcM::QueryImage, 898
  - gdcM::QueryPatient, 900
  - gdcM::QuerySeries, 903
  - gdcM::QueryStudy, 905
- GetOrderedValues
  - gdcM::Scanner, 941
  - gdcM::StrictScanner, 1054
- GetOrigin
  - gdcM::Image, 533
  - gdcM::Overlay, 784
- GetOriginValue
  - gdcM::ImageHelper, 573
- GetOutput
  - gdcM::ImageConverter, 566
- GetOutput
  - gdcM::BitmapToBitmapFilter, 207
  - gdcM::ImageToImageFilter, 589
  - gdcM::PixmapToPixmapFilter, 841
- GetOutputAsBitmap
  - gdcM::BitmapToBitmapFilter, 207
- GetOutputAsPixmap
  - gdcM::PixmapToPixmapFilter, 841
- GetOutputFilename
  - gdcM::DictConverter, 362
- GetOutputType
  - gdcM::DictConverter, 362
- GetOverlay
  - gdcM::Pixmap, 833
  - vtkGDCMImageReader, 1311
  - vtkGDCMImageReader2, 1323
- GetOverlayData
  - gdcM::Overlay, 784
- GetOverlayPort
  - vtkGDCMImageReader2, 1323
- GetOverlayTypeAsString
  - gdcM::Overlay, 784
- GetOverlayTypeFromString
  - gdcM::Overlay, 784
- GetOverlayVisibility
  - vtkImageColorViewer, 1370
- GetOwner
  - gdcM::PrivateTag, 881
- GetPath
  - gdcM::Filename, 482
- GetPattern
  - gdcM::FilenameGenerator, 490
- GetPDBEEnd
  - gdcM::PDBHeader, 802
- GetPDBElementByName
  - gdcM::PDBHeader, 802
- GetPDBInfoTag
  - gdcM::PDBHeader, 802
- GetPDUs
  - gdcM::network::ULEvent, 1257
- GetPDVs
  - gdcM::network::PDUFactory, 809
- GetPermissions
  - gdcM::System, 1112
- GetPhotometricInterpretation
  - gdcM::Bitmap, 198
  - gdcM::ImageChangePhotometricInterpretation, 542
  - gdcM::ImageCodec, 561
- GetPhotometricInterpretationValue
  - gdcM::ImageHelper, 573
- GetPIString
  - gdcM::PhotometricInterpretation, 818
- GetPIType
  - gdcM::PhotometricInterpretation, 818
- GetPixelFormat
  - gdcM::Bitmap, 199
  - gdcM::ImageCodec, 561
- GetPixelFormatValue
  - gdcM::ImageHelper, 573
- GetPixelRepresentation
  - gdcM::PixelFormat, 823
- GetPixelSize
  - gdcM::PixelFormat, 824
- GetPixelSpacingDataRoot
  - gdcM::Testing, 1139
- GetPixmap
  - gdcM::FileDecompressLookupTable, 464
  - gdcM::IconImageGenerator, 525
  - gdcM::PixmapReader, 838
  - gdcM::PixmapWriter, 845
- GetPlanarConfiguration
  - gdcM::Bitmap, 199
  - gdcM::ImageChangePlanarConfiguration, 547
  - gdcM::ImageCodec, 561
- GetPlanarConfigurationValue
  - gdcM::ImageHelper, 574
- GetPMSRescaleInterceptSlope
  - gdcM::ImageHelper, 574
- GetPMTFInformationDataTag
  - gdcM::MEC\_MR3, 688
- GetPointCoordinatesData
  - gdcM::Surface, 1086
- GetPointer
  - gdcM::ByteValue, 222

- gdcm::LookupTable, [676](#)
- gdcm::SmartPointer< ObjectType >, [1019](#)
- vtkLookupTable16, [1395](#)
- GetPointerFromElement
  - gdcm::ImageHelper, [574](#)
- GetPointPositionAccuracy
  - gdcm::Surface, [1086](#)
- GetPointsBoundingBoxCoordinates
  - gdcm::Surface, [1087](#)
- GetPosition
  - vtkImageColorViewer, [1370](#)
- GetPreamble
  - gdcm::FileMetaInformation, [477](#)
- GetPrefix
  - gdcm::FilenameGenerator, [490](#)
- GetPresentationContext
  - gdcm::network::AAssociateRQPDU, [97](#)
- GetPresentationContextAC
  - gdcm::network::AAssociateACPDU, [90](#)
- GetPresentationContextACByID
  - gdcm::network::ULConnection, [1242](#)
- GetPresentationContextByAbstractSyntax
  - gdcm::network::AAssociateRQPDU, [97](#)
- GetPresentationContextByID
  - gdcm::network::AAssociateRQPDU, [97](#)
- GetPresentationContextID
  - gdcm::network::PresentationContextAC, [859](#)
  - gdcm::network::PresentationContextRQ, [866](#)
  - gdcm::network::PresentationDataValue, [869](#)
  - gdcm::PresentationContext, [857](#)
- GetPresentationContextIDFromPresentationContext
  - gdcm::network::ULConnection, [1243](#)
- GetPresentationContextRQByID
  - gdcm::network::ULConnection, [1243](#)
- GetPresentationContexts
  - gdcm::network::AAssociateRQPDU, [97](#)
  - gdcm::network::ULConnection, [1243](#)
  - gdcm::PresentationContextGenerator, [863](#)
- GetPresentationDataValue
  - gdcm::network::PDataTFPDU, [796](#)
- GetPrettyPrint
  - gdcm::JSON, [663](#)
- GetPrimitiveData
  - gdcm::MeshPrimitive, [708](#), [709](#)
- GetPrimitivesData
  - gdcm::MeshPrimitive, [709](#)
- GetPrimitiveType
  - gdcm::MeshPrimitive, [709](#)
- GetPrintStyle
  - gdcm::Printer, [873](#)
  - gdcm::XMLPrinter, [1421](#)
- GetPrivateCreator
  - gdcm::DataSet, [332](#)
  - gdcm::Tag, [1127](#)
- GetPrivateDict
  - gdcm::Dicts, [374](#)
  - gdcm::XMLPrivateDictReader, [1425](#)
- GetPrivateMapping
  - gdcm::Scanner2, [950](#)
  - gdcm::StrictScanner2, [1064](#)
- GetPrivateMappings
  - gdcm::Scanner2, [950](#)
  - gdcm::StrictScanner2, [1064](#)
- GetPrivateOrderedValues
  - gdcm::Scanner2, [950](#)
  - gdcm::StrictScanner2, [1064](#)
- GetPrivateTag
  - gdcm::DataSet, [332](#)
- GetPrivateValue
  - gdcm::Scanner2, [950](#)
  - gdcm::StrictScanner2, [1064](#)
- GetPrivateValues
  - gdcm::Scanner2, [951](#)
  - gdcm::StrictScanner2, [1064](#)
- GetProcessingAlgorithm
  - gdcm::Surface, [1087](#)
- GetProgress
  - gdcm::ProgressEvent, [885](#)
- GetPropertyCategory
  - gdcm::Segment, [958](#)
- GetPropertyType
  - gdcm::Segment, [958](#)
- GetPropertyTypeModifiers
  - gdcm::Segment, [958](#)
- GetProtocol
  - gdcm::network::ULConnection, [1243](#)
- GetPublicDict
  - gdcm::Dicts, [374](#)
- GetPublicMapping
  - gdcm::Scanner2, [951](#)
  - gdcm::StrictScanner2, [1064](#)
- GetPublicMappings
  - gdcm::Scanner2, [951](#)
  - gdcm::StrictScanner2, [1064](#)
- GetPublicOrderedValues
  - gdcm::Scanner2, [951](#)
  - gdcm::StrictScanner2, [1065](#)
- GetPublicValue
  - gdcm::Scanner2, [951](#)
  - gdcm::StrictScanner2, [1065](#)
- GetPublicValues
  - gdcm::Scanner2, [951](#)
  - gdcm::StrictScanner2, [1065](#)
- GetQuality
  - gdcm::JPEG2000Codec, [638](#)
  - gdcm::JPEGCodec, [652](#)
- GetQueryDataSet
  - gdcm::BaseQuery, [177](#), [178](#)



- GetQueryLevel
  - gdcm::QueryBase, [894](#)
  - gdcm::QueryImage, [898](#)
  - gdcm::QueryPatient, [900](#)
  - gdcm::QuerySeries, [903](#)
  - gdcm::QueryStudy, [905](#)
- GetQueryLevelFromQueryRoot
  - gdcm::BaseRootQuery, [183](#)
- GetQueryLevelFromString
  - gdcm::BaseRootQuery, [183](#)
- GetQueryLevelString
  - gdcm::BaseRootQuery, [183](#)
- GetRate
  - gdcm::JPEG2000Codec, [638](#)
- GetRAWMD5FromFile
  - vtkGDCMTesting, [1353](#)
- GetRealWorldValueMappingContent
  - gdcm::ImageHelper, [574](#)
- GetReason
  - gdcm::network::PresentationContextAC, [859](#)
- GetRecommendedDisplayCIELabValue
  - gdcm::Surface, [1087](#)
- GetRecommendedDisplayGrayscaleValue
  - gdcm::Surface, [1087](#)
- GetRecommendedPresentationOpacity
  - gdcm::Surface, [1087](#)
- GetRecommendedPresentationType
  - gdcm::Surface, [1087](#)
- GetRef
  - gdcm::IODEntry, [607](#)
- GetReferencedFrameOfReferenceClassUID
  - vtkRTStructSetProperties, [1400](#)
- GetReferencedFrameOfReferenceInstanceUID
  - vtkRTStructSetProperties, [1400](#)
- GetRegion
  - gdcm::ImageRegionReader, [586](#)
- GetRequiredDataSet
  - gdcm::ModalityPerformedProcedureStepCreateQuery, [713](#)
  - gdcm::ModalityPerformedProcedureStepSetQuery, [716](#)
- GetRequiredTags
  - gdcm::QueryBase, [894](#)
  - gdcm::QueryImage, [898](#)
  - gdcm::QueryPatient, [900](#)
  - gdcm::QuerySeries, [903](#)
  - gdcm::QueryStudy, [905](#)
- GetRescaleInterceptSlopeValue
  - gdcm::ImageHelper, [574](#)
- GetReserved43\_74
  - gdcm::network::AAssociateRQPDU, [97](#)
- GetResponses
  - gdcm::network::ULBasicCallback, [1239](#)
- GetRetired
  - gdcm::DictEntry, [366](#)
- GetRoot
  - gdcm::UIDGenerator, [1160](#)
- GetRows
  - gdcm::Bitmap, [199](#)
  - gdcm::Overlay, [784](#)
- GetRTStructSeriesUIDs
  - gdcm::DirectoryHelper, [384](#)
- GetSamplesPerPixel
  - gdcm::PhotometricInterpretation, [818](#)
  - gdcm::PixelFormat, [824](#)
- GetScalarType
  - gdcm::PixelFormat, [824](#)
- GetScalarTypeAsString
  - gdcm::PixelFormat, [824](#)
- GetScanner
  - gdcm::DICOMDIRGenerator, [355](#)
- GetSecondaryCaptureImagePlaneModule
  - gdcm::ImageHelper, [574](#)
- GetSegment
  - gdcm::SegmentWriter, [973](#)
- GetSegmentAlgorithmName
  - gdcm::Segment, [958](#)
- GetSegmentAlgorithmType
  - gdcm::Segment, [958](#)
- GetSegmentDescription
  - gdcm::Segment, [959](#)
- GetSegmentLabel
  - gdcm::Segment, [959](#)
- GetSegmentNumber
  - gdcm::Segment, [959](#)
- GetSegments
  - gdcm::SegmentReader, [969](#)
  - gdcm::SegmentWriter, [973](#)
- GetSelectedPrivateGroupOffsetFromFile
  - gdcm::Testing, [1139](#)
- GetSelectedTagsOffsetFromFile
  - gdcm::Testing, [1139](#)
- GetSequenceOfFragments
  - gdcm::DataElement, [312](#), [313](#)
- GetSeriesUIDsBySOPClassUID
  - gdcm::DirectoryHelper, [384](#)
- GetSize
  - gdcm::VR, [1298](#)
  - vtkImageColorViewer, [1370](#)
- GetSiz eof
  - gdcm::VR, [1298](#)
- GetSliceArray
  - gdcm::MrProtocol, [736](#)
- GetSliceMax
  - vtkImageColorViewer, [1370](#)
- GetSliceMin
  - vtkImageColorViewer, [1371](#)
- GetSliceRange

- vtkImageColorViewer, [1371](#)
- GetSlope
  - gdcm::Image, [533](#)
  - gdcm::Rescaler, [924](#)
- GetSOPClassUID
  - gdcm::DirectoryHelper, [384](#)
- GetSOPClassUIDFromIOD
  - gdcm::SOPClassUIDToIOD, [1023](#)
- GetSOPClassUIDToIOD
  - gdcm::SOPClassUIDToIOD, [1023](#)
- GetSOPClassUIDToIODs
  - gdcm::SOPClassUIDToIOD, [1024](#)
- GetSOPInstanceUID
  - gdcm::BaseQuery, [178](#)
- GetSourceApplicationEntityTitle
  - gdcm::FileMetaInformation, [477](#)
- GetSourceDirectory
  - gdcm::Testing, [1140](#)
- GetSpacing
  - gdcm::Image, [533](#)
- GetSpacingTagFromMediaStorage
  - gdcm::ImageHelper, [575](#)
- GetSpacingValue
  - gdcm::ImageHelper, [575](#)
- GetStart
  - gdcm::ByteBuffer, [214](#)
- GetState
  - gdcm::network::ULConnection, [1243](#)
- GetStateIndex
  - gdcm::network, [80](#)
- GetSTATES
  - gdcm::Surface, [1088](#)
- GetSTATESString
  - gdcm::Surface, [1088](#)
- GetStream
  - gdcm::Trace, [1144](#)
- GetStreamCurrentPosition
  - gdcm::Reader, [914](#)
- GetStreamOffsetFromFile
  - gdcm::Testing, [1140](#)
- GetStreamPtr
  - gdcm::Reader, [915](#)
  - gdcm::Writer, [1414](#)
- GetString
  - gdcm::MediaStorage, [696](#)
  - gdcm::PhotometricInterpretation, [818](#)
  - gdcm::TransferSyntax, [1150](#)
  - gdcm::UIDs, [1196](#)
- GetStringValueFromTag
  - gdcm::DirectoryHelper, [385](#)
- GetStructureSetObservationNumber
  - vtkRTStructSetProperties, [1401](#)
- GetStructureSetROIDescription
  - vtkRTStructSetProperties, [1401](#)
- GetStructureSetROIGenerationAlgorithm
  - vtkRTStructSetProperties, [1401](#)
- GetStructureSetROIName
  - vtkRTStructSetProperties, [1401](#)
- GetStructureSetROINumber
  - vtkRTStructSetProperties, [1401](#)
- GetStructureSetROIObservationLabel
  - vtkRTStructSetProperties, [1401](#)
- GetStructureSetROIRefFrameRefUID
  - vtkRTStructSetProperties, [1401](#)
- GetStructureSetRTROIInterpretedType
  - vtkRTStructSetProperties, [1401](#)
- GetSurface
  - gdcm::Segment, [959](#)
- GetSurfaceComments
  - gdcm::Surface, [1088](#)
- GetSurfaceCount
  - gdcm::Segment, [959](#)
- GetSurfaceNumber
  - gdcm::Surface, [1088](#)
- GetSurfaceProcessing
  - gdcm::Surface, [1088](#)
- GetSurfaceProcessingDescription
  - gdcm::Surface, [1088](#)
- GetSurfaceProcessingRatio
  - gdcm::Surface, [1088](#)
- GetSurfaces
  - gdcm::Segment, [959](#)
- GetSwapCode
  - gdcm::TransferSyntax, [1150](#)
- GetSwapCodeString
  - gdcm::SwapCode, [1105](#)
- GetSyngoDT
  - gdcm::CSAElement, [284](#)
- GetTable
  - gdcm::SequenceOfFragments, [979](#)
- GetTableEntry
  - gdcm::Table, [1116](#)
- GetTag
  - gdcm::AnonymizeEvent, [105](#)
  - gdcm::Attribute< Group, Element, TVR, TVM >, [133](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [140](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [150](#)
  - gdcm::DataElement, [313](#)
- GetTagListByLevel
  - gdcm::BaseRootQuery, [183](#)
  - gdcm::FindPatientRootQuery, [504](#)
  - gdcm::FindStudyRootQuery, [508](#)
  - gdcm::MovePatientRootQuery, [729](#)
  - gdcm::MoveStudyRootQuery, [733](#)
  - gdcm::WLMFindQuery, [1409](#)
- GetTempDirectory



- gdcmm::Testing, 1140
- GetTempDirectoryW
  - gdcmm::Testing, 1140
- GetTempFilename
  - gdcmm::Testing, 1140
- GetTempFilenameW
  - gdcmm::Testing, 1141
- GetTimeout
  - gdcmm::network::ARTIMTimer, 126
  - gdcmm::ServiceClassUser, 1001
- GetTimer
  - gdcmm::network::ULConnection, 1243
- GetTimezoneOffsetFromUTC
  - gdcmm::System, 1112
- GetToplevel
  - gdcmm::Directory, 382
- GetToshibaMECMR3Tag
  - gdcmm::MEC\_MR3, 689
- GetTransferSyntax
  - gdcmm::Bitmap, 199
  - gdcmm::ImageChangeTransferSyntax, 551
  - gdcmm::network::PresentationContextAC, 860
  - gdcmm::network::PresentationContextRQ, 866
  - gdcmm::PresentationContext, 857
- GetTransferSyntaxes
  - gdcmm::network::PresentationContextRQ, 866
- GetTransferSyntaxString
  - gdcmm::UIDs, 1196
- GetTransferSyntaxStrings
  - gdcmm::UIDs, 1196
- GetTSString
  - gdcmm::TransferSyntax, 1150
- GetTSType
  - gdcmm::TransferSyntax, 1150
- GetType
  - gdcmm::ModuleEntry, 724
  - gdcmm::Orientation, 778
  - gdcmm::Overlay, 784
  - gdcmm::PhotometricInterpretation, 819
- GetTypeAsEnum
  - gdcmm::Overlay, 784
- GetTypeFromTag
  - gdcmm::Defs, 346
  - gdcmm::IOD, 605
- GetTypeOfData
  - gdcmm::Curve, 306
- GetTypeOfDataDescription
  - gdcmm::Curve, 306
- GetTypeString
  - gdcmm::Type, 1157
- GetTypeType
  - gdcmm::Type, 1157
- GetUIDName
  - gdcmm::UIDs, 1196
- GetUIDString
  - gdcmm::UIDs, 1196
- GetUniqueTags
  - gdcmm::QueryBase, 894
  - gdcmm::QueryImage, 898
  - gdcmm::QueryPatient, 901
  - gdcmm::QuerySeries, 903
  - gdcmm::QueryStudy, 906
- GetUnpackBuffer
  - gdcmm::Overlay, 785
- GetUnpackBufferLength
  - gdcmm::Overlay, 785
- GetUsage
  - gdcmm::IODEntry, 607
- GetUsageString
  - gdcmm::Usage, 1272
- GetUsageType
  - gdcmm::IODEntry, 607
  - gdcmm::Usage, 1272
- GetUserData
  - gdcmm::Parser, 793
- GetUserInformation
  - gdcmm::network::AAssociateACPDU, 90
  - gdcmm::network::AAssociateRQPDU, 97
- GetValidatedFile
  - gdcmm::Validate, 1278
- GetValidDataSet
  - gdcmm::WLMFindQuery, 1410
- GetValue
  - gdcmm::Attribute< Group, Element, TVR, TVM >, 133, 134
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 141
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, 151
  - gdcmm::CSAElement, 284
  - gdcmm::DataElement, 313
  - gdcmm::Element< TVR, TVM >, 394
  - gdcmm::Element< TVR, VM::VM1\_n >, 401
  - gdcmm::PDSElement, 799
  - gdcmm::Scanner, 941
  - gdcmm::StrictScanner, 1054
- GetValueAsSQ
  - gdcmm::DataElement, 314
- GetValues
  - gdcmm::Attribute< Group, Element, TVR, TVM >, 134
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 141
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, 151
  - gdcmm::Element< TVR, TVM >, 394
  - gdcmm::Scanner, 941
  - gdcmm::Scanner2, 952
  - gdcmm::StrictScanner, 1054, 1055

- gdcmm::StrictScanner2, 1065
- GetVectorAccuracy
  - gdcmm::Surface, 1088
- GetVectorCoordinateData
  - gdcmm::Surface, 1088, 1089
- GetVectorDimensionality
  - gdcmm::Surface, 1089
- GetVersion
  - gdcmm::MrProtocol, 736
  - gdcmm::Version, 1285
- GetVIEWType
  - gdcmm::Surface, 1089
- GetVIEWTypeString
  - gdcmm::Surface, 1089
- GetVL
  - gdcmm::DataElement, 314
- GetVL16Max
  - gdcmm::VL, 1287
- GetVL32Max
  - gdcmm::VL, 1287
- GetVM
  - gdcmm::Attribute< Group, Element, TVR, TVM >, 134
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 141
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_3 >, 146
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_8 >, 148
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, 151
  - gdcmm::Attribute< Group, Element, TVR, VM::VM2\_2n >, 156
  - gdcmm::Attribute< Group, Element, TVR, VM::VM2\_n >, 158
  - gdcmm::Attribute< Group, Element, TVR, VM::VM3\_3n >, 161
  - gdcmm::Attribute< Group, Element, TVR, VM::VM3\_n >, 163
  - gdcmm::CSAElement, 284
  - gdcmm::CSAHeaderDictEntry, 297
  - gdcmm::DictEntry, 366
  - gdcmm::Element< TVR, TVM >, 394
  - gdcmm::Element< TVR, VM::VM1\_n >, 401
- GetVMString
  - gdcmm::VM, 1293
- GetVMType
  - gdcmm::VM, 1293
- GetVMTypeFromLength
  - gdcmm::VM, 1293
- GetVoidPointer
  - gdcmm::ByteValue, 222
- GetVR
  - gdcmm::Attribute< Group, Element, TVR, TVM >, 134
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 141
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, 151
  - gdcmm::CSAElement, 285
  - gdcmm::CSAHeaderDictEntry, 298
  - gdcmm::DataElement, 314
  - gdcmm::DictEntry, 366
  - gdcmm::Element< TVR, TVM >, 394
  - gdcmm::Element< TVR, VM::VM1\_n >, 401
- GetVRFromTag
  - gdcmm, 64
- GetVRString
  - gdcmm::VR, 1298
- GetVRStringFromFile
  - gdcmm::VR, 1298
- GetVRType
  - gdcmm::VR, 1298
- GetVRTypeFromFile
  - gdcmm::VR, 1299
- GetVTKDataRoot
  - vtkGDCMTesting, 1353
- GetWarningFlag
  - gdcmm::Trace, 1144
- GetWarningStream
  - gdcmm::Trace, 1144
- GetWindowName
  - vtkImageColorViewer, 1371
- GetXMax
  - gdcmm::BoxRegion, 211
- GetXMin
  - gdcmm::BoxRegion, 211
- GetYMax
  - gdcmm::BoxRegion, 211
- GetYMin
  - gdcmm::BoxRegion, 211
- GetZMax
  - gdcmm::BoxRegion, 212
- GetZMin
  - gdcmm::BoxRegion, 212
- GetZSpacing
  - gdcmm::IPPSorter, 613
- GetZSpacingTagFromMediaStorage
  - gdcmm::ImageHelper, 575
- GetZSpacingTolerance
  - gdcmm::IPPSorter, 614
- Global
  - gdcmm::Defs, 347
  - gdcmm::Dicts, 375
  - gdcmm::Global, 515
- GlobalInstance
  - gdcmm, 74
- GrabOverlayFromPixelData
  - gdcmm::Overlay, 785

- Graphics
  - gdcm::Overlay, [782](#)
- GRAY
  - gdcm::LookupTable, [674](#)
- GrayscalePlanarMPRVolumetricPresentationStateStorage
  - gdcm::UIDs, [1184](#)
- GrayscaleSoftcopyPresentationStateStorageSOPClass
  - gdcm::MediaStorage, [693](#)
  - gdcm::UIDs, [1180](#)
- GREEN
  - gdcm::LookupTable, [674](#)
- green
  - gdcm::terminal, [82](#)
- GroupDict
  - gdcm::GroupDict, [519](#)
- GroupStringVector
  - gdcm::GroupDict, [518](#)
- GuessFromModality
  - gdcm::MediaStorage, [697](#)
- HandleBulkData
  - gdcm::XMLPrinter, [1421](#)
- HandleDataSet
  - gdcm::network::ULBasicCallback, [1240](#)
  - gdcm::network::ULConnectionCallback, [1246](#)
  - gdcm::network::ULWritingCallback, [1261](#)
- HandleDescription
  - gdcm::XMLDictReader, [1419](#)
  - gdcm::XMLPrivateDictReader, [1425](#)
- HandleEntry
  - gdcm::XMLDictReader, [1419](#)
  - gdcm::XMLPrivateDictReader, [1425](#)
- HandleEvent
  - gdcm::network::ULTransitionTable, [1259](#)
- HandleIOD
  - gdcm::TableReader, [1120](#)
- HandleIODEntry
  - gdcm::TableReader, [1120](#)
- HandleMacro
  - gdcm::TableReader, [1120](#)
- HandleMacroEntry
  - gdcm::TableReader, [1120](#)
- HandleMacroEntryDescription
  - gdcm::TableReader, [1120](#)
- HandleModule
  - gdcm::TableReader, [1120](#)
- HandleModuleEntry
  - gdcm::TableReader, [1121](#)
- HandleModuleEntryDescription
  - gdcm::TableReader, [1121](#)
- HandleModuleInclude
  - gdcm::TableReader, [1121](#)
- HandleResponse
  - gdcm::network::ULBasicCallback, [1240](#)
  - gdcm::network::ULConnectionCallback, [1246](#)
  - gdcm::network::ULWritingCallback, [1261](#)
- HangingProtocolInformationModelFIND
  - gdcm::UIDs, [1182](#)
- HangingProtocolInformationModelGET
  - gdcm::UIDs, [1186](#)
- HangingProtocolInformationModelMOVE
  - gdcm::UIDs, [1182](#)
- HangingProtocolStorage
  - gdcm::MediaStorage, [694](#)
  - gdcm::UIDs, [1182](#)
- HardcopyColorImageStorage
  - gdcm::MediaStorage, [695](#)
- HardcopyColorImageStorageSOPClassRetired
  - gdcm::UIDs, [1179](#)
- HardcopyGrayscaleImageStorage
  - gdcm::MediaStorage, [694](#)
- HardcopyGrayscaleImageStorageSOPClassRetired
  - gdcm::UIDs, [1179](#)
- HasObserver
  - gdcm::Subject, [1079](#)
- HemodynamicWaveformStorage
  - gdcm::MediaStorage, [693](#)
  - gdcm::UIDs, [1180](#)
- HEVCH\_265Main10ProfileLevel5\_1
  - gdcm::UIDs, [1184](#)
- HEVCH\_265MainProfileLevel5\_1
  - gdcm::UIDs, [1184](#)
- hidden
  - gdcm::terminal, [82](#)
- HITACHI
  - gdcm::EquipmentManufacturer, [431](#)
- HotIronColorPaletteSOPInstance
  - gdcm::UIDs, [1184](#)
- HotMetalBlueColorPaletteSOPInstance
  - gdcm::UIDs, [1183](#)
- HSV
  - gdcm::PhotometricInterpretation, [817](#)
- ICBM452T1FrameofReference
  - gdcm::UIDs, [1178](#)
- ICBMSingleSubjectMRIFrameofReference
  - gdcm::UIDs, [1178](#)
- ICD11
  - gdcm::UIDs, [1183](#)
- Icon
  - gdcm::Pixmap, [834](#)
- IconDataScalarType
  - vtkGDCMImageReader, [1318](#)
  - vtkGDCMImageReader2, [1330](#)
- IconImage
  - gdcm, [59](#)
- IconImageDataExtent
  - vtkGDCMImageReader, [1318](#)

- vtkGDCMImageReader2, [1330](#)
- IconImageFilter
  - gdcm::IconImageFilter, [521](#)
- IconImageGenerator
  - gdcm::IconImageGenerator, [524](#)
- IconNumberOfScalarComponents
  - vtkGDCMImageReader, [1318](#)
  - vtkGDCMImageReader2, [1330](#)
- ID
  - gdcm::PresentationContext, [858](#)
- ignore\_char
  - gdcm::ignore\_char, [527](#)
- Image
  - gdcm::Image, [532](#)
- ImageActor
  - vtkImageColorViewer, [1377](#)
- ImageApplyLookupTable
  - gdcm::ImageApplyLookupTable, [538](#)
- ImageBiomarkerStandardisationInitiative
  - gdcm::UIDs, [1184](#)
- ImageChangePhotometricInterpretation
  - gdcm::ImageChangePhotometricInterpretation, [542](#)
  - gdcm::ImageCodec, [564](#)
- ImageChangePlanarConfiguration
  - gdcm::ImageChangePlanarConfiguration, [547](#)
- ImageChangeTransferSyntax
  - gdcm::Bitmap, [204](#)
  - gdcm::ImageChangeTransferSyntax, [551](#)
- ImageCodec
  - gdcm::ImageCodec, [557](#)
- ImageConverter
  - gdcm::ImageConverter, [566](#)
- ImageFormat
  - vtkGDCMImageReader, [1318](#)
  - vtkGDCMImageReader2, [1330](#)
- ImageFragmentSplitter
  - gdcm::ImageFragmentSplitter, [569](#)
- ImageNumberOrdering
  - gdcm::SerieHelper, [993](#)
- ImageOrientationPatient
  - vtkGDCMImageReader, [1318](#)
  - vtkGDCMImageReader2, [1330](#)
- ImageOverlayBoxSOPClassRetired
  - gdcm::UIDs, [1179](#)
- ImagePositionPatient
  - vtkGDCMImageReader, [1318](#)
  - vtkGDCMImageReader2, [1330](#)
- ImagePositionPatientOrdering
  - gdcm::SerieHelper, [994](#)
- ImageReader
  - gdcm::ImageReader, [580](#)
- ImageRegionReader
  - gdcm::ImageRegionReader, [585](#)
  - gdcm::JPEG2000Codec, [640](#)
- gdcm::JPEGCodec, [654](#)
  - gdcm::JPEGLSCodec, [662](#)
  - gdcm::RLECodec, [933](#)
- ImageToImageFilter
  - gdcm::ImageToImageFilter, [589](#)
- ImageWriter
  - gdcm::ImageWriter, [593](#)
- ImplantAssemblyTemplateInformationModelFIND
  - gdcm::UIDs, [1186](#)
- ImplantAssemblyTemplateInformationModelGET
  - gdcm::UIDs, [1186](#)
- ImplantAssemblyTemplateInformationModelMOVE
  - gdcm::UIDs, [1186](#)
- ImplantAssemblyTemplateStorage
  - gdcm::UIDs, [1186](#)
- ImplantationPlanSRStorage
  - gdcm::UIDs, [1185](#)
- ImplantTemplateGroupInformationModelFIND
  - gdcm::UIDs, [1186](#)
- ImplantTemplateGroupInformationModelGET
  - gdcm::UIDs, [1186](#)
- ImplantTemplateGroupInformationModelMOVE
  - gdcm::UIDs, [1186](#)
- ImplantTemplateGroupStorage
  - gdcm::UIDs, [1186](#)
- ImplementationClassUIDSub
  - gdcm::network::ImplementationClassUIDSub, [595](#)
- ImplementationUIDSub
  - gdcm::network::ImplementationUIDSub, [596](#)
- ImplementationVersionNameSub
  - gdcm::network::ImplementationVersionNameSub, [597](#)
- Implicit
  - gdcm::TransferSyntax, [1148](#)
- ImplicitVRBigEndianACRNEMA
  - gdcm::TransferSyntax, [1149](#)
- ImplicitVRBigEndianPrivateGE
  - gdcm::TransferSyntax, [1149](#)
- ImplicitVRLittleEndian
  - gdcm::TransferSyntax, [1149](#)
- ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM
  - gdcm::UIDs, [1177](#)
- IncompleteLUT
  - gdcm::LookupTable, [679](#)
- InitFromRQ
  - gdcm::network::AAssociateACPDU, [90](#)
- Initialize
  - gdcm::network::ULConnectionInfo, [1248](#)
- InitializeBlueLUT
  - gdcm::LookupTable, [677](#)
- InitializeConnection
  - gdcm::network::ULConnection, [1243](#)
  - gdcm::ServiceClassUser, [1001](#)
- Initialized

- gdcmm::LookupTable, 677
- InitializeDataSet
  - gdcmm::BaseRootQuery, 183
  - gdcmm::FindPatientRootQuery, 505
  - gdcmm::FindStudyRootQuery, 509
  - gdcmm::MovePatientRootQuery, 730
  - gdcmm::MoveStudyRootQuery, 734
  - gdcmm::WLMFindQuery, 1410
- InitializeGreenLUT
  - gdcmm::LookupTable, 677
- InitializeIncomingConnection
  - gdcmm::network::ULConnection, 1243
- InitializeLUT
  - gdcmm::LookupTable, 677
- InitializeRedLUT
  - gdcmm::LookupTable, 677
- InitializeRTStructSet
  - vtkGDCMPolyDataWriter, 1349
- InitOpenSSL
  - gdcmm::OpenSSLCryptoFactory, 767
- Input
  - gdcmm::BitmapToBitmapFilter, 208
- Insert
  - gdcmm::CommandDataSet, 263
  - gdcmm::DataSet, 332
  - gdcmm::FileMetaInformation, 478
  - gdcmm::GroupDict, 519
- InsertDataElement
  - gdcmm::DataSet, 332
  - gdcmm::Item, 620
- InsertEntry
  - gdcmm::Table, 1116
- InstallPipeline
  - vtkImageColorViewer, 1371
- InstanceAvailabilityNotificationSOPClass
  - gdcmm::UIDs, 1182
- INT12
  - gdcmm::PixelFormat, 822
- INT16
  - gdcmm::PixelFormat, 822
- INT32
  - gdcmm::PixelFormat, 822
- INT64
  - gdcmm::PixelFormat, 822
- INT8
  - gdcmm::PixelFormat, 822
- IntegratedTaxonomicInformationSystemITISTaxonomicSerialNumber
  - gdcmm::UIDs, 1183
- Interactor
  - vtkImageColorViewer, 1377
- InteractorStyle
  - vtkImageColorViewer, 1377
- INTERFILE
  - gdcmm::CSAHeader, 290
- Internal
  - gdcmm::ApplicationEntity, 120
  - gdcmm::Attribute< Group, Element, TVR, TVM >, 137
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 143
  - gdcmm::Element< TVR, TVM >, 396
  - gdcmm::Element< VR::AS, VM::VM5 >, 417
  - gdcmm::LookupTable, 679
  - gdcmm::UI, 1158
- InternalCode
  - gdcmm::Coder, 252
  - gdcmm::JPEG12Codec, 627
  - gdcmm::JPEG16Codec, 632
  - gdcmm::JPEG8Codec, 645
- Internals
  - vtkRTStructSetProperties, 1404
- IntraocularLensCalculationsStorage
  - gdcmm::UIDs, 1185
- IntravascularOpticalCoherenceTomographyImageStorageForPresentation
  - gdcmm::UIDs, 1184
- IntravascularOpticalCoherenceTomographyImageStorageForProcessing
  - gdcmm::UIDs, 1184
- INVALID
  - gdcmm::VR, 1296
- Invalid
  - gdcmm::Overlay, 782
  - gdcmm::Usage, 1271
- InverseRescale
  - gdcmm::Rescaler, 924
- InverseRescaleFunctionIntoBestFit
  - gdcmm::Rescaler, 924
- InvokeEvent
  - gdcmm::Subject, 1079, 1080
- IOD
  - gdcmm::IOD, 604
- IODEntry
  - gdcmm::IODEntry, 607
- IODMapType
  - gdcmm::IODs, 609
- IODMapTypeConstIterator
  - gdcmm::IODs, 609
- IODName
  - gdcmm::IODs, 609
- IODs
  - gdcmm::IODs, 610
- IPPSorter
  - gdcmm::IPPSorter, 613
- IS
  - gdcmm::VR, 1296
- IsAETitleValid
  - gdcmm::network::AAAssociateRQPDU, 97
- IsASCII
  - gdcmm::VR, 1299
- IsASCII2

- gdcm::VR, [1299](#)
- IsBinary
  - gdcm::VR, [1299](#)
- IsBinary2
  - gdcm::VR, [1299](#)
- IsCompatible
  - gdcm::PixelFormat, [825](#)
- IsDual
  - gdcm::VR, [1299](#)
- IsEmpty
  - gdcm::Bitmap, [199](#)
  - gdcm::ByteValue, [222](#)
  - gdcm::CSAElement, [285](#)
  - gdcm::CSAHeaderDict, [295](#)
  - gdcm::Curve, [306](#)
  - gdcm::DataElement, [315](#)
  - gdcm::DataSet, [333](#)
  - gdcm::Defs, [346](#)
  - gdcm::Dict, [359](#)
  - gdcm::Dicts, [374](#)
  - gdcm::Filename, [482](#)
  - gdcm::Macros, [685](#)
  - gdcm::Modules, [726](#)
  - gdcm::Overlay, [785](#)
  - gdcm::Preamble, [853](#)
  - gdcm::PrivateDict, [876](#)
  - gdcm::SegmentHelper::BasicCodedEntry, [187](#)
  - gdcm::SequenceOfItems, [987](#)
- IsEncapsulated
  - gdcm::TransferSyntax, [1150](#)
- IsEncoded
  - gdcm::TransferSyntax, [1151](#)
- IsExplicit
  - gdcm::TransferSyntax, [1151](#)
- IsFrameEncoder
  - gdcm::ImageCodec, [561](#)
  - gdcm::JPEG2000Codec, [638](#)
  - gdcm::JPEGCodec, [652](#)
  - gdcm::JPEGLSCodec, [661](#)
  - gdcm::RLECodec, [932](#)
- IsGroupLength
  - gdcm::Tag, [1127](#)
- IsGroupXX
  - gdcm::Tag, [1127](#)
- IsIdentical
  - gdcm::Filename, [483](#)
- IsIllegal
  - gdcm::Tag, [1127](#)
- IsImage
  - gdcm::MediaStorage, [697](#)
- IsImplicit
  - gdcm::TransferSyntax, [1151](#)
- IsInPixelData
  - gdcm::Overlay, [785](#)
- IsKey
  - gdcm::Scanner, [941](#)
  - gdcm::Scanner2, [952](#)
  - gdcm::StrictScanner, [1055](#)
  - gdcm::StrictScanner2, [1065](#)
- IsLastFragment
  - gdcm::network::AAAbortPDU, [86](#)
  - gdcm::network::AAssociateACPDU, [90](#)
  - gdcm::network::AAssociateRJPDU, [93](#)
  - gdcm::network::AAssociateRQPDU, [98](#)
  - gdcm::network::AReleaseRPPDU, [122](#)
  - gdcm::network::AReleaseRQPDU, [124](#)
  - gdcm::network::BasePDU, [174](#)
  - gdcm::network::PDataTFPDU, [796](#)
- IsLossless
  - gdcm::PhotometricInterpretation, [819](#)
  - gdcm::TransferSyntax, [1151](#)
- IsLossy
  - gdcm::Bitmap, [200](#)
  - gdcm::ImageCodec, [561](#)
  - gdcm::PhotometricInterpretation, [819](#)
  - gdcm::TransferSyntax, [1151](#)
- IsOdd
  - gdcm::VL, [1287](#)
- IsPresentationContextAccepted
  - gdcm::ServiceClassUser, [1001](#)
- IsPrintable
  - gdcm::ByteValue, [223](#)
- IsPrivate
  - gdcm::Tag, [1127](#)
- IsPrivateCreator
  - gdcm::Tag, [1127](#)
- IsPublic
  - gdcm::Tag, [1128](#)
- IsRetired
  - gdcm::PhotometricInterpretation, [819](#)
- IsRGB8
  - gdcm::LookupTable, [677](#)
- IsRowEncoder
  - gdcm::ImageCodec, [562](#)
  - gdcm::JPEG2000Codec, [639](#)
  - gdcm::JPEGCodec, [652](#)
  - gdcm::JPEGLSCodec, [661](#)
  - gdcm::RLECodec, [932](#)
- IsSameColorSpace
  - gdcm::PhotometricInterpretation, [819](#)
- IsStateSuspension
  - gdcm::JPEG12Codec, [627](#)
  - gdcm::JPEG16Codec, [632](#)
  - gdcm::JPEG8Codec, [645](#)
  - gdcm::JPEGCodec, [652](#)
- IsSwap
  - gdcm::VR, [1299](#)
- IsTransferSyntaxCompatible

- gdcM::Bitmap, 200
- IsUndefined
  - gdcM::MediaStorage, 697
  - gdcM::VL, 1287
- IsUndefinedLength
  - gdcM::DataElement, 315
  - gdcM::SequenceOfItems, 987
- IsUnique
  - gdcM::DictEntry, 366
- IsValid
  - gdcM::ApplicationEntity, 119
  - gdcM::BoxRegion, 212
  - gdcM::CodeString, 256
  - gdcM::DirectionCosines, 379
  - gdcM::DPath, 387
  - gdcM::FileMetaInformation, 478
  - gdcM::ImageCodec, 562
  - gdcM::JPEGCodec, 653
  - gdcM::LO, 671
  - gdcM::PixelFormat, 825
  - gdcM::Preamble, 854
  - gdcM::Region, 921
  - gdcM::String< TDelimiter, TMaxLength, TPadChar  
>, 1071
  - gdcM::TagPath, 1134
  - gdcM::TransferSyntax, 1151
  - gdcM::UIDGenerator, 1160
  - gdcM::UUIDGenerator, 1277
  - gdcM::VM, 1293
  - gdcM::VR, 1299, 1300
- IsVRFile
  - gdcM::VR, 1300
- IsZero
  - gdcM::Overlay, 785
- Item
  - gdcM::Item, 619
- Items
  - gdcM::SequenceOfItems, 989
- ItemVector
  - gdcM::SequenceOfItems, 985
- Iterator
  - gdcM::CSAHeaderDict, 294
  - gdcM::DataSet, 328
  - gdcM::Dict, 357
  - gdcM::SequenceOfFragments, 977
  - gdcM::SequenceOfItems, 985
- iterator
  - gdcM::CodeString, 255
  - gdcM::LO, 670
  - gdcM::String< TDelimiter, TMaxLength, TPadChar  
>, 1069
- ItFileSetHt
  - gdcM::SerieHelper, 994
- IVOCTForPresentation
  - gdcM::MediaStorage, 695
- IVOCTForProcessing
  - gdcM::MediaStorage, 695
- Join
  - gdcM::Filename, 483
- JPEG12Codec
  - gdcM::JPEG12Codec, 627
- JPEG16Codec
  - gdcM::JPEG16Codec, 632
- JPEG2000
  - gdcM::TransferSyntax, 1149
- JPEG2000\_COMPRESSION
  - vtkGDCMImageWriter, 1334
- JPEG2000Codec
  - gdcM::JPEG2000Codec, 636
- JPEG2000ImageCompression
  - gdcM::UIDs, 1177
- JPEG2000ImageCompressionLosslessOnly
  - gdcM::UIDs, 1177
- JPEG2000Lossless
  - gdcM::TransferSyntax, 1149
- JPEG2000Part2
  - gdcM::TransferSyntax, 1149
- JPEG2000Part2Lossless
  - gdcM::TransferSyntax, 1149
- JPEG2000Part2MulticomponentImageCompression
  - gdcM::UIDs, 1177
- JPEG2000Part2MulticomponentImageCompressionLosslessOnly
  - gdcM::UIDs, 1177
- JPEG8Codec
  - gdcM::JPEG8Codec, 644
- JPEG\_COMPRESSION
  - vtkGDCMImageWriter, 1334
- JPEGBaselineProcess1
  - gdcM::TransferSyntax, 1149
- JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImageComp
  - gdcM::UIDs, 1177
- JPEGCodec
  - gdcM::JPEGCodec, 649
- JPEGExtendedHierarchicalProcess1618Retired
  - gdcM::UIDs, 1177
- JPEGExtendedHierarchicalProcess1719Retired
  - gdcM::UIDs, 1177
- JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG12BitImageC
  - gdcM::UIDs, 1177
- JPEGExtendedProcess2\_4
  - gdcM::TransferSyntax, 1149
- JPEGExtendedProcess35Retired
  - gdcM::UIDs, 1177
- JPEGExtendedProcess3\_5
  - gdcM::TransferSyntax, 1149
- JPEGFullProgressionHierarchicalProcess2426Retired
  - gdcM::UIDs, 1177



- JPEGFullProgressionHierarchicalProcess2527Retired
  - gdcm::UIDs, [1177](#)
- JPEGFullProgressionNonHierarchicalProcess1012Retired
  - gdcm::UIDs, [1177](#)
- JPEGFullProgressionNonHierarchicalProcess1113Retired
  - gdcm::UIDs, [1177](#)
- JPEGFullProgressionProcess10\_12
  - gdcm::TransferSyntax, [1149](#)
- JPEGLosslessHierarchicalProcess28Retired
  - gdcm::UIDs, [1177](#)
- JPEGLosslessHierarchicalProcess29Retired
  - gdcm::UIDs, [1177](#)
- JPEGLosslessNonHierarchicalFirstOrderPredictionProcess143Retired
  - gdcm::UIDs, [1177](#)
- JPEGLosslessNonHierarchicalProcess14
  - gdcm::UIDs, [1177](#)
- JPEGLosslessNonHierarchicalProcess15Retired
  - gdcm::UIDs, [1177](#)
- JPEGLosslessProcess14
  - gdcm::TransferSyntax, [1149](#)
- JPEGLosslessProcess14\_1
  - gdcm::TransferSyntax, [1149](#)
- JPEGLS\_COMPRESSION
  - vtkGDCMImageWriter, [1334](#)
- JPEGLSCodec
  - gdcm::JPEGLSCodec, [658](#)
- JPEGLSLossless
  - gdcm::TransferSyntax, [1149](#)
- JPEGLSLosslessImageCompression
  - gdcm::UIDs, [1177](#)
- JPEGLSLossyNearLosslessImageCompression
  - gdcm::UIDs, [1177](#)
- JPEGLSNearLossless
  - gdcm::TransferSyntax, [1149](#)
- JPEGSpectralSelectionHierarchicalProcess2022Retired
  - gdcm::UIDs, [1177](#)
- JPEGSpectralSelectionHierarchicalProcess2123Retired
  - gdcm::UIDs, [1177](#)
- JPEGSpectralSelectionNonHierarchicalProcess68Retired
  - gdcm::UIDs, [1177](#)
- JPEGSpectralSelectionNonHierarchicalProcess79Retired
  - gdcm::UIDs, [1177](#)
- JPEGSpectralSelectionProcess6\_8
  - gdcm::TransferSyntax, [1149](#)
- JPIPReferenced
  - gdcm::TransferSyntax, [1149](#)
  - gdcm::UIDs, [1177](#)
- JPIPReferencedDeflate
  - gdcm::UIDs, [1178](#)
- JSON
  - gdcm::JSON, [662](#)
- JunkAfterDocElementError
  - gdcm::Parser, [792](#)
- KAKADUCodec
  - gdcm::KAKADUCodec, [667](#)
- KeratometryMeasurementsStorage
  - gdcm::UIDs, [1184](#)
- KeyField
  - gdcm::CSAElement, [287](#)
- KeyObjectSelectionDocument
  - gdcm::MediaStorage, [694](#)
- KeyObjectSelectionDocumentStorage
  - gdcm::UIDs, [1181](#)
- KeyValuePairArrayType
  - gdcm::CompositeNetworkFunctions, [267](#)
- KeySelectionType
  - gdcm::CompositeNetworkFunctions, [267](#)
- KODAK
  - gdcm::EquipmentManufacturer, [431](#)
- LD\_ALL
  - gdcm, [63](#)
- LD\_NOSEQ
  - gdcm, [63](#)
- LD\_NOSHADOW
  - gdcm, [63](#)
- LD\_NOSHADOWSEQ
  - gdcm, [63](#)
- LeadECGWaveformStorage
  - gdcm::MediaStorage, [693](#)
- LegacyConvertedEnhancedCTImageStorage
  - gdcm::MediaStorage, [695](#)
  - gdcm::UIDs, [1183](#)
- LegacyConvertedEnhancedMRImageStorage
  - gdcm::MediaStorage, [695](#)
  - gdcm::UIDs, [1183](#)
- LegacyConvertedEnhancedPETImageStorage
  - gdcm::MediaStorage, [695](#)
  - gdcm::UIDs, [1183](#)
- LensometryMeasurementsStorage
  - gdcm::UIDs, [1184](#)
- Level
  - vtkImageMapToWindowLevelColors2, [1387](#)
- LINE
  - gdcm::MeshPrimitive, [707](#)
- ListCharSets
  - gdcm::QueryFactory, [895](#)
- LittleEndian
  - gdcm::SwapCode, [1105](#)
- LO
  - gdcm::LO, [670](#), [671](#)
  - gdcm::VR, [1296](#)
- Load
  - gdcm::Directory, [382](#)
  - gdcm::MrProtocol, [736](#)
- LOADBULKDATA
  - gdcm::XMLPrinter, [1421](#)



- LoadDefault
  - gdcm::CSAHeaderDict, [295](#)
  - gdcm::Dict, [359](#)
  - gdcm::PrivateDict, [876](#)
- LoadDefaults
  - gdcm::Defs, [346](#)
  - gdcm::Dicts, [374](#)
- LoadFromDataElement
  - gdcm::CSAHeader, [292](#)
  - gdcm::PDBHeader, [802](#)
- LoadFromFile
  - gdcm::Defs, [346](#)
- LoadIconImage
  - vtkGDCMImageReader, [1318](#)
  - vtkGDCMImageReader2, [1330](#)
- LoadImageFromFiles
  - gdcm::DirectoryHelper, [385](#)
- LoadOverlays
  - vtkGDCMImageReader, [1318](#)
  - vtkGDCMImageReader2, [1330](#)
- LoadResourcesFiles
  - gdcm::Global, [516](#)
- LoadSingleFile
  - vtkGDCMImageReader, [1311](#)
  - vtkGDCMImageReader2, [1323](#)
- Locate
  - gdcm::Global, [517](#)
- LOComp
  - gdcm, [59](#)
- LodModeType
  - gdcm, [63](#)
- LookupTable
  - gdcm::LookupTable, [674](#), [675](#)
  - vtkImageMapToColors16, [1383](#)
- LookupTableType
  - gdcm::LookupTable, [674](#)
- LossyFlag
  - gdcm::Bitmap, [205](#)
  - gdcm::ImageCodec, [564](#)
  - vtkGDCMImageReader, [1319](#)
  - vtkGDCMImageReader2, [1331](#)
- LT
  - gdcm::VR, [1296](#)
- LTCComp
  - gdcm, [59](#)
- LUT
  - gdcm::Bitmap, [205](#)
  - gdcm::ImageCodec, [564](#)
- LUTPtr
  - gdcm::Bitmap, [195](#)
  - gdcm::ImageCodec, [557](#)
- m\_char
  - gdcm::ignore\_char, [527](#)
- m\_ConstMemberFunction
  - gdcm::MemberCommand< T >, [704](#)
- m\_DataSet
  - gdcm::DataSetEvent, [340](#)
- m\_MemberFunction
  - gdcm::MemberCommand< T >, [704](#)
  - gdcm::SimpleMemberCommand< T >, [1012](#)
- m\_This
  - gdcm::MemberCommand< T >, [704](#)
  - gdcm::SimpleMemberCommand< T >, [1012](#)
- Macro
  - gdcm::Macro, [682](#)
- MacroEntry
  - gdcm, [59](#)
- Macros
  - gdcm::Macros, [685](#)
- mAction
  - gdcm::network::Transition, [1155](#)
- MacularGridThicknessandVolumeReportStorage
  - gdcm::UIDs, [1185](#)
- magenta
  - gdcm::terminal, [82](#)
- MAGNIFIED
  - gdcm::Spacing, [1030](#)
- MakeDirectory
  - gdcm::System, [1112](#)
- MakeNew
  - gdcm::network::Transition, [1155](#)
- MakeObject
  - gdcm::AnonymizeEvent, [105](#)
  - gdcm::DataEvent, [324](#)
  - gdcm::DataSetEvent, [339](#)
  - gdcm::Event, [434](#)
  - gdcm::FileNameEvent, [486](#)
  - gdcm::ProgressEvent, [885](#)
- MammographyCADSR
  - gdcm::MediaStorage, [694](#)
- MammographyCADSRStorage
  - gdcm::UIDs, [1181](#)
- Mandatory
  - gdcm::Usage, [1271](#)
- MANUAL
  - gdcm::Segment, [957](#)
- MapCSAHeaderDictEntry
  - gdcm::CSAHeaderDict, [294](#)
- MapDictEntry
  - gdcm::Dict, [357](#)
- MapIODEntry
  - gdcm::IOD, [604](#)
- MapModuleEntry
  - gdcm::Macro, [682](#)
  - gdcm::Module, [719](#)
- MappingType
  - gdcm::Scanner, [938](#)

- gdcM::StrictScanner, 1051
- MapScalarsThroughTable2
  - vtkLookupTable16, 1395
- MapTableEntry
  - gdcM::Table, 1116
- MARCONI
  - gdcM::EquipmentManufacturer, 431
- Match
  - gdcM::DPath, 387
- MaximumLengthSub
  - gdcM::network::MaximumLengthSub, 686
- MaxLength
  - gdcM::ApplicationEntity, 120
  - gdcM::PersonName, 811
- MaxNumberOfComponents
  - gdcM::ApplicationEntity, 120
  - gdcM::PersonName, 811
- MaxPrintLength
  - gdcM::Printer, 874
- MayoClinicNonradiologicalImagesSBSAnatomicalSurfaceRegionGDCM
  - gdcM::UIDs, 1184
- mConnection
  - gdcM::network::ULConnectionManager, 1255
- MD5DataImagesType
  - gdcM::Testing, 1136
- MD5MetalImagesType
  - vtkGDCMTesting, 1352
- mDataSet
  - gdcM::BaseQuery, 179
- MediaCreationManagementSOPClassUID
  - gdcM::UIDs, 1179
- MediaStorage
  - gdcM::MediaStorage, 695
- MediaStorageDataFilesType
  - gdcM::Testing, 1136
- MediaStorageDirectoryStorage
  - gdcM::MediaStorage, 693
  - gdcM::UIDs, 1178
- MedicalImageProperties
  - vtkGDCMImageReader, 1319
  - vtkGDCMPolyDataReader, 1346
  - vtkGDCMPolyDataWriter, 1350
- mElementOffsets
  - gdcM::StreamImageWriter, 1046
- mElementOffsets1
  - gdcM::StreamImageWriter, 1046
- MemberCommand
  - gdcM::MemberCommand< T >, 702
- mEnd
  - gdcM::network::Transition, 1155
- MeshPrimitive
  - gdcM::MeshPrimitive, 708
- MessageID
  - gdcM::network::CEchoRQ, 232
- MetaInformationTS
  - gdcM::FileMetaInformation, 480
- mHelpDescription
  - gdcM::BaseRootQuery, 184
- mImage
  - gdcM::BaseRootQuery, 184
- mImplicit
  - gdcM::network::ULConnectionCallback, 1247
- ModalityPerformedProcedureStepCreateQuery
  - gdcM::ModalityPerformedProcedureStepCreateQuery, 712
- ModalityPerformedProcedureStepNotificationSOPClass
  - gdcM::UIDs, 1179
- ModalityPerformedProcedureStepRetrieveSOPClass
  - gdcM::UIDs, 1178
- ModalityPerformedProcedureStepSetQuery
  - gdcM::ModalityPerformedProcedureStepSetQuery, 716
- ModalityPerformedProcedureStepSOPClass
  - gdcM::MediaStorage, 694
  - gdcM::UIDs, 1178
- ModalityWorklistInformationModelFIND
  - gdcM::UIDs, 1182
- Mode
  - gdcM::terminal, 82
- Module
  - gdcM::Module, 720
- ModuleEntry
  - gdcM::ModuleEntry, 723
- ModuleMapType
  - gdcM::Macros, 684
  - gdcM::Modules, 726
- Modules
  - gdcM::Modules, 726
- MONOCHROME1
  - gdcM::PhotometricInterpretation, 817
- MONOCHROME2
  - gdcM::PhotometricInterpretation, 817
- MouseGenomeInitiativeMGI
  - gdcM::UIDs, 1183
- MovePatientRootQuery
  - gdcM::MovePatientRootQuery, 729
- MoveStudyRootQuery
  - gdcM::MoveStudyRootQuery, 733
- mPatient
  - gdcM::BaseRootQuery, 184
- MPEG2MainProfile
  - gdcM::TransferSyntax, 1149
- MPEG2MainProfileHighLevel
  - gdcM::TransferSyntax, 1149
  - gdcM::UIDs, 1183
- MPEG2MainProfileMainLevel
  - gdcM::UIDs, 1178
- MPEG4AVCH264BDcompatibleHighProfileLevel4\_1

- gdcM::TransferSyntax, [1149](#)
- MPEG4AVCH264HighProfileLevel4\_1
  - gdcM::TransferSyntax, [1149](#)
- MPEG4AVCH\_264BDcompatibleHighProfileLevel4\_1
  - gdcM::UIDs, [1183](#)
- MPEG4AVCH\_264HighProfileLevel4\_1
  - gdcM::UIDs, [1183](#)
- MPEG4AVCH\_264HighProfileLevel4\_2For2DVideo
  - gdcM::UIDs, [1184](#)
- MPEG4AVCH\_264HighProfileLevel4\_2For3DVideo
  - gdcM::UIDs, [1184](#)
- MPEG4AVCH\_264StereoHighProfileLevel4\_2
  - gdcM::UIDs, [1184](#)
- MPTType
  - gdcM::MeshPrimitive, [707](#)
- MPTType\_END
  - gdcM::MeshPrimitive, [707](#)
- MRImageStorage
  - gdcM::MediaStorage, [693](#)
  - gdcM::UIDs, [1180](#)
- mRootType
  - gdcM::BaseRootQuery, [185](#)
- MrProtocol
  - gdcM::MrProtocol, [735](#)
- MRSpectroscopyStorage
  - gdcM::MediaStorage, [693](#)
  - gdcM::UIDs, [1180](#)
- MS\_END
  - gdcM::MediaStorage, [695](#)
- mSecondaryConnection
  - gdcM::network::ULConnectionManager, [1255](#)
- mSeries
  - gdcM::BaseRootQuery, [185](#)
- mSopInstanceUID
  - gdcM::BaseQuery, [179](#)
- mSPFile
  - gdcM::StreamImageWriter, [1046](#)
- mStudy
  - gdcM::BaseRootQuery, [185](#)
- MSType
  - gdcM::MediaStorage, [692](#)
- mTransitions
  - gdcM::network::ULConnectionManager, [1256](#)
- MultiframeGrayscaleByteSecondaryCaptureImageStorage
  - gdcM::MediaStorage, [693](#)
  - gdcM::UIDs, [1180](#)
- MultiframeGrayscaleWordSecondaryCaptureImageStorage
  - gdcM::MediaStorage, [693](#)
  - gdcM::UIDs, [1180](#)
- MultiframeSingleBitSecondaryCaptureImageStorage
  - gdcM::MediaStorage, [693](#)
  - gdcM::UIDs, [1180](#)
- MultiframeTrueColorSecondaryCaptureImageStorage
  - gdcM::MediaStorage, [693](#)
- gdcM::UIDs, [1180](#)
- MultipleVolumeRenderingVolumetricPresentationStateStorage
  - gdcM::UIDs, [1184](#)
- mWriter
  - gdcM::StreamImageWriter, [1047](#)
- mXMax
  - gdcM::StreamImageWriter, [1047](#)
- mXMin
  - gdcM::StreamImageWriter, [1047](#)
- mYMax
  - gdcM::StreamImageWriter, [1047](#)
- mYMin
  - gdcM::StreamImageWriter, [1047](#)
- mZMax
  - gdcM::StreamImageWriter, [1047](#)
- mZMin
  - gdcM::StreamImageWriter, [1047](#)
- N\_ACTION\_RQ
  - gdcM::network::DIMSE, [376](#)
- N\_ACTION\_RSP
  - gdcM::network::DIMSE, [376](#)
- N\_CREATE\_RQ
  - gdcM::network::DIMSE, [376](#)
- N\_CREATE\_RSP
  - gdcM::network::DIMSE, [376](#)
- N\_DELETE\_RQ
  - gdcM::network::DIMSE, [376](#)
- N\_DELETE\_RSP
  - gdcM::network::DIMSE, [376](#)
- N\_EVENT\_REPORT\_RQ
  - gdcM::network::DIMSE, [376](#)
- N\_EVENT\_REPORT\_RSP
  - gdcM::network::DIMSE, [376](#)
- N\_GET\_RQ
  - gdcM::network::DIMSE, [376](#)
- N\_GET\_RSP
  - gdcM::network::DIMSE, [376](#)
- N\_SET\_RQ
  - gdcM::network::DIMSE, [376](#)
- N\_SET\_RSP
  - gdcM::network::DIMSE, [376](#)
- NAction
  - gdcM::NormalizedNetworkFunctions, [758](#)
- Name
  - gdcM::ModuleEntry, [725](#)
- NameField
  - gdcM::CSAElement, [287](#)
  - gdcM::PDBelement, [800](#)
- NativeDICOMModel
  - gdcM::UIDs, [1186](#)
- NCreate
  - gdcM::NormalizedNetworkFunctions, [759](#)
- NDelete

- gdcmm::NormalizedNetworkFunctions, 759
- NeedByteSwap
  - gdcmm::Bitmap, 205
  - gdcmm::ImageCodec, 565
- NeedOverlayCleanup
  - gdcmm::ImageCodec, 565
- NegotiatedType
  - gdcmm::TransferSyntax, 1148
- NestedMacroEntries
  - gdcmm, 59
- NestedModuleEntries
  - gdcmm::NestedModuleEntries, 748
- NEventReport
  - gdcmm::NormalizedNetworkFunctions, 759
- New
  - gdcmm::Anonymizer, 112
  - gdcmm::Cleaner, 243
  - gdcmm::FileChangeTransferSyntax, 460
  - gdcmm::FileStreamer, 497
  - gdcmm::MemberCommand< T >, 703
  - gdcmm::Scanner, 942
  - gdcmm::Scanner2, 952
  - gdcmm::SequenceOfFragments, 980
  - gdcmm::SequenceOfItems, 987
  - gdcmm::ServiceClassUser, 1001
  - gdcmm::SimpleMemberCommand< T >, 1011
  - gdcmm::StrictScanner, 1055
  - gdcmm::StrictScanner2, 1065
  - vtkGDCMImageReader, 1311
  - vtkGDCMImageReader2, 1323
  - vtkGDCMImageWriter, 1335
  - vtkGDCMMedicalImageProperties, 1341
  - vtkGDCMPolyDataReader, 1344
  - vtkGDCMPolyDataWriter, 1349
  - vtkGDCMTesting, 1353
  - vtkGDCMThreadedImageReader, 1358
  - vtkGDCMThreadedImageReader2, 1361
  - vtkImageColorViewer, 1371
  - vtkImageMapToColors16, 1380
  - vtkImageMapToWindowLevelColors2, 1385
  - vtkImagePlanarComponentsToComponents, 1389
  - vtkImageRGBToYBR, 1391
  - vtkImageYBRToRGB, 1393
  - vtkLookupTable16, 1395
  - vtkRTStructSetProperties, 1402
- NewYorkUniversityMelanomaClinicalCooperativeGroup
  - gdcmm::UIDs, 1184
- NGet
  - gdcmm::NormalizedNetworkFunctions, 759
- NO
  - gdcmm::Surface, 1084
- NO\_COMPRESSION
  - vtkGDCMImageWriter, 1334
- NoElementsError
  - gdcmm::Parser, 792
- NoError
  - gdcmm::Parser, 792
- NOMAGIC
  - gdcmm::CSAHeader, 290
- NoMemoryError
  - gdcmm::Parser, 792
- NoObject
  - gdcmm::MediaStorage, 695
- NoOfItemsField
  - gdcmm::CSAElement, 287
- Norm
  - gdcmm::DirectionCosines, 379
- Normal
  - gdcmm::MrProtocol::Slice, 1016
- Normalize
  - gdcmm::DirectionCosines, 379
- NSet
  - gdcmm::NormalizedNetworkFunctions, 759
- NuclearMedicineImageStorage
  - gdcmm::MediaStorage, 693
  - gdcmm::UIDs, 1180
- NuclearMedicineImageStorageRetired
  - gdcmm::MediaStorage, 693
  - gdcmm::UIDs, 1180
- Null0
  - gdcmm::UIDs, 1184
- Null1
  - gdcmm::UIDs, 1184
- NumberOfDimensions
  - gdcmm::Bitmap, 205
  - gdcmm::ImageCodec, 565
- NumberOfIconImages
  - vtkGDCMImageReader, 1319
  - vtkGDCMImageReader2, 1331
- NumberOfOverlays
  - vtkGDCMImageReader, 1319
  - vtkGDCMImageReader2, 1331
- NumberOfSurfaces
  - gdcmm::SurfaceWriter, 1104
- OB
  - gdcmm::VR, 1296
- OB\_OW
  - gdcmm::VR, 1297
- Object
  - gdcmm::Object, 764
- ObjectEnd
  - gdcmm::MediaStorage, 695
- ObjectType
  - gdcmm::MediaStorage, 695
- OBLIQUE
  - gdcmm::Orientation, 777
- OD

- gdcmm::VR, [1296](#)
- OF
  - gdcmm::VR, [1296](#)
- Ofstream
  - gdcmm::Writer, [1416](#)
- OL
  - gdcmm::VR, [1296](#)
- OnlyUUID
  - gdcmm::XMLPrinter, [1421](#)
- op
  - gdcmm::SerieHelper, [995](#)
- OPENSSL
  - gdcmm::CryptoFactory, [277](#)
- OpenSSLCryptoFactory
  - gdcmm::OpenSSLCryptoFactory, [767](#)
- OpenSSLCryptographicMessageSyntax
  - gdcmm::OpenSSLCryptographicMessageSyntax, [769](#)
- OPENSSL7
  - gdcmm::CryptoFactory, [277](#)
- OpenSSL7CryptoFactory
  - gdcmm::OpenSSL7CryptoFactory, [772](#)
- OpenSSL7CryptographicMessageSyntax
  - gdcmm::OpenSSL7CryptographicMessageSyntax, [774](#)
- operator const char \*
  - gdcmm::ConstCharWrapper, [271](#)
  - gdcmm::Filename, [483](#)
  - gdcmm::String< TDelimiter, TMaxLength, TPadChar >, [1071](#)
- operator const double \*
  - gdcmm::DirectionCosines, [379](#)
- operator const std::vector< char > &
  - gdcmm::ByteValue, [223](#)
- operator MStype
  - gdcmm::MediaStorage, [697](#)
- operator ObjectType \*
  - gdcmm::SmartPointer< ObjectType >, [1019](#)
- operator PType
  - gdcmm::PhotometricInterpretation, [819](#)
- operator ScalarType
  - gdcmm::PixelFormat, [825](#)
- operator SwapCode::SwapCodeType
  - gdcmm::SwapCode, [1105](#)
- operator TStype
  - gdcmm::TransferSyntax, [1151](#)
  - gdcmm::UIDs, [1197](#)
- operator TypeType
  - gdcmm::Type, [1157](#)
- operator uint32\_t
  - gdcmm::VL, [1287](#)
- operator UsageType
  - gdcmm::Usage, [1272](#)
- operator VMType
  - gdcmm::VM, [1293](#)
- operator VRType
  - gdcmm::VR, [1300](#)
- operator!=
  - gdcmm, [64](#)
  - gdcmm::Attribute< Group, Element, TVR, TVM >, [134](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [141](#)
  - gdcmm::CodeString, [256](#)
  - gdcmm::PixelFormat, [825](#)
  - gdcmm::PrivateTag, [881](#)
  - gdcmm::Tag, [1128](#)
- operator<
  - gdcmm::Attribute< Group, Element, TVR, TVM >, [135](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [141](#)
  - gdcmm::CSAElement, [285](#)
  - gdcmm::CSAHeaderDictEntry, [298](#)
  - gdcmm::DataElement, [315](#)
  - gdcmm::DPath, [387](#)
  - gdcmm::PrivateTag, [881](#)
  - gdcmm::Tag, [1128](#)
- operator<<
  - gdcmm, [64–73](#)
  - gdcmm::BasicOffsetTable, [192](#)
  - gdcmm::CodeString, [256](#)
  - gdcmm::CommandDataSet, [264](#)
  - gdcmm::CSAElement, [287](#)
  - gdcmm::CSAHeader, [293](#)
  - gdcmm::CSAHeaderDict, [295](#)
  - gdcmm::CSAHeaderDictEntry, [299](#)
  - gdcmm::DataElement, [319](#)
  - gdcmm::DataSet, [336](#)
  - gdcmm::Dict, [360](#)
  - gdcmm::DictEntry, [368](#)
  - gdcmm::Dicts, [375](#)
  - gdcmm::Directory, [383](#)
  - gdcmm::DPath, [387](#)
  - gdcmm::File, [452](#)
  - gdcmm::FileMetaInformation, [480](#)
  - gdcmm::FileSet, [493](#)
  - gdcmm::Fragment, [514](#)
  - gdcmm::Global, [517](#)
  - gdcmm::GroupDict, [520](#)
  - gdcmm::IOD, [605](#)
  - gdcmm::IODEntry, [608](#)
  - gdcmm::IODs, [611](#)
  - gdcmm::Item, [621](#)
  - gdcmm::Macro, [683](#)
  - gdcmm::Macros, [685](#)
  - gdcmm::MediaStorage, [699](#)
  - gdcmm::Module, [721](#)
  - gdcmm::ModuleEntry, [724](#)
  - gdcmm::Modules, [726](#)
  - gdcmm::MrProtocol, [736](#)

- gdcm::NestedModuleEntries, 749
- gdcm::Object, 765
- gdcm::Orientation, 779
- gdcm::PDBElement, 800
- gdcm::PDBHeader, 803
- gdcm::PhotometricInterpretation, 819
- gdcm::PixelFormat, 827
- gdcm::Preamble, 855
- gdcm::PrivateDict, 877
- gdcm::PrivateTag, 882
- gdcm::Scanner, 943
- gdcm::Scanner2, 953
- gdcm::Sorter, 1028
- gdcm::StrictScanner, 1056
- gdcm::StrictScanner2, 1067
- gdcm::SwapCode, 1106
- gdcm::Table, 1117
- gdcm::Tag, 1132
- gdcm::TransferSyntax, 1152
- gdcm::Type, 1157
- gdcm::UI, 1158
- gdcm::Usage, 1272
- gdcm::Version, 1285
- gdcm::VL, 1289
- gdcm::VM, 1293
- gdcm::VR, 1300
- operator<=
  - gdcm::Tag, 1128
- operator>>
  - gdcm, 73
  - gdcm::Tag, 1132
- operator()
  - gdcm::DataSet, 333
  - gdcm::Scanner2::Itstr, 679
  - gdcm::Scanner::Itstr, 680
  - gdcm::StrictScanner2::Itstr, 680
  - gdcm::StrictScanner::Itstr, 681
- operator++
  - gdcm::VL, 1287, 1288
- operator+=
  - gdcm::VL, 1288
- operator->
  - gdcm::SmartPointer< ObjectType >, 1020
- operator=
  - gdcm::AnonymizeEvent, 106
  - gdcm::ASN1, 127
  - gdcm::Base64, 168
  - gdcm::BoxRegion, 212
  - gdcm::ByteSwapFilter, 217
  - gdcm::ByteValue, 223
  - gdcm::Command, 260
  - gdcm::CryptographicMessageSyntax, 280
  - gdcm::CSAElement, 285
  - gdcm::CSAHeaderDict, 295
  - gdcm::DataElement, 315
  - gdcm::DataEvent, 324
  - gdcm::DataSet, 333
  - gdcm::DataSetEvent, 339
  - gdcm::Defs, 347
  - gdcm::Dict, 359
  - gdcm::Dicts, 374
  - gdcm::Element< TVR, VM::VM1\_n >, 401
  - gdcm::Event, 435
  - gdcm::FileMetaInformation, 478
  - gdcm::FileNameEvent, 487
  - gdcm::Global, 517
  - gdcm::MemberCommand< T >, 703
  - gdcm::network::ULAction, 1200
  - gdcm::network::ULConnection, 1244
  - gdcm::network::UserInformation, 1275
  - gdcm::Object, 765
  - gdcm::Overlay, 786
  - gdcm::ParseException, 790
  - gdcm::Preamble, 854
  - gdcm::PrivateTag, 881
  - gdcm::ProgressEvent, 886
  - gdcm::SequenceOfItems, 988
  - gdcm::ServiceClassUser, 1001
  - gdcm::SHA1, 1006
  - gdcm::SimpleMemberCommand< T >, 1011
  - gdcm::SimpleSubjectWatcher, 1014
  - gdcm::SmartPointer< ObjectType >, 1020
  - gdcm::Table, 1116
  - gdcm::Tag, 1128
- operator==
  - gdcm, 73
  - gdcm::Attribute< Group, Element, TVR, TVM >, 135
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, 142
  - gdcm::ByteValue, 223
  - gdcm::CodeString, 257
  - gdcm::CSAElement, 285
  - gdcm::DataElement, 316
  - gdcm::network::AbstractSyntax, 101
  - gdcm::network::PresentationContextRQ, 866
  - gdcm::network::TransferSyntaxSub, 1153
  - gdcm::PDBElement, 799
  - gdcm::PixelFormat, 825
  - gdcm::PresentationContext, 857
  - gdcm::PrivateTag, 881, 882
  - gdcm::SequenceOfFragments, 980
  - gdcm::SequenceOfItems, 988
  - gdcm::Tag, 1128
  - gdcm::Value, 1281
- operator[]
  - gdcm::Attribute< Group, Element, TVR, TVM >, 135
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, 151

- gdcM::DataSet, 333
- gdcM::Element< TVR, TVM >, 394
- gdcM::Element< TVR, VM::VM1\_n >, 402
- gdcM::Tag, 1129
- operator\*
  - gdcM::SmartPointer< ObjectType >, 1020
- OphthalmicAxialMeasurementsStorage
  - gdcM::UIDs, 1184
- OphthalmicOpticalCoherenceTomographyBscanVolumeAnalysisStorage
  - gdcM::UIDs, 1184
- OphthalmicOpticalCoherenceTomographyEnFaceImageStorage
  - gdcM::UIDs, 1184
- OphthalmicPhotography16BitImageStorage
  - gdcM::MediaStorage, 695
  - gdcM::UIDs, 1181
- OphthalmicPhotography8BitImageStorage
  - gdcM::MediaStorage, 694
  - gdcM::UIDs, 1181
- OphthalmicThicknessMapStorage
  - gdcM::UIDs, 1185
- OphthalmicTomographyImageStorage
  - gdcM::MediaStorage, 694
  - gdcM::UIDs, 1181
- OphthalmicVisualFieldStaticPerimetryMeasurementsStorage
  - gdcM::UIDs, 1185
- OrderFileList
  - gdcM::SerieHelper, 994
- Orientation
  - gdcM::Orientation, 777
- OrientationType
  - gdcM::Orientation, 777
- Output
  - gdcM::BitmapToBitmapFilter, 208
- OutputFormat
  - vtkImageMapToColors16, 1383
- OutputTypes
  - gdcM::DictConverter, 361
- OV
  - gdcM::VR, 1296
- Overlay
  - gdcM::Overlay, 782, 783
- OverlayImageActor
  - vtkImageColorViewer, 1377
- Overlays
  - gdcM::Pixmap, 834
- OverlayType
  - gdcM::Overlay, 782
- OW
  - gdcM::VR, 1296
- Pack
  - gdcM::Unpacker12Bits, 1270
- Padding
  - gdcM::ApplicationEntity, 120
- gdcM::PersonName, 811
- PALETTE\_COLOR
  - gdcM::PhotometricInterpretation, 817
- Papyrus3ImplicitVRLittleEndian
  - gdcM::UIDs, 1183
- ParametricMapStorage
  - gdcM::UIDs, 1184
- Parent
  - gdcM::Element< TVR, VM::VM1\_2 >, 398
  - gdcM::Element< TVR, VM::VM2\_2n >, 406
  - gdcM::Element< TVR, VM::VM2\_n >, 408
  - gdcM::Element< TVR, VM::VM3\_3n >, 411
  - gdcM::Element< TVR, VM::VM3\_4 >, 413
  - gdcM::Element< TVR, VM::VM3\_n >, 415
- Parse
  - gdcM::Parser, 793
- ParseBuffer
  - gdcM::Parser, 793
- ParseCertificateFile
  - gdcM::CAPICryptographicMessageSyntax, 229
  - gdcM::CryptographicMessageSyntax, 280
  - gdcM::OpenSSLCryptographicMessageSyntax, 770
  - gdcM::OpenSSL7CryptographicMessageSyntax, 775
- ParseDateTime
  - gdcM::System, 1112, 1113
- ParseDump
  - gdcM::ASN1, 127
- ParseDumpFile
  - gdcM::ASN1, 128
- ParseException
  - gdcM::ParseException, 789, 790
- ParseKeyFile
  - gdcM::CAPICryptographicMessageSyntax, 230
  - gdcM::CryptographicMessageSyntax, 280
  - gdcM::OpenSSLCryptographicMessageSyntax, 770
  - gdcM::OpenSSL7CryptographicMessageSyntax, 775
- Parser
  - gdcM::Parser, 792
- PassAlphaToOutput
  - vtkImageMapToColors16, 1383
- Patient
  - gdcM::Patient, 794
- PatientRadiationDoseSRStorage
  - gdcM::UIDs, 1185
- PatientRootQueryRetrieveInformationModelFIND
  - gdcM::UIDs, 1181
- PatientRootQueryRetrieveInformationModelGET
  - gdcM::UIDs, 1181
- PatientRootQueryRetrieveInformationModelMOVE
  - gdcM::UIDs, 1181
- PatientStudyOnlyQueryRetrieveInformationModelFINDRetired
  - gdcM::UIDs, 1181



- PatientStudyOnlyQueryRetrieveInformationModelGETRetired
  - gdcm::UIDs, [1182](#)
- PatientStudyOnlyQueryRetrieveInformationModelIMOVERetired
  - gdcm::UIDs, [1182](#)
- PDataTFPDU
  - gdcm::network::PDataTFPDU, [796](#)
- PDBElement
  - gdcm::PDBElement, [799](#)
- PDBHeader
  - gdcm::PDBHeader, [802](#)
- PDF
  - gdcm::MediaStorage, [695](#)
- PDFCodec
  - gdcm::PDFCodec, [805](#)
- PerformAction
  - gdcm::network::ULAction, [1200](#)
  - gdcm::network::ULActionAA1, [1201](#)
  - gdcm::network::ULActionAA2, [1203](#)
  - gdcm::network::ULActionAA3, [1204](#)
  - gdcm::network::ULActionAA4, [1205](#)
  - gdcm::network::ULActionAA5, [1207](#)
  - gdcm::network::ULActionAA6, [1208](#)
  - gdcm::network::ULActionAA7, [1209](#)
  - gdcm::network::ULActionAA8, [1211](#)
  - gdcm::network::ULActionAE1, [1212](#)
  - gdcm::network::ULActionAE2, [1213](#)
  - gdcm::network::ULActionAE3, [1215](#)
  - gdcm::network::ULActionAE4, [1216](#)
  - gdcm::network::ULActionAE5, [1217](#)
  - gdcm::network::ULActionAE6, [1219](#)
  - gdcm::network::ULActionAE7, [1220](#)
  - gdcm::network::ULActionAE8, [1221](#)
  - gdcm::network::ULActionAR1, [1223](#)
  - gdcm::network::ULActionAR10, [1224](#)
  - gdcm::network::ULActionAR2, [1225](#)
  - gdcm::network::ULActionAR3, [1227](#)
  - gdcm::network::ULActionAR4, [1228](#)
  - gdcm::network::ULActionAR5, [1229](#)
  - gdcm::network::ULActionAR6, [1231](#)
  - gdcm::network::ULActionAR7, [1232](#)
  - gdcm::network::ULActionAR8, [1233](#)
  - gdcm::network::ULActionAR9, [1235](#)
  - gdcm::network::ULActionDT1, [1236](#)
  - gdcm::network::ULActionDT2, [1237](#)
- PerformedImagingAgentAdministrationSRStorage
  - gdcm::UIDs, [1185](#)
- PET20StepColorPaletteSOPInstance
  - gdcm::UIDs, [1183](#)
- PETColorPaletteSOPInstance
  - gdcm::UIDs, [1183](#)
- PETImageStorage
  - gdcm::MediaStorage, [693](#)
- PF
  - gdcm::Bitmap, [205](#)
- gdcm::ImageCodec, [565](#)
- PGXCodec
- gdcm::PGXCodec, [815](#)
- PHILIPS
  - gdcm::Dicts, [372](#)
- Philips3D
  - gdcm::MediaStorage, [694](#)
- PhilipsPrivateMRSyntheticImageStorage
  - gdcm::MediaStorage, [694](#)
- PhotometricInterpretation
  - gdcm::PhotometricInterpretation, [818](#)
- PI
  - gdcm::Bitmap, [205](#)
  - gdcm::ImageCodec, [565](#)
- PI\_END
  - gdcm::PhotometricInterpretation, [818](#)
- PIType
  - gdcm::PhotometricInterpretation, [817](#)
- PixelData
  - gdcm::Bitmap, [205](#)
  - gdcm::PixmapReader, [839](#)
  - gdcm::PixmapWriter, [846](#)
- PixelFormat
  - gdcm::PixelFormat, [822](#)
- Pixmap
  - gdcm::Pixmap, [832](#)
- PixmapReader
  - gdcm::Bitmap, [204](#)
  - gdcm::PixmapReader, [838](#)
- PixmapToPixmapFilter
  - gdcm::PixmapToPixmapFilter, [841](#)
- PixmapWriter
  - gdcm::PixmapWriter, [845](#)
- PlanarConfiguration
  - gdcm::Bitmap, [205](#)
  - gdcm::ImageCodec, [565](#)
  - vtkGDCMImageReader, [1319](#)
  - vtkGDCMImageReader2, [1331](#)
- PlannedImagingAgentAdministrationSRStorage
  - gdcm::UIDs, [1185](#)
- PMS
  - gdcm::EquipmentManufacturer, [431](#)
- PN
  - gdcm::VR, [1296](#)
- PNComp
  - gdcm, [59](#)
- PNMCodec
  - gdcm::PNMCodec, [850](#)
- pointer
  - gdcm::CodeString, [255](#)
  - gdcm::LO, [670](#)
  - gdcm::String< TDelimiter, TMaxLength, TPadChar >, [1070](#)
- POINTS



- gdcmm::Surface, [1084](#)
- Position
  - gdcmm::MrProtocol::Slice, [1016](#)
- PositronEmissionTomographyImageStorage
  - gdcmm::UIDs, [1181](#)
- Preamble
  - gdcmm::Preamble, [853](#)
- PrepareWrite
  - gdcmm::PixmapWriter, [845](#)
  - gdcmm::SegmentWriter, [973](#)
  - gdcmm::SurfaceWriter, [1103](#)
- PrepareWritePointMacro
  - gdcmm::SurfaceWriter, [1103](#)
- Prepend
  - gdcmm::Global, [517](#)
- PresentationContext
  - gdcmm::PresentationContext, [857](#)
- PresentationContextAC
  - gdcmm::network::PresentationContextAC, [859](#)
- PresentationContextArrayType
  - gdcmm::network::AAssociateRQPDU, [96](#)
  - gdcmm::PresentationContextGenerator, [862](#)
- PresentationContextGenerator
  - gdcmm::PresentationContextGenerator, [862](#)
- PresentationContextRQ
  - gdcmm::network::PresentationContextRQ, [865](#)
- PresentationDataValue
  - gdcmm::network::PresentationDataValue, [868](#)
- PresentationLUTSOPClass
  - gdcmm::UIDs, [1179](#)
- Preserve
  - gdcmm::Cleaner, [243](#)
- PrettyPrintOff
  - gdcmm::JSON, [663](#)
- PrettyPrintOn
  - gdcmm::JSON, [663](#)
- PrimitiveData
  - gdcmm::MeshPrimitive, [710](#)
- PrimitivesData
  - gdcmm::MeshPrimitive, [707](#)
- PrimitiveType
  - gdcmm::MeshPrimitive, [710](#)
- Print
  - gdcmm::ApplicationEntity, [119](#)
  - gdcmm::Attribute< Group, Element, TVR, TVM >, [135](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [142](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, [151](#)
  - gdcmm::BaseQuery, [178](#)
  - gdcmm::Bitmap, [200](#)
  - gdcmm::BoxRegion, [212](#)
  - gdcmm::ByteValue, [223](#)
  - gdcmm::CSAHeader, [292](#)
  - gdcmm::Curve, [306](#)
  - gdcmm::DataSet, [333](#)
  - gdcmm::DictPrinter, [371](#)
  - gdcmm::DirectionCosines, [379](#)
  - gdcmm::Directory, [382](#)
  - gdcmm::DPath, [387](#)
  - gdcmm::Element< TVR, TVM >, [395](#)
  - gdcmm::Element< TVR, VM::VM1\_n >, [402](#)
  - gdcmm::Element< VR::AS, VM::VM5 >, [416](#)
  - gdcmm::Event, [435](#)
  - gdcmm::Image, [534](#)
  - gdcmm::LookupTable, [678](#)
  - gdcmm::MEC\_MR3, [689](#)
  - gdcmm::MrProtocol, [736](#)
  - gdcmm::network::AAbortPDU, [86](#)
  - gdcmm::network::AAssociateACPDU, [90](#)
  - gdcmm::network::AAssociateRJPDU, [93](#)
  - gdcmm::network::AAssociateRQPDU, [98](#)
  - gdcmm::network::AbstractSyntax, [101](#)
  - gdcmm::network::ApplicationContext, [117](#)
  - gdcmm::network::AReleaseRPPDU, [122](#)
  - gdcmm::network::AReleaseRQPDU, [124](#)
  - gdcmm::network::AsynchronousOperationsWindowSub, [129](#)
  - gdcmm::network::BasePDU, [174](#)
  - gdcmm::network::ImplementationClassUIDSub, [595](#)
  - gdcmm::network::ImplementationVersionNameSub, [597](#)
  - gdcmm::network::MaximumLengthSub, [686](#)
  - gdcmm::network::PDataTFPDU, [797](#)
  - gdcmm::network::PresentationContextAC, [860](#)
  - gdcmm::network::PresentationContextRQ, [866](#)
  - gdcmm::network::PresentationDataValue, [869](#)
  - gdcmm::network::RoleSelectionSub, [933](#)
  - gdcmm::network::ServiceClassApplicationInformation, [996](#)
  - gdcmm::network::SOPClassExtendedNegotiationSub, [1021](#)
  - gdcmm::network::TransferSyntaxSub, [1153](#)
  - gdcmm::network::UserInformation, [1275](#)
  - gdcmm::Object, [765](#)
  - gdcmm::Orientation, [778](#)
  - gdcmm::Overlay, [786](#)
  - gdcmm::PDBHeader, [803](#)
  - gdcmm::PersonName, [810](#)
  - gdcmm::PixelFormat, [825](#)
  - gdcmm::Pixmap, [833](#)
  - gdcmm::Preamble, [854](#)
  - gdcmm::PresentationContext, [858](#)
  - gdcmm::Printer, [873](#)
  - gdcmm::Region, [921](#)
  - gdcmm::Scanner, [942](#)
  - gdcmm::Scanner2, [952](#)
  - gdcmm::SegmentedPaletteColorLookupTable, [966](#)

- gdcm::SequenceOfFragments, 980
- gdcm::SequenceOfItems, 988
- gdcm::Sorter, 1026
- gdcm::StrictScanner, 1055
- gdcm::StrictScanner2, 1066
- gdcm::TagPath, 1134
- gdcm::Testing, 1141
- gdcm::Version, 1285
- gdcm::XMLPrinter, 1421
- PrintASCII
  - gdcm::ByteValue, 223
- PrintASCIIXML
  - gdcm::ByteValue, 224
- PrintAsContinuousString
  - gdcm::Tag, 1129
- PrintAsContinuousUpperCaseString
  - gdcm::Tag, 1129
- PrintAsPipeSeparatedString
  - gdcm::Tag, 1129
- PrintDataElement
  - gdcm::Printer, 873
  - gdcm::XMLPrinter, 1422
- PrintDataElement2
  - gdcm::DictPrinter, 371
- PrintDataSet
  - gdcm::Printer, 873
  - gdcm::XMLPrinter, 1422
- PrintDataSet2
  - gdcm::DictPrinter, 371
- Printer
  - gdcm::Printer, 873
- PrinterConfigurationRetrievalSOPClass
  - gdcm::UIDs, 1179
- PrinterConfigurationRetrievalSOPInstance
  - gdcm::UIDs, 1179
- PrinterSOPClass
  - gdcm::UIDs, 1179
- PrinterSOPInstance
  - gdcm::UIDs, 1179
- PrintGroupLength
  - gdcm::ByteValue, 224
- PrintHex
  - gdcm::ByteValue, 224
- PrintHexXML
  - gdcm::ByteValue, 224
- PrintJobSOPClass
  - gdcm::UIDs, 1179
- PrintPNXML
  - gdcm::ByteValue, 224
- PrintQueueManagementSOPClassRetired
  - gdcm::UIDs, 1179
- PrintQueueSOPInstanceRetired
  - gdcm::UIDs, 1179
- PrintSelf
- vtkGDCMImageReader, 1311
- vtkGDCMImageReader2, 1324
- vtkGDCMImageWriter, 1335
- vtkGDCMMedicalImageProperties, 1342
- vtkGDCMPolyDataReader, 1345
- vtkGDCMPolyDataWriter, 1349
- vtkGDCMTesting, 1353
- vtkGDCMThreadedImageReader, 1358
- vtkGDCMThreadedImageReader2, 1361
- vtkImageColorViewer, 1371
- vtkImageMapToColors16, 1380
- vtkImageMapToWindowLevelColors2, 1385
- vtkImagePlanarComponentsToComponents, 1389
- vtkImageRGBToYBR, 1391
- vtkImageYBRToRGB, 1393
- vtkLookupTable16, 1395
- vtkRTStructSetProperties, 1402
- PrintSQ
  - gdcm::Printer, 874
  - gdcm::XMLPrinter, 1422
- PrintStyle
  - gdcm::Printer, 875
  - gdcm::XMLPrinter, 1422
- PrintStyles
  - gdcm::Printer, 872
  - gdcm::XMLPrinter, 1421
- PrintTable
  - gdcm::network::ULTransitionTable, 1259
  - gdcm::Scanner, 942
  - gdcm::Scanner2, 952
  - gdcm::StrictScanner, 1055
  - gdcm::StrictScanner2, 1066
- PrintXML
  - gdcm::PrivateDict, 876
- PrivateBegin
  - gdcm::Scanner2, 952
  - gdcm::StrictScanner2, 1066
- PrivateConstIterator
  - gdcm::Scanner2, 947
  - gdcm::StrictScanner2, 1060
- PrivateDict
  - gdcm::PrivateDict, 876
- PrivateEnd
  - gdcm::Scanner2, 953
  - gdcm::StrictScanner2, 1066
- PrivateMappingType
  - gdcm::Scanner2, 947
  - gdcm::StrictScanner2, 1060
- PrivateTag
  - gdcm::PrivateTag, 880
- PrivateTagToValue
  - gdcm::Scanner2, 947
  - gdcm::StrictScanner2, 1060
- PrivateTagToValueValueType

- gdcm::Scanner2, [947](#)
- gdcm::StrictScanner2, [1060](#)
- ProceduralEventLoggingSOPClass
  - gdcm::UIDs, [1178](#)
- ProceduralEventLoggingSOPInstance
  - gdcm::UIDs, [1178](#)
- ProcedureLogStorage
  - gdcm::UIDs, [1181](#)
- Process
  - gdcm::Parser, [793](#)
- ProcessDataSet
  - gdcm::FileExplicitFilter, [470](#)
- ProcessPrivateTag
  - gdcm::Scanner2, [953](#)
  - gdcm::StrictScanner2, [1066](#)
- ProcessPublicTag
  - gdcm::Scanner, [942](#)
  - gdcm::Scanner2, [953](#)
  - gdcm::StrictScanner, [1056](#)
  - gdcm::StrictScanner2, [1066](#)
- ProcessRequest
  - vtkGDCMImageReader2, [1324](#)
- ProduceCharacterSetDataElement
  - gdcm::QueryFactory, [895](#)
- ProduceQuery
  - gdcm::QueryFactory, [896](#)
- ProductCharacteristicsQuerySOPClass
  - gdcm::UIDs, [1182](#)
- ProgressEvent
  - gdcm::ProgressEvent, [885](#)
- PropertyCategory
  - gdcm::Segment, [961](#)
- PropertyType
  - gdcm::Segment, [961](#)
- PropertyTypeModifiers
  - gdcm::Segment, [961](#)
- ProtocolApprovalInformationModelFIND
  - gdcm::UIDs, [1185](#)
- ProtocolApprovalInformationModelGET
  - gdcm::UIDs, [1185](#)
- ProtocolApprovalInformationModelMOVE
  - gdcm::UIDs, [1185](#)
- ProtocolApprovalStorage
  - gdcm::UIDs, [1185](#)
- PseudoColorSoftcopyPresentationStateStorageSOPClass
  - gdcm::UIDs, [1180](#)
- PubChemCompoundCID
  - gdcm::UIDs, [1183](#)
- PublicConstIterator
  - gdcm::Scanner2, [947](#)
  - gdcm::StrictScanner2, [1061](#)
- PublicMappingType
  - gdcm::Scanner2, [947](#)
  - gdcm::StrictScanner2, [1061](#)
- PublicTagToValue
  - gdcm::Scanner2, [947](#)
  - gdcm::StrictScanner2, [1061](#)
- PublicTagToValueValueType
  - gdcm::Scanner2, [947](#)
  - gdcm::StrictScanner2, [1061](#)
- PullPrintRequestSOPClassRetired
  - gdcm::UIDs, [1179](#)
- PullStoredPrintManagementMetaSOPClassRetired
  - gdcm::UIDs, [1179](#)
- Push
  - gdcm::TagPath, [1134](#)
- PushBackFile
  - vtkGDCMMedicalImageProperties, [1342](#)
- PVRGCodec
  - gdcm::PVRGCodec, [889](#)
- PythonFilter
  - gdcm::PythonFilter, [891](#)
- Quality
  - gdcm::JPEGCodec, [654](#)
- QueryFactory
  - gdcm::BaseQuery, [179](#)
  - gdcm::BaseRootQuery, [184](#)
  - gdcm::FindPatientRootQuery, [505](#)
  - gdcm::FindStudyRootQuery, [509](#)
  - gdcm::ModalityPerformedProcedureStepCreateQuery, [713](#)
  - gdcm::ModalityPerformedProcedureStepSetQuery, [717](#)
  - gdcm::MovePatientRootQuery, [730](#)
  - gdcm::MoveStudyRootQuery, [734](#)
  - gdcm::WLMFindQuery, [1410](#)
- RadiomicsOntology
  - gdcm::UIDs, [1184](#)
- RadiopharmaceuticalRadiationDoseSRStorage
  - gdcm::UIDs, [1185](#)
- RAWCodec
  - gdcm::RAWCodec, [909](#)
- RawDataStorage
  - gdcm::MediaStorage, [693](#)
  - gdcm::UIDs, [1180](#)
- Read
  - gdcm::BasicOffsetTable, [191](#)
  - gdcm::ByteValue, [224](#)
  - gdcm::CommandDataSet, [263](#)
  - gdcm::CP246ExplicitDataElement, [274](#)
  - gdcm::DataElement, [316](#)
  - gdcm::DataSet, [333](#)
  - gdcm::Element< TVR, TVM >, [395](#)
  - gdcm::Element< TVR, VM::VM1\_n >, [402](#)
  - gdcm::EncodingImplementation< VR::VRASCII >, [426](#)

gdcm::EncodingImplementation< VR::VRBINARY >, 428  
 gdcm::ExplicitDataElement, 442  
 gdcm::ExplicitImplicitDataElement, 446  
 gdcm::File, 451  
 gdcm::FileMetaInformation, 478  
 gdcm::Fragment, 513  
 gdcm::ImageReader, 581  
 gdcm::ImageRegionReader, 586  
 gdcm::ImplicitDataElement, 601  
 gdcm::Item, 621  
 gdcm::network::AAabortPDU, 86  
 gdcm::network::AAAssociateACPDU, 90  
 gdcm::network::AAAssociateRJPDU, 93  
 gdcm::network::AAAssociateRQPDU, 98  
 gdcm::network::AbstractSyntax, 102  
 gdcm::network::ApplicationContext, 117  
 gdcm::network::AReleaseRPPDU, 122  
 gdcm::network::AReleaseRQPDU, 124  
 gdcm::network::AsynchronousOperationsWindowSub, 129  
 gdcm::network::BasePDU, 174  
 gdcm::network::ImplementationClassUIDSub, 595  
 gdcm::network::ImplementationVersionNameSub, 597  
 gdcm::network::MaximumLengthSub, 686  
 gdcm::network::PDataTFPDU, 797  
 gdcm::network::PresentationContextAC, 860  
 gdcm::network::PresentationContextRQ, 866  
 gdcm::network::PresentationDataValue, 869  
 gdcm::network::RoleSelectionSub, 933  
 gdcm::network::ServiceClassApplicationInformation, 996  
 gdcm::network::SOPClassExtendedNegociationSub, 1021  
 gdcm::network::TransferSyntaxSub, 1153  
 gdcm::network::UserInformation, 1276  
 gdcm::PGXCodec, 815  
 gdcm::PixmapReader, 838  
 gdcm::PNMCodec, 851  
 gdcm::Preamble, 854  
 gdcm::Reader, 915  
 gdcm::SegmentReader, 969  
 gdcm::SequenceOfFragments, 980  
 gdcm::SequenceOfItems, 988  
 gdcm::StreamImageReader, 1040  
 gdcm::SurfaceReader, 1099  
 gdcm::TableReader, 1121  
 gdcm::Tag, 1129  
 gdcm::UNExplicitDataElement, 1264  
 gdcm::UNExplicitImplicitDataElement, 1268  
 gdcm::ValueIO< TDE, TSwap, TType >, 1282  
 gdcm::VL, 1288  
 gdcm::VR, 1300  
 gdcm::VR16ExplicitDataElement, 1303  
 gdcm::VRVLSIZE< 0 >, 1306  
 gdcm::VRVLSIZE< 1 >, 1306  
 Read16  
   gdcm::VL, 1288  
 ReadACRNEMAIImage  
   gdcm::ImageReader, 581  
   gdcm::PixmapReader, 838  
 ReadBacktrack  
   gdcm::Fragment, 513  
 ReadCompat  
   gdcm::FileMetaInformation, 478  
 ReadCompatInternal  
   gdcm::FileMetaInformation, 478  
 ReadComputeLength  
   gdcm::EncodingImplementation< VR::VRASCII >, 426  
   gdcm::EncodingImplementation< VR::VRBINARY >, 428  
 ReadDataSet  
   gdcm::Reader, 915  
 Reader  
   gdcm::Reader, 914  
 ReadFiles  
   vtkGDCMThreadedImageReader, 1358  
 ReadFromCommaSeparatedString  
   gdcm::PrivateTag, 882  
   gdcm::Tag, 1130  
 ReadFromContinuousString  
   gdcm::Tag, 1130  
 ReadFromPipeSeparatedString  
   gdcm::Tag, 1130  
 ReadImage  
   gdcm::ImageReader, 581  
   gdcm::PixmapReader, 839  
 ReadImageInformation  
   gdcm::StreamImageReader, 1040  
 ReadImageInternal  
   gdcm::PixmapReader, 839  
 ReadInformation  
   gdcm::ImageRegionReader, 586  
 ReadInto  
   gdcm::network::PDataTFPDU, 797  
   gdcm::network::PresentationDataValue, 869  
 ReadIntoBuffer  
   gdcm::ImageRegionReader, 586  
 README.txt, 1427  
 ReadMetaInformation  
   gdcm::Reader, 915  
 ReadNested  
   gdcm::DataSet, 333  
 ReadNoSwap  
   gdcm::EncodingImplementation< VR::VRASCII >, 427

- gdcm::EncodingImplementation< VR::VRBINARY >, 428
- ReadOrSkip
  - gdcm::DataElement, 316
- ReadPointMacro
  - gdcm::SurfaceReader, 1099
- ReadPreamble
  - gdcm::Reader, 915
- ReadPreValue
  - gdcm::CP246ExplicitDataElement, 275
  - gdcm::DataElement, 316
  - gdcm::ExplicitDataElement, 442
  - gdcm::ExplicitImplicitDataElement, 446
  - gdcm::Fragment, 513
  - gdcm::ImplicitDataElement, 601
  - gdcm::SequenceOfFragments, 980
  - gdcm::UNExplicitDataElement, 1265
  - gdcm::UNExplicitImplicitDataElement, 1268
  - gdcm::VR16ExplicitDataElement, 1304
- ReadSegment
  - gdcm::SegmentReader, 969
- ReadSegments
  - gdcm::SegmentReader, 970
- ReadSelectedPrivateTags
  - gdcm::DataSet, 334
  - gdcm::Reader, 916
- ReadSelectedPrivateTagsWithLength
  - gdcm::DataSet, 334
- ReadSelectedTags
  - gdcm::DataSet, 334
  - gdcm::Reader, 916
- ReadSelectedTagsWithLength
  - gdcm::DataSet, 334
- ReadSurface
  - gdcm::SurfaceReader, 1100
- ReadSurfaces
  - gdcm::SurfaceReader, 1100
- Readuint16
  - gdcm::DictConverter, 363
- ReadUpToTag
  - gdcm::DataSet, 334
  - gdcm::Reader, 916
- ReadUpToTagWithLength
  - gdcm::DataSet, 334
- ReadValue
  - gdcm::CP246ExplicitDataElement, 275
  - gdcm::DataElement, 316
  - gdcm::ExplicitDataElement, 442
  - gdcm::ExplicitImplicitDataElement, 446
  - gdcm::Fragment, 513
  - gdcm::ImplicitDataElement, 601
  - gdcm::SequenceOfFragments, 980
  - gdcm::UNExplicitDataElement, 1265
  - gdcm::UNExplicitImplicitDataElement, 1269
- gdcm::VR16ExplicitDataElement, 1304
- ReadValueWithLength
  - gdcm::DataElement, 316
  - gdcm::ImplicitDataElement, 601
- ReadVM
  - gdcm::DictConverter, 363
- ReadVR
  - gdcm::DictConverter, 363
- ReadWithLength
  - gdcm::CP246ExplicitDataElement, 275
  - gdcm::DataElement, 317
  - gdcm::DataSet, 335
  - gdcm::ExplicitDataElement, 442
  - gdcm::ExplicitImplicitDataElement, 446
  - gdcm::ImplicitDataElement, 601
  - gdcm::UNExplicitDataElement, 1265
  - gdcm::VR16ExplicitDataElement, 1304
- RealWorldValueIntercept
  - gdcm::RealWorldValueMappingContent, 918
- RealWorldValueMappingStorage
  - gdcm::UIDs, 1180
- RealWorldValueSlope
  - gdcm::RealWorldValueMappingContent, 919
- RecommendedDisplayCIELabToRGB
  - gdcm::SurfaceHelper, 1094
- RecurseDataSet
  - gdcm::Anonymizer, 112
- RED
  - gdcm::LookupTable, 674
- red
  - gdcm::terminal, 82
- reference
  - gdcm::CodeString, 255
  - gdcm::LO, 670
  - gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1070
- ReferencedColorPrintManagementMetaSOPClassRetired
  - gdcm::UIDs, 1179
- ReferencedGrayscalePrintManagementMetaSOPClassRetired
  - gdcm::UIDs, 1179
- ReferencedImageBoxSOPClassRetired
  - gdcm::UIDs, 1179
- ReferenceFrameOfReferenceUID
  - vtkRTStructSetProperties, 1404
- ReferenceSeriesInstanceUID
  - vtkRTStructSetProperties, 1405
- Region
  - gdcm::Region, 920
- Register
  - gdcm::Object, 765
- Remove
  - gdcm::Anonymizer, 112
  - gdcm::Cleaner, 243, 244
  - gdcm::DataSet, 335

- gdcmm::FileAnonymizer, [455](#)
- gdcmm::Preamble, [854](#)
- RemoveAllGroupLength
  - gdcmm::Cleaner, [244](#)
- RemoveAllIllegal
  - gdcmm::Cleaner, [244](#)
- RemoveAllMissingPrivateCreator
  - gdcmm::Cleaner, [244](#)
- RemoveAllObservers
  - gdcmm::Subject, [1080](#)
- RemoveDictEntry
  - gdcmm::PrivateDict, [877](#)
- RemoveFile
  - gdcmm::System, [1113](#)
- RemoveGroupLength
  - gdcmm::Anonymizer, [112](#)
- RemoveItemByIndex
  - gdcmm::SequenceOfItems, [988](#)
- RemoveMissingPrivateCreator
  - gdcmm::Cleaner, [244](#)
- RemoveObserver
  - gdcmm::Subject, [1080](#)
- RemoveOverlay
  - gdcmm::Pixmap, [833](#)
- RemovePrivateTags
  - gdcmm::Anonymizer, [113](#)
- RemoveRetired
  - gdcmm::Anonymizer, [113](#)
- Render
  - vtkImageColorViewer, [1372](#)
- Renderer
  - vtkImageColorViewer, [1377](#)
- RenderWindow
  - vtkImageColorViewer, [1377](#)
- Replace
  - gdcmm::Anonymizer, [113](#), [114](#)
  - gdcmm::CommandDataSet, [264](#)
  - gdcmm::DataSet, [335](#)
  - gdcmm::FileAnonymizer, [456](#)
  - gdcmm::FileMetaInformation, [478](#)
- ReplaceEmpty
  - gdcmm::DataSet, [335](#)
- RequestData
  - vtkGDCMImageReader2, [1324](#)
  - vtkGDCMPolyDataReader, [1345](#)
  - vtkImageMapToColors16, [1380](#)
  - vtkImageMapToWindowLevelColors2, [1385](#)
  - vtkImagePlanarComponentsToComponents, [1389](#)
- RequestData\_HemodynamicWaveformStorage
  - vtkGDCMPolyDataReader, [1345](#)
- RequestData\_RTStructureSetStorage
  - vtkGDCMPolyDataReader, [1345](#)
- RequestDataCompat
  - vtkGDCMImageReader, [1311](#)
- vtkGDCMImageReader2, [1324](#)
- vtkGDCMThreadedImageReader, [1358](#)
- RequestInformation
  - vtkGDCMImageReader2, [1324](#)
  - vtkGDCMPolyDataReader, [1345](#)
  - vtkGDCMThreadedImageReader2, [1361](#)
  - vtkImageMapToColors16, [1380](#)
  - vtkImageMapToWindowLevelColors2, [1385](#)
- RequestInformation\_HemodynamicWaveformStorage
  - vtkGDCMPolyDataReader, [1345](#)
- RequestInformation\_RTStructureSetStorage
  - vtkGDCMPolyDataReader, [1345](#)
- RequestInformationCompat
  - vtkGDCMImageReader, [1312](#)
  - vtkGDCMImageReader2, [1324](#)
- RequestPaddedCompositePixelCode
  - gdcmm::ImageCodec, [565](#)
- RequestPlanarConfiguration
  - gdcmm::ImageCodec, [565](#)
- Rescale
  - gdcmm::Rescaler, [924](#)
- RescaleFunctionIntoBestFit
  - gdcmm::Rescaler, [924](#)
- Rescaler
  - gdcmm::Rescaler, [923](#)
- ReserveDataElement
  - gdcmm::FileStreamer, [497](#)
- ReserveGroupDataElement
  - gdcmm::FileStreamer, [497](#)
- reset
  - gdcmm::terminal, [82](#)
- ResetHandledDataSet
  - gdcmm::network::ULConnectionCallback, [1246](#)
- RespiratoryWaveformStorage
  - gdcmm::UIDs, [1184](#)
- RetrieveSOPInstanceUIDFromIndex
  - gdcmm::DirectoryHelper, [385](#)
- RetrieveSOPInstanceUIDFromZPosition
  - gdcmm::DirectoryHelper, [385](#)
- reverse
  - gdcmm::terminal, [82](#)
- reverse\_iterator
  - gdcmm::CodeString, [255](#)
  - gdcmm::LO, [670](#)
  - gdcmm::String< TDelimiter, TMaxLength, TPadChar >, [1070](#)
- RFC2557MIMEencapsulation
  - gdcmm::UIDs, [1178](#)
- RGB
  - gdcmm::PhotometricInterpretation, [817](#)
- RGB2YBR
  - gdcmm::ImageChangePhotometricInterpretation, [542](#)
- RGBPixelsToRGBPlanes
  - gdcmm::ImageChangePlanarConfiguration, [547](#)



- RGBPlanesToRGBPixels
  - gdcm::ImageChangePlanarConfiguration, [547](#)
- RGBToRecommendedDisplayCIELab
  - gdcm::SurfaceHelper, [1095](#)
- RGBToRecommendedDisplayGrayscale
  - gdcm::SurfaceHelper, [1095](#)
- RLE\_COMPRESSION
  - vtkGDCMImageWriter, [1334](#)
- RLECodec
  - gdcm::RLECodec, [929](#)
- RLELossless
  - gdcm::TransferSyntax, [1149](#)
  - gdcm::UIDs, [1178](#)
- ROI
  - gdcm::Overlay, [782](#)
- RoleSelectionSub
  - gdcm::network::RoleSelectionSub, [933](#)
- Round
  - gdcm, [74](#)
- roundat
  - gdcm, [74](#)
- RTBeamsDeliveryInstructionStorage
  - gdcm::UIDs, [1186](#)
- RTBeamsDeliveryInstructionStorageSupplement74FrozenDraft
  - gdcm::UIDs, [1182](#)
- RTBeamsTreatmentRecordStorage
  - gdcm::UIDs, [1181](#)
- RTBrachyApplicationSetupDeliveryInstructionStorage
  - gdcm::UIDs, [1186](#)
- RTBrachyTreatmentRecordStorage
  - gdcm::UIDs, [1181](#)
- RTConventionalMachineVerification
  - gdcm::UIDs, [1186](#)
- RTConventionalMachineVerificationSupplement74FrozenDraft
  - gdcm::UIDs, [1182](#)
- RTDoseStorage
  - gdcm::MediaStorage, [694](#)
  - gdcm::UIDs, [1181](#)
- RTImageStorage
  - gdcm::MediaStorage, [693](#)
  - gdcm::UIDs, [1181](#)
- RTIonBeamsTreatmentRecordStorage
  - gdcm::MediaStorage, [694](#)
  - gdcm::UIDs, [1181](#)
- RTIonMachineVerification
  - gdcm::UIDs, [1186](#)
- RTIonMachineVerificationSupplement74FrozenDraft
  - gdcm::UIDs, [1182](#)
- RTIonPlanStorage
  - gdcm::MediaStorage, [694](#)
  - gdcm::UIDs, [1181](#)
- RTPhysicianIntentStorage
  - gdcm::UIDs, [1185](#)
- RTPlanStorage
  - gdcm::MediaStorage, [694](#)
  - gdcm::UIDs, [1181](#)
- RTSegmentAnnotationStorage
  - gdcm::UIDs, [1185](#)
- RTStructSetProperties
  - vtkGDCMPolyDataReader, [1347](#)
  - vtkGDCMPolyDataWriter, [1350](#)
- RTStructureSetStorage
  - gdcm::MediaStorage, [694](#)
  - gdcm::UIDs, [1181](#)
- RTTreatmentSummaryRecordStorage
  - gdcm::MediaStorage, [694](#)
  - gdcm::UIDs, [1181](#)
- Rule
  - gdcm::SerieHelper, [992](#)
- RunEventLoop
  - gdcm::network::ULConnectionManager, [1252](#)
- RunMoveEventLoop
  - gdcm::network::ULConnectionManager, [1253](#)
- SAGITTAL
  - gdcm::Orientation, [777](#)
- SAMSUNG
  - gdcm::EquipmentManufacturer, [431](#)
- ScalarType
  - gdcm::PixelFormat, [822](#)
- Scale
  - vtkGDCMImageReader, [1319](#)
  - vtkGDCMImageReader2, [1331](#)
- Scan
  - gdcm::Scanner, [942](#)
  - gdcm::Scanner2, [953](#)
  - gdcm::StrictScanner, [1056](#)
  - gdcm::StrictScanner2, [1066](#)
- Scanner
  - gdcm::Scanner, [938](#)
- Scanner2
  - gdcm::Scanner2, [948](#)
- Scrub
  - gdcm::Cleaner, [244](#), [245](#)
- SecondaryCaptureImageStorage
  - gdcm::MediaStorage, [693](#)
  - gdcm::UIDs, [1180](#)
- Segment
  - gdcm::Segment, [957](#)
- SegmentAlgorithmName
  - gdcm::Segment, [961](#)
- SegmentAlgorithmType
  - gdcm::Segment, [961](#)
- Segmentation
  - gdcm::MediaStorage, [695](#)
- SegmentationStorage
  - gdcm::MediaStorage, [694](#)
  - gdcm::UIDs, [1180](#)

- SegmentDescription
  - gdcm::Segment, [962](#)
- SegmentedPaletteColorLookupTable
  - gdcm::SegmentedPaletteColorLookupTable, [965](#)
- SegmentedVolumeRenderingVolumetricPresentationStateStorage
  - gdcm::UIDs, [1184](#)
- SegmentLabel
  - gdcm::Segment, [962](#)
- SegmentMap
  - gdcm::SegmentReader, [969](#)
- SegmentNumber
  - gdcm::Segment, [962](#)
- SegmentReader
  - gdcm::SegmentReader, [969](#)
- Segments
  - gdcm::SegmentReader, [970](#)
  - gdcm::SegmentWriter, [974](#)
- SegmentVector
  - gdcm::SegmentReader, [969](#)
  - gdcm::SegmentWriter, [972](#)
- SegmentWriter
  - gdcm::SegmentWriter, [972](#)
- Selection
  - gdcm::Sorter, [1028](#)
- SelectionMap
  - gdcm::Sorter, [1026](#)
- Self
  - gdcm::AnonymizeEvent, [104](#)
  - gdcm::DataEvent, [323](#)
  - gdcm::DataSetEvent, [338](#)
  - gdcm::FileNameEvent, [485](#)
  - gdcm::MemberCommand< T >, [702](#)
  - gdcm::ProgressEvent, [884](#)
  - gdcm::SimpleMemberCommand< T >, [1010](#)
- SEMIAUTOMATIC
  - gdcm::Segment, [957](#)
- SendEcho
  - gdcm::network::ULConnectionManager, [1253](#)
  - gdcm::ServiceClassUser, [1001](#)
- SendFind
  - gdcm::network::ULConnectionManager, [1253](#)
  - gdcm::ServiceClassUser, [1001](#)
- SendMove
  - gdcm::network::ULConnectionManager, [1253](#)
  - gdcm::ServiceClassUser, [1002](#)
- SendNAction
  - gdcm::network::ULConnectionManager, [1253](#), [1254](#)
- SendNCreate
  - gdcm::network::ULConnectionManager, [1254](#)
- SendNDelete
  - gdcm::network::ULConnectionManager, [1254](#)
- SendNEventReport
  - gdcm::network::ULConnectionManager, [1254](#)
- SendNGet
  - gdcm::network::ULConnectionManager, [1254](#), [1255](#)
- SendNSet
  - gdcm::network::ULConnectionManager, [1255](#)
- SendStore
  - gdcm::network::ULConnectionManager, [1255](#)
  - gdcm::ServiceClassUser, [1002](#), [1003](#)
- Separator
  - gdcm::ApplicationEntity, [120](#)
  - gdcm::PersonName, [811](#)
- SequenceLengthField
  - gdcm::SequenceOfItems, [989](#)
- SequenceOfFragments
  - gdcm::SequenceOfFragments, [977](#)
- SequenceOfItems
  - gdcm::SequenceOfItems, [985](#)
- SerieHelper
  - gdcm::SerieHelper, [992](#)
- SerieRestrictions
  - gdcm::SerieHelper, [992](#)
- Series
  - gdcm::Series, [995](#)
- SeriesInstanceUID
  - vtkRTStructSetProperties, [1405](#)
- ServiceClassApplicationInformation
  - gdcm::network::ServiceClassApplicationInformation, [996](#)
- ServiceClassUser
  - gdcm::ServiceClassUser, [1000](#)
- Set
  - gdcm::Attribute< Group, Element, TVR, TVM >, [135](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [142](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [152](#)
  - gdcm::Element< TVR, TVM >, [395](#)
  - gdcm::Element< TVR, VM::VM1\_n >, [402](#)
- SetAbstractSyntax
  - gdcm::network::PresentationContextRQ, [867](#)
  - gdcm::PresentationContext, [858](#)
- SetAETitle
  - gdcm::ServiceClassUser, [1003](#)
- SetAlgorithmFamily
  - gdcm::Surface, [1089](#)
- SetAlgorithmName
  - gdcm::Surface, [1089](#)
- SetAlgorithmVersion
  - gdcm::Surface, [1089](#)
- SetAnatomicRegion
  - gdcm::Segment, [959](#)
- SetAnatomicRegionModifiers
  - gdcm::Segment, [959](#)
- SetAppendDerivationHistory
  - gdcm::FileDerivation, [467](#)
- SetArray



- gdcM::Element< TVR, VM::VM1\_n >, 402
- setAttribute
  - gdcM::terminal, 83
- SetAxisOfRotation
  - gdcM::Surface, 1089
- setbgcolor
  - gdcM::terminal, 83
- SetBitPosition
  - gdcM::Overlay, 786
- SetBitsAllocated
  - gdcM::Overlay, 786
  - gdcM::PixelFormat, 826
- SetBitSample
  - gdcM::JPEGCodec, 653
- SetBitsStored
  - gdcM::PixelFormat, 826
- SetBlob
  - gdcM::ApplicationEntity, 119
  - gdcM::network::PresentationDataValue, 869
  - gdcM::PersonName, 810
- SetBlueLUT
  - gdcM::LookupTable, 678
- SetBufferLength
  - gdcM::JPEGLSCodec, 661
  - gdcM::PNMCodec, 851
  - gdcM::RLECodec, 932
- SetByteSwapTag
  - gdcM::ByteSwapFilter, 217
- SetByteValue
  - gdcM::Attribute< Group, Element, TVR, TVM >, 136
  - gdcM::Attribute< Group, Element, TVR, VM::VM1 >, 142
  - gdcM::Attribute< Group, Element, TVR, VM::VM1\_n >, 152
  - gdcM::CSAElement, 285
  - gdcM::DataElement, 317
- SetByteValueNoSwap
  - gdcM::Attribute< Group, Element, TVR, TVM >, 136
  - gdcM::Attribute< Group, Element, TVR, VM::VM1 >, 142
- SetCallbackFunction
  - gdcM::MemberCommand< T >, 703, 704
  - gdcM::SimpleMemberCommand< T >, 1011
- SetCalledAETitle
  - gdcM::network::AAssociateACPDU, 91
  - gdcM::network::AAssociateRQPDU, 98
  - gdcM::ServiceClassUser, 1003
- SetCallingAETitle
  - gdcM::network::AAssociateACPDU, 91
  - gdcM::network::AAssociateRQPDU, 98
- SetCenterOfRotation
  - gdcM::Surface, 1089
- SetChangePrivateTags
  - gdcM::FileExplicitFilter, 470
- SetCheckFileMetaInformation
  - gdcM::Writer, 1414
- SetCipherType
  - gdcM::CAPICryptographicMessageSyntax, 230
  - gdcM::CryptographicMessageSyntax, 280
  - gdcM::OpenSSLCryptographicMessageSyntax, 770
  - gdcM::OpenSSL7CryptographicMessageSyntax, 775
- SetColor
  - gdcM::Printer, 874
- SetColorLevel
  - vtkImageColorViewer, 1372
- SetColorWindow
  - vtkImageColorViewer, 1372
- SetColumns
  - gdcM::Bitmap, 200
  - gdcM::Overlay, 786
- SetCommand
  - gdcM::network::PresentationDataValue, 870
- SetComponents
  - gdcM::PersonName, 810
- SetCompressIconImage
  - gdcM::ImageChangeTransferSyntax, 552
- SetComputeZSpacing
  - gdcM::IPPSorter, 614
- SetCoordinateStartValue
  - gdcM::Curve, 306
- SetCoordinateStepValue
  - gdcM::Curve, 306
- SetCryptographicMessageSyntax
  - gdcM::Anonymizer, 114
- SetCurve
  - gdcM::Curve, 306
  - vtkGDCMImageReader, 1312
  - vtkGDCMImageReader2, 1324
- SetCurveDataDescriptor
  - gdcM::Curve, 307
- SetCurveDescription
  - gdcM::Curve, 307
- SetData
  - gdcM::DataEvent, 325
- SetDataElement
  - gdcM::Bitmap, 200
- SetDataSet
  - gdcM::File, 451
  - gdcM::network::PresentationDataValue, 870
- SetDataSetTransferSyntax
  - gdcM::FileMetaInformation, 479
- SetDataValueRepresentation
  - gdcM::Curve, 307
- SetDebug
  - gdcM::Trace, 1144
- SetDebugStream
  - gdcM::Trace, 1144

- SetDefaultTransferSyntax
  - gdcm::PresentationContextGenerator, [863](#)
- SetDerivationCodeSequenceCodeValue
  - gdcm::FileDerivation, [467](#)
- SetDerivationDescription
  - gdcm::FileDerivation, [467](#)
- SetDescription
  - gdcm::CSAHeaderDictEntry, [298](#)
  - gdcm::ModuleEntry, [724](#)
  - gdcm::Overlay, [786](#)
- SetDescriptor
  - gdcm::DICOMDIRGenerator, [355](#)
- SetDictName
  - gdcm::DictConverter, [363](#)
- SetDicts
  - gdcm::PythonFilter, [891](#)
  - gdcm::StringFilter, [1074](#)
- SetDimension
  - gdcm::Bitmap, [200](#)
- SetDimensions
  - gdcm::Bitmap, [201](#)
  - gdcm::Curve, [307](#)
  - gdcm::ImageCodec, [562](#)
- SetDimensionsValue
  - gdcm::ImageHelper, [575](#)
- SetDirectionCosines
  - gdcm::Image, [534](#)
  - vtkGDCMImageWriter, [1335](#)
- SetDirectionCosinesFromImageOrientationPatient
  - vtkGDCMImageWriter, [1335](#)
- SetDirectionCosinesTolerance
  - gdcm::IPPSorter, [614](#)
- SetDirectionCosinesValue
  - gdcm::ImageHelper, [575](#)
- SetDirectory
  - gdcm::network::ULWritingCallback, [1261](#)
  - gdcm::SerieHelper, [994](#)
- SetDisplayId
  - vtkImageColorViewer, [1372](#)
- SetDomain
  - gdcm::BoxRegion, [212](#)
- SetDropDuplicatePositions
  - gdcm::IPPSorter, [614](#)
- SetElement
  - gdcm::Tag, [1130](#)
- SetElementHandler
  - gdcm::Parser, [793](#)
- SetElementTag
  - gdcm::Tag, [1130](#), [1131](#)
- SetElementXX
  - gdcm::DictEntry, [366](#)
- SetError
  - gdcm::Trace, [1145](#)
- SetErrorStream
  - gdcm::Trace, [1145](#)
- SetEvent
  - gdcm::network::ULEvent, [1257](#)
- setfgcolor
  - gdcm::terminal, [83](#)
- SetFile
  - gdcm::Anonymizer, [114](#)
  - gdcm::Cleaner, [245](#)
  - gdcm::DICOMDIRGenerator, [355](#)
  - gdcm::FileDecompressLookupTable, [464](#)
  - gdcm::FileDerivation, [467](#)
  - gdcm::FileExplicitFilter, [470](#)
  - gdcm::IconImageFilter, [523](#)
  - gdcm::Printer, [874](#)
  - gdcm::PythonFilter, [891](#)
  - gdcm::Reader, [916](#)
  - gdcm::SplitMosaicFilter, [1034](#)
  - gdcm::StreamImageWriter, [1044](#)
  - gdcm::StringFilter, [1074](#)
  - gdcm::Validate, [1278](#)
  - gdcm::Writer, [1415](#)
  - gdcm::XMLPrinter, [1422](#)
- SetFileName
  - gdcm::FileNameEvent, [487](#)
  - gdcm::Reader, [916](#)
  - gdcm::StreamImageReader, [1040](#)
  - gdcm::StreamImageWriter, [1045](#)
  - gdcm::Writer, [1415](#)
  - vtkGDCMThreadedImageReader2, [1362](#)
- SetFilename
  - gdcm::TableReader, [1121](#)
- SetFileNames
  - vtkGDCMImageReader, [1312](#)
  - vtkGDCMImageWriter, [1335](#)
  - vtkGDCMThreadedImageReader2, [1362](#)
- SetFileNames
  - gdcm::DICOMDIRGenerator, [355](#)
- SetFilePattern
  - vtkGDCMImageReader, [1312](#)
  - vtkGDCMImageReader2, [1325](#)
- SetFilePrefix
  - vtkGDCMImageReader, [1312](#)
  - vtkGDCMImageReader2, [1325](#)
- SetFiles
  - gdcm::FileSet, [492](#)
- SetFiniteVolume
  - gdcm::Surface, [1090](#)
- SetForce
  - gdcm::ImageChangeTransferSyntax, [552](#)
  - gdcm::ImageFragmentSplitter, [570](#)
- SetForcePixelSpacing
  - gdcm::ImageHelper, [575](#)
- SetForceRescaleInterceptSlope
  - gdcm::ImageHelper, [575](#)

- SetFragmentSizeMax
  - gdcm::ImageFragmentSplitter, [570](#)
- SetFrameOrigin
  - gdcm::Overlay, [787](#)
- SetFromDataElement
  - gdcm::Attribute< Group, Element, TVR, TVM >, [136](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [142](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [152](#)
  - gdcm::Element< TVR, TVM >, [395](#)
  - gdcm::Element< TVR, VM::VM1\_n >, [402](#)
- SetFromDataSet
  - gdcm::Attribute< Group, Element, TVR, TVM >, [136](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [143](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [152](#)
  - gdcm::MediaStorage, [698](#)
- SetFromFile
  - gdcm::MediaStorage, [698](#)
- SetFromHeader
  - gdcm::MediaStorage, [698](#)
- SetFromModality
  - gdcm::MediaStorage, [698](#)
- SetFromSourceImageSequence
  - gdcm::MediaStorage, [698](#)
- SetFromString
  - gdcm::DirectionCosines, [379](#)
- SetFromUID
  - gdcm::UIDs, [1197](#)
- SetGreenLUT
  - gdcm::LookupTable, [678](#)
- SetGroup
  - gdcm::Curve, [307](#)
  - gdcm::Overlay, [787](#)
  - gdcm::Tag, [1131](#)
- SetGroupXX
  - gdcm::DictEntry, [367](#)
- SetHeader
  - gdcm::File, [452](#)
- SetHighBit
  - gdcm::PixelFormat, [826](#)
- SetHostname
  - gdcm::ServiceClassUser, [1003](#)
- SetIconImage
  - gdcm::Pixmap, [833](#)
- SetIE
  - gdcm::IODEntry, [607](#)
- SetImage
  - gdcm::PixmapWriter, [846](#)
  - gdcm::SplitMosaicFilter, [1034](#)
- SetImplementationClassUID
  - gdcm::FileMetaInformation, [479](#)
- SetImplementationVersionName
  - gdcm::FileMetaInformation, [479](#)
- SetImplicitFlag
  - gdcm::network::ULConnectionCallback, [1247](#)
- SetInput
  - gdcm::BitmapToBitmapFilter, [207](#)
  - gdcm::ImageConverter, [567](#)
  - vtkImageColorViewer, [1372](#)
- SetInputConnection
  - vtkImageColorViewer, [1372](#)
- SetInputDirectory
  - gdcm::EmptyMaskGenerator, [424](#)
- SetInputFileName
  - gdcm::DictConverter, [363](#)
  - gdcm::FileAnonymizer, [456](#)
  - gdcm::FileChangeTransferSyntax, [460](#)
- SetIntercept
  - gdcm::Image, [534](#)
  - gdcm::Rescaler, [925](#)
- SetKey
  - gdcm::CSAElement, [286](#)
- SetKeyword
  - gdcm::DictEntry, [367](#)
- SetLastElement
  - gdcm::ParseException, [790](#)
- SetLastFragment
  - gdcm::network::PresentationDataValue, [870](#)
- SetLength
  - gdcm::ByteValue, [225](#)
  - gdcm::Element< TVR, VM::VM1\_2 >, [399](#)
  - gdcm::Element< TVR, VM::VM1\_n >, [403](#)
  - gdcm::Element< TVR, VM::VM2\_2n >, [406](#)
  - gdcm::Element< TVR, VM::VM2\_n >, [408](#)
  - gdcm::Element< TVR, VM::VM3\_3n >, [411](#)
  - gdcm::Element< TVR, VM::VM3\_4 >, [413](#)
  - gdcm::Element< TVR, VM::VM3\_n >, [416](#)
  - gdcm::RLECodec, [932](#)
  - gdcm::SequenceOfFragments, [981](#)
  - gdcm::SequenceOfItems, [989](#)
  - gdcm::Value, [1281](#)
- SetLengthOnly
  - gdcm::ByteValue, [225](#)
  - gdcm::Value, [1281](#)
- SetLengthToUndefined
  - gdcm::SequenceOfItems, [989](#)
- SetLoadMode
  - gdcm::SerieHelper, [994](#)
- SetLookupTable
  - vtkImageMapToColors16, [1381](#)
- SetLossless
  - gdcm::JPEGCodec, [653](#)
  - gdcm::JPEGLSCodec, [661](#)
- SetLossyError
  - gdcm::JPEGLSCodec, [661](#)

- SetLossyFlag
  - gdcm::Bitmap, 201
  - gdcm::ImageCodec, 562
  - gdcm::PVRGCodec, 890
- SetLUT
  - gdcm::Bitmap, 201
  - gdcm::ImageCodec, 562
  - gdcm::LookupTable, 678
  - gdcm::SegmentedPaletteColorLookupTable, 966
- SetManifold
  - gdcm::Surface, 1090
- SetMaximumLength
  - gdcm::network::MaximumLengthSub, 687
- SetMaximumPointDistance
  - gdcm::Surface, 1090
- SetMaxPDULength
  - gdcm::network::ULConnectionInfo, 1248
- SetMaxPDUSize
  - gdcm::network::ULConnection, 1244
- SetMCT
  - gdcm::JPEG2000Codec, 639
- SetMeanPointDistance
  - gdcm::Surface, 1090
- SetMedicalImageProperties
  - vtkGDCMImageReader, 1312
  - vtkGDCMImageReader2, 1325
  - vtkGDCMImageWriter, 1335
  - vtkGDCMPolyDataWriter, 1349
- SetMergeModeToAbstractSyntax
  - gdcm::PresentationContextGenerator, 864
- SetMergeModeToTransferSyntax
  - gdcm::PresentationContextGenerator, 864
- SetMeshPrimitive
  - gdcm::Surface, 1090
- SetMessageHeader
  - gdcm::network::PresentationDataValue, 870
- SetMinMaxForPixelFormat
  - gdcm::Rescaler, 925
- setmode
  - gdcm::terminal, 83
- SetName
  - gdcm::CSAElement, 286
  - gdcm::CSAHeaderDictEntry, 298
  - gdcm::DictEntry, 367
  - gdcm::IODEntry, 607
  - gdcm::Macro, 683
  - gdcm::Module, 721
  - gdcm::ModuleEntry, 724
  - gdcm::network::AbstractSyntax, 102
  - gdcm::network::ApplicationContext, 117
  - gdcm::network::TransferSyntaxSub, 1153
  - gdcm::PDBElement, 799
- SetNameFromUID
  - gdcm::network::AbstractSyntax, 102
- gdcm::network::TransferSyntaxSub, 1153
- SetNeedByteSwap
  - gdcm::Bitmap, 201
  - gdcm::ImageCodec, 562
- SetNeedOverlayCleanup
  - gdcm::ImageCodec, 563
- SetNestedDataSet
  - gdcm::Item, 621
- SetNoOfItems
  - gdcm::CSAElement, 286
- SetNoSwap
  - gdcm::Element< TVR, TVM >, 395
  - gdcm::Element< TVR, VM::VM1\_n >, 403
- SetNumberOfCurves
  - gdcm::Pixmap, 833
- SetNumberOfDimensions
  - gdcm::Bitmap, 201
  - gdcm::ImageCodec, 563
- SetNumberOfFilenames
  - gdcm::FilenameGenerator, 490
- SetNumberOfFrames
  - gdcm::Overlay, 787
- SetNumberOfInputPorts
  - vtkGDCMPolyDataWriter, 1349
- SetNumberOfItems
  - gdcm::SequenceOfItems, 989
- SetNumberOfOverlays
  - gdcm::Pixmap, 833
- SetNumberOfPoints
  - gdcm::Curve, 307
- SetNumberOfResolutions
  - gdcm::JPEG2000Codec, 639
- SetNumberOfSegments
  - gdcm::SegmentWriter, 973
- SetNumberOfSurfacePoints
  - gdcm::Surface, 1090
- SetNumberOfSurfaces
  - gdcm::SurfaceWriter, 1103
- SetNumberOfTableValues
  - vtkLookupTable16, 1395
- SetNumberOfThreadsForDecompression
  - gdcm::JPEG2000Codec, 639
- SetNumberOfValues
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, 152
- SetNumberOfVectors
  - gdcm::Surface, 1090
- SetObliquityThresholdCosineValue
  - gdcm::Orientation, 778
- SetOffScreenRendering
  - vtkImageColorViewer, 1372
- SetOrigin
  - gdcm::Image, 534, 535
  - gdcm::Overlay, 787

- SetOriginValue
  - gdcm::ImageHelper, 576
- SetOutputDimensions
  - gdcm::IconImageGenerator, 525
- SetOutputDirectory
  - gdcm::EmptyMaskGenerator, 424
- SetOutputFileName
  - gdcm::DictConverter, 363
  - gdcm::FileAnonymizer, 456
  - gdcm::FileChangeTransferSyntax, 460
  - gdcm::FileStreamer, 497
- SetOutputFormatToLuminance
  - vtkImageMapToColors16, 1381
- SetOutputFormatToLuminanceAlpha
  - vtkImageMapToColors16, 1381
- SetOutputFormatToRGB
  - vtkImageMapToColors16, 1381
- SetOutputFormatToRGBA
  - vtkImageMapToColors16, 1381
- SetOutputType
  - gdcm::DictConverter, 363
- SetOutsideValuePixel
  - gdcm::IconImageGenerator, 525
- SetOverlay
  - gdcm::Overlay, 787
- SetOverlayVisibility
  - vtkImageColorViewer, 1373
- SetOwner
  - gdcm::PrivateTag, 882
- SetParentId
  - vtkImageColorViewer, 1373
- SetPassword
  - gdcm::CAPICryptographicMessageSyntax, 230
  - gdcm::CryptographicMessageSyntax, 281
  - gdcm::OpenSSLCryptographicMessageSyntax, 770
  - gdcm::OpenSSL7CryptographicMessageSyntax, 775
- SetPattern
  - gdcm::FilenameGenerator, 490
- SetPDU
  - gdcm::network::ULEvent, 1258
- SetPermissions
  - gdcm::System, 1113
- SetPhotometricInterpretation
  - gdcm::Bitmap, 202
  - gdcm::ImageChangePhotometricInterpretation, 543
  - gdcm::ImageCodec, 563
- SetPixelFormat
  - gdcm::Bitmap, 202
  - gdcm::ImageCodec, 563
  - gdcm::JPEGCodec, 653
  - gdcm::Rescaler, 925
- SetPixelMinMax
  - gdcm::IconImageGenerator, 526
- SetPixelRepresentation
  - gdcm::PixelFormat, 826
- SetPixmap
  - gdcm::FileDecompressLookupTable, 464
  - gdcm::IconImageGenerator, 526
  - gdcm::PixmapWriter, 846
- SetPlanarConfiguration
  - gdcm::Bitmap, 202
  - gdcm::ImageChangePlanarConfiguration, 548
  - gdcm::ImageCodec, 563
- SetPMSRescaleInterceptSlope
  - gdcm::ImageHelper, 576
- SetPointCoordinatesData
  - gdcm::Surface, 1090
- SetPointPositionAccuracy
  - gdcm::Surface, 1091
- SetPointsBoundingBoxCoordinates
  - gdcm::Surface, 1091
- SetPort
  - gdcm::ServiceClassUser, 1003
- SetPortSCP
  - gdcm::ServiceClassUser, 1004
- SetPosition
  - vtkImageColorViewer, 1373
- SetPreamble
  - gdcm::FileMetaInformation, 479
- SetPrefix
  - gdcm::FilenameGenerator, 490
- SetPresentationContextID
  - gdcm::network::PresentationContextAC, 860
  - gdcm::network::PresentationContextRQ, 867
  - gdcm::network::PresentationDataValue, 870
  - gdcm::PresentationContext, 858
- SetPresentationContexts
  - gdcm::network::ULConnection, 1244
  - gdcm::ServiceClassUser, 1004
- SetPrettyPrint
  - gdcm::JSON, 663
- SetPrimitiveData
  - gdcm::MeshPrimitive, 709
- SetPrimitivesData
  - gdcm::MeshPrimitive, 709
- SetPrimitiveType
  - gdcm::MeshPrimitive, 709
- SetPrivateCreator
  - gdcm::Tag, 1131
- SetProcessingAlgorithm
  - gdcm::Surface, 1091
- SetProgress
  - gdcm::ProgressEvent, 886
- SetPropertyCategory
  - gdcm::Segment, 960
- SetPropertyType
  - gdcm::Segment, 960

- SetPropertyTypeModifiers
  - gdcm::Segment, [960](#)
- SetPurposeOfReferenceCodeSequenceCodeValue
  - gdcm::FileDerivation, [468](#)
- SetQuality
  - gdcm::JPEG2000Codec, [639](#)
  - gdcm::JPEGCodec, [653](#)
- SetRate
  - gdcm::JPEG2000Codec, [639](#)
- SetReason
  - gdcm::network::AAAbortPDU, [87](#)
  - gdcm::network::PresentationContextAC, [860](#)
- SetRecommendedDisplayCIELabValue
  - gdcm::Surface, [1091](#)
- SetRecommendedDisplayGrayscaleValue
  - gdcm::Surface, [1091](#)
- SetRecommendedPresentationOpacity
  - gdcm::Surface, [1091](#)
- SetRecommendedPresentationType
  - gdcm::Surface, [1092](#)
- SetRecomputeItemLength
  - gdcm::FileExplicitFilter, [470](#)
- SetRecomputeSequenceLength
  - gdcm::FileExplicitFilter, [470](#)
- SetRedLUT
  - gdcm::LookupTable, [678](#)
- SetRef
  - gdcm::IODEntry, [608](#)
- SetRegion
  - gdcm::ImageRegionReader, [587](#)
- SetRenderer
  - vtkImageColorViewer, [1373](#)
- SetRenderWindow
  - vtkImageColorViewer, [1373](#)
- SetRescaleInterceptSlopeValue
  - gdcm::ImageHelper, [576](#)
- SetRetired
  - gdcm::DictEntry, [367](#)
- SetReversible
  - gdcm::JPEG2000Codec, [640](#)
- SetRGB8
  - gdcm::ImageApplyLookupTable, [539](#)
- SetRoot
  - gdcm::UIDGenerator, [1160](#)
- SetRootDirectory
  - gdcm::DICOMDIRGenerator, [356](#)
- SetRows
  - gdcm::Bitmap, [202](#)
  - gdcm::Overlay, [787](#)
- SetRTStructSetProperties
  - vtkGDCMPolyDataWriter, [1349](#)
- SetSamplesPerPixel
  - gdcm::PixelFormat, [826](#)
- SetScalarType
  - gdcm::PixelFormat, [826](#)
- SetSearchParameter
  - gdcm::BaseQuery, [178](#)
- SetSecondaryCaptureImagePlaneModule
  - gdcm::ImageHelper, [576](#)
- SetSegmentAlgorithmName
  - gdcm::Segment, [960](#)
- SetSegmentAlgorithmType
  - gdcm::Segment, [960](#)
- SetSegmentDescription
  - gdcm::Segment, [960](#)
- SetSegmentLabel
  - gdcm::Segment, [960](#)
- SetSegmentNumber
  - gdcm::Segment, [961](#)
- SetSegments
  - gdcm::SegmentWriter, [973](#)
- SetSize
  - vtkImageColorViewer, [1373](#), [1374](#)
- SetSlice
  - vtkImageColorViewer, [1374](#)
- SetSliceOrientation
  - vtkImageColorViewer, [1374](#)
- SetSliceOrientationToXY
  - vtkImageColorViewer, [1374](#)
- SetSliceOrientationToXZ
  - vtkImageColorViewer, [1374](#)
- SetSliceOrientationToYZ
  - vtkImageColorViewer, [1374](#)
- SetSlope
  - gdcm::Image, [535](#)
  - gdcm::Rescaler, [925](#)
- SetSOPClassUIDMode
  - gdcm::EmptyMaskGenerator, [424](#)
- SetSOPInstanceUID
  - gdcm::BaseQuery, [178](#)
- SetSortFunction
  - gdcm::Sorter, [1027](#)
- SetSource
  - gdcm::network::AAAbortPDU, [87](#)
- SetSourceApplicationEntityTitle
  - gdcm::FileMetaInformation, [479](#)
- SetSpacing
  - gdcm::Image, [535](#)
- SetSpacingValue
  - gdcm::ImageHelper, [576](#)
- SetState
  - gdcm::network::ULConnection, [1244](#)
- SetStream
  - gdcm::Reader, [917](#)
  - gdcm::StreamImageReader, [1041](#)
  - gdcm::StreamImageWriter, [1045](#)
  - gdcm::Trace, [1145](#)
  - gdcm::Writer, [1415](#)



- SetStreamToFile
  - gdcm::Trace, [1145](#)
- SetStyle
  - gdcm::Printer, [874](#)
  - gdcm::XMLPrinter, [1422](#)
- SetSurfaceComments
  - gdcm::Surface, [1092](#)
- SetSurfaceCount
  - gdcm::Segment, [961](#)
- SetSurfaceNumber
  - gdcm::Surface, [1092](#)
- SetSurfaceProcessing
  - gdcm::Surface, [1092](#)
- SetSurfaceProcessingDescription
  - gdcm::Surface, [1092](#)
- SetSurfaceProcessingRatio
  - gdcm::Surface, [1092](#)
- SetSyngoDT
  - gdcm::CSAElement, [286](#)
- SetTag
  - gdcm::AnonymizeEvent, [106](#)
  - gdcm::DataElement, [317](#)
- SetTagsToRead
  - gdcm::Sorter, [1027](#)
- SetTargetPixelFormat
  - gdcm::Rescaler, [925](#)
- SetTemplateFileName
  - gdcm::FileStreamer, [497](#)
- SetTileSize
  - gdcm::JPEG2000Codec, [640](#)
- SetTimeout
  - gdcm::network::ARTIMTimer, [126](#)
  - gdcm::ServiceClassUser, [1004](#)
- SetToUndefined
  - gdcm::VL, [1288](#)
- SetTransferSyntax
  - gdcm::Bitmap, [203](#)
  - gdcm::FileChangeTransferSyntax, [461](#)
  - gdcm::ImageChangeTransferSyntax, [552](#)
  - gdcm::network::PresentationContextAC, [860](#)
- SetTuple
  - gdcm::network::RoleSelectionSub, [934](#)
  - gdcm::network::ServiceClassApplicationInformation, [996](#)
  - gdcm::network::SOPClassExtendedNegociationSub, [1021](#)
- SetType
  - gdcm::ModuleEntry, [724](#)
  - gdcm::Overlay, [788](#)
- SetTypeOfData
  - gdcm::Curve, [307](#)
- SetupInteractor
  - vtkImageColorViewer, [1375](#)
- SetUsage
  - gdcm::IODEntry, [608](#)
- SetUserCodec
  - gdcm::ImageChangeTransferSyntax, [552](#)
- SetUserData
  - gdcm::Parser, [794](#)
- SetUserInformation
  - gdcm::network::AAssociateRQPDU, [98](#)
- SetUseSeriesDetails
  - gdcm::SerieHelper, [994](#)
- SetUseTargetPixelFormat
  - gdcm::Rescaler, [926](#)
- SetUseVRUN
  - gdcm::FileExplicitFilter, [470](#)
- SetValue
  - gdcm::Attribute< Group, Element, TVR, TVM >, [137](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [143](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [152](#), [153](#)
  - gdcm::CSAElement, [286](#)
  - gdcm::DataElement, [318](#)
  - gdcm::Element< TVR, TVM >, [396](#)
  - gdcm::Element< TVR, VM::VM1\_n >, [403](#)
  - gdcm::PDBelement, [799](#)
- SetValueFieldLength
  - gdcm::DataElement, [318](#)
- SetValues
  - gdcm::Attribute< Group, Element, TVR, TVM >, [137](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [153](#)
- SetVectorAccuracy
  - gdcm::Surface, [1092](#)
- SetVectorCoordinateData
  - gdcm::Surface, [1092](#)
- SetVectorDimensionality
  - gdcm::Surface, [1093](#)
- SetVL
  - gdcm::DataElement, [318](#)
- SetVLToUndefined
  - gdcm::DataElement, [318](#)
- SetVM
  - gdcm::CSAElement, [286](#)
  - gdcm::CSAHeaderDictEntry, [298](#)
  - gdcm::DictEntry, [367](#)
- SetVR
  - gdcm::CSAElement, [286](#)
  - gdcm::CSAHeaderDictEntry, [298](#)
  - gdcm::DataElement, [319](#)
  - gdcm::DictEntry, [367](#)
- SetWarning
  - gdcm::Trace, [1145](#)
- SetWarningStream
  - gdcm::Trace, [1146](#)
- SetWindowId

- vtkImageColorViewer, 1375
- SetWriteDataSetOnly
  - gdcm::Writer, 1416
- SetZSpacingTolerance
  - gdcm::IPPSorter, 615
- SH
  - gdcm::VR, 1296
- SHA1
  - gdcm::SHA1, 1006
- SHComp
  - gdcm, 59
- Shift
  - vtkGDCMImageReader, 1319
  - vtkGDCMImageReader2, 1331
- ShiftEnd
  - gdcm::ByteBuffer, 214
- ShowAbort
  - gdcm::SimpleSubjectWatcher, 1014
- ShowAnonymization
  - gdcm::SimpleSubjectWatcher, 1014
- ShowData
  - gdcm::SimpleSubjectWatcher, 1014
- ShowDataSet
  - gdcm::SimpleSubjectWatcher, 1014
- ShowFileName
  - gdcm::SimpleSubjectWatcher, 1014
- ShowIteration
  - gdcm::SimpleSubjectWatcher, 1014
- ShowProgress
  - gdcm::SimpleSubjectWatcher, 1015
- SIEMENS
  - gdcm::Dicts, 372
  - gdcm::EquipmentManufacturer, 431
- SimpleMemberCommand
  - gdcm::SimpleMemberCommand< T >, 1010
- SimpleSubjectWatcher
  - gdcm::SimpleSubjectWatcher, 1013
- SimplifiedAdultEchoSRStorage
  - gdcm::UIDs, 1185
- SINGLEBIT
  - gdcm::PixelFormat, 822
- SingleSerieUIDFileSetHT
  - gdcm::SerieHelper, 995
- SingleSerieUIDFileSetmap
  - gdcm::SerieHelper, 992
- Size
  - gdcm::CodeString, 256
  - gdcm::DataSet, 336
  - gdcm::GroupDict, 519
  - gdcm::network::AAAbortPDU, 87
  - gdcm::network::AAssociateACPDU, 91
  - gdcm::network::AAssociateRJPDU, 93
  - gdcm::network::AAssociateRQPDU, 98
  - gdcm::network::AbstractSyntax, 102
  - gdcm::network::ApplicationContext, 117
  - gdcm::network::AReleaseRPPDU, 122
  - gdcm::network::AReleaseRQPDU, 124
  - gdcm::network::AsynchronousOperationsWindowSub, 129
  - gdcm::network::BasePDU, 174
  - gdcm::network::ImplementationClassUIDSub, 595
  - gdcm::network::ImplementationVersionNameSub, 597
  - gdcm::network::MaximumLengthSub, 687
  - gdcm::network::PDataTFPDU, 797
  - gdcm::network::PresentationContextAC, 860
  - gdcm::network::PresentationContextRQ, 867
  - gdcm::network::PresentationDataValue, 870
  - gdcm::network::RoleSelectionSub, 934
  - gdcm::network::ServiceClassApplicationInformation, 996
  - gdcm::network::SOPClassExtendedNegociationSub, 1022
  - gdcm::network::TransferSyntaxSub, 1153
  - gdcm::network::UserInformation, 1276
- size\_type
  - gdcm::CodeString, 255
  - gdcm::LO, 670
  - gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1070
- SizeType
  - gdcm::DataSet, 328
  - gdcm::FilenameGenerator, 488
  - gdcm::IOD, 604
  - gdcm::NestedModuleEntries, 748
  - gdcm::network::AAssociateACPDU, 89
  - gdcm::network::AAssociateRQPDU, 96
  - gdcm::network::PDataTFPDU, 796
  - gdcm::network::PresentationContextRQ, 865
  - gdcm::PresentationContext, 856
  - gdcm::PresentationContextGenerator, 862
  - gdcm::SequenceOfFragments, 977
  - gdcm::SequenceOfItems, 985
- SL
  - gdcm::VR, 1296
- Slice
  - vtkImageColorViewer, 1378
- SLICE\_ORIENTATION\_XY
  - vtkImageColorViewer, 1369
- SLICE\_ORIENTATION\_XZ
  - vtkImageColorViewer, 1369
- SLICE\_ORIENTATION\_YZ
  - vtkImageColorViewer, 1369
- SliceOrientation
  - vtkImageColorViewer, 1378
- Slices
  - gdcm::MrProtocol::SliceArray, 1017
- SmartPointer



- gdcm::Object, [765](#)
- gdcm::SmartPointer< ObjectType >, [1019](#)
- SOPClassExtendedNegociationSub
  - gdcm::network::SOPClassExtendedNegociationSub, [1021](#)
- SOPClassUIDMode
  - gdcm::EmptyMaskGenerator, [423](#)
- SOPInstanceUID
  - vtkRTStructSetProperties, [1405](#)
- Sort
  - gdcm::IPPSorter, [615](#)
  - gdcm::Sorter, [1027](#)
- Sorter
  - gdcm::Sorter, [1026](#)
- SortFunc
  - gdcm::Sorter, [1028](#)
- SortFunction
  - gdcm::Sorter, [1026](#)
- SpacialFiducialsStorage
  - gdcm::MediaStorage, [693](#)
- SpacialRegistrationStorage
  - gdcm::MediaStorage, [693](#)
- Spacing
  - gdcm::Spacing, [1030](#)
- SpacingType
  - gdcm::Spacing, [1030](#)
- SpatialFiducialsStorage
  - gdcm::UIDs, [1180](#)
- SpatialRegistrationStorage
  - gdcm::UIDs, [1180](#)
- SpectaclePrescriptionReportStorage
  - gdcm::UIDs, [1184](#)
- Spectroscopy
  - gdcm::Spectroscopy, [1031](#)
- Split
  - gdcm::ImageFragmentSplitter, [570](#)
  - gdcm::SplitMosaicFilter, [1034](#)
- SplitExtent
  - vtkGDCMThreadedImageReader2, [1362](#)
- SplitMosaicFilter
  - gdcm::SplitMosaicFilter, [1033](#)
- SPM2AVG152PDFFrameofReference
  - gdcm::UIDs, [1178](#)
- SPM2AVG152T1FrameofReference
  - gdcm::UIDs, [1178](#)
- SPM2AVG152T2FrameofReference
  - gdcm::UIDs, [1178](#)
- SPM2AVG305T1FrameofReference
  - gdcm::UIDs, [1178](#)
- SPM2BRAINMASKFrameofReference
  - gdcm::UIDs, [1178](#)
- SPM2CSFFrameofReference
  - gdcm::UIDs, [1178](#)
- SPM2EPIFrameofReference
  - gdcm::UIDs, [1178](#)
- SPM2FILT1FrameofReference
  - gdcm::UIDs, [1178](#)
- SPM2GRAYFrameofReference
  - gdcm::UIDs, [1178](#)
- SPM2PDFFrameofReference
  - gdcm::UIDs, [1178](#)
- SPM2PETFrameofReference
  - gdcm::UIDs, [1178](#)
- SPM2SINGLESUBJT1FrameofReference
  - gdcm::UIDs, [1178](#)
- SPM2SPECTFrameofReference
  - gdcm::UIDs, [1178](#)
- SPM2T1FrameofReference
  - gdcm::UIDs, [1178](#)
- SPM2T2FrameofReference
  - gdcm::UIDs, [1178](#)
- SPM2TRANSMFrameofReference
  - gdcm::UIDs, [1178](#)
- SPM2WHITEFrameofReference
  - gdcm::UIDs, [1178](#)
- SpringColorPaletteSOPInstance
  - gdcm::UIDs, [1183](#)
- SQ
  - gdcm::VR, [1296](#)
- Squeeze
  - gdcm::ApplicationEntity, [119](#)
- SS
  - gdcm::VR, [1296](#)
- ST
  - gdcm::VR, [1297](#)
- StableSort
  - gdcm::Sorter, [1027](#)
- StandaloneCurveStorage
  - gdcm::MediaStorage, [693](#)
- StandaloneCurveStorageRetired
  - gdcm::UIDs, [1180](#)
- StandaloneModalityLUTStorage
  - gdcm::MediaStorage, [693](#)
- StandaloneModalityLUTStorageRetired
  - gdcm::UIDs, [1180](#)
- StandaloneOverlayStorage
  - gdcm::MediaStorage, [693](#)
- StandaloneOverlayStorageRetired
  - gdcm::UIDs, [1180](#)
- StandalonePETCurveStorageRetired
  - gdcm::UIDs, [1181](#)
- StandaloneVOILUTStorage
  - gdcm::MediaStorage, [693](#)
- StandaloneVOILUTStorageRetired
  - gdcm::UIDs, [1180](#)
- Start
  - gdcm::network::ARTIMTimer, [126](#)
- StartAssociation

- gdcmm::ServiceClassUser, 1004
- StartDataElement
  - gdcmm::FileStreamer, 498
- StartElement
  - gdcmm::TableReader, 1121
  - gdcmm::XMLDictReader, 1419
  - gdcmm::XMLPrivateDictReader, 1425
- StartElementHandler
  - gdcmm::Parser, 792
- StartEncode
  - gdcmm::ImageCodec, 563
  - gdcmm::JPEG2000Codec, 640
  - gdcmm::JPEGCodec, 653
  - gdcmm::JPEGLSCodec, 661
  - gdcmm::RLECodec, 932
- StartFilter
  - gdcmm::SimpleSubjectWatcher, 1015
- StartGroupDataElement
  - gdcmm::FileStreamer, 498
- STATES
  - gdcmm::Surface, 1084
- STATES\_END
  - gdcmm::Surface, 1084
- STComp
  - gdcmm, 59
- StereometricRelationshipStorage
  - gdcmm::UIDs, 1181
- Stop
  - gdcmm::network::ARTIMTimer, 126
- StopAssociation
  - gdcmm::ServiceClassUser, 1005
- StopDataElement
  - gdcmm::FileStreamer, 498
- StopEncode
  - gdcmm::ImageCodec, 564
  - gdcmm::JPEG2000Codec, 640
  - gdcmm::JPEGCodec, 654
  - gdcmm::JPEGLSCodec, 661
  - gdcmm::RLECodec, 932
- StopGroupDataElement
  - gdcmm::FileStreamer, 498
- StopProtocol
  - gdcmm::network::ULConnection, 1244
- StorageCommitmentPullModelSOPClassRetired
  - gdcmm::UIDs, 1178
- StorageCommitmentPullModelSOPInstanceRetired
  - gdcmm::UIDs, 1178
- StorageCommitmentPushModelSOPClass
  - gdcmm::UIDs, 1178
- StorageCommitmentPushModelSOPInstance
  - gdcmm::UIDs, 1178
- StorageServiceClass
  - gdcmm::UIDs, 1179
- StoredPrintStorageSOPClassRetired
  - gdcmm::UIDs, 1179
- StrCaseCmp
  - gdcmm::System, 1113
- Stream
  - gdcmm::Writer, 1416
- StreamImageReader
  - gdcmm::Reader, 917
  - gdcmm::StreamImageReader, 1038
- StreamImageWriter
  - gdcmm::StreamImageWriter, 1043
  - gdcmm::Writer, 1416
- StrictScanner
  - gdcmm::StrictScanner, 1052
- StrictScanner2
  - gdcmm::StrictScanner2, 1061
- String
  - gdcmm::String< TDelimiter, TMaxLength, TPadChar >, 1070, 1071
- StringFilter
  - gdcmm::StringFilter, 1073
- StrNCaseCmp
  - gdcmm::System, 1113
- StrSep
  - gdcmm::System, 1114
- StrTokR
  - gdcmm::System, 1114
- StructureSetDate
  - vtkRTStructSetProperties, 1405
- StructureSetLabel
  - vtkRTStructSetProperties, 1405
- StructureSetName
  - vtkRTStructSetProperties, 1405
- StructureSetTime
  - vtkRTStructSetProperties, 1405
- Study
  - gdcmm::Study, 1076
- StudyComponentManagementSOPClass
  - gdcmm::MediaStorage, 694
- StudyComponentManagementSOPClassRetired
  - gdcmm::UIDs, 1178
- StudyInstanceUID
  - vtkRTStructSetProperties, 1405
- StudyRootQueryRetrieveInformationModelFIND
  - gdcmm::UIDs, 1181
- StudyRootQueryRetrieveInformationModelGET
  - gdcmm::UIDs, 1181
- StudyRootQueryRetrieveInformationModelMOVE
  - gdcmm::UIDs, 1181
- Subject
  - gdcmm::Subject, 1078
- SubjectiveRefractionMeasurementsStorage
  - gdcmm::UIDs, 1184
- SubstanceAdministrationLoggingSOPClass
  - gdcmm::UIDs, 1178

- SubstanceAdministrationLoggingSOPInstance
  - gdcm::UIDs, [1178](#)
- SubstanceApprovalQuerySOPClass
  - gdcm::UIDs, [1182](#)
- SummerColorPaletteSOPInstance
  - gdcm::UIDs, [1183](#)
- Superclass
  - gdcm::AnonymizeEvent, [104](#)
  - gdcm::DataEvent, [323](#)
  - gdcm::DataSetEvent, [338](#)
  - gdcm::FileNameEvent, [485](#)
  - gdcm::LO, [670](#)
  - gdcm::ProgressEvent, [884](#)
- SURFACE
  - gdcm::Surface, [1084](#)
- Surface
  - gdcm::Surface, [1084](#)
- SurfaceCount
  - gdcm::Segment, [962](#)
- SurfaceReader
  - gdcm::SurfaceReader, [1099](#)
- Surfaces
  - gdcm::Segment, [962](#)
- SurfaceScanMeshStorage
  - gdcm::UIDs, [1184](#)
- SurfaceScanPointCloudStorage
  - gdcm::UIDs, [1184](#)
- SurfaceSegmentationStorage
  - gdcm::MediaStorage, [694](#)
  - gdcm::UIDs, [1183](#)
- SurfaceVector
  - gdcm::Segment, [956](#)
- SurfaceWriter
  - gdcm::SurfaceWriter, [1103](#)
- SV
  - gdcm::VR, [1297](#)
- SV10
  - gdcm::CSAHeader, [290](#)
- Swap
  - gdcm::ByteSwap< T >, [215](#)
  - gdcm::SwapperDoOp, [1106](#)
  - gdcm::SwapperNoOp, [1107](#)
- SwapArray
  - gdcm::SwapperDoOp, [1106](#)
  - gdcm::SwapperNoOp, [1107](#)
- SwapCode
  - gdcm::SwapCode, [1105](#)
- SwapCodeType
  - gdcm::SwapCode, [1105](#)
- SwapFromSwapCodeIntoSystem
  - gdcm::ByteSwap< T >, [215](#)
- SwapRange
  - gdcm::ByteSwap< T >, [215](#)
- SwapRangeFromSwapCodeIntoSystem
  - gdcm::ByteSwap< T >, [215](#)
- SyngoDTField
  - gdcm::CSAElement, [287](#)
- SyntaxError
  - gdcm::Parser, [792](#)
- SystemIsBigEndian
  - gdcm::ByteSwap< T >, [216](#)
- SystemIsLittleEndian
  - gdcm::ByteSwap< T >, [216](#)
- T1
  - gdcm::Type, [1157](#)
- T1C
  - gdcm::Type, [1157](#)
- T2
  - gdcm::Type, [1157](#)
- T2C
  - gdcm::Type, [1157](#)
- T3
  - gdcm::Type, [1157](#)
- Table
  - gdcm::Table, [1116](#)
- Table16
  - vtkLookupTable16, [1396](#)
- TableEntry
  - gdcm::TableEntry, [1118](#)
- TableInternal
  - gdcm::Table, [1117](#)
- TableReader
  - gdcm::TableReader, [1119](#)
- TableRow
  - gdcm::network::TableRow, [1122](#)
- Tag
  - gdcm::Tag, [1125](#)
- tag
  - gdcm::Tag, [1132](#)
- TagField
  - gdcm::DataElement, [320](#)
- TagMismatchError
  - gdcm::Parser, [792](#)
- TagPath
  - gdcm::TagPath, [1133](#)
- tags
  - gdcm::Tag, [1132](#)
- TagsToRead
  - gdcm::Sorter, [1028](#)
- TagToValue
  - gdcm::Scanner, [938](#)
  - gdcm::StrictScanner, [1051](#)
- TagToValueValueType
  - gdcm::Scanner, [938](#)
  - gdcm::StrictScanner, [1051](#)
- TalairachBrainAtlasFrameofReference
  - gdcm::UIDs, [1178](#)

- TConstMemberFunctionPointer
  - gdcm::MemberCommand< T >, 702
- TestAbortOff
  - gdcm::SimpleSubjectWatcher, 1015
- TestAbortOn
  - gdcm::SimpleSubjectWatcher, 1015
- Testing
  - gdcm::Testing, 1136
- TestPBKDF2
  - gdcm::ASN1, 128
- TestsList.txt, 1427
- TextSRStorageTrialRetired
  - gdcm::UIDs, 1181
- ThreadedExecute
  - vtkImageRGBToYBR, 1391
  - vtkImageYBRToRGB, 1393
- ThreadedRequestData
  - vtkGDCMThreadedImageReader2, 1362
  - vtkImageMapToColors16, 1381
  - vtkImageMapToWindowLevelColors2, 1386
- TM
  - gdcm::VR, 1297
- TMComp
  - gdcm, 59
- TMemberFunctionPointer
  - gdcm::MemberCommand< T >, 702
  - gdcm::SimpleMemberCommand< T >, 1010
- Todo List, 3
- ToPyObject
  - gdcm::PythonFilter, 892
- TOSHIBA
  - gdcm::EquipmentManufacturer, 431
- ToshibaPrivateDataStorage
  - gdcm::MediaStorage, 694
- ToString
  - gdcm::StringFilter, 1075
- ToStringPair
  - gdcm::StringFilter, 1075, 1076
- ToUnixSlashes
  - gdcm::Filename, 483
- ToWindowsSlashes
  - gdcm::Filename, 483
- Trace
  - gdcm::Trace, 1143
- TractographyResultsStorage
  - gdcm::UIDs, 1184
- TransferSyntax
  - gdcm::TransferSyntax, 1149
- TransferSyntaxArrayType
  - gdcm::PresentationContext, 856
- TransferSyntaxes
  - gdcm::PresentationContext, 858
- TransferSyntaxStringsType
  - gdcm::UIDs, 1177
- TransferSyntaxSub
  - gdcm::network::TransferSyntaxSub, 1152
- Transition
  - gdcm::network::Transition, 1154, 1155
- transitions
  - gdcm::network::TableRow, 1123
- TRIANGLE
  - gdcm::MeshPrimitive, 707
- TRIANGLE\_FAN
  - gdcm::MeshPrimitive, 707
- TRIANGLE\_STRIP
  - gdcm::MeshPrimitive, 707
- Trim
  - gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1071, 1072
- TrimInternal
  - gdcm::CodeString, 256
- Truncate
  - gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1072
- TryJPEG2000Codec
  - gdcm::Bitmap, 203
  - gdcm::ImageChangeTransferSyntax, 553
- TryJPEG2000Codec2
  - gdcm::Bitmap, 203
- TryJPEGCodec
  - gdcm::Bitmap, 203
  - gdcm::ImageChangeTransferSyntax, 553
- TryJPEGCodec2
  - gdcm::Bitmap, 203
- TryJPEGLSCodec
  - gdcm::Bitmap, 203
  - gdcm::ImageChangeTransferSyntax, 553
- TryKAKADUCodec
  - gdcm::Bitmap, 203
- TryPVRGCodec
  - gdcm::Bitmap, 204
- TryRAWCodec
  - gdcm::Bitmap, 204
  - gdcm::ImageChangeTransferSyntax, 553
- TryRLECodec
  - gdcm::Bitmap, 204
  - gdcm::ImageChangeTransferSyntax, 553
- TS
  - gdcm::Bitmap, 205
- TS\_END
  - gdcm::TransferSyntax, 1149
- TSName
  - gdcm::UIDs, 1177
- TSType
  - gdcm::TransferSyntax, 1149
  - gdcm::UIDs, 1186
- Type
  - gdcm::Element< TVR, TVM >, 393

- gdcM::Element< TVR, VM::VM1\_n >, [400](#)
- gdcM::EquipmentManufacturer, [431](#)
- gdcM::Type, [1157](#)
- gdcM::VL, [1286](#)
- TYPETOENCODING
  - gdcMVR.h, [1673](#)
- TYPETOLENGTH
  - gdcMVM.h, [1670](#)
- TypeToString
  - gdcM::EquipmentManufacturer, [432](#)
- TypeType
  - gdcM::Type, [1156](#)
- UberonOntology
  - gdcM::UIDs, [1183](#)
- UC
  - gdcM::VR, [1297](#)
- UCComp
  - gdcM, [60](#)
- UI
  - gdcM::VR, [1297](#)
- UIComp
  - gdcM, [60](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_1
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_10
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_11
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_12
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_13
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_14
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_15
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_16
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_17
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_18
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_19
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_2
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_20
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_21
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_22
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_23
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_24
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_25
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_26
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_27
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_28
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_29
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_3
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_30
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_31
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_4
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_5
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_6
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_7
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_8
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_9
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_4\_1
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_4\_2
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_4\_3
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_4\_4
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_4\_5
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_4\_6
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_4\_7
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_0\_4\_8
  - gdcM::UIDs, [1192](#)
- uid\_1\_2\_840\_10008\_15\_1\_1
  - gdcM::UIDs, [1195](#)
- uid\_1\_2\_840\_10008\_1\_1
  - gdcM::UIDs, [1186](#)
- uid\_1\_2\_840\_10008\_1\_2
  - gdcM::UIDs, [1186](#)
- uid\_1\_2\_840\_10008\_1\_20

gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_1\_20\_1  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_1\_20\_1\_1  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_1\_20\_2  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_1\_20\_2\_1  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_1\_2\_1  
gdcn::UIDs, [1186](#)  
uid\_1\_2\_840\_10008\_1\_2\_1\_99  
gdcn::UIDs, [1186](#)  
uid\_1\_2\_840\_10008\_1\_2\_2  
gdcn::UIDs, [1186](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_100  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_101  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_102  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_103  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_104  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_105  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_106  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_107  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_108  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_50  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_51  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_52  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_53  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_54  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_55  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_56  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_57  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_58  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_59  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_60

gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_61  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_62  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_63  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_64  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_65  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_66  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_70  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_80  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_81  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_90  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_91  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_92  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_93  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_94  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_95  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_5  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_6\_1  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_2\_6\_2  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_3\_10  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_40  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_1\_40\_1  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_1\_42  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_1\_42\_1  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_1  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_10  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_11  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_12

gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_13  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_14  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_15  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_16  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_17  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_18  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_2  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_3  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_4  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_5  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_6  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_7  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_8  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_9  
gdcn::UIDs, [1187](#)  
uid\_1\_2\_840\_10008\_1\_4\_2\_1  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_1\_4\_2\_2  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_1\_5\_1  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_1\_5\_2  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_1\_5\_3  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_1\_5\_4  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_1\_5\_5  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_1\_5\_6  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_1\_5\_7  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_1\_5\_8  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_1\_9  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_2\_16\_10  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_2\_16\_11

gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_2\_16\_12  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_2\_16\_13  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_2\_16\_14  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_2\_16\_4  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_2\_16\_5  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_2\_16\_6  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_2\_16\_7  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_2\_16\_8  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_2\_16\_9  
gdcn::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_2\_6\_1  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_1\_1  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_1\_1  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_1\_4  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_2\_1  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_3\_1  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_3\_2  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_3\_3  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_3\_4  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_3\_5  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_5\_1  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_5\_4  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_5\_5  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_6\_1  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_4\_2  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_1  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_14  
gdcn::UIDs, [1188](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_15

gdc::UIDs, [1188](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_16  
 gdc::UIDs, [1188](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_16\_376  
 gdc::UIDs, [1188](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_17  
 gdc::UIDs, [1188](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_17\_376  
 gdc::UIDs, [1188](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_18  
 gdc::UIDs, [1188](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_18\_1  
 gdc::UIDs, [1188](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_2  
 gdc::UIDs, [1188](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_22  
 gdc::UIDs, [1188](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_23  
 gdc::UIDs, [1189](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_24  
 gdc::UIDs, [1189](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_24\_1  
 gdc::UIDs, [1189](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_25  
 gdc::UIDs, [1189](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_26  
 gdc::UIDs, [1189](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_27  
 gdc::UIDs, [1189](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_29  
 gdc::UIDs, [1189](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_30  
 gdc::UIDs, [1189](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_31  
 gdc::UIDs, [1189](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_32  
 gdc::UIDs, [1189](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_33  
 gdc::UIDs, [1189](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_4  
 gdc::UIDs, [1188](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_40  
 gdc::UIDs, [1193](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_40\_1  
 gdc::UIDs, [1193](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_4\_1  
 gdc::UIDs, [1188](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_4\_2  
 gdc::UIDs, [1188](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_9  
 gdc::UIDs, [1188](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_9\_1  
 gdc::UIDs, [1188](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1

gdc::UIDs, [1189](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_10  
 gdc::UIDs, [1189](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_104\_1  
 gdc::UIDs, [1191](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_104\_2  
 gdc::UIDs, [1191](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_104\_3  
 gdc::UIDs, [1194](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11  
 gdc::UIDs, [1189](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_1  
 gdc::UIDs, [1189](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_10  
 gdc::UIDs, [1193](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_11  
 gdc::UIDs, [1193](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_2  
 gdc::UIDs, [1189](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_3  
 gdc::UIDs, [1190](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_4  
 gdc::UIDs, [1190](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_5  
 gdc::UIDs, [1193](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_6  
 gdc::UIDs, [1193](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_7  
 gdc::UIDs, [1193](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_8  
 gdc::UIDs, [1193](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_9  
 gdc::UIDs, [1193](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_128  
 gdc::UIDs, [1191](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_128\_1  
 gdc::UIDs, [1192](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_129  
 gdc::UIDs, [1191](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_1  
 gdc::UIDs, [1190](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_1\_1  
 gdc::UIDs, [1190](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_2  
 gdc::UIDs, [1190](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_2\_1  
 gdc::UIDs, [1190](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_3  
 gdc::UIDs, [1190](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_77  
 gdc::UIDs, [1193](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_130  
 gdc::UIDs, [1194](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_131



gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_1  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_2  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_3  
gdcm::UIDs, [1192](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_4  
gdcm::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_5  
gdcm::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_14\_1  
gdcm::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_14\_2  
gdcm::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_1  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_1\_1  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_2  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_2\_1  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_3  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_3\_1  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_2  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_20  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_1  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_2  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_3  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_4  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_5  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_6  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_2\_1  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_2\_2  
gdcm::UIDs, [1192](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_3  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_30  
gdcm::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_3\_1  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4

gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_40  
gdcm::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_1  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_10  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_11  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_2  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_3  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_4  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_5  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_6  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_7  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_8  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_9  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4\_1  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4\_2  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4\_3  
gdcm::UIDs, [1195](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4\_4  
gdcm::UIDs, [1192](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_5  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_1  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_2\_1  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_2\_2  
gdcm::UIDs, [1195](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_3  
gdcm::UIDs, [1195](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_4  
gdcm::UIDs, [1195](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_5  
gdcm::UIDs, [1195](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_6  
gdcm::UIDs, [1195](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_6  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_601\_1  
gdcm::UIDs, [1195](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_601\_2

gdcm::UIDs, [1195](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_1  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_2  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_3  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_4  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_5  
gdcm::UIDs, [1192](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_6  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_67  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_68\_1  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_68\_2  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_6\_1  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_6\_2  
gdcm::UIDs, [1192](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_7  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_1  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_1\_1  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_2  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_2\_1  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_3  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_4  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_4\_1  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_1  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_2  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_3  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_4  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_5  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_6

gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_7  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_8  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_6  
gdcm::UIDs, [1192](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_2  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_78\_1  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_78\_2  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_78\_3  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_78\_4  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_78\_5  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_78\_6  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_78\_7  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_78\_8  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_79\_1  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_7\_1  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_7\_2  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_7\_3  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_7\_4  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_8  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_80\_1  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_81\_1  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_82\_1  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_1  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_11  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_2  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_22  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_3  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_33

gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_34  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_35  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_4  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_40  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_50  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_59  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_65  
gdcm::UIDs, [1190](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_67  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_68  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_69  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_70  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_71  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_72  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_73  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_74  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_75  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_90\_1  
gdcm::UIDs, [1194](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_1  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_1\_1  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_1\_2  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_1\_3  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_2\_1  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_3\_1  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_4\_1  
gdcm::UIDs, [1189](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_4\_2  
gdcm::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_5\_1  
gdcm::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_6\_1  
gdcm::UIDs, [1193](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_1\_1  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_1\_2  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_1\_3  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_2\_1  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_2\_2  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_2\_3  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_3\_1  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_3\_2  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_3\_3  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_4\_2  
gdcm::UIDs, [1195](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_4\_3  
gdcm::UIDs, [1195](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_5\_3  
gdcm::UIDs, [1195](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_20\_1  
gdcm::UIDs, [1195](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_20\_2  
gdcm::UIDs, [1195](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_20\_3  
gdcm::UIDs, [1195](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_31  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_32  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_32\_1  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_32\_2  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_32\_3  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_33  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_1  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_10  
gdcm::UIDs, [1195](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_2  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_3  
gdcm::UIDs, [1191](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4

gdcmm::UIDs, [1191](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_1  
   gdcmm::UIDs, [1191](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_2  
   gdcmm::UIDs, [1191](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_3  
   gdcmm::UIDs, [1191](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_4  
   gdcmm::UIDs, [1191](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_34\_5  
   gdcmm::UIDs, [1191](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_34\_5\_1  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_34\_6  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_34\_6\_1  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_34\_6\_2  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_34\_6\_3  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_34\_6\_4  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_34\_7  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_34\_8  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_34\_9  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_37\_1  
   gdcmm::UIDs, [1191](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_37\_2  
   gdcmm::UIDs, [1191](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_37\_3  
   gdcmm::UIDs, [1191](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_38\_1  
   gdcmm::UIDs, [1191](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_38\_2  
   gdcmm::UIDs, [1191](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_38\_3  
   gdcmm::UIDs, [1191](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_38\_4  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_39\_1  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_39\_2  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_39\_3  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_39\_4  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_41  
   gdcmm::UIDs, [1191](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_42

gdcmm::UIDs, [1191](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_43\_1  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_43\_2  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_43\_3  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_43\_4  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_44\_1  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_44\_2  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_44\_3  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_44\_4  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_45\_1  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_45\_2  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_45\_3  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_45\_4  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_7\_1\_1  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_7\_1\_2  
   gdcmm::UIDs, [1195](#)  
 uid\_1\_2\_840\_10008\_8\_1\_1  
   gdcmm::UIDs, [1195](#)  
 UIDGenerator  
   gdcmm::UIDGenerator, [1159](#)  
 UIH  
   gdcmm::EquipmentManufacturer, [431](#)  
 UINT12  
   gdcmm::PixelFormat, [822](#)  
 UINT16  
   gdcmm::PixelFormat, [822](#)  
 UINT32  
   gdcmm::PixelFormat, [822](#)  
 UINT64  
   gdcmm::PixelFormat, [822](#)  
 UINT8  
   gdcmm::PixelFormat, [822](#)  
 UL  
   gdcmm::VR, [1297](#)  
 ULAction  
   gdcmm::network::ULAction, [1199](#)  
 ULActionAE6  
   gdcmm::network::ULConnection, [1244](#)  
 ULBasicCallback  
   gdcmm::network::ULBasicCallback, [1239](#)  
 ULConnection

- gdcm::network::ULConnection, [1241](#), [1242](#)
- ULConnectionCallback
  - gdcm::network::ULConnectionCallback, [1246](#)
- ULConnectionInfo
  - gdcm::network::ULConnectionInfo, [1248](#)
- ULConnectionManager
  - gdcm::network::ULConnection, [1244](#)
  - gdcm::network::ULConnectionManager, [1251](#)
- ULEvent
  - gdcm::network::ULEvent, [1257](#)
- ULTransitionTable
  - gdcm::network::ULTransitionTable, [1258](#)
- UltrasoundImageStorage
  - gdcm::MediaStorage, [693](#)
  - gdcm::UIDs, [1180](#)
- UltrasoundImageStorageRetired
  - gdcm::MediaStorage, [693](#)
  - gdcm::UIDs, [1180](#)
- UltrasoundMultiFramedImageStorage
  - gdcm::MediaStorage, [693](#)
- UltrasoundMultiframeImageStorage
  - gdcm::UIDs, [1180](#)
- UltrasoundMultiFramedImageStorageRetired
  - gdcm::MediaStorage, [693](#)
- UltrasoundMultiframeImageStorageRetired
  - gdcm::UIDs, [1179](#)
- ULWritingCallback
  - gdcm::network::ULWritingCallback, [1261](#)
- UN
  - gdcm::VR, [1297](#)
- UndefinedEntityError
  - gdcm::Parser, [792](#)
- underline
  - gdcm::terminal, [82](#)
- UnexpectedStateError
  - gdcm::Parser, [792](#)
- UnifiedProcedureStepEventSOPClass
  - gdcm::UIDs, [1182](#)
- UnifiedProcedureStepEventSOPClass1
  - gdcm::UIDs, [1186](#)
- UnifiedProcedureStepPullSOPClass
  - gdcm::UIDs, [1182](#)
- UnifiedProcedureStepPullSOPClass1
  - gdcm::UIDs, [1186](#)
- UnifiedProcedureStepPushSOPClass
  - gdcm::UIDs, [1182](#)
- UnifiedProcedureStepPushSOPClass1
  - gdcm::UIDs, [1186](#)
- UnifiedProcedureStepWatchSOPClass
  - gdcm::UIDs, [1182](#)
- UnifiedProcedureStepWatchSOPClass1
  - gdcm::UIDs, [1186](#)
- UnifiedWorklistandProcedureStepServiceClass
  - gdcm::UIDs, [1182](#)
- UnifiedWorklistandProcedureStepServiceClass1
  - gdcm::UIDs, [1186](#)
- UnifiedWorklistandProcedureStepSOPInstance
  - gdcm::UIDs, [1182](#)
- UnInstallPipeline
  - vtkImageColorViewer, [1375](#)
- UniversalCoordinatedTime
  - gdcm::UIDs, [1186](#)
- UNKNOWN
  - gdcm::CSAHeader, [290](#)
  - gdcm::EquipmentManufacturer, [431](#)
  - gdcm::LookupTable, [674](#)
  - gdcm::Orientation, [777](#)
  - gdcm::PhotometricInterpretation, [817](#)
  - gdcm::PixelFormat, [822](#)
  - gdcm::Spacing, [1030](#)
  - gdcm::Surface, [1084](#)
  - gdcm::Type, [1157](#)
- Unknown
  - gdcm::SwapCode, [1105](#)
  - gdcm::TransferSyntax, [1148](#)
- Unpack
  - gdcm::Unpacker12Bits, [1270](#)
- UnRegister
  - gdcm::Object, [765](#)
- UnusedBitsPresentInPixelData
  - gdcm::Bitmap, [204](#)
  - gdcm::Pixmap, [834](#)
- Update
  - gdcm::Curve, [307](#)
  - gdcm::Overlay, [788](#)
- UpdateDisplayExtent
  - vtkImageColorViewer, [1375](#)
- UpdateOrientation
  - vtkImageColorViewer, [1375](#)
- UpdatePosition
  - gdcm::ByteBuffer, [214](#)
- UPSFilteredGlobalSubscriptionSOPInstance
  - gdcm::UIDs, [1185](#)
- UR
  - gdcm::VR, [1297](#)
- URComp
  - gdcm, [60](#)
- URI
  - gdcm::MediaStorage, [695](#)
- US
  - gdcm::VR, [1297](#)
- US\_OW
  - gdcm::VR, [1297](#)
- US\_SS
  - gdcm::VR, [1297](#)
- US\_SS\_OW
  - gdcm::VR, [1297](#)
- Usage

- gdcmm::Usage, 1272
- UsageType
  - gdcmm::Usage, 1271
- UseDictAlways
  - gdcmm::PythonFilter, 892
  - gdcmm::StringFilter, 1076
- UseGrayscaleSecondaryImageStorage
  - gdcmm::EmptyMaskGenerator, 423
- UseOriginalSOPClassUID
  - gdcmm::EmptyMaskGenerator, 423
- UserInformation
  - gdcmm::network::UserInformation, 1275
- UserOption
  - gdcmm::Usage, 1271
- UserOrdering
  - gdcmm::SerieHelper, 994
- UT
  - gdcmm::VR, 1297
- UTComp
  - gdcmm, 60
- UV
  - gdcmm::VR, 1297
- V
  - gdcmm::Validate, 1279
- Valid
  - gdcmm::Preamble, 854
- Validate
  - gdcmm::PixelFormat, 827
  - gdcmm::Validate, 1278
- ValidateQuery
  - gdcmm::BaseQuery, 179
  - gdcmm::BaseRootQuery, 184
  - gdcmm::FindPatientRootQuery, 505
  - gdcmm::FindStudyRootQuery, 509
  - gdcmm::ModalityPerformedProcedureStepCreateQuery, 713
  - gdcmm::ModalityPerformedProcedureStepSetQuery, 716
  - gdcmm::MovePatientRootQuery, 730
  - gdcmm::MoveStudyRootQuery, 734
  - gdcmm::WLMFindQuery, 1410
- Validation
  - gdcmm::Validate, 1278
- ValidDataSet
  - gdcmm::BaseQuery, 179
- Value
  - gdcmm::Value, 1280
- value
  - gdcmm::SerieHelper, 995
  - gdcmm::STATIC\_ASSERTION\_FAILURE< true >, 1037
- value\_type
  - gdcmm::CodeString, 255
- gdcmm::LO, 670
- gdcmm::String< TDelimiter, TMaxLength, TPadChar >, 1070
- ValueField
  - gdcmm::DataElement, 320
  - gdcmm::PDBelement, 800
- ValueLengthField
  - gdcmm::DataElement, 320
- ValueMultiplicityField
  - gdcmm::CSAElement, 287
- ValuePtr
  - gdcmm::DataElement, 311
- ValueType
  - gdcmm::Scanner, 938
  - gdcmm::Scanner2, 947
  - gdcmm::StrictScanner, 1052
  - gdcmm::StrictScanner2, 1061
- VERBOSE\_STYLE
  - gdcmm::Printer, 873
- VerificationSOPClass
  - gdcmm::UIDs, 1177
- Verify
  - gdcmm::Defs, 347
  - gdcmm::Macro, 683
  - gdcmm::Module, 721
- Version
  - gdcmm::Version, 1284
- VERTEX
  - gdcmm::MeshPrimitive, 707
- Video
  - gdcmm::MediaStorage, 695
- VideoEndoscopicImageStorage
  - gdcmm::MediaStorage, 694
  - gdcmm::UIDs, 1181
- VideoMicroscopicImageStorage
  - gdcmm::MediaStorage, 695
  - gdcmm::UIDs, 1181
- VideoPhotographicImageStorage
  - gdcmm::MediaStorage, 694
  - gdcmm::UIDs, 1181
- VIEWType
  - gdcmm::Surface, 1084
- VIEWType\_END
  - gdcmm::Surface, 1084
- VisualAcuityMeasurementsStorage
  - gdcmm::UIDs, 1184
- VL
  - gdcmm::VL, 1287
- VL16
  - gdcmm::VR, 1297
- VL32
  - gdcmm::VR, 1297
- VLEndoscopicImageStorage
  - gdcmm::MediaStorage, 694

- gdcM::UIDs, [1181](#)
- VImageStorageTrialRetired
  - gdcM::UIDs, [1180](#)
- VLMicroscopicImageStorage
  - gdcM::MediaStorage, [694](#)
  - gdcM::UIDs, [1181](#)
- VMultiframeImageStorageTrialRetired
  - gdcM::UIDs, [1180](#)
- VPhotographicImageStorage
  - gdcM::MediaStorage, [694](#)
  - gdcM::UIDs, [1181](#)
- VSlideCoordinatesMicroscopicImageStorage
  - gdcM::UIDs, [1181](#)
- VWholeSlideMicroscopyImageStorage
  - gdcM::MediaStorage, [694](#)
  - gdcM::UIDs, [1183](#)
- VM
  - gdcM::VM, [1292](#)
- VM0
  - gdcM::VM, [1291](#)
- VM1
  - gdcM::VM, [1291](#)
- VM10
  - gdcM::VM, [1291](#)
- VM12
  - gdcM::VM, [1291](#)
- VM16
  - gdcM::VM, [1291](#)
- VM18
  - gdcM::VM, [1291](#)
- VM1\_2
  - gdcM::VM, [1291](#)
- VM1\_3
  - gdcM::VM, [1291](#)
- VM1\_32
  - gdcM::VM, [1291](#)
- VM1\_4
  - gdcM::VM, [1291](#)
- VM1\_5
  - gdcM::VM, [1291](#)
- VM1\_8
  - gdcM::VM, [1291](#)
- VM1\_99
  - gdcM::VM, [1291](#)
- VM1\_n
  - gdcM::VM, [1291](#)
- VM2
  - gdcM::VM, [1291](#)
- VM24
  - gdcM::VM, [1291](#)
- VM256
  - gdcM::VM, [1291](#)
- VM28
  - gdcM::VM, [1291](#)
- VM2\_2n
  - gdcM::VM, [1291](#)
- VM2\_n
  - gdcM::VM, [1292](#)
- VM3
  - gdcM::VM, [1291](#)
- VM30\_30n
  - gdcM::VM, [1292](#)
- VM32
  - gdcM::VM, [1291](#)
- VM35
  - gdcM::VM, [1291](#)
- VM3\_3n
  - gdcM::VM, [1292](#)
- VM3\_4
  - gdcM::VM, [1292](#)
- VM3\_n
  - gdcM::VM, [1292](#)
- VM4
  - gdcM::VM, [1291](#)
- VM47\_47n
  - gdcM::VM, [1292](#)
- VM4\_4n
  - gdcM::VM, [1292](#)
- VM5
  - gdcM::VM, [1291](#)
- VM6
  - gdcM::VM, [1291](#)
- VM6\_6n
  - gdcM::VM, [1292](#)
- VM6\_n
  - gdcM::VM, [1292](#)
- VM7\_7n
  - gdcM::VM, [1292](#)
- VM8
  - gdcM::VM, [1291](#)
- VM9
  - gdcM::VM, [1291](#)
- VM99
  - gdcM::VM, [1291](#)
- VM\_END
  - gdcM::VM, [1292](#)
- VMType
  - gdcM::Attribute< Group, Element, TVR, TVM >, [132](#)
  - gdcM::Attribute< Group, Element, TVR, VM::VM1 >, [139](#)
  - gdcM::VM, [1291](#)
- VOILUTBoxSOPClass
  - gdcM::UIDs, [1179](#)
- VolumeRenderingVolumetricPresentationStateStorage
  - gdcM::UIDs, [1184](#)
- VR
  - gdcM::VR, [1297](#)
- VR\_END

- gdcmm::VR, [1297](#)
- VR\_VM1
  - gdcmm::VR, [1297](#)
- VRALL
  - gdcmm::VR, [1297](#)
- VRASCI
  - gdcmm::VR, [1297](#)
- VRBINARY
  - gdcmm::VR, [1297](#)
- VRField
  - gdcmm::CSAElement, [287](#)
  - gdcmm::DataElement, [320](#)
- VRType
  - gdcmm::VR, [1296](#)
- VRTypeTemplateCase
  - gdcmmVR.h, [1673](#)
- VT100
  - gdcmm::terminal, [82](#)
- VTK\_CMYK
  - vtkGDCMImageReader.h, [2051](#)
  - vtkGDCMImageReader2.h, [2057](#)
- VTK\_INVERSE\_LUMINANCE
  - vtkGDCMImageReader.h, [2051](#)
  - vtkGDCMImageReader2.h, [2057](#)
- VTK\_LOOKUP\_TABLE
  - vtkGDCMImageReader.h, [2052](#)
  - vtkGDCMImageReader2.h, [2057](#)
- VTK\_YBR
  - vtkGDCMImageReader.h, [2052](#)
  - vtkGDCMImageReader2.h, [2057](#)
- vtkBooleanMacro
  - vtkGDCMImageReader, [1312](#), [1313](#)
  - vtkGDCMImageReader2, [1325](#)
  - vtkGDCMImageWriter, [1336](#)
  - vtkGDCMThreadedImageReader, [1358](#)
  - vtkGDCMThreadedImageReader2, [1362](#), [1363](#)
  - vtkImageColorViewer, [1375](#)
  - vtkImageMapToColors16, [1381](#)
- vtkGDCMImageReader, [1307](#)
  - ~vtkGDCMImageReader, [1310](#)
  - ApplyInverseVideo, [1317](#)
  - ApplyLookupTable, [1317](#)
  - ApplyPlanarConfiguration, [1317](#)
  - ApplyShiftScale, [1317](#)
  - ApplyYBRToRGB, [1317](#)
  - CanReadFile, [1310](#)
  - Curve, [1317](#)
  - DirectionCosines, [1317](#)
  - ExecuteData, [1310](#)
  - ExecuteInformation, [1310](#)
  - FileNames, [1318](#)
  - FillMedicalImageInformation, [1310](#)
  - ForceRescale, [1318](#)
  - GetDescriptiveName, [1310](#)
  - GetFileExtensions, [1311](#)
  - GetIconImage, [1311](#)
  - GetOverlay, [1311](#)
  - IconDataScalarType, [1318](#)
  - IconImageDataExtent, [1318](#)
  - IconNumberOfScalarComponents, [1318](#)
  - ImageFormat, [1318](#)
  - ImageOrientationPatient, [1318](#)
  - ImagePositionPatient, [1318](#)
  - LoadIconImage, [1318](#)
  - LoadOverlays, [1318](#)
  - LoadSingleFile, [1311](#)
  - LossyFlag, [1319](#)
  - MedicalImageProperties, [1319](#)
  - New, [1311](#)
  - NumberOfIconImages, [1319](#)
  - NumberOfOverlays, [1319](#)
  - PlanarConfiguration, [1319](#)
  - PrintSelf, [1311](#)
  - RequestDataCompat, [1311](#)
  - RequestInformationCompat, [1312](#)
  - Scale, [1319](#)
  - SetCurve, [1312](#)
  - SetFileNames, [1312](#)
  - SetFilePattern, [1312](#)
  - SetFilePrefix, [1312](#)
  - SetMedicalImageProperties, [1312](#)
  - Shift, [1319](#)
  - vtkBooleanMacro, [1312](#), [1313](#)
  - vtkGDCMImageReader, [1310](#)
  - vtkGDCMMedicalImageProperties, [1342](#)
  - vtkGetMacro, [1313](#)–[1315](#)
  - vtkGetObjectMacro, [1315](#)
  - vtkGetStringMacro, [1315](#)
  - vtkGetVector3Macro, [1316](#)
  - vtkGetVector6Macro, [1316](#)
  - vtkSetMacro, [1316](#)
  - vtkSetVector6Macro, [1316](#)
  - vtkTypeMacro, [1317](#)
- vtkGDCMImageReader.h, [2050](#), [2052](#)
  - VTK\_CMYK, [2051](#)
  - VTK\_INVERSE\_LUMINANCE, [2051](#)
  - VTK\_LOOKUP\_TABLE, [2052](#)
  - VTK\_YBR, [2052](#)
- vtkGDCMImageReader2, [1320](#)
  - ~vtkGDCMImageReader2, [1322](#)
  - ApplyInverseVideo, [1329](#)
  - ApplyLookupTable, [1329](#)
  - ApplyPlanarConfiguration, [1329](#)
  - ApplyShiftScale, [1329](#)
  - ApplyYBRToRGB, [1329](#)
  - CanReadFile, [1322](#)
  - Curve, [1329](#)
  - DirectionCosines, [1330](#)



- FillMedicalImageInformation, [1322](#)
- ForceRescale, [1330](#)
- GetDescriptiveName, [1323](#)
- GetFileExtensions, [1323](#)
- GetIconImage, [1323](#)
- GetIconImagePort, [1323](#)
- GetOverlay, [1323](#)
- GetOverlayPort, [1323](#)
- IconDataScalarType, [1330](#)
- IconImageDataExtent, [1330](#)
- IconNumberOfScalarComponents, [1330](#)
- ImageFormat, [1330](#)
- ImageOrientationPatient, [1330](#)
- ImagePositionPatient, [1330](#)
- LoadIconImage, [1330](#)
- LoadOverlays, [1330](#)
- LoadSingleFile, [1323](#)
- LossyFlag, [1331](#)
- New, [1323](#)
- NumberOfIconImages, [1331](#)
- NumberOfOverlays, [1331](#)
- PlanarConfiguration, [1331](#)
- PrintSelf, [1324](#)
- ProcessRequest, [1324](#)
- RequestData, [1324](#)
- RequestDataCompat, [1324](#)
- RequestInformation, [1324](#)
- RequestInformationCompat, [1324](#)
- Scale, [1331](#)
- SetCurve, [1324](#)
- SetFilePattern, [1325](#)
- SetFilePrefix, [1325](#)
- SetMedicalImageProperties, [1325](#)
- Shift, [1331](#)
- vtkBooleanMacro, [1325](#)
- vtkGDCMImageReader2, [1322](#)
- vtkGDCMMedicalImageProperties, [1342](#)
- vtkGetMacro, [1326](#), [1327](#)
- vtkGetObjectMacro, [1327](#)
- vtkGetStringMacro, [1327](#), [1328](#)
- vtkGetVector3Macro, [1328](#)
- vtkGetVector6Macro, [1328](#)
- vtkSetMacro, [1328](#)
- vtkSetVector6Macro, [1329](#)
- vtkTypeMacro, [1329](#)
- vtkGDCMImageReader2.h, [2056](#), [2057](#)
  - VTK\_CMYK, [2057](#)
  - VTK\_INVERSE\_LUMINANCE, [2057](#)
  - VTK\_LOOKUP\_TABLE, [2057](#)
  - VTK\_YBR, [2057](#)
- vtkGDCMImageWriter, [1332](#)
  - ~vtkGDCMImageWriter, [1334](#)
  - CompressionTypes, [1334](#)
  - GetDescriptiveName, [1334](#)
  - GetFileExtensions, [1334](#)
  - GetFileName, [1334](#)
  - JPEG2000\_COMPRESSION, [1334](#)
  - JPEG\_COMPRESSION, [1334](#)
  - JPEGLS\_COMPRESSION, [1334](#)
  - New, [1335](#)
  - NO\_COMPRESSION, [1334](#)
  - PrintSelf, [1335](#)
  - RLE\_COMPRESSION, [1334](#)
  - SetDirectionCosines, [1335](#)
  - SetDirectionCosinesFromImageOrientationPatient, [1335](#)
  - SetFileNames, [1335](#)
  - SetMedicalImageProperties, [1335](#)
  - vtkBooleanMacro, [1336](#)
  - vtkGDCMImageWriter, [1334](#)
  - vtkGDCMMedicalImageProperties, [1342](#)
  - vtkGetMacro, [1336](#), [1337](#)
  - vtkGetObjectMacro, [1337](#)
  - vtkGetStringMacro, [1337](#), [1338](#)
  - vtkSetMacro, [1338](#), [1339](#)
  - vtkSetStringMacro, [1339](#)
  - vtkTypeMacro, [1339](#)
  - Write, [1339](#)
  - WriteGDCMData, [1339](#)
  - WriteSlice, [1339](#)
- vtkGDCMImageWriter.h, [2061](#)
- vtkGDCMMedicalImageProperties, [1340](#)
  - ~vtkGDCMMedicalImageProperties, [1341](#)
  - Clear, [1341](#)
  - GetFile, [1341](#)
  - New, [1341](#)
  - PrintSelf, [1342](#)
  - PushBackFile, [1342](#)
  - vtkGDCMImageReader, [1342](#)
  - vtkGDCMImageReader2, [1342](#)
  - vtkGDCMImageWriter, [1342](#)
  - vtkGDCMMedicalImageProperties, [1341](#)
  - vtkTypeMacro, [1342](#)
- vtkGDCMMedicalImageProperties.h, [2064](#)
- vtkGDCMPolyDataReader, [1343](#)
  - ~vtkGDCMPolyDataReader, [1344](#)
  - FileName, [1346](#)
  - FillMedicalImageInformation, [1344](#)
  - MedicalImageProperties, [1346](#)
  - New, [1344](#)
  - PrintSelf, [1345](#)
  - RequestData, [1345](#)
  - RequestData\_HemodynamicWaveformStorage, [1345](#)
  - RequestData\_RTStructureSetStorage, [1345](#)
  - RequestInformation, [1345](#)
  - RequestInformation\_HemodynamicWaveformStorage, [1345](#)

- RequestInformation\_RTStructureSetStorage, 1345
- RTStructSetProperties, 1347
- vtkGDCMPolyDataReader, 1344
- vtkGetObjectMacro, 1346
- vtkGetStringMacro, 1346
- vtkSetStringMacro, 1346
- vtkTypeMacro, 1346
- vtkGDCMPolyDataReader.h, 2069, 2070
- vtkGDCMPolyDataWriter, 1347
  - ~vtkGDCMPolyDataWriter, 1348
  - InitializeRTStructSet, 1349
  - MedicalImageProperties, 1350
  - New, 1349
  - PrintSelf, 1349
  - RTStructSetProperties, 1350
  - SetMedicalImageProperties, 1349
  - SetNumberOfInputPorts, 1349
  - SetRTStructSetProperties, 1349
  - vtkGDCMPolyDataWriter, 1348
  - vtkTypeMacro, 1350
  - WriteData, 1350
  - WriteRTSTRUCTData, 1350
  - WriteRTSTRUCTInfo, 1350
- vtkGDCMPolyDataWriter.h, 2071, 2072
- vtkGDCMTesting, 1351
  - ~vtkGDCMTesting, 1352
  - GetGDCMDataRoot, 1352
  - GetMD5MetalImage, 1352
  - GetMHDMD5FromFile, 1353
  - GetNumberOfMD5MetalImages, 1353
  - GetRAWMD5FromFile, 1353
  - GetVTKDataRoot, 1353
  - MD5MetalImagesType, 1352
  - New, 1353
  - PrintSelf, 1353
  - vtkGDCMTesting, 1352
  - vtkTypeMacro, 1354
- vtkGDCMTesting.h, 2073
- vtkGDCMThreadedImageReader, 1354
  - ~vtkGDCMThreadedImageReader, 1357
  - ExecuteData, 1357
  - ExecuteInformation, 1357
  - New, 1358
  - PrintSelf, 1358
  - ReadFiles, 1358
  - RequestDataCompat, 1358
  - vtkBooleanMacro, 1358
  - vtkGDCMThreadedImageReader, 1357
  - vtkGetMacro, 1358
  - vtkSetMacro, 1358, 1359
  - vtkTypeMacro, 1359
- vtkGDCMThreadedImageReader.h, 2074, 2075
- vtkGDCMThreadedImageReader2, 1359
  - ~vtkGDCMThreadedImageReader2, 1361
- GetFileName, 1361
- New, 1361
- PrintSelf, 1361
- RequestInformation, 1361
- SetFileName, 1362
- SetFileNames, 1362
- SplitExtent, 1362
- ThreadedRequestData, 1362
- vtkBooleanMacro, 1362, 1363
- vtkGDCMThreadedImageReader2, 1361
- vtkGetMacro, 1363, 1364
- vtkGetObjectMacro, 1364
- vtkGetVector3Macro, 1364
- vtkGetVector6Macro, 1364
- vtkSetMacro, 1364, 1365
- vtkSetVector3Macro, 1365, 1366
- vtkSetVector6Macro, 1366
- vtkTypeMacro, 1366
- vtkGDCMThreadedImageReader2.h, 2076
- vtkGetMacro
  - vtkGDCMImageReader, 1313–1315
  - vtkGDCMImageReader2, 1326, 1327
  - vtkGDCMImageWriter, 1336, 1337
  - vtkGDCMThreadedImageReader, 1358
  - vtkGDCMThreadedImageReader2, 1363, 1364
  - vtkImageColorViewer, 1375, 1376
  - vtkImageMapToColors16, 1382
  - vtkImageMapToWindowLevelColors2, 1386
- vtkGetObjectMacro
  - vtkGDCMImageReader, 1315
  - vtkGDCMImageReader2, 1327
  - vtkGDCMImageWriter, 1337
  - vtkGDCMPolyDataReader, 1346
  - vtkGDCMThreadedImageReader2, 1364
  - vtkImageColorViewer, 1376
  - vtkImageMapToColors16, 1382
- vtkGetStringMacro
  - vtkGDCMImageReader, 1315
  - vtkGDCMImageReader2, 1327, 1328
  - vtkGDCMImageWriter, 1337, 1338
  - vtkGDCMPolyDataReader, 1346
  - vtkRTStructSetProperties, 1402, 1403
- vtkGetVector3Macro
  - vtkGDCMImageReader, 1316
  - vtkGDCMImageReader2, 1328
  - vtkGDCMThreadedImageReader2, 1364
- vtkGetVector6Macro
  - vtkGDCMImageReader, 1316
  - vtkGDCMImageReader2, 1328
  - vtkGDCMThreadedImageReader2, 1364
- vtkImageColorViewer, 1366
  - ~vtkImageColorViewer, 1369
  - AddInput, 1370
  - AddInputConnection, 1370

FirstRender, [1377](#)  
GetColorLevel, [1370](#)  
GetColorWindow, [1370](#)  
GetInput, [1370](#)  
GetOffScreenRendering, [1370](#)  
GetOverlayVisibility, [1370](#)  
GetPosition, [1370](#)  
GetSize, [1370](#)  
GetSliceMax, [1370](#)  
GetSliceMin, [1371](#)  
GetSliceRange, [1371](#)  
GetWindowName, [1371](#)  
ImageActor, [1377](#)  
InstallPipeline, [1371](#)  
Interactor, [1377](#)  
InteractorStyle, [1377](#)  
New, [1371](#)  
OverlayImageActor, [1377](#)  
PrintSelf, [1371](#)  
Render, [1372](#)  
Renderer, [1377](#)  
RenderWindow, [1377](#)  
SetColorLevel, [1372](#)  
SetColorWindow, [1372](#)  
SetDisplayId, [1372](#)  
SetInput, [1372](#)  
SetInputConnection, [1372](#)  
SetOffScreenRendering, [1372](#)  
SetOverlayVisibility, [1373](#)  
SetParentId, [1373](#)  
SetPosition, [1373](#)  
SetRenderer, [1373](#)  
SetRenderWindow, [1373](#)  
SetSize, [1373](#), [1374](#)  
SetSlice, [1374](#)  
SetSliceOrientation, [1374](#)  
SetSliceOrientationToXY, [1374](#)  
SetSliceOrientationToXZ, [1374](#)  
SetSliceOrientationToYZ, [1374](#)  
SetupInteractor, [1375](#)  
SetWindowId, [1375](#)  
Slice, [1378](#)  
SLICE\_ORIENTATION\_XY, [1369](#)  
SLICE\_ORIENTATION\_XZ, [1369](#)  
SLICE\_ORIENTATION\_YZ, [1369](#)  
SliceOrientation, [1378](#)  
UnInstallPipeline, [1375](#)  
UpdateDisplayExtent, [1375](#)  
UpdateOrientation, [1375](#)  
vtkBooleanMacro, [1375](#)  
vtkGetMacro, [1375](#), [1376](#)  
vtkGetObjectMacro, [1376](#)  
vtkImageColorViewer, [1369](#)  
vtkImageColorViewerCallback, [1377](#)  
vtkTypeMacro, [1376](#)  
WindowLevel, [1378](#)  
vtkImageColorViewer.h, [2078](#), [2079](#)  
vtkImageColorViewerCallback  
  vtkImageColorViewer, [1377](#)  
vtkImageMapToColors16, [1378](#)  
  ~vtkImageMapToColors16, [1380](#)  
  ActiveComponent, [1383](#)  
  DataWasPassed, [1383](#)  
  GetMTime, [1380](#)  
  LookupTable, [1383](#)  
  New, [1380](#)  
  OutputFormat, [1383](#)  
  PassAlphaToOutput, [1383](#)  
  PrintSelf, [1380](#)  
  RequestData, [1380](#)  
  RequestInformation, [1380](#)  
  SetLookupTable, [1381](#)  
  SetOutputFormatToLuminance, [1381](#)  
  SetOutputFormatToLuminanceAlpha, [1381](#)  
  SetOutputFormatToRGB, [1381](#)  
  SetOutputFormatToRGBA, [1381](#)  
  ThreadedRequestData, [1381](#)  
  vtkBooleanMacro, [1381](#)  
  vtkGetMacro, [1382](#)  
  vtkGetObjectMacro, [1382](#)  
  vtkImageMapToColors16, [1380](#)  
  vtkSetMacro, [1382](#)  
  vtkTypeMacro, [1383](#)  
vtkImageMapToColors16.h, [2082](#), [2083](#)  
vtkImageMapToWindowLevelColors2, [1384](#)  
  ~vtkImageMapToWindowLevelColors2, [1385](#)  
  Level, [1387](#)  
  New, [1385](#)  
  PrintSelf, [1385](#)  
  RequestData, [1385](#)  
  RequestInformation, [1385](#)  
  ThreadedRequestData, [1386](#)  
  vtkGetMacro, [1386](#)  
  vtkImageMapToWindowLevelColors2, [1385](#)  
  vtkSetMacro, [1386](#)  
  vtkTypeMacro, [1386](#)  
  Window, [1387](#)  
vtkImageMapToWindowLevelColors2.h, [2084](#), [2085](#)  
vtkImagePlanarComponentsToComponents, [1387](#)  
  ~vtkImagePlanarComponentsToComponents, [1388](#)  
  New, [1389](#)  
  PrintSelf, [1389](#)  
  RequestData, [1389](#)  
  vtkImagePlanarComponentsToComponents, [1388](#)  
  vtkTypeMacro, [1389](#)  
vtkImagePlanarComponentsToComponents.h, [2086](#)  
vtkImageRGBToYBR, [1389](#)  
  ~vtkImageRGBToYBR, [1390](#)

- New, [1391](#)
- PrintSelf, [1391](#)
- ThreadedExecute, [1391](#)
- vtkImageRGBToYBR, [1390](#)
- vtkTypeMacro, [1391](#)
- vtkImageRGBToYBR.h, [2088](#)
- vtkImageYBRToRGB, [1391](#)
  - ~vtkImageYBRToRGB, [1392](#)
  - New, [1393](#)
  - PrintSelf, [1393](#)
  - ThreadedExecute, [1393](#)
  - vtkImageYBRToRGB, [1392](#)
  - vtkTypeMacro, [1393](#)
- vtkImageYBRToRGB.h, [2089](#), [2090](#)
- vtkLookupTable16, [1393](#)
  - ~vtkLookupTable16, [1394](#)
  - Build, [1395](#)
  - GetPointer, [1395](#)
  - MapScalarsThroughTable2, [1395](#)
  - New, [1395](#)
  - PrintSelf, [1395](#)
  - SetNumberOfTableValues, [1395](#)
  - Table16, [1396](#)
  - vtkLookupTable16, [1394](#)
  - vtkTypeMacro, [1395](#)
  - WritePointer, [1396](#)
- vtkLookupTable16.h, [2090](#), [2091](#)
- vtkRTStructSetProperties, [1396](#)
  - ~vtkRTStructSetProperties, [1398](#)
  - AddContourReferencedFrameOfReference, [1399](#)
  - AddReferencedFrameOfReference, [1399](#)
  - AddStructureSetROI, [1399](#)
  - AddStructureSetROIObservation, [1399](#)
  - Clear, [1399](#)
  - DeepCopy, [1399](#)
  - GetContourReferencedFrameOfReferenceClassUID, [1400](#)
  - GetContourReferencedFrameOfReferenceInstanceUID, [1400](#)
  - GetNumberOfContourReferencedFrameOfReferences, [1400](#)
  - GetNumberOfReferencedFrameOfReferences, [1400](#)
  - GetNumberOfStructureSetROIs, [1400](#)
  - GetReferencedFrameOfReferenceClassUID, [1400](#)
  - GetReferencedFrameOfReferenceInstanceUID, [1400](#)
  - GetStructureSetObservationNumber, [1401](#)
  - GetStructureSetROIDescription, [1401](#)
  - GetStructureSetROIGenerationAlgorithm, [1401](#)
  - GetStructureSetROIName, [1401](#)
  - GetStructureSetROINumber, [1401](#)
  - GetStructureSetROIObservationLabel, [1401](#)
  - GetStructureSetROIRefFrameRefUID, [1401](#)
  - GetStructureSetRTROIInterpretedType, [1401](#)
  - Internals, [1404](#)
  - New, [1402](#)
  - PrintSelf, [1402](#)
  - ReferenceFrameOfReferenceUID, [1404](#)
  - ReferenceSeriesInstanceUID, [1405](#)
  - SeriesInstanceUID, [1405](#)
  - SOPInstanceUID, [1405](#)
  - StructureSetDate, [1405](#)
  - StructureSetLabel, [1405](#)
  - StructureSetName, [1405](#)
  - StructureSetTime, [1405](#)
  - StudyInstanceUID, [1405](#)
  - vtkGetStringMacro, [1402](#), [1403](#)
  - vtkRTStructSetProperties, [1398](#)
  - vtkSetStringMacro, [1403](#), [1404](#)
  - vtkTypeMacro, [1404](#)
- vtkRTStructSetProperties.h, [2092](#), [2093](#)
- vtkSetMacro
  - vtkGDCMImageReader, [1316](#)
  - vtkGDCMImageReader2, [1328](#)
  - vtkGDCMImageWriter, [1338](#), [1339](#)
  - vtkGDCMThreadedImageReader, [1358](#), [1359](#)
  - vtkGDCMThreadedImageReader2, [1364](#), [1365](#)
  - vtkImageMapToColors16, [1382](#)
  - vtkImageMapToWindowLevelColors2, [1386](#)
- vtkSetStringMacro
  - vtkGDCMImageWriter, [1339](#)
  - vtkGDCMPolyDataReader, [1346](#)
  - vtkRTStructSetProperties, [1403](#), [1404](#)
- vtkSetVector3Macro
  - vtkGDCMThreadedImageReader2, [1365](#), [1366](#)
- vtkSetVector6Macro
  - vtkGDCMImageReader, [1316](#)
  - vtkGDCMImageReader2, [1329](#)
  - vtkGDCMThreadedImageReader2, [1366](#)
- vtkTypeMacro
  - vtkGDCMImageReader, [1317](#)
  - vtkGDCMImageReader2, [1329](#)
  - vtkGDCMImageWriter, [1339](#)
  - vtkGDCMMedicalImageProperties, [1342](#)
  - vtkGDCMPolyDataReader, [1346](#)
  - vtkGDCMPolyDataWriter, [1350](#)
  - vtkGDCMTesting, [1354](#)
  - vtkGDCMThreadedImageReader, [1359](#)
  - vtkGDCMThreadedImageReader2, [1366](#)
  - vtkImageColorViewer, [1376](#)
  - vtkImageMapToColors16, [1383](#)
  - vtkImageMapToWindowLevelColors2, [1386](#)
  - vtkImagePlanarComponentsToComponents, [1389](#)
  - vtkImageRGBToYBR, [1391](#)
  - vtkImageYBRToRGB, [1393](#)
  - vtkLookupTable16, [1395](#)
  - vtkRTStructSetProperties, [1404](#)
- WarningOff

- gdcmm::Trace, 1146
- WarningOn
  - gdcmm::Trace, 1146
- Waveform
  - gdcmm::MediaStorage, 695
  - gdcmm::Waveform, 1406
- WaveformStorageTrialRetired
  - gdcmm::UIDs, 1180
- WeirdPapryus
  - gdcmm::TransferSyntax, 1149
- what
  - gdcmm::Exception, 437
- white
  - gdcmm::terminal, 82
- WideFieldOphthalmicPhotography3DCoordinatesImageStorage
  - gdcmm::UIDs, 1184
- WideFieldOphthalmicPhotographyStereographicProjectionImageStorage
  - gdcmm::UIDs, 1184
- Window
  - vtkImageMapToWindowLevelColors2, 1387
- WindowLevel
  - vtkImageColorViewer, 1378
- WinterColorPaletteSOPInstance
  - gdcmm::UIDs, 1183
- WIREFRAME
  - gdcmm::Surface, 1084
- WLMFindQuery
  - gdcmm::WLMFindQuery, 1409
- Write
  - gdcmm::ByteValue, 225
  - gdcmm::CommandDataSet, 264
  - gdcmm::DataElement, 319
  - gdcmm::DataSet, 336
  - gdcmm::Element< TVR, TVM >, 396
  - gdcmm::Element< TVR, VM::VM1\_n >, 403
  - gdcmm::EncodingImplementation< VR::VRASCII >, 427
  - gdcmm::EncodingImplementation< VR::VRBINARY >, 428
  - gdcmm::ExplicitDataElement, 443
  - gdcmm::File, 452
  - gdcmm::FileAnonymizer, 457
  - gdcmm::FileMetaInformation, 480
  - gdcmm::Fragment, 514
  - gdcmm::ImageWriter, 594
  - gdcmm::ImplicitDataElement, 601
  - gdcmm::Item, 621
  - gdcmm::network::AAAbortPDU, 87
  - gdcmm::network::AAAssociateACPDU, 91
  - gdcmm::network::AAAssociateRJPDU, 93
  - gdcmm::network::AAAssociateRQPDU, 99
  - gdcmm::network::AbstractSyntax, 102
  - gdcmm::network::ApplicationContext, 117
  - gdcmm::network::AResponseRPPDU, 122
  - gdcmm::network::AResponseRQPDU, 124
  - gdcmm::network::AsynchronousOperationsWindowSub, 129
  - gdcmm::network::BasePDU, 175
  - gdcmm::network::ImplementationClassUIDSub, 595
  - gdcmm::network::ImplementationUIDSub, 596
  - gdcmm::network::ImplementationVersionNameSub, 597
  - gdcmm::network::MaximumLengthSub, 687
  - gdcmm::network::PDataTFPDU, 797
  - gdcmm::network::PresentationContextAC, 860
  - gdcmm::network::PresentationContextRQ, 867
  - gdcmm::network::PresentationDataValue, 870
  - gdcmm::network::RoleSelectionSub, 934
  - gdcmm::network::ServiceClassApplicationInformation, 997
  - gdcmm::network::SOPClassExtendedNegotiationSub, 1022
  - gdcmm::network::TransferSyntaxSub, 1153
  - gdcmm::network::UserInformation, 1276
  - gdcmm::PGXCodec, 816
  - gdcmm::PixmapWriter, 846
  - gdcmm::PNMCodec, 851
  - gdcmm::Preamble, 854
  - gdcmm::SegmentWriter, 973
  - gdcmm::SequenceOfFragments, 981
  - gdcmm::SequenceOfItems, 989
  - gdcmm::StreamImageWriter, 1045
  - gdcmm::SurfaceWriter, 1103
  - gdcmm::Tag, 1131
  - gdcmm::ValueIO< TDE, TSwap, TType >, 1282
  - gdcmm::VL, 1288
  - gdcmm::VR, 1300
  - gdcmm::VRVLSIZE< 0 >, 1306
  - gdcmm::VRVLSIZE< 1 >, 1306
  - gdcmm::Writer, 1416
  - vtkGDCMImageWriter, 1339
- Write16
  - gdcmm::VL, 1288
- WriteASCII
  - gdcmm::Element< TVR, VM::VM1\_n >, 403
- WriteBuffer
  - gdcmm::ByteValue, 225
  - gdcmm::SequenceOfFragments, 981
- WriteBufferAsRGBA
  - gdcmm::LookupTable, 678
- WriteData
  - vtkGDCMPolyDataWriter, 1350
- WriteFooter
  - gdcmm::DictConverter, 363
- WriteGDCMData
  - vtkGDCMImageWriter, 1339
- WriteHeader
  - gdcmm::DictConverter, 364

- WriteHelpFile
  - gdcm::BaseQuery, [179](#)
- WriteImageInformation
  - gdcm::StreamImageWriter, [1045](#)
- WriteImageSubregionRAW
  - gdcm::StreamImageWriter, [1046](#)
- WritePointer
  - vtkLookupTable16, [1396](#)
- WriteQuery
  - gdcm::BaseQuery, [179](#)
- Writer
  - gdcm::Writer, [1414](#)
- WriteRawHeader
  - gdcm::StreamImageWriter, [1046](#)
- WriteRTSTRUCTData
  - vtkGDCMPolyDataWriter, [1350](#)
- WriteRTSTRUCTInfo
  - vtkGDCMPolyDataWriter, [1350](#)
- WriteSlice
  - vtkGDCMImageWriter, [1339](#)
- x16printf
  - gdcm, [74](#)
- XAXRFGayscaleSoftcopyPresentationStateStorage
  - gdcm::UIDs, [1184](#)
- XML
  - gdcm::Printer, [873](#)
- XMLDictReader
  - gdcm::XMLDictReader, [1418](#)
- XMLEncoding
  - gdcm::UIDs, [1178](#)
- XMLPrinter
  - gdcm::XMLPrinter, [1421](#)
- XMLPrivateDictReader
  - gdcm::XMLPrivateDictReader, [1424](#)
- XRay3DAngiographicImageStorage
  - gdcm::MediaStorage, [694](#)
  - gdcm::UIDs, [1180](#)
- XRay3DCraniofacialImageStorage
  - gdcm::MediaStorage, [695](#)
  - gdcm::UIDs, [1180](#)
- XRayAngiographicBiPlaneImageStorageRetired
  - gdcm::MediaStorage, [693](#)
  - gdcm::UIDs, [1180](#)
- XRayAngiographicImageStorage
  - gdcm::MediaStorage, [693](#)
  - gdcm::UIDs, [1180](#)
- XRayRadiationDoseSR
  - gdcm::MediaStorage, [694](#)
- XRayRadiationDoseSRStorage
  - gdcm::UIDs, [1181](#)
- XRayRadiofluoroscopicImageStorage
  - gdcm::UIDs, [1180](#)
- XRayRadiofluoroscopicImageStorage
  - gdcm::MediaStorage, [693](#)
- YBR2RGB
  - gdcm::ImageChangePhotometricInterpretation, [543](#)
- YBR\_FULL
  - gdcm::PhotometricInterpretation, [818](#)
- YBR\_FULL\_422
  - gdcm::PhotometricInterpretation, [818](#)
- YBR\_ICT
  - gdcm::PhotometricInterpretation, [818](#)
- YBR\_PARTIAL\_420
  - gdcm::PhotometricInterpretation, [818](#)
- YBR\_PARTIAL\_422
  - gdcm::PhotometricInterpretation, [818](#)
- YBR\_RCT
  - gdcm::PhotometricInterpretation, [818](#)
- yellow
  - gdcm::terminal, [82](#)
- YES
  - gdcm::Surface, [1084](#)
- ZEROED\_OUT
  - gdcm::CSAHeader, [290](#)
- ZSpacing
  - gdcm::IPPSorter, [616](#)
- ZTolerance
  - gdcm::IPPSorter, [616](#)