

GDCM  
3.0.8

Generated by Doxygen 1.9.1



---

<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 . . . . .	57
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 . . . . .	58
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 . . . . .	60
9.1.2.15 STComp . . . . .	60
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	61
9.1.3.3 ENQueryType	61
9.1.3.4 EQueryLevel	62
9.1.3.5 EQueryType	62
9.1.3.6 ERootType	62
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()	63
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<<() [1/56]	65
9.1.4.10 operator<<() [2/56]	65
9.1.4.11 operator<<() [3/56]	65
9.1.4.12 operator<<() [4/56]	65
9.1.4.13 operator<<() [5/56]	65
9.1.4.14 operator<<() [6/56]	65
9.1.4.15 operator<<() [7/56]	66
9.1.4.16 operator<<() [8/56]	66
9.1.4.17 operator<<() [9/56]	66
9.1.4.18 operator<<() [10/56]	66
9.1.4.19 operator<<() [11/56]	66
9.1.4.20 operator<<() [12/56]	66
9.1.4.21 operator<<() [13/56]	67
9.1.4.22 operator<<() [14/56]	67
9.1.4.23 operator<<() [15/56]	67
9.1.4.24 operator<<() [16/56]	67
9.1.4.25 operator<<() [17/56]	67
9.1.4.26 operator<<() [18/56]	68
9.1.4.27 operator<<() [19/56]	68
9.1.4.28 operator<<() [20/56]	68



9.1.4.29 operator<<()	[21/56]	68
9.1.4.30 operator<<()	[22/56]	68
9.1.4.31 operator<<()	[23/56]	68
9.1.4.32 operator<<()	[24/56]	69
9.1.4.33 operator<<()	[25/56]	69
9.1.4.34 operator<<()	[26/56]	69
9.1.4.35 operator<<()	[27/56]	69
9.1.4.36 operator<<()	[28/56]	69
9.1.4.37 operator<<()	[29/56]	69
9.1.4.38 operator<<()	[30/56]	70
9.1.4.39 operator<<()	[31/56]	70
9.1.4.40 operator<<()	[32/56]	70
9.1.4.41 operator<<()	[33/56]	70
9.1.4.42 operator<<()	[34/56]	70
9.1.4.43 operator<<()	[35/56]	70
9.1.4.44 operator<<()	[36/56]	71
9.1.4.45 operator<<()	[37/56]	71
9.1.4.46 operator<<()	[38/56]	71
9.1.4.47 operator<<()	[39/56]	71
9.1.4.48 operator<<()	[40/56]	71
9.1.4.49 operator<<()	[41/56]	71
9.1.4.50 operator<<()	[42/56]	72
9.1.4.51 operator<<()	[43/56]	72
9.1.4.52 operator<<()	[44/56]	72
9.1.4.53 operator<<()	[45/56]	72
9.1.4.54 operator<<()	[46/56]	72
9.1.4.55 operator<<()	[47/56]	72
9.1.4.56 operator<<()	[48/56]	73
9.1.4.57 operator<<()	[49/56]	73
9.1.4.58 operator<<()	[50/56]	73
9.1.4.59 operator<<()	[51/56]	73
9.1.4.60 operator<<()	[52/56]	73
9.1.4.61 operator<<()	[53/56]	74
9.1.4.62 operator<<()	[54/56]	74
9.1.4.63 operator<<()	[55/56]	74
9.1.4.64 operator<<()	[56/56]	74
9.1.4.65 operator==( )		74
9.1.4.66 operator>>()	[1/3]	75
9.1.4.67 operator>>()	[2/3]	75

9.1.4.68 operator>>() [3/3]	75
9.1.4.69 Round()	75
9.1.4.70 roundat()	75
9.1.4.71 TYPETOENCODING()	76
9.1.4.72 x16printf()	76
9.1.5 Variable Documentation	76
9.1.5.1 GlobalInstance	76
9.1.5.2 VRBINARY	76
9.2 gdcmm::network Namespace Reference	77
9.2.1 Enumeration Type Documentation	81
9.2.1.1 EEventID	81
9.2.1.2 EStateID	81
9.2.2 Function Documentation	83
9.2.2.1 GetStateIndex()	83
9.2.3 Variable Documentation	83
9.2.3.1 cMaxEventID	83
9.2.3.2 cMaxStateID	84
9.3 gdcmm::SegmentHelper Namespace Reference	84
9.4 gdcmm::terminal Namespace Reference	84
9.4.1 Detailed Description	85
9.4.2 Enumeration Type Documentation	85
9.4.2.1 Attribute	85
9.4.2.2 Color	85
9.4.2.3 Mode	86
9.4.3 Function Documentation	86
9.4.3.1 setattribute()	86
9.4.3.2 setbgcolor()	86
9.4.3.3 setfgcolor()	86
9.4.3.4 setmode()	86
<b>10 Class Documentation</b>	<b>87</b>
10.1 gdcmm::network::AAbortPDU Class Reference	87
10.1.1 Detailed Description	88
10.1.2 Constructor & Destructor Documentation	88
10.1.2.1 AAbortPDU()	88
10.1.3 Member Function Documentation	88
10.1.3.1 IsLastFragment()	88
10.1.3.2 Print()	88
10.1.3.3 Read()	89

10.1.3.4 SetReason()	89
10.1.3.5 SetSource()	89
10.1.3.6 Size()	89
10.1.3.7 Write()	89
10.2 gdcmm::network::AAssociateACPDU Class Reference	90
10.2.1 Detailed Description	91
10.2.2 Member Typedef Documentation	91
10.2.2.1 SizeType	91
10.2.3 Constructor & Destructor Documentation	91
10.2.3.1 AAssociateACPDU()	92
10.2.4 Member Function Documentation	92
10.2.4.1 AddPresentationContextAC()	92
10.2.4.2 GetNumberOfPresentationContextAC()	92
10.2.4.3 GetPresentationContextAC()	92
10.2.4.4 GetUserInfoInformation()	92
10.2.4.5 InitFromRQ()	92
10.2.4.6 IsLastFragment()	93
10.2.4.7 Print()	93
10.2.4.8 Read()	93
10.2.4.9 SetCalledAETitle()	93
10.2.4.10 SetCallingAETitle()	93
10.2.4.11 Size()	93
10.2.4.12 Write()	94
10.2.5 Friends And Related Function Documentation	94
10.2.5.1 AAssociateRQPDU	94
10.3 gdcmm::network::AAssociateRJPDU Class Reference	94
10.3.1 Detailed Description	95
10.3.2 Constructor & Destructor Documentation	95
10.3.2.1 AAssociateRJPDU()	95
10.3.3 Member Function Documentation	95
10.3.3.1 IsLastFragment()	96
10.3.3.2 Print()	96
10.3.3.3 Read()	96
10.3.3.4 Size()	96
10.3.3.5 Write()	96
10.4 gdcmm::network::AAssociateRQPDU Class Reference	97
10.4.1 Detailed Description	98
10.4.2 Member Typedef Documentation	98
10.4.2.1 PresentationContextArrayType	99

---

10.4.2.2 SizeType . . . . .	99
10.4.3 Constructor & Destructor Documentation . . . . .	99
10.4.3.1 AAssociateRQPDU() [1/2] . . . . .	99
10.4.3.2 AAssociateRQPDU() [2/2] . . . . .	99
10.4.4 Member Function Documentation . . . . .	99
10.4.4.1 AddPresentationContext() . . . . .	99
10.4.4.2 GetCalledAETitle() . . . . .	99
10.4.4.3 GetCallingAETitle() . . . . .	100
10.4.4.4 GetNumberOfPresentationContext() . . . . .	100
10.4.4.5 GetPresentationContext() . . . . .	100
10.4.4.6 GetPresentationContextByAbstractSyntax() . . . . .	100
10.4.4.7 GetPresentationContextByID() . . . . .	100
10.4.4.8 GetPresentationContexts() . . . . .	100
10.4.4.9 GetReserved43_74() . . . . .	101
10.4.4.10 GetUserInfo() . . . . .	101
10.4.4.11 IsAETitleValid() . . . . .	101
10.4.4.12 IsLastFragment() . . . . .	101
10.4.4.13 Print() . . . . .	101
10.4.4.14 Read() . . . . .	102
10.4.4.15 SetCalledAETitle() . . . . .	102
10.4.4.16 SetCallingAETitle() . . . . .	102
10.4.4.17 SetUserInfo() . . . . .	102
10.4.4.18 Size() . . . . .	102
10.4.4.19 Write() . . . . .	103
10.4.5 Friends And Related Function Documentation . . . . .	103
10.4.5.1 AAssociateACPDU . . . . .	103
10.5 gdcm::AbortEvent Class Reference . . . . .	103
10.6 gdcm::network::AbstractSyntax Class Reference . . . . .	104
10.6.1 Detailed Description . . . . .	105
10.6.2 Constructor & Destructor Documentation . . . . .	105
10.6.2.1 AbstractSyntax() . . . . .	105
10.6.3 Member Function Documentation . . . . .	105
10.6.3.1 GetAsDataElement() . . . . .	105
10.6.3.2 GetName() . . . . .	105
10.6.3.3 operator==( ) . . . . .	105
10.6.3.4 Print() . . . . .	105
10.6.3.5 Read() . . . . .	106
10.6.3.6 SetName() . . . . .	106
10.6.3.7 SetNameFromUID() . . . . .	106

10.6.3.8 Size()	106
10.6.3.9 Write()	106
10.7 gdcm::AnonymizeEvent Class Reference	107
10.7.1 Detailed Description	108
10.7.2 Member Typedef Documentation	108
10.7.2.1 Self	108
10.7.2.2 Superclass	108
10.7.3 Constructor & Destructor Documentation	108
10.7.3.1 AnonymizeEvent() [1/2]	109
10.7.3.2 ~AnonymizeEvent()	109
10.7.3.3 AnonymizeEvent() [2/2]	109
10.7.4 Member Function Documentation	109
10.7.4.1 CheckEvent()	109
10.7.4.2 GetEventName()	109
10.7.4.3 GetTag()	109
10.7.4.4 MakeObject()	110
10.7.4.5 operator=()	110
10.7.4.6 SetTag()	110
10.8 gdcm::Anonymizer Class Reference	110
10.8.1 Detailed Description	112
10.8.2 Constructor & Destructor Documentation	113
10.8.2.1 Anonymizer()	113
10.8.2.2 ~Anonymizer()	113
10.8.3 Member Function Documentation	113
10.8.3.1 BALCPPProtect()	113
10.8.3.2 BasicApplicationLevelConfidentialityProfile()	114
10.8.3.3 CanEmptyTag()	114
10.8.3.4 ClearInternalUIDs()	114
10.8.3.5 Empty()	114
10.8.3.6 GetBasicApplicationLevelConfidentialityProfileAttributes()	115
10.8.3.7 GetCryptographicMessageSyntax()	115
10.8.3.8 GetFile()	115
10.8.3.9 New()	115
10.8.3.10 RecurseDataSet()	115
10.8.3.11 Remove()	115
10.8.3.12 RemoveGroupLength()	116
10.8.3.13 RemovePrivateTags()	116
10.8.3.14 RemoveRetired()	116
10.8.3.15 Replace() [1/2]	116

10.8.3.16 Replace() [2/2]	117
10.8.3.17 SetCryptographicMessageSyntax()	117
10.8.3.18 SetFile()	117
10.9 gdcmm::AnyEvent Class Reference	118
10.10 gdcmm::network::ApplicationContext Class Reference	119
10.10.1 Detailed Description	119
10.10.2 Constructor & Destructor Documentation	120
10.10.2.1 ApplicationContext()	120
10.10.3 Member Function Documentation	120
10.10.3.1 GetName()	120
10.10.3.2 Print()	120
10.10.3.3 Read()	120
10.10.3.4 SetName()	120
10.10.3.5 Size()	121
10.10.3.6 Write()	121
10.11 gdcmm::ApplicationEntity Class Reference	121
10.11.1 Detailed Description	122
10.11.2 Member Function Documentation	122
10.11.2.1 IsValid()	122
10.11.2.2 Print()	122
10.11.2.3 SetBlob()	123
10.11.2.4 Squeeze()	123
10.11.3 Member Data Documentation	123
10.11.3.1 Internal	123
10.11.3.2 MaxLength	123
10.11.3.3 MaxNumberOfComponents	123
10.11.3.4 Padding	123
10.11.3.5 Separator	124
10.12 gdcmm::network::AReleaseRPPDU Class Reference	124
10.12.1 Detailed Description	125
10.12.2 Constructor & Destructor Documentation	125
10.12.2.1 AReleaseRPPDU()	125
10.12.3 Member Function Documentation	125
10.12.3.1 IsLastFragment()	125
10.12.3.2 Print()	125
10.12.3.3 Read()	126
10.12.3.4 Size()	126
10.12.3.5 Write()	126
10.13 gdcmm::network::AReleaseRQPDU Class Reference	126

10.13.1 Detailed Description	127
10.13.2 Constructor & Destructor Documentation	127
10.13.2.1 AReleaseRQPDU()	127
10.13.3 Member Function Documentation	127
10.13.3.1 IsLastFragment()	128
10.13.3.2 Print()	128
10.13.3.3 Read()	128
10.13.3.4 Size()	128
10.13.3.5 Write()	128
10.14 gdcmm::network::ARTIMTimer Class Reference	129
10.14.1 Detailed Description	129
10.14.2 Constructor & Destructor Documentation	129
10.14.2.1 ARTIMTimer()	129
10.14.3 Member Function Documentation	129
10.14.3.1 GetElapsedTime()	130
10.14.3.2 GetHasExpired()	130
10.14.3.3 GetTimeout()	130
10.14.3.4 SetTimeout()	130
10.14.3.5 Start()	130
10.14.3.6 Stop()	130
10.15 gdcmm::ASN1 Class Reference	130
10.15.1 Detailed Description	131
10.15.2 Constructor & Destructor Documentation	131
10.15.2.1 ASN1() [1/2]	131
10.15.2.2 ~ASN1()	131
10.15.2.3 ASN1() [2/2]	131
10.15.3 Member Function Documentation	132
10.15.3.1 operator=()	132
10.15.3.2 ParseDump()	132
10.15.3.3 ParseDumpFile()	132
10.15.3.4 TestPBKDF2()	132
10.16 gdcmm::network::AsynchronousOperationsWindowSub Class Reference	132
10.16.1 Detailed Description	133
10.16.2 Constructor & Destructor Documentation	133
10.16.2.1 AsynchronousOperationsWindowSub()	133
10.16.3 Member Function Documentation	133
10.16.3.1 Print()	133
10.16.3.2 Read()	133
10.16.3.3 Size()	134

10.16.3.4 Write()	134
10.17 gdcmm::Attribute< Group, Element, TVR, TVM > Class Template Reference	134
10.17.1 Detailed Description	136
10.17.2 Member Typedef Documentation	136
10.17.2.1 ArrayType	136
10.17.3 Member Enumeration Documentation	136
10.17.3.1 anonymous enum	136
10.17.4 Member Function Documentation	137
10.17.4.1 GDCM_STATIC_ASSERT() [1/3]	137
10.17.4.2 GDCM_STATIC_ASSERT() [2/3]	137
10.17.4.3 GDCM_STATIC_ASSERT() [3/3]	137
10.17.4.4 GetAsDataElement()	137
10.17.4.5 GetDictVM()	138
10.17.4.6 GetDictVR()	138
10.17.4.7 GetNumberOfValues()	138
10.17.4.8 GetTag()	138
10.17.4.9 GetValue() [1/2]	139
10.17.4.10 GetValue() [2/2]	139
10.17.4.11 GetValues()	139
10.17.4.12 GetVM()	139
10.17.4.13 GetVR()	140
10.17.4.14 operator!=(())	140
10.17.4.15 operator<()	140
10.17.4.16 operator==(())	140
10.17.4.17 operator[]() [1/2]	140
10.17.4.18 operator[]() [2/2]	141
10.17.4.19 Print()	141
10.17.4.20 Set()	141
10.17.4.21 SetByteValue()	141
10.17.4.22 SetByteValueNoSwap()	142
10.17.4.23 SetFromDataElement()	142
10.17.4.24 SetFromDataSet()	142
10.17.4.25 SetValue()	143
10.17.4.26 SetValues()	143
10.17.5 Member Data Documentation	143
10.17.5.1 Internal	143
10.18 gdcmm::Attribute< Group, Element, TVR, VM::VM1 > Class Template Reference	144
10.18.1 Member Typedef Documentation	145
10.18.1.1 ArrayType	145



10.18.2 Member Enumeration Documentation	145
10.18.2.1 anonymous enum	145
10.18.3 Member Function Documentation	146
10.18.3.1 GDCM_STATIC_ASSERT() [1/4]	146
10.18.3.2 GDCM_STATIC_ASSERT() [2/4]	146
10.18.3.3 GDCM_STATIC_ASSERT() [3/4]	146
10.18.3.4 GDCM_STATIC_ASSERT() [4/4]	146
10.18.3.5 GetAsDataElement()	147
10.18.3.6 GetDictVM()	147
10.18.3.7 GetDictVR()	147
10.18.3.8 GetNumberOfValues()	147
10.18.3.9 GetTag()	147
10.18.3.10 GetValue() [1/2]	147
10.18.3.11 GetValue() [2/2]	148
10.18.3.12 GetValues()	148
10.18.3.13 GetVM()	148
10.18.3.14 GetVR()	148
10.18.3.15 operator!=(())	148
10.18.3.16 operator<()	148
10.18.3.17 operator==(())	149
10.18.3.18 Print()	149
10.18.3.19 Set()	149
10.18.3.20 SetByteValue()	149
10.18.3.21 SetByteValueNoSwap()	149
10.18.3.22 SetFromDataElement()	150
10.18.3.23 SetFromDataSet()	150
10.18.3.24 SetValue()	150
10.18.4 Member Data Documentation	150
10.18.4.1 Internal	150
10.19 gdcM::Attribute< Group, Element, TVR, VM::VM1_3 > Class Template Reference	151
10.19.1 Member Function Documentation	151
10.19.1.1 GetVM()	152
10.20 gdcM::Attribute< Group, Element, TVR, VM::VM1_8 > Class Template Reference	152
10.20.1 Member Function Documentation	153
10.20.1.1 GetVM()	153
10.21 gdcM::Attribute< Group, Element, TVR, VM::VM1_n > Class Template Reference	153
10.21.1 Member Typedef Documentation	154
10.21.1.1 ArrayType	154
10.21.2 Constructor & Destructor Documentation	155

10.21.2.1 Attribute()	155
10.21.2.2 ~Attribute()	155
10.21.3 Member Function Documentation	155
10.21.3.1 GDCM_STATIC_ASSERT() [1/3]	155
10.21.3.2 GDCM_STATIC_ASSERT() [2/3]	155
10.21.3.3 GDCM_STATIC_ASSERT() [3/3]	155
10.21.3.4 GetAsDataElement()	156
10.21.3.5 GetDictVM()	156
10.21.3.6 GetDictVR()	156
10.21.3.7 GetNumberOfValues()	156
10.21.3.8 GetTag()	156
10.21.3.9 GetValue() [1/2]	156
10.21.3.10 GetValue() [2/2]	157
10.21.3.11 GetValues()	157
10.21.3.12 GetVM()	157
10.21.3.13 GetVR()	157
10.21.3.14 operator[]() [1/2]	157
10.21.3.15 operator[]() [2/2]	157
10.21.3.16 Print()	158
10.21.3.17 Set()	158
10.21.3.18 SetByteValue()	158
10.21.3.19 SetFromDataElement()	158
10.21.3.20 SetFromDataSet()	158
10.21.3.21 SetNumberOfValues()	159
10.21.3.22 SetValue() [1/2]	159
10.21.3.23 SetValue() [2/2]	159
10.21.3.24 SetValues()	159
10.22 gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n > Class Template Reference	160
10.22.1 Member Function Documentation	161
10.22.1.1 GetVM()	161
10.23 gdcmm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference	161
10.23.1 Member Function Documentation	162
10.23.1.1 GetVM()	162
10.24 gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n > Class Template Reference	163
10.24.1 Member Function Documentation	164
10.24.1.1 GetVM()	164
10.25 gdcmm::Attribute< Group, Element, TVR, VM::VM3_n > Class Template Reference	164
10.25.1 Member Function Documentation	165
10.25.1.1 GetVM()	165

10.26 gdcmm::AudioCodec Class Reference	166
10.26.1 Detailed Description	167
10.26.2 Constructor & Destructor Documentation	167
10.26.2.1 AudioCodec()	167
10.26.2.2 ~AudioCodec()	167
10.26.3 Member Function Documentation	167
10.26.3.1 CanCode()	167
10.26.3.2 CanDecode()	168
10.26.3.3 Decode()	168
10.27 gdcmm::Base64 Class Reference	168
10.27.1 Detailed Description	169
10.27.2 Constructor & Destructor Documentation	169
10.27.2.1 Base64()	169
10.27.3 Member Function Documentation	169
10.27.3.1 Decode()	169
10.27.3.2 Encode()	170
10.27.3.3 GetDecodeLength()	170
10.27.3.4 GetEncodeLength()	170
10.27.3.5 operator=()	171
10.28 gdcmm::network::BaseCompositeMessage Class Reference	171
10.28.1 Detailed Description	172
10.28.2 Constructor & Destructor Documentation	172
10.28.2.1 ~BaseCompositeMessage()	172
10.28.3 Member Function Documentation	172
10.28.3.1 ConstructPDV()	172
10.29 gdcmm::network::BaseNormalizedMessage Class Reference	173
10.29.1 Detailed Description	174
10.29.2 Constructor & Destructor Documentation	174
10.29.2.1 ~BaseNormalizedMessage()	174
10.29.3 Member Function Documentation	174
10.29.3.1 ConstructPDV()	174
10.30 gdcmm::network::BasePDU Class Reference	175
10.30.1 Detailed Description	175
10.30.2 Constructor & Destructor Documentation	176
10.30.2.1 ~BasePDU()	176
10.30.3 Member Function Documentation	176
10.30.3.1 IsLastFragment()	176
10.30.3.2 Print()	176
10.30.3.3 Read()	177

10.30.3.4 Size()	177
10.30.3.5 Write()	177
10.31 gdcmm::BaseQuery Class Reference	177
10.31.1 Detailed Description	179
10.31.2 Constructor & Destructor Documentation	179
10.31.2.1 BaseQuery()	179
10.31.2.2 ~BaseQuery()	179
10.31.3 Member Function Documentation	179
10.31.3.1 AddQueryDataSet()	180
10.31.3.2 GetAbstractSyntaxUID()	180
10.31.3.3 GetQueryDataSet() [1/2]	180
10.31.3.4 GetQueryDataSet() [2/2]	180
10.31.3.5 GetSOPInstanceUID()	180
10.31.3.6 Print()	180
10.31.3.7 SetSearchParameter() [1/3]	181
10.31.3.8 SetSearchParameter() [2/3]	181
10.31.3.9 SetSearchParameter() [3/3]	181
10.31.3.10 SetSOPInstanceUID()	181
10.31.3.11 ValidateQuery()	181
10.31.3.12 ValidDataSet()	182
10.31.3.13 WriteHelpFile()	182
10.31.3.14 WriteQuery()	182
10.31.4 Friends And Related Function Documentation	182
10.31.4.1 QueryFactory	182
10.31.5 Member Data Documentation	182
10.31.5.1 mDataSet	182
10.31.5.2 mSopInstanceUID	183
10.32 gdcmm::BaseRootQuery Class Reference	183
10.32.1 Detailed Description	184
10.32.2 Constructor & Destructor Documentation	185
10.32.2.1 BaseRootQuery()	185
10.32.2.2 ~BaseRootQuery()	185
10.32.3 Member Function Documentation	185
10.32.3.1 Construct()	185
10.32.3.2 GetQueryLevelFromQueryRoot()	185
10.32.3.3 GetQueryLevelFromString()	185
10.32.3.4 GetQueryLevelString()	186
10.32.3.5 GetTagListByLevel()	186
10.32.3.6 InitializeDataSet()	186

10.32.3.7 ValidateQuery()	186
10.32.4 Friends And Related Function Documentation	187
10.32.4.1 QueryFactory	187
10.32.5 Member Data Documentation	187
10.32.5.1 mHelpDescription	187
10.32.5.2 mImage	187
10.32.5.3 mPatient	187
10.32.5.4 mRootType	187
10.32.5.5 mSeries	187
10.32.5.6 mStudy	188
10.33 gdcm::SegmentHelper::BasicCodedEntry Struct Reference	188
10.33.1 Detailed Description	189
10.33.2 Constructor & Destructor Documentation	189
10.33.2.1 BasicCodedEntry() [1/3]	189
10.33.2.2 BasicCodedEntry() [2/3]	189
10.33.2.3 BasicCodedEntry() [3/3]	190
10.33.3 Member Function Documentation	190
10.33.3.1 IsEmpty()	190
10.33.4 Member Data Documentation	190
10.33.4.1 CM	190
10.33.4.2 CSD	190
10.33.4.3 CSV	191
10.33.4.4 CV	191
10.34 gdcm::BasicOffsetTable Class Reference	191
10.34.1 Detailed Description	192
10.34.2 Constructor & Destructor Documentation	192
10.34.2.1 BasicOffsetTable()	193
10.34.3 Member Function Documentation	193
10.34.3.1 Read()	193
10.34.4 Friends And Related Function Documentation	193
10.34.4.1 operator<<	193
10.35 gdcm::Bitmap Class Reference	194
10.35.1 Detailed Description	196
10.35.2 Member Typedef Documentation	197
10.35.2.1 LUTPtr	197
10.35.3 Constructor & Destructor Documentation	197
10.35.3.1 Bitmap()	197
10.35.3.2 ~Bitmap()	197
10.35.4 Member Function Documentation	197

10.35.4.1 AreOverlaysInPixelData()	197
10.35.4.2 Clear()	197
10.35.4.3 ComputeLossyFlag()	198
10.35.4.4 GetBuffer()	198
10.35.4.5 GetBuffer2()	198
10.35.4.6 GetBufferLength()	198
10.35.4.7 GetColumns()	198
10.35.4.8 GetDataElement() [1/2]	199
10.35.4.9 GetDataElement() [2/2]	199
10.35.4.10 GetDimension()	199
10.35.4.11 GetDimensions()	199
10.35.4.12 GetLUT() [1/2]	199
10.35.4.13 GetLUT() [2/2]	200
10.35.4.14 GetNeedByteSwap()	200
10.35.4.15 GetNumberOfDimensions()	200
10.35.4.16 GetPhotometricInterpretation()	200
10.35.4.17 GetPixelFormat() [1/2]	201
10.35.4.18 GetPixelFormat() [2/2]	201
10.35.4.19 GetPlanarConfiguration()	201
10.35.4.20 GetRows()	201
10.35.4.21 GetTransferSyntax()	201
10.35.4.22 IsEmpty()	202
10.35.4.23 IsLossy()	202
10.35.4.24 IsTransferSyntaxCompatible()	202
10.35.4.25 Print()	202
10.35.4.26 SetColumns()	202
10.35.4.27 SetDataElement()	203
10.35.4.28 SetDimension()	203
10.35.4.29 SetDimensions()	203
10.35.4.30 SetLossyFlag()	203
10.35.4.31 SetLUT()	204
10.35.4.32 SetNeedByteSwap()	204
10.35.4.33 SetNumberOfDimensions()	204
10.35.4.34 SetPhotometricInterpretation()	204
10.35.4.35 SetPixelFormat()	205
10.35.4.36 SetPlanarConfiguration()	205
10.35.4.37 SetRows()	205
10.35.4.38 SetTransferSyntax()	205
10.35.4.39 TryJPEG2000Codec()	206

10.35.4.40 TryJPEG2000Codec2()	206
10.35.4.41 TryJPEGCodec()	206
10.35.4.42 TryJPEGCodec2()	206
10.35.4.43 TryJPEGLSCodec()	206
10.35.4.44 TryKAKADUCodec()	206
10.35.4.45 TryPVRGCodec()	207
10.35.4.46 TryRAWCodec()	207
10.35.4.47 TryRLECodec()	207
10.35.4.48 UnusedBitsPresentInPixelData()	207
10.35.5 Friends And Related Function Documentation	207
10.35.5.1 ImageChangeTransferSyntax	207
10.35.5.2 PixmapReader	207
10.35.6 Member Data Documentation	208
10.35.6.1 Dimensions	208
10.35.6.2 LossyFlag	208
10.35.6.3 LUT	208
10.35.6.4 NeedByteSwap	208
10.35.6.5 NumberOfDimensions	208
10.35.6.6 PF	208
10.35.6.7 PI	209
10.35.6.8 PixelData	209
10.35.6.9 PlanarConfiguration	209
10.35.6.10 TS	209
10.36 gdcm::BitmapToBitmapFilter Class Reference	209
10.36.1 Detailed Description	210
10.36.2 Constructor & Destructor Documentation	211
10.36.2.1 BitmapToBitmapFilter()	211
10.36.2.2 ~BitmapToBitmapFilter()	211
10.36.3 Member Function Documentation	211
10.36.3.1 GetOutput()	211
10.36.3.2 GetOutputAsBitmap()	211
10.36.3.3 SetInput()	211
10.36.4 Member Data Documentation	212
10.36.4.1 Input	212
10.36.4.2 Output	212
10.37 gdcm::BoxRegion Class Reference	212
10.37.1 Detailed Description	214
10.37.2 Constructor & Destructor Documentation	214
10.37.2.1 BoxRegion() [1/2]	214

10.37.2.2 ~BoxRegion()	214
10.37.2.3 BoxRegion() [2/2]	214
10.37.3 Member Function Documentation	214
10.37.3.1 Area()	214
10.37.3.2 BoundingBox()	215
10.37.3.3 Clone()	215
10.37.3.4 ComputeBoundingBox()	215
10.37.3.5 Empty()	215
10.37.3.6 GetXMax()	215
10.37.3.7 GetXMin()	216
10.37.3.8 GetYMax()	216
10.37.3.9 GetYMin()	216
10.37.3.10 GetZMax()	216
10.37.3.11 GetZMin()	216
10.37.3.12 IsValid()	216
10.37.3.13 operator=()	217
10.37.3.14 Print()	217
10.37.3.15 SetDomain()	217
10.38 gdcm::ByteBuffer Class Reference	217
10.38.1 Detailed Description	218
10.38.2 Constructor & Destructor Documentation	218
10.38.2.1 ByteBuffer()	218
10.38.3 Member Function Documentation	218
10.38.3.1 Get()	218
10.38.3.2 GetStart()	218
10.38.3.3 ShiftEnd()	218
10.38.3.4 UpdatePosition()	219
10.39 gdcm::ByteSwap< T > Class Template Reference	219
10.39.1 Detailed Description	219
10.39.2 Member Function Documentation	219
10.39.2.1 Swap()	220
10.39.2.2 SwapFromSwapCodeIntoSystem()	220
10.39.2.3 SwapRange()	220
10.39.2.4 SwapRangeFromSwapCodeIntoSystem()	220
10.39.2.5 SystemIsBigEndian()	221
10.39.2.6 SystemIsLittleEndian()	221
10.40 gdcm::ByteSwapFilter Class Reference	221
10.40.1 Detailed Description	221
10.40.2 Constructor & Destructor Documentation	221



10.40.2.1 ByteSwapFilter() [1/2]	222
10.40.2.2 ~ByteSwapFilter()	222
10.40.2.3 ByteSwapFilter() [2/2]	222
10.40.3 Member Function Documentation	222
10.40.3.1 ByteSwap()	222
10.40.3.2 operator=()	222
10.40.3.3 SetByteSwapTag()	222
10.41 gdcm::ByteValue Class Reference	223
10.41.1 Detailed Description	225
10.41.2 Constructor & Destructor Documentation	225
10.41.2.1 ByteValue() [1/2]	225
10.41.2.2 ByteValue() [2/2]	225
10.41.2.3 ~ByteValue()	225
10.41.3 Member Function Documentation	226
10.41.3.1 Append()	226
10.41.3.2 Clear()	226
10.41.3.3 ComputeLength()	226
10.41.3.4 Fill()	226
10.41.3.5 GetBuffer()	226
10.41.3.6 GetLength()	227
10.41.3.7 GetPointer()	227
10.41.3.8 GetVoidPointer() [1/2]	227
10.41.3.9 GetVoidPointer() [2/2]	228
10.41.3.10 IsEmpty()	228
10.41.3.11 IsPrintable()	228
10.41.3.12 operator const std::vector< char > &()	228
10.41.3.13 operator=()	228
10.41.3.14 operator==( [1/2]	229
10.41.3.15 operator==( [2/2]	229
10.41.3.16 Print()	229
10.41.3.17 PrintASCII()	229
10.41.3.18 PrintASCIIXML()	229
10.41.3.19 PrintGroupLength()	229
10.41.3.20 PrintHex()	230
10.41.3.21 PrintHexXML()	230
10.41.3.22 PrintPNXML()	230
10.41.3.23 Read() [1/2]	230
10.41.3.24 Read() [2/2]	230
10.41.3.25 SetLength()	230

10.41.3.26 SetLengthOnly()	231
10.41.3.27 Write() [1/2]	231
10.41.3.28 Write() [2/2]	231
10.41.3.29 WriteBuffer()	231
10.42 gdcmm::CAPICryptoFactory Class Reference	232
10.42.1 Constructor & Destructor Documentation	232
10.42.1.1 CAPICryptoFactory()	233
10.42.2 Member Function Documentation	233
10.42.2.1 CreateCMSProvider()	233
10.43 gdcmm::CAPICryptographicMessageSyntax Class Reference	233
10.43.1 Constructor & Destructor Documentation	234
10.43.1.1 CAPICryptographicMessageSyntax()	234
10.43.1.2 ~CAPICryptographicMessageSyntax()	235
10.43.2 Member Function Documentation	235
10.43.2.1 Decrypt()	235
10.43.2.2 Encrypt()	235
10.43.2.3 GetCipherType()	235
10.43.2.4 GetInitialized()	236
10.43.2.5 ParseCertificateFile()	236
10.43.2.6 ParseKeyFile()	236
10.43.2.7 SetCipherType()	236
10.43.2.8 SetPassword()	236
10.44 gdcmm::network::CEchoRQ Class Reference	237
10.44.1 Detailed Description	238
10.44.2 Member Function Documentation	238
10.44.2.1 ConstructPDV()	238
10.44.3 Member Data Documentation	238
10.44.3.1 AffectedSOPClassUID	238
10.44.3.2 MessageID	238
10.45 gdcmm::network::CEchoRSP Class Reference	239
10.45.1 Detailed Description	239
10.45.2 Member Function Documentation	240
10.45.2.1 ConstructPDVByDataSet()	240
10.46 gdcmm::network::CFind Class Reference	240
10.46.1 Detailed Description	240
10.47 gdcmm::network::CFindCancelRQ Class Reference	240
10.47.1 Detailed Description	241
10.47.2 Member Function Documentation	241
10.47.2.1 ConstructPDVByDataSet()	241

10.48 gdcmm::network::CFindRQ Class Reference . . . . .	242
10.48.1 Detailed Description . . . . .	242
10.48.2 Member Function Documentation . . . . .	243
10.48.2.1 ConstructPDV() . . . . .	243
10.49 gdcmm::network::CFindRSP Class Reference . . . . .	243
10.49.1 Detailed Description . . . . .	244
10.49.2 Member Function Documentation . . . . .	244
10.49.2.1 ConstructPDVByDataSet() . . . . .	244
10.50 gdcmm::network::CMoveCancelRq Class Reference . . . . .	245
10.50.1 Member Function Documentation . . . . .	245
10.50.1.1 ConstructPDVByDataSet() . . . . .	246
10.51 gdcmm::network::CMoveRQ Class Reference . . . . .	246
10.51.1 Detailed Description . . . . .	247
10.51.2 Member Function Documentation . . . . .	247
10.51.2.1 ConstructPDV() . . . . .	247
10.52 gdcmm::network::CMoveRSP Class Reference . . . . .	247
10.52.1 Detailed Description . . . . .	248
10.52.2 Member Function Documentation . . . . .	248
10.52.2.1 ConstructPDVByDataSet() . . . . .	248
10.53 gdcmm::Codec Class Reference . . . . .	249
10.53.1 Detailed Description . . . . .	249
10.54 gdcmm::Coder Class Reference . . . . .	250
10.54.1 Detailed Description . . . . .	250
10.54.2 Constructor & Destructor Documentation . . . . .	250
10.54.2.1 ~Coder() . . . . .	251
10.54.3 Member Function Documentation . . . . .	251
10.54.3.1 CanCode() . . . . .	251
10.54.3.2 Code() . . . . .	251
10.54.3.3 InternalCode() . . . . .	251
10.55 gdcmm::CodeString Class Reference . . . . .	252
10.55.1 Detailed Description . . . . .	253
10.55.2 Member Typedef Documentation . . . . .	253
10.55.2.1 const_iterator . . . . .	253
10.55.2.2 const_reference . . . . .	253
10.55.2.3 const_reverse_iterator . . . . .	253
10.55.2.4 difference_type . . . . .	253
10.55.2.5 iterator . . . . .	254
10.55.2.6 pointer . . . . .	254
10.55.2.7 reference . . . . .	254

10.55.2.8 reverse_iterator . . . . .	254
10.55.2.9 size_type . . . . .	254
10.55.2.10 value_type . . . . .	254
10.55.3 Constructor & Destructor Documentation . . . . .	254
10.55.3.1 CodeString() [1/4] . . . . .	255
10.55.3.2 CodeString() [2/4] . . . . .	255
10.55.3.3 CodeString() [3/4] . . . . .	255
10.55.3.4 CodeString() [4/4] . . . . .	255
10.55.4 Member Function Documentation . . . . .	255
10.55.4.1 GetAsString() . . . . .	255
10.55.4.2 IsValid() . . . . .	256
10.55.4.3 Size() . . . . .	256
10.55.4.4 TrimInternal() . . . . .	256
10.55.5 Friends And Related Function Documentation . . . . .	256
10.55.5.1 operator"!=" . . . . .	256
10.55.5.2 operator<< . . . . .	256
10.55.5.3 operator== . . . . .	257
10.56 gdcmm::Command Class Reference . . . . .	257
10.56.1 Detailed Description . . . . .	258
10.56.2 Constructor & Destructor Documentation . . . . .	258
10.56.2.1 Command() [1/2] . . . . .	259
10.56.2.2 Command() [2/2] . . . . .	259
10.56.2.3 ~Command() . . . . .	259
10.56.3 Member Function Documentation . . . . .	259
10.56.3.1 Execute() [1/2] . . . . .	259
10.56.3.2 Execute() [2/2] . . . . .	259
10.56.3.3 operator=() . . . . .	260
10.57 gdcmm::CommandDataSet Class Reference . . . . .	260
10.57.1 Detailed Description . . . . .	261
10.57.2 Constructor & Destructor Documentation . . . . .	261
10.57.2.1 CommandDataSet() . . . . .	261
10.57.2.2 ~CommandDataSet() . . . . .	261
10.57.3 Member Function Documentation . . . . .	261
10.57.3.1 Insert() . . . . .	262
10.57.3.2 Read() . . . . .	262
10.57.3.3 Replace() . . . . .	262
10.57.3.4 Write() . . . . .	262
10.57.4 Friends And Related Function Documentation . . . . .	262
10.57.4.1 operator<< . . . . .	262

10.58 gdcmm::network::CompositeMessageFactory Class Reference	263
10.58.1 Detailed Description	263
10.58.2 Member Function Documentation	263
10.58.2.1 ConstructCEchoRQ()	263
10.58.2.2 ConstructCFindRQ()	263
10.58.2.3 ConstructCMoveRQ()	264
10.58.2.4 ConstructCStoreRQ()	264
10.58.2.5 ConstructCStoreRSP()	264
10.59 gdcmm::CompositeNetworkFunctions Class Reference	264
10.59.1 Detailed Description	265
10.59.2 Member Typedef Documentation	265
10.59.2.1 KeyValuePairArrayType	265
10.59.2.2 KeyValuePairType	265
10.59.3 Member Function Documentation	266
10.59.3.1 CEcho()	266
10.59.3.2 CFind()	266
10.59.3.3 CMove()	267
10.59.3.4 ConstructQuery() [1/2]	267
10.59.3.5 ConstructQuery() [2/2]	268
10.59.3.6 CStore()	268
10.60 gdcmm::ConstCharWrapper Class Reference	269
10.60.1 Detailed Description	269
10.60.2 Constructor & Destructor Documentation	269
10.60.2.1 ConstCharWrapper()	269
10.60.3 Member Function Documentation	269
10.60.3.1 operator const char *()	269
10.61 gdcmm::CP246ExplicitDataElement Class Reference	270
10.61.1 Detailed Description	271
10.61.2 Member Function Documentation	271
10.61.2.1 GetLength()	271
10.61.2.2 Read()	271
10.61.2.3 ReadPreValue()	271
10.61.2.4 ReadValue()	272
10.61.2.5 ReadWithLength()	272
10.62 gdcmm::CryptoFactory Class Reference	272
10.62.1 Detailed Description	273
10.62.2 Member Enumeration Documentation	273
10.62.2.1 CryptoLib	273
10.62.3 Constructor & Destructor Documentation	274

10.62.3.1 CryptoFactory() [1/2]	274
10.62.3.2 CryptoFactory() [2/2]	274
10.62.3.3 ~CryptoFactory()	274
10.62.4 Member Function Documentation	274
10.62.4.1 CreateCMSProvider()	274
10.62.4.2 GetFactoryInstance()	274
10.63 gdcM::CryptographicMessageSyntax Class Reference	275
10.63.1 Member Enumeration Documentation	275
10.63.1.1 CipherTypes	275
10.63.2 Constructor & Destructor Documentation	276
10.63.2.1 CryptographicMessageSyntax() [1/2]	276
10.63.2.2 ~CryptographicMessageSyntax()	276
10.63.2.3 CryptographicMessageSyntax() [2/2]	276
10.63.3 Member Function Documentation	276
10.63.3.1 Decrypt()	276
10.63.3.2 Encrypt()	277
10.63.3.3 GetCipherType()	277
10.63.3.4 operator=()	277
10.63.3.5 ParseCertificateFile()	277
10.63.3.6 ParseKeyFile()	277
10.63.3.7 SetCipherType()	278
10.63.3.8 SetPassword()	278
10.64 gdcM::CSAElement Class Reference	278
10.64.1 Detailed Description	280
10.64.2 Member Typedef Documentation	280
10.64.2.1 DataPtr	280
10.64.3 Constructor & Destructor Documentation	280
10.64.3.1 CSAElement() [1/2]	280
10.64.3.2 CSAElement() [2/2]	280
10.64.4 Member Function Documentation	281
10.64.4.1 GetByteValue()	281
10.64.4.2 GetKey()	281
10.64.4.3 GetName()	281
10.64.4.4 GetNoOfItems()	281
10.64.4.5 GetSyngoDT()	282
10.64.4.6 GetValue() [1/2]	282
10.64.4.7 GetValue() [2/2]	282
10.64.4.8 GetVM()	282
10.64.4.9 GetVR()	282

10.64.4.10 IsEmpty()	283
10.64.4.11 operator<()	283
10.64.4.12 operator=()	283
10.64.4.13 operator==()	283
10.64.4.14 SetByteValue()	283
10.64.4.15 SetKey()	284
10.64.4.16 SetName()	284
10.64.4.17 SetNoOfItems()	284
10.64.4.18 SetSyngoDT()	284
10.64.4.19 SetValue()	284
10.64.4.20 SetVM()	284
10.64.4.21 SetVR()	285
10.64.5 Friends And Related Function Documentation	285
10.64.5.1 operator<<	285
10.64.6 Member Data Documentation	285
10.64.6.1 DataField	285
10.64.6.2 KeyField	285
10.64.6.3 NameField	285
10.64.6.4 NoOfItemsField	286
10.64.6.5 SyngoDTField	286
10.64.6.6 ValueMultiplicityField	286
10.64.6.7 VRField	286
10.65 gdcM::CSAHeader Class Reference	286
10.65.1 Detailed Description	287
10.65.2 Member Enumeration Documentation	288
10.65.2.1 CSAHeaderType	288
10.65.3 Constructor & Destructor Documentation	288
10.65.3.1 CSAHeader()	289
10.65.3.2 ~CSAHeader()	289
10.65.4 Member Function Documentation	289
10.65.4.1 FindCSAElementByName()	289
10.65.4.2 GetCSADataInfo()	289
10.65.4.3 GetCSAEEnd()	289
10.65.4.4 GetCSAElementByName()	290
10.65.4.5 GetCSAImageHeaderInfoTag()	290
10.65.4.6 GetCSASeriesHeaderInfoTag()	290
10.65.4.7 GetDataSet()	291
10.65.4.8 GetFormat()	291
10.65.4.9 GetInterfile()	291

10.65.4.10 GetMrProtocol()	291
10.65.4.11 LoadFromDataElement()	291
10.65.4.12 Print()	292
10.65.5 Friends And Related Function Documentation	292
10.65.5.1 operator<<	292
10.66 gdcM::CSAHeaderDict Class Reference	292
10.66.1 Detailed Description	293
10.66.2 Member Typedef Documentation	293
10.66.2.1 ConstIterator	293
10.66.2.2 Iterator	293
10.66.2.3 MapCSAHeaderDictEntry	293
10.66.3 Constructor & Destructor Documentation	293
10.66.3.1 CSAHeaderDict() [1/2]	294
10.66.3.2 CSAHeaderDict() [2/2]	294
10.66.4 Member Function Documentation	294
10.66.4.1 AddCSAHeaderDictEntry()	294
10.66.4.2 Begin()	294
10.66.4.3 End()	294
10.66.4.4 GetCSAHeaderDictEntry()	294
10.66.4.5 IsEmpty()	295
10.66.4.6 LoadDefault()	295
10.66.4.7 operator=()	295
10.66.5 Friends And Related Function Documentation	295
10.66.5.1 Dicts	295
10.66.5.2 operator<<	295
10.67 gdcM::CSAHeaderDictEntry Class Reference	295
10.67.1 Detailed Description	296
10.67.2 Constructor & Destructor Documentation	296
10.67.2.1 CSAHeaderDictEntry()	297
10.67.3 Member Function Documentation	297
10.67.3.1 GetDescription()	297
10.67.3.2 GetName()	297
10.67.3.3 GetVM()	297
10.67.3.4 GetVR()	297
10.67.3.5 operator<()	298
10.67.3.6 SetDescription()	298
10.67.3.7 SetName()	298
10.67.3.8 SetVM()	298
10.67.3.9 SetVR()	298



---

10.67.4 Friends And Related Function Documentation	298
10.67.4.1 operator<<	299
10.68 gdcm::CSAHeaderDictException Class Reference	299
10.69 gdcm::network::CStoreRQ Class Reference	300
10.69.1 Detailed Description	300
10.69.2 Member Function Documentation	301
10.69.2.1 ConstructPDV()	301
10.70 gdcm::network::CStoreRSP Class Reference	301
10.70.1 Detailed Description	302
10.70.2 Member Function Documentation	302
10.70.2.1 ConstructPDV()	302
10.71 gdcm::Curve Class Reference	303
10.71.1 Detailed Description	304
10.71.2 Constructor & Destructor Documentation	304
10.71.2.1 Curve() [1/2]	305
10.71.2.2 ~Curve()	305
10.71.2.3 Curve() [2/2]	305
10.71.3 Member Function Documentation	305
10.71.3.1 Decode()	305
10.71.3.2 GetAsPoints()	305
10.71.3.3 GetCurveDataDescriptor()	305
10.71.3.4 GetDataValueRepresentation()	306
10.71.3.5 GetDimensions()	306
10.71.3.6 GetGroup()	306
10.71.3.7 GetNumberOfCurves()	306
10.71.3.8 GetNumberOfPoints()	306
10.71.3.9 GetTypeInfoData()	306
10.71.3.10 GetTypeInfoDataDescription()	306
10.71.3.11 IsEmpty()	307
10.71.3.12 Print()	307
10.71.3.13 SetCoordinateStartValue()	307
10.71.3.14 SetCoordinateStepValue()	307
10.71.3.15 SetCurve()	307
10.71.3.16 SetCurveDataDescriptor()	307
10.71.3.17 SetCurveDescription()	308
10.71.3.18 SetDataValueRepresentation()	308
10.71.3.19 SetDimensions()	308
10.71.3.20 SetGroup()	308
10.71.3.21 SetNumberOfPoints()	308

10.71.3.22 SetTypeOfData()	308
10.71.3.23 Update()	309
10.72 gdcm::DataElement Class Reference	309
10.72.1 Detailed Description	312
10.72.2 Member Typedef Documentation	312
10.72.2.1 ValuePtr	312
10.72.3 Constructor & Destructor Documentation	312
10.72.3.1 DataElement() [1/2]	313
10.72.3.2 DataElement() [2/2]	313
10.72.4 Member Function Documentation	313
10.72.4.1 Clear()	313
10.72.4.2 Empty()	313
10.72.4.3 GetByteValue()	313
10.72.4.4 GetLength()	314
10.72.4.5 GetSequenceOfFragments() [1/2]	314
10.72.4.6 GetSequenceOfFragments() [2/2]	314
10.72.4.7 GetTag() [1/2]	314
10.72.4.8 GetTag() [2/2]	314
10.72.4.9 GetValue() [1/2]	315
10.72.4.10 GetValue() [2/2]	315
10.72.4.11 GetValueAsSQ()	315
10.72.4.12 GetVL() [1/2]	316
10.72.4.13 GetVL() [2/2]	316
10.72.4.14 GetVR()	316
10.72.4.15 IsEmpty()	316
10.72.4.16 IsUndefinedLength()	317
10.72.4.17 operator<()	317
10.72.4.18 operator=()	317
10.72.4.19 operator==(())	317
10.72.4.20 Read()	317
10.72.4.21 ReadOrSkip()	318
10.72.4.22 ReadPreValue()	318
10.72.4.23 ReadValue()	318
10.72.4.24 ReadValueWithLength()	318
10.72.4.25 ReadWithLength()	318
10.72.4.26 SetByteValue()	319
10.72.4.27 SetTag()	319
10.72.4.28 SetValue()	320
10.72.4.29 SetValueFieldLength()	320

10.72.4.30 SetVL()	320
10.72.4.31 SetVLToUndefined()	320
10.72.4.32 SetVR()	321
10.72.4.33 Write()	321
10.72.5 Friends And Related Function Documentation	321
10.72.5.1 operator<<	321
10.72.6 Member Data Documentation	321
10.72.6.1 TagField	322
10.72.6.2 ValueField	322
10.72.6.3 ValueLengthField	322
10.72.6.4 VRField	322
10.73 gdcm::DataElementException Class Reference	323
10.74 gdcm::DataEvent Class Reference	323
10.74.1 Detailed Description	325
10.74.2 Member Typedef Documentation	325
10.74.2.1 Self	325
10.74.2.2 Superclass	325
10.74.3 Constructor & Destructor Documentation	325
10.74.3.1 DataEvent() [1/2]	325
10.74.3.2 ~DataEvent()	326
10.74.3.3 DataEvent() [2/2]	326
10.74.4 Member Function Documentation	326
10.74.4.1 CheckEvent()	326
10.74.4.2 GetData()	326
10.74.4.3 GetDataLength()	326
10.74.4.4 GetEventName()	326
10.74.4.5 MakeObject()	327
10.74.4.6 operator=()	327
10.74.4.7 SetData()	327
10.75 gdcm::DataSet Class Reference	327
10.75.1 Detailed Description	329
10.75.2 Member Typedef Documentation	330
10.75.2.1 ConstIterator	330
10.75.2.2 DataElementSet	330
10.75.2.3 Iterator	330
10.75.2.4 SizeType	330
10.75.3 Member Function Documentation	331
10.75.3.1 Begin() [1/2]	331
10.75.3.2 Begin() [2/2]	331

10.75.3.3 Clear()	331
10.75.3.4 ComputeDataElement()	331
10.75.3.5 ComputeGroupLength()	331
10.75.3.6 End() [1/2]	332
10.75.3.7 End() [2/2]	332
10.75.3.8 FindDataElement() [1/2]	332
10.75.3.9 FindDataElement() [2/2]	332
10.75.3.10 FindNextDataElement()	333
10.75.3.11 GetDataElement() [1/2]	333
10.75.3.12 GetDataElement() [2/2]	333
10.75.3.13 GetDEEnd()	334
10.75.3.14 GetDES() [1/2]	334
10.75.3.15 GetDES() [2/2]	334
10.75.3.16 GetLength()	334
10.75.3.17 GetMediaStorage()	334
10.75.3.18 GetPrivateCreator()	334
10.75.3.19 Insert()	335
10.75.3.20 InsertDataElement()	335
10.75.3.21 IsEmpty()	335
10.75.3.22 operator>()	335
10.75.3.23 operator=()	336
10.75.3.24 operator[]()	336
10.75.3.25 Print()	336
10.75.3.26 Read()	336
10.75.3.27 ReadNested()	336
10.75.3.28 ReadSelectedPrivateTags()	337
10.75.3.29 ReadSelectedPrivateTagsWithLength()	337
10.75.3.30 ReadSelectedTags()	337
10.75.3.31 ReadSelectedTagsWithLength()	337
10.75.3.32 ReadUpToTag()	337
10.75.3.33 ReadUpToTagWithLength()	338
10.75.3.34 ReadWithLength()	338
10.75.3.35 Remove()	338
10.75.3.36 Replace()	338
10.75.3.37 ReplaceEmpty()	339
10.75.3.38 Size()	339
10.75.3.39 Write()	339
10.75.4 Friends And Related Function Documentation	339
10.75.4.1 CSAHeader	339

10.75.4.2 operator<<	340
10.76 gdcm::DataSetEvent Class Reference	340
10.76.1 Detailed Description	341
10.76.2 Member Typedef Documentation	342
10.76.2.1 Self	342
10.76.2.2 Superclass	342
10.76.3 Constructor & Destructor Documentation	342
10.76.3.1 DataSetEvent() [1/2]	342
10.76.3.2 ~DataSetEvent()	342
10.76.3.3 DataSetEvent() [2/2]	342
10.76.4 Member Function Documentation	342
10.76.4.1 CheckEvent()	343
10.76.4.2 GetDataSet()	343
10.76.4.3 GetEventName()	343
10.76.4.4 MakeObject()	343
10.76.4.5 operator=()	343
10.76.5 Member Data Documentation	343
10.76.5.1 m_DataSet	344
10.77 gdcm::DataSetHelper Class Reference	344
10.77.1 Detailed Description	344
10.77.2 Member Function Documentation	344
10.77.2.1 ComputeVR()	344
10.78 gdcm::Decoder Class Reference	345
10.78.1 Detailed Description	345
10.78.2 Constructor & Destructor Documentation	345
10.78.2.1 ~Decoder()	346
10.78.3 Member Function Documentation	346
10.78.3.1 CanDecode()	346
10.78.3.2 Decode()	346
10.78.3.3 DecodeByStreams()	346
10.79 gdcm::DefinedTerms Class Reference	347
10.79.1 Detailed Description	347
10.79.2 Constructor & Destructor Documentation	347
10.79.2.1 DefinedTerms()	347
10.80 gdcm::Defs Class Reference	347
10.80.1 Detailed Description	348
10.80.2 Constructor & Destructor Documentation	348
10.80.2.1 Defs() [1/2]	349
10.80.2.2 ~Defs()	349

10.80.2.3 Defs() [2/2] . . . . .	349
10.80.3 Member Function Documentation . . . . .	349
10.80.3.1 GetIODFromFile() . . . . .	349
10.80.3.2 GetIODNameFromMediaStorage() . . . . .	349
10.80.3.3 GetIODs() [1/2] . . . . .	349
10.80.3.4 GetIODs() [2/2] . . . . .	350
10.80.3.5 GetMacros() [1/2] . . . . .	350
10.80.3.6 GetMacros() [2/2] . . . . .	350
10.80.3.7 GetModules() [1/2] . . . . .	350
10.80.3.8 GetModules() [2/2] . . . . .	350
10.80.3.9 GetTypeFromTag() . . . . .	350
10.80.3.10 IsEmpty() . . . . .	350
10.80.3.11 LoadDefaults() . . . . .	351
10.80.3.12 LoadFromFile() . . . . .	351
10.80.3.13 operator=() . . . . .	351
10.80.3.14 Verify() [1/2] . . . . .	351
10.80.3.15 Verify() [2/2] . . . . .	351
10.80.4 Friends And Related Function Documentation . . . . .	351
10.80.4.1 Global . . . . .	351
10.81 gdcmm::DeltaEncodingCodec Class Reference . . . . .	352
10.81.1 Detailed Description . . . . .	353
10.81.2 Constructor & Destructor Documentation . . . . .	353
10.81.2.1 DeltaEncodingCodec() . . . . .	353
10.81.2.2 ~DeltaEncodingCodec() . . . . .	353
10.81.3 Member Function Documentation . . . . .	353
10.81.3.1 CanDecode() . . . . .	353
10.81.3.2 Decode() [1/2] . . . . .	353
10.81.3.3 Decode() [2/2] . . . . .	354
10.82 gdcmm::DICOMDIR Class Reference . . . . .	354
10.82.1 Detailed Description . . . . .	354
10.82.2 Constructor & Destructor Documentation . . . . .	354
10.82.2.1 DICOMDIR() [1/2] . . . . .	354
10.82.2.2 DICOMDIR() [2/2] . . . . .	354
10.83 gdcmm::DICOMDIRGenerator Class Reference . . . . .	355
10.83.1 Detailed Description . . . . .	355
10.83.2 Member Typedef Documentation . . . . .	356
10.83.2.1 FilenamesType . . . . .	356
10.83.2.2 FilenameType . . . . .	356
10.83.3 Constructor & Destructor Documentation . . . . .	356

10.83.3.1 DICOMDIRGenerator()	356
10.83.3.2 ~DICOMDIRGenerator()	357
10.83.4 Member Function Documentation	357
10.83.4.1 AddImageDirectoryRecord()	357
10.83.4.2 AddPatientDirectoryRecord()	357
10.83.4.3 AddSeriesDirectoryRecord()	357
10.83.4.4 AddStudyDirectoryRecord()	357
10.83.4.5 Generate()	357
10.83.4.6 GetFile()	358
10.83.4.7 GetScanner()	358
10.83.4.8 SetDescriptor()	358
10.83.4.9 SetFile()	358
10.83.4.10 SetFilenames()	358
10.83.4.11 SetRootDirectory()	359
10.84 gdcmm::Dict Class Reference	359
10.84.1 Detailed Description	360
10.84.2 Member Typedef Documentation	360
10.84.2.1 ConstIterator	360
10.84.2.2 Iterator	360
10.84.2.3 MapDictEntry	360
10.84.3 Constructor & Destructor Documentation	360
10.84.3.1 Dict() [1/2]	361
10.84.3.2 Dict() [2/2]	361
10.84.4 Member Function Documentation	361
10.84.4.1 AddDictEntry()	361
10.84.4.2 Begin()	361
10.84.4.3 End()	361
10.84.4.4 GetDictEntry()	362
10.84.4.5 GetDictEntryByKeyword()	362
10.84.4.6 GetDictEntryByName()	362
10.84.4.7 GetKeywordFromTag()	362
10.84.4.8 IsEmpty()	363
10.84.4.9 LoadDefault()	363
10.84.4.10 operator=()	363
10.84.5 Friends And Related Function Documentation	363
10.84.5.1 Dicts	363
10.84.5.2 operator<<	363
10.85 gdcmm::DictConverter Class Reference	363
10.85.1 Detailed Description	364

10.85.2 Member Enumeration Documentation	365
10.85.2.1 OutputTypes	365
10.85.3 Constructor & Destructor Documentation	365
10.85.3.1 DictConverter()	365
10.85.3.2 ~DictConverter()	365
10.85.4 Member Function Documentation	365
10.85.4.1 AddGroupLength()	365
10.85.4.2 Convert()	366
10.85.4.3 ConvertToCXX()	366
10.85.4.4 ConvertToXML()	366
10.85.4.5 GetDictName()	366
10.85.4.6 GetInputFilename()	366
10.85.4.7 GetOutputFilename()	366
10.85.4.8 GetOutputType()	366
10.85.4.9 Readuint16()	367
10.85.4.10 ReadVM()	367
10.85.4.11 ReadVR()	367
10.85.4.12 SetDictName()	367
10.85.4.13 SetInputFileName()	367
10.85.4.14 SetOutputFileName()	367
10.85.4.15 SetOutputType()	368
10.85.4.16 WriteFooter()	368
10.85.4.17 WriteHeader()	368
10.86 gdcm::DictEntry Class Reference	368
10.86.1 Detailed Description	369
10.86.2 Constructor & Destructor Documentation	369
10.86.2.1 DictEntry()	369
10.86.3 Member Function Documentation	369
10.86.3.1 GetKeyword()	370
10.86.3.2 GetName()	370
10.86.3.3 GetRetired()	370
10.86.3.4 GetVM()	370
10.86.3.5 GetVR()	371
10.86.3.6 IsUnique()	371
10.86.3.7 SetElementXX()	371
10.86.3.8 SetGroupXX()	371
10.86.3.9 SetKeyword()	371
10.86.3.10 SetName()	372
10.86.3.11 SetRetired()	372



10.86.3.12 SetVM()	372
10.86.3.13 SetVR()	372
10.86.4 Friends And Related Function Documentation	372
10.86.4.1 Dict	372
10.86.4.2 operator<<	373
10.87 gdcmm::DictPrinter Class Reference	373
10.87.1 Detailed Description	374
10.87.2 Constructor & Destructor Documentation	374
10.87.2.1 DictPrinter()	375
10.87.2.2 ~DictPrinter()	375
10.87.3 Member Function Documentation	375
10.87.3.1 Print()	375
10.87.3.2 PrintDataElement2()	375
10.87.3.3 PrintDataSet2()	375
10.88 gdcmm::Dicts Class Reference	376
10.88.1 Detailed Description	377
10.88.2 Member Enumeration Documentation	377
10.88.2.1 ConstructorType	377
10.88.3 Constructor & Destructor Documentation	377
10.88.3.1 Dicts() [1/2]	377
10.88.3.2 ~Dicts()	377
10.88.3.3 Dicts() [2/2]	378
10.88.4 Member Function Documentation	378
10.88.4.1 GetConstructorString()	378
10.88.4.2 GetCSAHeaderDict()	378
10.88.4.3 GetDictEntry() [1/2]	378
10.88.4.4 GetDictEntry() [2/2]	378
10.88.4.5 GetPrivateDict() [1/2]	379
10.88.4.6 GetPrivateDict() [2/2]	379
10.88.4.7 GetPublicDict()	379
10.88.4.8 IsEmpty()	379
10.88.4.9 LoadDefaults()	379
10.88.4.10 operator=()	379
10.88.5 Friends And Related Function Documentation	379
10.88.5.1 Global	380
10.88.5.2 operator<<	380
10.89 gdcmm::network::DIMSE Class Reference	380
10.89.1 Detailed Description	381
10.89.2 Member Enumeration Documentation	381

10.89.2.1 CommandTypes	381
10.90 gdcM::DirectionCosines Class Reference	382
10.90.1 Detailed Description	382
10.90.2 Constructor & Destructor Documentation	383
10.90.2.1 DirectionCosines() [1/2]	383
10.90.2.2 DirectionCosines() [2/2]	383
10.90.2.3 ~DirectionCosines()	383
10.90.3 Member Function Documentation	383
10.90.3.1 ComputeDistAlongNormal()	383
10.90.3.2 Cross()	383
10.90.3.3 CrossDot()	384
10.90.3.4 Dot() [1/2]	384
10.90.3.5 Dot() [2/2]	384
10.90.3.6 IsValid()	384
10.90.3.7 Normalize() [1/2]	384
10.90.3.8 Normalize() [2/2]	385
10.90.3.9 operator const double *()	385
10.90.3.10 Print()	385
10.90.3.11 SetFromString()	385
10.91 gdcM::Directory Class Reference	385
10.91.1 Detailed Description	386
10.91.2 Member Typedef Documentation	387
10.91.2.1 FilenamesType	387
10.91.2.2 FilenameType	387
10.91.3 Constructor & Destructor Documentation	387
10.91.3.1 Directory()	387
10.91.3.2 ~Directory()	387
10.91.4 Member Function Documentation	387
10.91.4.1 Explore()	388
10.91.4.2 GetDirectories()	388
10.91.4.3 GetFilenames()	388
10.91.4.4 GetToplevel()	388
10.91.4.5 Load()	389
10.91.4.6 Print()	389
10.91.5 Friends And Related Function Documentation	389
10.91.5.1 operator<<	389
10.92 gdcM::DirectoryHelper Class Reference	390
10.92.1 Detailed Description	390
10.92.2 Member Function Documentation	390

10.92.2.1 GetCTImageSeriesUIDs()	390
10.92.2.2 GetFileNamesFromSeriesUIDs()	391
10.92.2.3 GetFrameOfReference()	391
10.92.2.4 GetMRIImageSeriesUIDs()	391
10.92.2.5 GetRTStructSeriesUIDs()	391
10.92.2.6 GetSeriesUIDsBySOPClassUID()	391
10.92.2.7 GetSOPClassUID()	392
10.92.2.8 GetStringValueFromTag()	392
10.92.2.9 LoadImageFromFiles()	392
10.92.2.10 RetrieveSOPInstanceUIDFromIndex()	392
10.92.2.11 RetrieveSOPInstanceUIDFromZPosition()	392
10.93 gdcm::DummyValueGenerator Class Reference	392
10.93.1 Detailed Description	393
10.93.2 Member Function Documentation	393
10.93.2.1 Generate()	393
10.94 gdcm::Dumper Class Reference	393
10.94.1 Detailed Description	394
10.94.2 Constructor & Destructor Documentation	394
10.94.2.1 Dumper()	395
10.94.2.2 ~Dumper()	395
10.95 gdcm::Element< TVR, TVM > Class Template Reference	395
10.95.1 Detailed Description	397
10.95.2 Member Typedef Documentation	397
10.95.2.1 Type	397
10.95.3 Member Function Documentation	397
10.95.3.1 GetAsDataElement()	397
10.95.3.2 GetLength()	398
10.95.3.3 GetValue() [1/2]	398
10.95.3.4 GetValue() [2/2]	398
10.95.3.5 GetValues()	398
10.95.3.6 GetVM()	398
10.95.3.7 GetVR()	399
10.95.3.8 operator[]()	399
10.95.3.9 Print()	399
10.95.3.10 Read()	399
10.95.3.11 Set()	399
10.95.3.12 SetFromDataElement()	400
10.95.3.13 SetNoSwap()	400
10.95.3.14 SetValue()	400

10.95.3.15 Write()	400
10.95.4 Member Data Documentation	400
10.95.4.1 Internal	401
10.96 gdcM::Element< TVR, VM::VM1_2 > Class Template Reference	401
10.96.1 Member Typedef Documentation	402
10.96.1.1 Parent	402
10.96.2 Member Function Documentation	402
10.96.2.1 SetLength()	402
10.97 gdcM::Element< TVR, VM::VM1_n > Class Template Reference	403
10.97.1 Member Typedef Documentation	404
10.97.1.1 Type	404
10.97.2 Constructor & Destructor Documentation	404
10.97.2.1 Element() [1/2]	404
10.97.2.2 ~Element()	404
10.97.2.3 Element() [2/2]	404
10.97.3 Member Function Documentation	404
10.97.3.1 GetAsDataElement()	405
10.97.3.2 GetLength()	405
10.97.3.3 GetValue() [1/2]	405
10.97.3.4 GetValue() [2/2]	405
10.97.3.5 GetVM()	405
10.97.3.6 GetVR()	405
10.97.3.7 operator=()	406
10.97.3.8 operator[]()	406
10.97.3.9 Print()	406
10.97.3.10 Read()	406
10.97.3.11 Set()	406
10.97.3.12 SetArray()	407
10.97.3.13 SetFromDataElement()	407
10.97.3.14 SetLength()	407
10.97.3.15 SetNoSwap()	407
10.97.3.16 SetValue()	407
10.97.3.17 Write()	408
10.97.3.18 WriteASCII()	408
10.98 gdcM::Element< TVR, VM::VM2_2n > Class Template Reference	408
10.98.1 Member Typedef Documentation	409
10.98.1.1 Parent	409
10.98.2 Member Function Documentation	409
10.98.2.1 SetLength()	410

10.99 gdcmm::Element< TVR, VM::VM2_n > Class Template Reference . . . . .	410
10.99.1 Member Typedef Documentation . . . . .	411
10.99.1.1 Parent . . . . .	411
10.99.2 Member Function Documentation . . . . .	411
10.99.2.1 SetLength() . . . . .	412
10.100 gdcmm::Element< TVR, VM::VM3_3n > Class Template Reference . . . . .	412
10.100.1 Member Typedef Documentation . . . . .	413
10.100.1.1 Parent . . . . .	413
10.100.2 Member Function Documentation . . . . .	413
10.100.2.1 SetLength() . . . . .	414
10.101 gdcmm::Element< TVR, VM::VM3_n > Class Template Reference . . . . .	414
10.101.1 Member Typedef Documentation . . . . .	415
10.101.1.1 Parent . . . . .	415
10.101.2 Member Function Documentation . . . . .	415
10.101.2.1 SetLength() . . . . .	416
10.102 gdcmm::Element< VR::AS, VM::VM5 > Class Reference . . . . .	416
10.102.1 Member Function Documentation . . . . .	416
10.102.1.1 GetLength() . . . . .	416
10.102.1.2 Print() . . . . .	416
10.102.2 Member Data Documentation . . . . .	416
10.102.2.1 Internal . . . . .	417
10.103 gdcmm::Element< VR::OB, VM::VM1 > Class Reference . . . . .	417
10.104 gdcmm::Element< VR::OW, VM::VM1 > Class Reference . . . . .	418
10.105 gdcmm::ElementDisableCombinations< TVR, TVM > Class Template Reference . . . . .	420
10.105.1 Detailed Description . . . . .	421
10.106 gdcmm::ElementDisableCombinations< VR::OB, VM::VM1_n > Class Reference . . . . .	421
10.107 gdcmm::ElementDisableCombinations< VR::OW, VM::VM1_n > Class Reference . . . . .	421
10.108 gdcmm::EmptyMaskGenerator Class Reference . . . . .	421
10.108.1 Detailed Description . . . . .	422
10.108.2 Member Enumeration Documentation . . . . .	423
10.108.2.1 SOPClassUIDMode . . . . .	423
10.108.3 Constructor & Destructor Documentation . . . . .	423
10.108.3.1 EmptyMaskGenerator() . . . . .	423
10.108.3.2 ~EmptyMaskGenerator() . . . . .	423
10.108.4 Member Function Documentation . . . . .	423
10.108.4.1 Execute() . . . . .	423
10.108.4.2 SetInputDirectory() . . . . .	424
10.108.4.3 SetOutputDirectory() . . . . .	424
10.108.4.4 SetSOPClassUIDMode() . . . . .	424

10.109 gdcM::EncapsulatedDocument Class Reference . . . . .	424
10.109.1 Detailed Description . . . . .	425
10.109.2 Constructor & Destructor Documentation . . . . .	425
10.109.2.1 EncapsulatedDocument() . . . . .	425
10.110 gdcM::EncodingImplementation< T > Class Template Reference . . . . .	425
10.110.1 Detailed Description . . . . .	425
10.111 gdcM::EncodingImplementation< VR::VRASCII > Class Reference . . . . .	426
10.111.1 Member Function Documentation . . . . .	426
10.111.1.1 Read() . . . . .	426
10.111.1.2 ReadComputeLength() . . . . .	426
10.111.1.3 ReadNoSwap() . . . . .	427
10.111.1.4 Write() [1/2] . . . . .	427
10.111.1.5 Write() [2/2] . . . . .	427
10.112 gdcM::EncodingImplementation< VR::VRBINARY > Class Reference . . . . .	427
10.112.1 Member Function Documentation . . . . .	428
10.112.1.1 Read() . . . . .	428
10.112.1.2 ReadComputeLength() . . . . .	428
10.112.1.3 ReadNoSwap() . . . . .	428
10.112.1.4 Write() . . . . .	428
10.113 gdcM::EndEvent Class Reference . . . . .	429
10.114 gdcM::EnumeratedValues Class Reference . . . . .	430
10.114.1 Detailed Description . . . . .	430
10.114.2 Constructor & Destructor Documentation . . . . .	430
10.114.2.1 EnumeratedValues() . . . . .	430
10.115 gdcM::EquipmentManufacturer Class Reference . . . . .	430
10.115.1 Detailed Description . . . . .	431
10.115.2 Member Enumeration Documentation . . . . .	431
10.115.2.1 Type . . . . .	431
10.115.3 Member Function Documentation . . . . .	432
10.115.3.1 Compute() . . . . .	432
10.116 gdcM::Event Class Reference . . . . .	432
10.116.1 Detailed Description . . . . .	434
10.116.2 Constructor & Destructor Documentation . . . . .	434
10.116.2.1 Event() [1/2] . . . . .	434
10.116.2.2 ~Event() . . . . .	434
10.116.2.3 Event() [2/2] . . . . .	434
10.116.3 Member Function Documentation . . . . .	434
10.116.3.1 CheckEvent() . . . . .	434
10.116.3.2 GetEventName() . . . . .	435

10.116.3.3 MakeObject()	435
10.116.3.4 operator=()	435
10.116.3.5 Print()	435
10.117 gdcM::Exception Class Reference	436
10.117.1 Detailed Description	437
10.117.2 Constructor & Destructor Documentation	437
10.117.2.1 Exception()	437
10.117.2.2 ~Exception()	437
10.117.3 Member Function Documentation	437
10.117.3.1 GetDescription()	437
10.117.3.2 what()	438
10.118 gdcM::ExitEvent Class Reference	438
10.119 gdcM::ExplicitDataElement Class Reference	439
10.119.1 Detailed Description	440
10.119.2 Member Function Documentation	441
10.119.2.1 GetLength()	441
10.119.2.2 Read()	441
10.119.2.3 ReadPreValue()	441
10.119.2.4 ReadValue()	441
10.119.2.5 ReadWithLength()	441
10.119.2.6 Write()	442
10.120 gdcM::ExplicitImplicitDataElement Class Reference	442
10.120.1 Detailed Description	443
10.120.2 Member Function Documentation	444
10.120.2.1 GetLength()	444
10.120.2.2 Read()	444
10.120.2.3 ReadPreValue()	444
10.120.2.4 ReadValue()	444
10.120.2.5 ReadWithLength()	444
10.121 gdcM::Fiducials Class Reference	445
10.121.1 Detailed Description	445
10.121.2 Constructor & Destructor Documentation	445
10.121.2.1 Fiducials()	445
10.122 gdcM::File Class Reference	445
10.122.1 Detailed Description	447
10.122.2 Constructor & Destructor Documentation	447
10.122.2.1 File()	447
10.122.2.2 ~File()	448
10.122.3 Member Function Documentation	448

10.122.3.1 GetDataSet() [1/2]	448
10.122.3.2 GetDataSet() [2/2]	448
10.122.3.3 GetHeader() [1/2]	448
10.122.3.4 GetHeader() [2/2]	449
10.122.3.5 Read()	449
10.122.3.6 SetDataSet()	449
10.122.3.7 SetHeader()	449
10.122.3.8 Write()	449
10.122.4 Friends And Related Function Documentation	450
10.122.4.1 operator<<	450
10.123 gdcm::FileAnonymizer Class Reference	450
10.123.1 Detailed Description	451
10.123.2 Constructor & Destructor Documentation	452
10.123.2.1 FileAnonymizer()	452
10.123.2.2 ~FileAnonymizer()	452
10.123.3 Member Function Documentation	452
10.123.3.1 Empty()	452
10.123.3.2 Remove()	453
10.123.3.3 Replace() [1/2]	453
10.123.3.4 Replace() [2/2]	453
10.123.3.5 SetInputFileName()	453
10.123.3.6 SetOutputFileName()	454
10.123.3.7 Write()	454
10.124 gdcm::FileChangeTransferSyntax Class Reference	454
10.124.1 Detailed Description	455
10.124.2 Constructor & Destructor Documentation	456
10.124.2.1 FileChangeTransferSyntax()	456
10.124.2.2 ~FileChangeTransferSyntax()	456
10.124.3 Member Function Documentation	456
10.124.3.1 Change()	456
10.124.3.2 GetCodec()	456
10.124.3.3 New()	457
10.124.3.4 SetInputFileName()	457
10.124.3.5 SetOutputFileName()	457
10.124.3.6 SetTransferSyntax()	457
10.125 gdcm::FileDecompressLookupTable Class Reference	458
10.125.1 Detailed Description	459
10.125.2 Constructor & Destructor Documentation	459
10.125.2.1 FileDecompressLookupTable()	459



---

10.125.2.2 ~FileDecompressLookupTable()	459
10.125.3 Member Function Documentation	459
10.125.3.1 Change()	460
10.125.3.2 GetFile()	460
10.125.3.3 GetPixmap() [1/2]	460
10.125.3.4 GetPixmap() [2/2]	460
10.125.3.5 SetFile()	460
10.125.3.6 SetPixmap()	460
10.126 gdcm::FileDerivation Class Reference	461
10.126.1 Detailed Description	461
10.126.2 Constructor & Destructor Documentation	462
10.126.2.1 FileDerivation()	462
10.126.2.2 ~FileDerivation()	462
10.126.3 Member Function Documentation	462
10.126.3.1 AddDerivationDescription()	462
10.126.3.2 AddPurposeOfReferenceCodeSequence()	462
10.126.3.3 AddReference()	462
10.126.3.4 AddSourceImageSequence()	463
10.126.3.5 Derive()	463
10.126.3.6 GetFile() [1/2]	463
10.126.3.7 GetFile() [2/2]	463
10.126.3.8 SetAppendDerivationHistory()	463
10.126.3.9 SetDerivationCodeSequenceCodeValue()	464
10.126.3.10 SetDerivationDescription()	464
10.126.3.11 SetFile()	464
10.126.3.12 SetPurposeOfReferenceCodeSequenceCodeValue()	464
10.127 gdcm::FileExplicitFilter Class Reference	465
10.127.1 Detailed Description	465
10.127.2 Constructor & Destructor Documentation	466
10.127.2.1 FileExplicitFilter()	466
10.127.2.2 ~FileExplicitFilter()	466
10.127.3 Member Function Documentation	466
10.127.3.1 Change()	466
10.127.3.2 ChangeFMI()	466
10.127.3.3 GetFile()	466
10.127.3.4 ProcessDataSet()	467
10.127.3.5 SetChangePrivateTags()	467
10.127.3.6 SetFile()	467
10.127.3.7 SetRecomputeItemLength()	467

10.127.3.8 SetRecomputeSequenceLength()	467
10.127.3.9 SetUseVRUN()	468
10.128 gdcm::FileMetaInformation Class Reference	468
10.128.1 Detailed Description	470
10.128.2 Constructor & Destructor Documentation	470
10.128.2.1 FileMetaInformation() [1/2]	471
10.128.2.2 ~FileMetaInformation()	471
10.128.2.3 FileMetaInformation() [2/2]	471
10.128.3 Member Function Documentation	471
10.128.3.1 AppendImplementationClassUID()	471
10.128.3.2 ComputeDataSetMediaStorageSOPClass()	471
10.128.3.3 ComputeDataSetTransferSyntax()	471
10.128.3.4 Default()	472
10.128.3.5 FillFromDataSet()	472
10.128.3.6 GetDataSetTransferSyntax()	472
10.128.3.7 GetFileMetaInformationVersion()	472
10.128.3.8 GetFullLength()	472
10.128.3.9 GetGDCMImplementationClassUID()	472
10.128.3.10 GetGDCMImplementationVersionName()	473
10.128.3.11 GetGDCMSourceApplicationEntityTitle()	473
10.128.3.12 GetImplementationClassUID()	473
10.128.3.13 GetImplementationVersionName()	473
10.128.3.14 GetMediaStorage()	473
10.128.3.15 GetMediaStorageAsString()	473
10.128.3.16 GetMetaInformationTS()	473
10.128.3.17 GetPreamble() [1/2]	474
10.128.3.18 GetPreamble() [2/2]	474
10.128.3.19 GetSourceApplicationEntityTitle()	474
10.128.3.20 Insert()	474
10.128.3.21 IsValid()	474
10.128.3.22 Read()	474
10.128.3.23 ReadCompat()	475
10.128.3.24 ReadCompatInternal()	475
10.128.3.25 Replace()	475
10.128.3.26 SetDataSetTransferSyntax()	475
10.128.3.27 SetImplementationClassUID()	476
10.128.3.28 SetImplementationVersionName()	476
10.128.3.29 SetPreamble()	476
10.128.3.30 SetSourceApplicationEntityTitle()	476

---

10.128.3.31 Write()	476
10.128.4 Friends And Related Function Documentation	476
10.128.4.1 operator<<	477
10.128.5 Member Data Documentation	477
10.128.5.1 DataSetMS	477
10.128.5.2 DataSetTS	477
10.128.5.3 MetaInformationTS	477
10.129 gdcM::Filename Class Reference	477
10.129.1 Detailed Description	478
10.129.2 Constructor & Destructor Documentation	478
10.129.2.1 Filename()	478
10.129.3 Member Function Documentation	479
10.129.3.1 EndWith()	479
10.129.3.2 GetExtension()	479
10.129.3.3 GetFileName()	479
10.129.3.4 GetName()	479
10.129.3.5 GetPath()	479
10.129.3.6 IsEmpty()	480
10.129.3.7 IsIdentical()	480
10.129.3.8 Join()	480
10.129.3.9 operator const char *()	480
10.129.3.10 ToUnixSlashes()	480
10.129.3.11 ToWindowsSlashes()	481
10.130 gdcM::FileNameEvent Class Reference	481
10.130.1 Detailed Description	482
10.130.2 Member Typedef Documentation	483
10.130.2.1 Self	483
10.130.2.2 Superclass	483
10.130.3 Constructor & Destructor Documentation	483
10.130.3.1 FileNameEvent() [1/2]	483
10.130.3.2 ~FileNameEvent()	483
10.130.3.3 FileNameEvent() [2/2]	483
10.130.4 Member Function Documentation	483
10.130.4.1 CheckEvent()	484
10.130.4.2 GetEventName()	484
10.130.4.3 GetFileName()	484
10.130.4.4 MakeObject()	484
10.130.4.5 operator=()	484
10.130.4.6 SetFileName()	485

10.131 gdcm::FilenameGenerator Class Reference	485
10.131.1 Detailed Description	486
10.131.2 Member Typedef Documentation	486
10.131.2.1 FilenamesType	486
10.131.2.2 FilenameType	486
10.131.2.3 SizeType	486
10.131.3 Constructor & Destructor Documentation	486
10.131.3.1 FilenameGenerator()	487
10.131.3.2 ~FilenameGenerator()	487
10.131.4 Member Function Documentation	487
10.131.4.1 Generate()	487
10.131.4.2 GetFilename()	487
10.131.4.3 GetFilenames()	487
10.131.4.4 GetNumberOfFilenames()	488
10.131.4.5 GetPattern()	488
10.131.4.6 GetPrefix()	488
10.131.4.7 SetNumberOfFilenames()	488
10.131.4.8 SetPattern()	488
10.131.4.9 SetPrefix()	489
10.132 gdcm::FileSet Class Reference	489
10.132.1 Detailed Description	489
10.132.2 Member Typedef Documentation	489
10.132.2.1 FileType	490
10.132.2.2 FileType	490
10.132.3 Constructor & Destructor Documentation	490
10.132.3.1 FileSet()	490
10.132.4 Member Function Documentation	490
10.132.4.1 AddFile() [1/2]	490
10.132.4.2 AddFile() [2/2]	490
10.132.4.3 GetFiles()	491
10.132.4.4 SetFiles()	491
10.132.5 Friends And Related Function Documentation	491
10.132.5.1 operator<<	491
10.133 gdcm::FileStreamer Class Reference	491
10.133.1 Detailed Description	493
10.133.2 Constructor & Destructor Documentation	493
10.133.2.1 FileStreamer()	493
10.133.2.2 ~FileStreamer()	493
10.133.3 Member Function Documentation	493

10.133.3.1 AppendToDataElement()	493
10.133.3.2 AppendToGroupDataElement()	494
10.133.3.3 CheckDataElement()	494
10.133.3.4 CheckTemplateFileName()	494
10.133.3.5 New()	494
10.133.3.6 ReserveDataElement()	494
10.133.3.7 ReserveGroupDataElement()	495
10.133.3.8 SetOutputFileName()	495
10.133.3.9 SetTemplateFileName()	495
10.133.3.10 StartDataElement()	495
10.133.3.11 StartGroupDataElement()	495
10.133.3.12 StopDataElement()	496
10.133.3.13 StopGroupDataElement()	496
10.134 gdcM::FileWithName Class Reference	496
10.134.1 Detailed Description	497
10.134.2 Constructor & Destructor Documentation	497
10.134.2.1 FileWithName()	498
10.134.3 Member Data Documentation	498
10.134.3.1 filename	498
10.135 gdcM::FindPatientRootQuery Class Reference	498
10.135.1 Detailed Description	499
10.135.2 Constructor & Destructor Documentation	499
10.135.2.1 FindPatientRootQuery()	499
10.135.3 Member Function Documentation	500
10.135.3.1 GetAbstractSyntaxUID()	500
10.135.3.2 GetTagListByLevel()	500
10.135.3.3 InitializeDataSet()	500
10.135.3.4 ValidateQuery()	500
10.135.4 Friends And Related Function Documentation	501
10.135.4.1 QueryFactory	501
10.136 gdcM::FindStudyRootQuery Class Reference	501
10.136.1 Detailed Description	502
10.136.2 Constructor & Destructor Documentation	502
10.136.2.1 FindStudyRootQuery()	502
10.136.3 Member Function Documentation	503
10.136.3.1 GetAbstractSyntaxUID()	503
10.136.3.2 GetTagListByLevel()	503
10.136.3.3 InitializeDataSet()	503
10.136.3.4 ValidateQuery()	503

10.136.4 Friends And Related Function Documentation	504
10.136.4.1 QueryFactory	504
10.137 gdcmm::Fragment Class Reference	504
10.137.1 Detailed Description	505
10.137.2 Constructor & Destructor Documentation	506
10.137.2.1 Fragment()	506
10.137.3 Member Function Documentation	506
10.137.3.1 ComputeLength()	506
10.137.3.2 GetLength()	506
10.137.3.3 Read()	506
10.137.3.4 ReadBacktrack()	506
10.137.3.5 ReadPreValue()	507
10.137.3.6 ReadValue()	507
10.137.3.7 Write()	507
10.137.4 Friends And Related Function Documentation	507
10.137.4.1 operator<<	507
10.138 gdcmm::Global Class Reference	507
10.138.1 Detailed Description	508
10.138.2 Constructor & Destructor Documentation	508
10.138.2.1 Global() [1/2]	509
10.138.2.2 ~Global()	509
10.138.2.3 Global() [2/2]	509
10.138.3 Member Function Documentation	509
10.138.3.1 Append()	509
10.138.3.2 GetDefs()	509
10.138.3.3 GetDicts() [1/2]	510
10.138.3.4 GetDicts() [2/2]	510
10.138.3.5 GetInstance()	510
10.138.3.6 LoadResourcesFiles()	510
10.138.3.7 Locate()	511
10.138.3.8 operator=()	511
10.138.3.9 Prepend()	511
10.138.4 Friends And Related Function Documentation	511
10.138.4.1 operator<<	511
10.139 gdcmm::GroupDict Class Reference	511
10.139.1 Detailed Description	512
10.139.2 Member Typedef Documentation	512
10.139.2.1 GroupStringVector	512
10.139.3 Constructor & Destructor Documentation	512

10.139.3.1 GroupDict()	513
10.139.3.2 ~GroupDict()	513
10.139.4 Member Function Documentation	513
10.139.4.1 Add()	513
10.139.4.2 GetAbbreviation()	513
10.139.4.3 GetName()	513
10.139.4.4 Insert()	513
10.139.4.5 Size()	514
10.139.5 Friends And Related Function Documentation	514
10.139.5.1 operator<<	514
10.140 gdcm::IconImageFilter Class Reference	514
10.140.1 Detailed Description	515
10.140.2 Constructor & Destructor Documentation	515
10.140.2.1 IconImageFilter()	515
10.140.2.2 ~IconImageFilter()	515
10.140.3 Member Function Documentation	516
10.140.3.1 Extract()	516
10.140.3.2 ExtractIconImages()	516
10.140.3.3 ExtractVeprolIconImages()	516
10.140.3.4 GetFile() [1/2]	516
10.140.3.5 GetFile() [2/2]	516
10.140.3.6 GetIconImage()	517
10.140.3.7 GetNumberOfIconImages()	517
10.140.3.8 SetFile()	517
10.141 gdcm::IconImageGenerator Class Reference	517
10.141.1 Detailed Description	518
10.141.2 Constructor & Destructor Documentation	518
10.141.2.1 IconImageGenerator()	519
10.141.2.2 ~IconImageGenerator()	519
10.141.3 Member Function Documentation	519
10.141.3.1 AutoPixelMinMax()	519
10.141.3.2 ConvertRGBToPaletteColor()	519
10.141.3.3 Generate()	519
10.141.3.4 GetIconImage()	520
10.141.3.5 GetPixmap() [1/2]	520
10.141.3.6 GetPixmap() [2/2]	520
10.141.3.7 SetOutputDimensions()	520
10.141.3.8 SetOutsideValuePixel()	520
10.141.3.9 SetPixelMinMax()	521

10.141.3.10 SetPixmap()	521
10.142 gdcm::ignore_char Struct Reference	521
10.142.1 Constructor & Destructor Documentation	521
10.142.1.1 ignore_char()	522
10.142.2 Member Data Documentation	522
10.142.2.1 m_char	522
10.143 gdcm::Image Class Reference	522
10.143.1 Detailed Description	524
10.143.2 Constructor & Destructor Documentation	524
10.143.2.1 Image()	524
10.143.2.2 ~Image()	525
10.143.3 Member Function Documentation	525
10.143.3.1 GetDirectionCosines() [1/2]	525
10.143.3.2 GetDirectionCosines() [2/2]	525
10.143.3.3 GetIntercept()	525
10.143.3.4 GetOrigin() [1/2]	525
10.143.3.5 GetOrigin() [2/2]	526
10.143.3.6 GetSlope()	526
10.143.3.7 GetSpacing() [1/2]	526
10.143.3.8 GetSpacing() [2/2]	526
10.143.3.9 Print()	526
10.143.3.10 SetDirectionCosines() [1/3]	527
10.143.3.11 SetDirectionCosines() [2/3]	527
10.143.3.12 SetDirectionCosines() [3/3]	527
10.143.3.13 SetIntercept()	527
10.143.3.14 SetOrigin() [1/3]	527
10.143.3.15 SetOrigin() [2/3]	528
10.143.3.16 SetOrigin() [3/3]	528
10.143.3.17 SetSlope()	528
10.143.3.18 SetSpacing() [1/2]	528
10.143.3.19 SetSpacing() [2/2]	528
10.144 gdcm::ImageApplyLookupTable Class Reference	529
10.144.1 Detailed Description	531
10.144.2 Constructor & Destructor Documentation	531
10.144.2.1 ImageApplyLookupTable()	531
10.144.2.2 ~ImageApplyLookupTable()	531
10.144.3 Member Function Documentation	531
10.144.3.1 Apply()	531
10.144.3.2 SetRGB8()	531



10.145	gdcm::ImageChangePhotometricInterpretation Class Reference	532
10.145.1	Detailed Description	534
10.145.2	Constructor & Destructor Documentation	534
10.145.2.1	ImageChangePhotometricInterpretation()	534
10.145.2.2	~ImageChangePhotometricInterpretation()	534
10.145.3	Member Function Documentation	534
10.145.3.1	Change()	535
10.145.3.2	ChangeMonochrome()	535
10.145.3.3	ChangeRGB2YBR()	535
10.145.3.4	ChangeYBR2RGB()	535
10.145.3.5	GetPhotometricInterpretation()	535
10.145.3.6	RGB2YBR()	535
10.145.3.7	SetPhotometricInterpretation()	536
10.145.3.8	YBR2RGB()	536
10.146	gdcm::ImageChangePlanarConfiguration Class Reference	536
10.146.1	Detailed Description	538
10.146.2	Constructor & Destructor Documentation	538
10.146.2.1	ImageChangePlanarConfiguration()	538
10.146.2.2	~ImageChangePlanarConfiguration()	538
10.146.3	Member Function Documentation	538
10.146.3.1	Change()	538
10.146.3.2	GetPlanarConfiguration()	539
10.146.3.3	RGBPixelsToRGBPlanes()	539
10.146.3.4	RGBPlanesToRGBPixels()	539
10.146.3.5	SetPlanarConfiguration()	539
10.147	gdcm::ImageChangeTransferSyntax Class Reference	540
10.147.1	Detailed Description	542
10.147.2	Constructor & Destructor Documentation	542
10.147.2.1	ImageChangeTransferSyntax()	542
10.147.2.2	~ImageChangeTransferSyntax()	542
10.147.3	Member Function Documentation	543
10.147.3.1	Change()	543
10.147.3.2	GetTransferSyntax()	543
10.147.3.3	SetCompressIconImage()	543
10.147.3.4	SetForce()	543
10.147.3.5	SetTransferSyntax()	544
10.147.3.6	SetUserCodec()	544
10.147.3.7	TryJPEG2000Codec()	544
10.147.3.8	TryJPEGCodec()	544

10.147.3.9 TryJPEGLSCodec()	545
10.147.3.10 TryRAWCodec()	545
10.147.3.11 TryRLECodec()	545
10.148 gdcmm::ImageCodec Class Reference	545
10.148.1 Detailed Description	547
10.148.2 Member Typedef Documentation	548
10.148.2.1 LUTPtr	548
10.148.3 Constructor & Destructor Documentation	548
10.148.3.1 ImageCodec()	548
10.148.3.2 ~ImageCodec()	548
10.148.4 Member Function Documentation	548
10.148.4.1 AppendFrameEncode()	548
10.148.4.2 AppendRowEncode()	549
10.148.4.3 CanCode()	549
10.148.4.4 CanDecode()	549
10.148.4.5 CleanupUnusedBits()	549
10.148.4.6 Clone()	550
10.148.4.7 Decode()	550
10.148.4.8 DecodeByStreams()	550
10.148.4.9 DoByteSwap()	550
10.148.4.10 DoInvertMonochrome()	551
10.148.4.11 DoOverlayCleanup()	551
10.148.4.12 DoPaddedCompositePixelCode()	551
10.148.4.13 DoPlanarConfiguration()	551
10.148.4.14 DoSimpleCopy()	551
10.148.4.15 DoYBR()	551
10.148.4.16 DoYBRFull422()	552
10.148.4.17 GetDimensions()	552
10.148.4.18 GetHeaderInfo()	552
10.148.4.19 GetLossyFlag()	552
10.148.4.20 GetLUT()	552
10.148.4.21 GetNeedByteSwap()	552
10.148.4.22 GetNumberOfDimensions()	553
10.148.4.23 GetPhotometricInterpretation()	553
10.148.4.24 GetPixelFormat() [1/2]	553
10.148.4.25 GetPixelFormat() [2/2]	553
10.148.4.26 GetPlanarConfiguration()	553
10.148.4.27 IsFrameEncoder()	553
10.148.4.28 IsLossy()	554

10.148.4.29 IsRowEncoder()	554
10.148.4.30 IsValid()	554
10.148.4.31 SetDimensions() [1/2]	554
10.148.4.32 SetDimensions() [2/2]	554
10.148.4.33 SetLossyFlag()	555
10.148.4.34 SetLUT()	555
10.148.4.35 SetNeedByteSwap()	555
10.148.4.36 SetNeedOverlayCleanup()	555
10.148.4.37 SetNumberOfDimensions()	555
10.148.4.38 SetPhotometricInterpretation()	556
10.148.4.39 SetPixelFormat()	556
10.148.4.40 SetPlanarConfiguration()	556
10.148.4.41 StartEncode()	556
10.148.4.42 StopEncode()	556
10.148.5 Friends And Related Function Documentation	557
10.148.5.1 FileChangeTransferSyntax	557
10.148.5.2 ImageChangePhotometricInterpretation	557
10.148.6 Member Data Documentation	557
10.148.6.1 Dimensions	557
10.148.6.2 LossyFlag	557
10.148.6.3 LUT	557
10.148.6.4 NeedByteSwap	558
10.148.6.5 NeedOverlayCleanup	558
10.148.6.6 NumberOfDimensions	558
10.148.6.7 PF	558
10.148.6.8 PI	558
10.148.6.9 PlanarConfiguration	558
10.148.6.10 RequestPaddedCompositePixelCode	558
10.148.6.11 RequestPlanarConfiguration	559
10.149 gdcm::ImageConverter Class Reference	559
10.149.1 Detailed Description	559
10.149.2 Constructor & Destructor Documentation	559
10.149.2.1 ImageConverter()	559
10.149.2.2 ~ImageConverter()	560
10.149.3 Member Function Documentation	560
10.149.3.1 Convert()	560
10.149.3.2 GetOutput()	560
10.149.3.3 SetInput()	560
10.150 gdcm::ImageFragmentSplitter Class Reference	561

10.150.1 Detailed Description	563
10.150.2 Constructor & Destructor Documentation	563
10.150.2.1 ImageFragmentSplitter()	563
10.150.2.2 ~ImageFragmentSplitter()	563
10.150.3 Member Function Documentation	563
10.150.3.1 GetFragmentSizeMax()	563
10.150.3.2 SetForce()	563
10.150.3.3 SetFragmentSizeMax()	564
10.150.3.4 Split()	564
10.151 gdcm::ImageHelper Class Reference	564
10.151.1 Detailed Description	565
10.151.2 Member Function Documentation	565
10.151.2.1 ComputeMediaStorageFromModality()	565
10.151.2.2 ComputeSpacingFromImagePositionPatient()	566
10.151.2.3 GetDimensionsValue()	566
10.151.2.4 GetDirectionCosinesFromDataSet()	566
10.151.2.5 GetDirectionCosinesValue()	566
10.151.2.6 GetForcePixelSpacing()	566
10.151.2.7 GetForceRescaleInterceptSlope()	567
10.151.2.8 GetLUT()	567
10.151.2.9 GetOriginValue()	567
10.151.2.10 GetPhotometricInterpretationValue()	567
10.151.2.11 GetPixelFormatValue()	567
10.151.2.12 GetPlanarConfigurationValue()	567
10.151.2.13 GetPMSRescaleInterceptSlope()	568
10.151.2.14 GetPointerFromElement()	568
10.151.2.15 GetRealWorldValueMappingContent()	568
10.151.2.16 GetRescaleInterceptSlopeValue()	568
10.151.2.17 GetSpacingTagFromMediaStorage()	568
10.151.2.18 GetSpacingValue()	569
10.151.2.19 GetZSpacingTagFromMediaStorage()	569
10.151.2.20 SetDimensionsValue()	569
10.151.2.21 SetDirectionCosinesValue()	569
10.151.2.22 SetForcePixelSpacing()	569
10.151.2.23 SetForceRescaleInterceptSlope()	570
10.151.2.24 SetOriginValue()	570
10.151.2.25 SetPMSRescaleInterceptSlope()	570
10.151.2.26 SetRescaleInterceptSlopeValue()	570
10.151.2.27 SetSpacingValue()	570

10.152 gdcm::ImageReader Class Reference . . . . .	571
10.152.1 Detailed Description . . . . .	573
10.152.2 Constructor & Destructor Documentation . . . . .	573
10.152.2.1 ImageReader() . . . . .	573
10.152.2.2 ~ImageReader() . . . . .	573
10.152.3 Member Function Documentation . . . . .	573
10.152.3.1 GetImage() [1/2] . . . . .	573
10.152.3.2 GetImage() [2/2] . . . . .	574
10.152.3.3 Read() . . . . .	574
10.152.3.4 ReadACRNEMAIImage() . . . . .	574
10.152.3.5 ReadImage() . . . . .	575
10.153 gdcm::ImageRegionReader Class Reference . . . . .	575
10.153.1 Detailed Description . . . . .	577
10.153.2 Constructor & Destructor Documentation . . . . .	577
10.153.2.1 ImageRegionReader() . . . . .	577
10.153.2.2 ~ImageRegionReader() . . . . .	577
10.153.3 Member Function Documentation . . . . .	577
10.153.3.1 ComputeBufferLength() . . . . .	578
10.153.3.2 GetRegion() . . . . .	578
10.153.3.3 Read() . . . . .	578
10.153.3.4 ReadInformation() . . . . .	578
10.153.3.5 ReadIntoBuffer() . . . . .	579
10.153.3.6 SetRegion() . . . . .	579
10.154 gdcm::ImageToImageFilter Class Reference . . . . .	579
10.154.1 Detailed Description . . . . .	580
10.154.2 Constructor & Destructor Documentation . . . . .	581
10.154.2.1 ImageToImageFilter() . . . . .	581
10.154.2.2 ~ImageToImageFilter() . . . . .	581
10.154.3 Member Function Documentation . . . . .	581
10.154.3.1 GetInput() . . . . .	581
10.154.3.2 GetOutput() . . . . .	581
10.155 gdcm::ImageWriter Class Reference . . . . .	582
10.155.1 Detailed Description . . . . .	584
10.155.2 Constructor & Destructor Documentation . . . . .	584
10.155.2.1 ImageWriter() . . . . .	584
10.155.2.2 ~ImageWriter() . . . . .	584
10.155.3 Member Function Documentation . . . . .	584
10.155.3.1 ComputeTargetMediaStorage() . . . . .	585
10.155.3.2 GetImage() [1/2] . . . . .	585

10.155.3.3 GetImage() [2/2]	585
10.155.3.4 Write()	585
10.156 gdcm::network::ImplementationClassUIDSub Class Reference	586
10.156.1 Detailed Description	586
10.156.2 Constructor & Destructor Documentation	586
10.156.2.1 ImplementationClassUIDSub()	586
10.156.3 Member Function Documentation	586
10.156.3.1 Print()	586
10.156.3.2 Read()	587
10.156.3.3 Size()	587
10.156.3.4 Write()	587
10.157 gdcm::network::ImplementationUIDSub Class Reference	587
10.157.1 Detailed Description	587
10.157.2 Constructor & Destructor Documentation	587
10.157.2.1 ImplementationUIDSub()	588
10.157.3 Member Function Documentation	588
10.157.3.1 Write()	588
10.158 gdcm::network::ImplementationVersionNameSub Class Reference	588
10.158.1 Detailed Description	588
10.158.2 Constructor & Destructor Documentation	588
10.158.2.1 ImplementationVersionNameSub()	589
10.158.3 Member Function Documentation	589
10.158.3.1 Print()	589
10.158.3.2 Read()	589
10.158.3.3 Size()	589
10.158.3.4 Write()	589
10.159 gdcm::ImplicitDataElement Class Reference	590
10.159.1 Detailed Description	591
10.159.2 Member Function Documentation	591
10.159.2.1 GetLength()	591
10.159.2.2 Read()	591
10.159.2.3 ReadPreValue()	591
10.159.2.4 ReadValue()	592
10.159.2.5 ReadValueWithLength()	592
10.159.2.6 ReadWithLength()	592
10.159.2.7 Write()	592
10.160 gdcm::InitializeEvent Class Reference	593
10.161 gdcm::IOD Class Reference	594
10.161.1 Detailed Description	594

10.161.2 Member Typedef Documentation	594
10.161.2.1 MapIODEntry	595
10.161.2.2 SizeType	595
10.161.3 Constructor & Destructor Documentation	595
10.161.3.1 IOD()	595
10.161.4 Member Function Documentation	595
10.161.4.1 AddIODEntry()	595
10.161.4.2 Clear()	595
10.161.4.3 GetIODEntry()	595
10.161.4.4 GetNumberOfIODs()	596
10.161.4.5 GetTypeFromTag()	596
10.161.5 Friends And Related Function Documentation	596
10.161.5.1 operator<<	596
10.162 gdcmm::IODEntry Class Reference	596
10.162.1 Detailed Description	597
10.162.2 Constructor & Destructor Documentation	597
10.162.2.1 IODEntry()	597
10.162.3 Member Function Documentation	597
10.162.3.1 GetIE()	598
10.162.3.2 GetName()	598
10.162.3.3 GetRef()	598
10.162.3.4 GetUsage()	598
10.162.3.5 GetUsageType()	598
10.162.3.6 SetIE()	598
10.162.3.7 SetName()	598
10.162.3.8 SetRef()	599
10.162.3.9 SetUsage()	599
10.162.4 Friends And Related Function Documentation	599
10.162.4.1 operator<<	599
10.163 gdcmm::IODs Class Reference	599
10.163.1 Detailed Description	600
10.163.2 Member Typedef Documentation	600
10.163.2.1 IODMapType	600
10.163.2.2 IODMapTypeConstIterator	600
10.163.2.3 IODName	601
10.163.3 Constructor & Destructor Documentation	601
10.163.3.1 IODs()	601
10.163.4 Member Function Documentation	601
10.163.4.1 AddIOD()	601

10.163.4.2 Begin()	601
10.163.4.3 Clear()	601
10.163.4.4 End()	602
10.163.4.5 GetIOD()	602
10.163.5 Friends And Related Function Documentation	602
10.163.5.1 operator<<	602
10.164 gdcmm::IPPSorter Class Reference	602
10.164.1 Detailed Description	603
10.164.2 Constructor & Destructor Documentation	604
10.164.2.1 IPPSorter()	604
10.164.3 Member Function Documentation	604
10.164.3.1 GetDirectionCosinesTolerance()	604
10.164.3.2 GetZSpacing()	605
10.164.3.3 GetZSpacingTolerance()	605
10.164.3.4 SetComputeZSpacing()	605
10.164.3.5 SetDirectionCosinesTolerance()	605
10.164.3.6 SetDropDuplicatePositions()	606
10.164.3.7 SetZSpacingTolerance()	606
10.164.3.8 Sort()	606
10.164.4 Member Data Documentation	606
10.164.4.1 ComputeZSpacing	606
10.164.4.2 DirCosTolerance	607
10.164.4.3 DropDuplicatePositions	607
10.164.4.4 ZSpacing	607
10.164.4.5 ZTolerance	607
10.165 gdcmm::Item Class Reference	607
10.165.1 Detailed Description	609
10.165.2 Constructor & Destructor Documentation	609
10.165.2.1 Item() [1/2]	609
10.165.2.2 Item() [2/2]	609
10.165.3 Member Function Documentation	609
10.165.3.1 Clear()	610
10.165.3.2 FindDataElement()	610
10.165.3.3 GetDataElement()	610
10.165.3.4 GetLength()	610
10.165.3.5 GetNestedDataSet() [1/2]	610
10.165.3.6 GetNestedDataSet() [2/2]	610
10.165.3.7 InsertDataElement()	611
10.165.3.8 Read()	611



---

10.165.3.9 SetNestedDataSet()	611
10.165.3.10 Write()	611
10.165.4 Friends And Related Function Documentation	611
10.165.4.1 operator<<	611
10.166 gdcm::IterationEvent Class Reference	612
10.167 gdcm::JPEG12Codec Class Reference	613
10.167.1 Detailed Description	614
10.167.2 Constructor & Destructor Documentation	614
10.167.2.1 JPEG12Codec()	614
10.167.2.2 ~JPEG12Codec()	614
10.167.3 Member Function Documentation	614
10.167.3.1 DecodeByStreams()	615
10.167.3.2 EncodeBuffer()	615
10.167.3.3 GetHeaderInfo()	615
10.167.3.4 InternalCode()	615
10.167.3.5 IsStateSuspension()	615
10.168 gdcm::JPEG16Codec Class Reference	616
10.168.1 Detailed Description	617
10.168.2 Constructor & Destructor Documentation	617
10.168.2.1 JPEG16Codec()	617
10.168.2.2 ~JPEG16Codec()	617
10.168.3 Member Function Documentation	617
10.168.3.1 DecodeByStreams()	618
10.168.3.2 EncodeBuffer()	618
10.168.3.3 GetHeaderInfo()	618
10.168.3.4 InternalCode()	618
10.168.3.5 IsStateSuspension()	618
10.169 gdcm::JPEG2000Codec Class Reference	619
10.169.1 Detailed Description	620
10.169.2 Constructor & Destructor Documentation	621
10.169.2.1 JPEG2000Codec()	621
10.169.2.2 ~JPEG2000Codec()	621
10.169.3 Member Function Documentation	621
10.169.3.1 AppendFrameEncode()	621
10.169.3.2 AppendRowEncode()	621
10.169.3.3 CanCode()	622
10.169.3.4 CanDecode()	622
10.169.3.5 Clone()	622
10.169.3.6 Code()	622

10.169.3.7 Decode()	623
10.169.3.8 DecodeByStreams()	623
10.169.3.9 DecodeExtent()	623
10.169.3.10 GetHeaderInfo()	623
10.169.3.11 GetQuality()	624
10.169.3.12 GetRate()	624
10.169.3.13 IsFrameEncoder()	624
10.169.3.14 IsRowEncoder()	624
10.169.3.15 SetNumberOfResolutions()	624
10.169.3.16 SetNumberOfThreadsForDecompression()	624
10.169.3.17 SetQuality()	625
10.169.3.18 SetRate()	625
10.169.3.19 SetReversible()	625
10.169.3.20 SetTileSize()	625
10.169.3.21 StartEncode()	625
10.169.3.22 StopEncode()	626
10.169.4 Friends And Related Function Documentation	626
10.169.4.1 Bitmap	626
10.169.4.2 ImageRegionReader	626
10.170 gdcmm::JPEG8Codec Class Reference	626
10.170.1 Detailed Description	628
10.170.2 Constructor & Destructor Documentation	628
10.170.2.1 JPEG8Codec()	628
10.170.2.2 ~JPEG8Codec()	628
10.170.3 Member Function Documentation	628
10.170.3.1 DecodeByStreams()	628
10.170.3.2 EncodeBuffer()	629
10.170.3.3 GetHeaderInfo()	629
10.170.3.4 InternalCode()	629
10.170.3.5 IsStateSuspension()	629
10.171 gdcmm::JPEGCodec Class Reference	630
10.171.1 Detailed Description	632
10.171.2 Constructor & Destructor Documentation	632
10.171.2.1 JPEGCodec()	632
10.171.2.2 ~JPEGCodec()	632
10.171.3 Member Function Documentation	632
10.171.3.1 AppendFrameEncode()	633
10.171.3.2 AppendRowEncode()	633
10.171.3.3 CanCode()	633

10.171.3.4 CanDecode()	633
10.171.3.5 Clone()	634
10.171.3.6 Code()	634
10.171.3.7 ComputeOffsetTable()	634
10.171.3.8 Decode()	634
10.171.3.9 DecodeByStreams()	634
10.171.3.10 DecodeExtent()	635
10.171.3.11 EncodeBuffer()	635
10.171.3.12 GetHeaderInfo()	635
10.171.3.13 GetLossless()	635
10.171.3.14 GetQuality()	636
10.171.3.15 IsFrameEncoder()	636
10.171.3.16 IsRowEncoder()	636
10.171.3.17 IsStateSuspension()	636
10.171.3.18 IsValid()	636
10.171.3.19 SetBitSample()	636
10.171.3.20 SetLossless()	637
10.171.3.21 SetPixelFormat()	637
10.171.3.22 SetQuality()	637
10.171.3.23 StartEncode()	637
10.171.3.24 StopEncode()	637
10.171.4 Friends And Related Function Documentation	638
10.171.4.1 ImageRegionReader	638
10.171.5 Member Data Documentation	638
10.171.5.1 BitSample	638
10.171.5.2 Quality	638
10.172 gdcm::JPEGLSCodec Class Reference	638
10.172.1 Detailed Description	640
10.172.2 Constructor & Destructor Documentation	640
10.172.2.1 JPEGLSCodec()	641
10.172.2.2 ~JPEGLSCodec()	641
10.172.3 Member Function Documentation	641
10.172.3.1 AppendFrameEncode()	641
10.172.3.2 AppendRowEncode()	641
10.172.3.3 CanCode()	641
10.172.3.4 CanDecode()	642
10.172.3.5 Clone()	642
10.172.3.6 Code()	642
10.172.3.7 Decode() [1/2]	642

10.172.3.8 Decode() [2/2]	643
10.172.3.9 DecodeExtent()	643
10.172.3.10 GetBufferLength()	643
10.172.3.11 GetHeaderInfo()	643
10.172.3.12 GetLossless()	643
10.172.3.13 IsFrameEncoder()	644
10.172.3.14 IsRowEncoder()	644
10.172.3.15 SetBufferLength()	644
10.172.3.16 SetLossless()	644
10.172.3.17 SetLossyError()	644
10.172.3.18 StartEncode()	644
10.172.3.19 StopEncode()	645
10.172.4 Friends And Related Function Documentation	645
10.172.4.1 ImageRegionReader	645
10.173 gdcmm::JSON Class Reference	645
10.173.1 Detailed Description	645
10.173.2 Constructor & Destructor Documentation	645
10.173.2.1 JSON()	646
10.173.2.2 ~JSON()	646
10.173.3 Member Function Documentation	646
10.173.3.1 Code()	646
10.173.3.2 Decode()	646
10.173.3.3 GetPrettyPrint()	646
10.173.3.4 PrettyPrintOff()	647
10.173.3.5 PrettyPrintOn()	647
10.173.3.6 SetPrettyPrint()	647
10.174 gdcmm::KAKADUCodec Class Reference	647
10.174.1 Detailed Description	649
10.174.2 Constructor & Destructor Documentation	649
10.174.2.1 KAKADUCodec()	649
10.174.2.2 ~KAKADUCodec()	649
10.174.3 Member Function Documentation	649
10.174.3.1 CanCode()	649
10.174.3.2 CanDecode()	649
10.174.3.3 Clone()	650
10.174.3.4 Code()	650
10.174.3.5 Decode()	650
10.175 gdcmm::LO Class Reference	651
10.175.1 Detailed Description	652

10.175.2 Member Typedef Documentation	652
10.175.2.1 const_iterator	652
10.175.2.2 const_reference	652
10.175.2.3 const_reverse_iterator	652
10.175.2.4 difference_type	652
10.175.2.5 iterator	653
10.175.2.6 pointer	653
10.175.2.7 reference	653
10.175.2.8 reverse_iterator	653
10.175.2.9 size_type	653
10.175.2.10 Superclass	653
10.175.2.11 value_type	653
10.175.3 Constructor & Destructor Documentation	654
10.175.3.1 LO() [1/4]	654
10.175.3.2 LO() [2/4]	654
10.175.3.3 LO() [3/4]	654
10.175.3.4 LO() [4/4]	654
10.175.4 Member Function Documentation	654
10.175.4.1 IsValid()	654
10.176 gdcm::LookupTable Class Reference	655
10.176.1 Detailed Description	657
10.176.2 Member Enumeration Documentation	657
10.176.2.1 LookupTableType	657
10.176.3 Constructor & Destructor Documentation	657
10.176.3.1 LookupTable() [1/2]	657
10.176.3.2 ~LookupTable()	657
10.176.3.3 LookupTable() [2/2]	658
10.176.4 Member Function Documentation	658
10.176.4.1 Allocate()	658
10.176.4.2 Clear()	658
10.176.4.3 Decode() [1/2]	658
10.176.4.4 Decode() [2/2]	658
10.176.4.5 Decode8()	659
10.176.4.6 GetBitSample()	659
10.176.4.7 GetBufferAsRGBA()	659
10.176.4.8 GetLUT()	659
10.176.4.9 GetLUTDescriptor()	660
10.176.4.10 GetLUTLength()	660
10.176.4.11 GetPointer()	660

10.176.4.12 InitializeBlueLUT()	660
10.176.4.13 Initialized()	660
10.176.4.14 InitializeGreenLUT()	661
10.176.4.15 InitializeLUT()	661
10.176.4.16 InitializeRedLUT()	661
10.176.4.17 IsRGB8()	661
10.176.4.18 Print()	661
10.176.4.19 SetBlueLUT()	662
10.176.4.20 SetGreenLUT()	662
10.176.4.21 SetLUT()	662
10.176.4.22 SetRedLUT()	662
10.176.4.23 WriteBufferAsRGBA()	662
10.176.5 Member Data Documentation	662
10.176.5.1 BitSample	663
10.176.5.2 IncompleteLUT	663
10.176.5.3 Internal	663
10.177 gdcmm::Scanner::ltstr Struct Reference	663
10.177.1 Member Function Documentation	663
10.177.1.1 operator>()	663
10.178 gdcmm::StrictScanner::ltstr Struct Reference	664
10.178.1 Member Function Documentation	664
10.178.1.1 operator>()	664
10.179 gdcmm::Macro Class Reference	664
10.179.1 Detailed Description	665
10.179.2 Member Typedef Documentation	665
10.179.2.1 ArrayIncludeMacrosType	665
10.179.2.2 MapModuleEntry	665
10.179.3 Constructor & Destructor Documentation	665
10.179.3.1 Macro()	665
10.179.4 Member Function Documentation	665
10.179.4.1 AddMacroEntry()	666
10.179.4.2 Clear()	666
10.179.4.3 FindMacroEntry()	666
10.179.4.4 GetMacroEntry()	666
10.179.4.5 GetName()	666
10.179.4.6 SetName()	666
10.179.4.7 Verify()	667
10.179.5 Friends And Related Function Documentation	667
10.179.5.1 operator<<	667

10.180 gdcmmacros Class Reference . . . . .	667
10.180.1 Detailed Description . . . . .	668
10.180.2 Member Typedef Documentation . . . . .	668
10.180.2.1 ModuleMapType . . . . .	668
10.180.3 Constructor & Destructor Documentation . . . . .	668
10.180.3.1 Macros() . . . . .	668
10.180.4 Member Function Documentation . . . . .	668
10.180.4.1 AddMacro() . . . . .	668
10.180.4.2 Clear() . . . . .	669
10.180.4.3 GetMacro() . . . . .	669
10.180.4.4 IsEmpty() . . . . .	669
10.180.5 Friends And Related Function Documentation . . . . .	669
10.180.5.1 operator<< . . . . .	669
10.181 gdcmmacros::network::MaximumLengthSub Class Reference . . . . .	669
10.181.1 Detailed Description . . . . .	670
10.181.2 Constructor & Destructor Documentation . . . . .	670
10.181.2.1 MaximumLengthSub() . . . . .	670
10.181.3 Member Function Documentation . . . . .	670
10.181.3.1 GetMaximumLength() . . . . .	670
10.181.3.2 Print() . . . . .	670
10.181.3.3 Read() . . . . .	671
10.181.3.4 SetMaximumLength() . . . . .	671
10.181.3.5 Size() . . . . .	671
10.181.3.6 Write() . . . . .	671
10.182 gdcmmacros::MD5 Class Reference . . . . .	671
10.182.1 Detailed Description . . . . .	672
10.182.2 Member Function Documentation . . . . .	672
10.182.2.1 Compute() . . . . .	672
10.182.2.2 ComputeFile() . . . . .	672
10.183 gdcmmacros::MediaStorage Class Reference . . . . .	672
10.183.1 Detailed Description . . . . .	676
10.183.2 Member Enumeration Documentation . . . . .	676
10.183.2.1 MStype . . . . .	676
10.183.2.2 ObjectType . . . . .	679
10.183.3 Constructor & Destructor Documentation . . . . .	679
10.183.3.1 MediaStorage() . . . . .	679
10.183.4 Member Function Documentation . . . . .	679
10.183.4.1 GetModality() . . . . .	679
10.183.4.2 GetModalityDimension() . . . . .	680

10.183.4.3 GetMSString()	680
10.183.4.4 GetMSType()	680
10.183.4.5 GetNumberOfModality()	680
10.183.4.6 GetNumberOfMSString()	680
10.183.4.7 GetNumberOfMSType()	681
10.183.4.8 GetString()	681
10.183.4.9 GuessFromModality()	681
10.183.4.10 IsImage()	681
10.183.4.11 IsUndefined()	682
10.183.4.12 operator MSType()	682
10.183.4.13 SetFromDataSet()	682
10.183.4.14 SetFromFile()	682
10.183.4.15 SetFromHeader()	682
10.183.4.16 SetFromModality()	683
10.183.4.17 SetFromSourceImageSequence()	683
10.183.5 Friends And Related Function Documentation	683
10.183.5.1 operator<<	683
10.184 gdcm::MemberCommand< T > Class Template Reference	683
10.184.1 Detailed Description	685
10.184.2 Member Typedef Documentation	685
10.184.2.1 Self	686
10.184.2.2 TConstMemberFunctionPointer	686
10.184.2.3 TMemberFunctionPointer	686
10.184.3 Constructor & Destructor Documentation	686
10.184.3.1 MemberCommand() [1/2]	686
10.184.3.2 MemberCommand() [2/2]	686
10.184.3.3 ~MemberCommand()	687
10.184.4 Member Function Documentation	687
10.184.4.1 Execute() [1/2]	687
10.184.4.2 Execute() [2/2]	687
10.184.4.3 New()	687
10.184.4.4 operator=()	688
10.184.4.5 SetCallbackFunction() [1/2]	688
10.184.4.6 SetCallbackFunction() [2/2]	688
10.184.5 Member Data Documentation	688
10.184.5.1 m_ConstMemberFunction	688
10.184.5.2 m_MemberFunction	689
10.184.5.3 m_This	689
10.185 gdcm::MeshPrimitive Class Reference	689



10.185.1 Detailed Description . . . . .	691
10.185.2 Member Typedef Documentation . . . . .	691
10.185.2.1 PrimitivesData . . . . .	691
10.185.3 Member Enumeration Documentation . . . . .	691
10.185.3.1 MPTType . . . . .	691
10.185.4 Constructor & Destructor Documentation . . . . .	692
10.185.4.1 MeshPrimitive() . . . . .	692
10.185.4.2 ~MeshPrimitive() . . . . .	692
10.185.5 Member Function Documentation . . . . .	692
10.185.5.1 AddPrimitiveData() . . . . .	692
10.185.5.2 GetMPTType() . . . . .	692
10.185.5.3 GetMPTTypeString() . . . . .	693
10.185.5.4 GetNumberOfPrimitivesData() . . . . .	693
10.185.5.5 GetPrimitiveData() [1/4] . . . . .	693
10.185.5.6 GetPrimitiveData() [2/4] . . . . .	693
10.185.5.7 GetPrimitiveData() [3/4] . . . . .	693
10.185.5.8 GetPrimitiveData() [4/4] . . . . .	693
10.185.5.9 GetPrimitivesData() [1/2] . . . . .	693
10.185.5.10 GetPrimitivesData() [2/2] . . . . .	694
10.185.5.11 GetPrimitiveType() . . . . .	694
10.185.5.12 SetPrimitiveData() [1/2] . . . . .	694
10.185.5.13 SetPrimitiveData() [2/2] . . . . .	694
10.185.5.14 SetPrimitivesData() . . . . .	694
10.185.5.15 SetPrimitiveType() . . . . .	694
10.185.6 Member Data Documentation . . . . .	694
10.185.6.1 PrimitiveData . . . . .	695
10.185.6.2 PrimitiveType . . . . .	695
10.186 gdcmm::ModalityPerformedProcedureStepCreateQuery Class Reference . . . . .	695
10.186.1 Detailed Description . . . . .	696
10.186.2 Constructor & Destructor Documentation . . . . .	697
10.186.2.1 ModalityPerformedProcedureStepCreateQuery() . . . . .	697
10.186.3 Member Function Documentation . . . . .	697
10.186.3.1 GetAbstractSyntaxUID() . . . . .	697
10.186.3.2 GetRequiredDataSet() . . . . .	697
10.186.3.3 ValidateQuery() . . . . .	697
10.186.4 Friends And Related Function Documentation . . . . .	697
10.186.4.1 QueryFactory . . . . .	698
10.187 gdcmm::ModalityPerformedProcedureStepSetQuery Class Reference . . . . .	698
10.187.1 Detailed Description . . . . .	699

10.187.2 Constructor & Destructor Documentation	700
10.187.2.1 ModalityPerformedProcedureStepSetQuery()	700
10.187.3 Member Function Documentation	700
10.187.3.1 GetAbstractSyntaxUID()	700
10.187.3.2 GetRequiredDataSet()	700
10.187.3.3 ValidateQuery()	700
10.187.4 Friends And Related Function Documentation	700
10.187.4.1 QueryFactory	701
10.188 gdcmm::ModifiedEvent Class Reference	701
10.189 gdcmm::Module Class Reference	702
10.189.1 Detailed Description	702
10.189.2 Member Typedef Documentation	703
10.189.2.1 ArrayIncludeMacrosType	703
10.189.2.2 MapModuleEntry	703
10.189.3 Constructor & Destructor Documentation	703
10.189.3.1 Module()	703
10.189.4 Member Function Documentation	703
10.189.4.1 AddMacro()	703
10.189.4.2 AddModuleEntry()	703
10.189.4.3 Clear()	704
10.189.4.4 FindModuleEntryInMacros()	704
10.189.4.5 GetModuleEntryInMacros()	704
10.189.4.6 GetName()	704
10.189.4.7 SetName()	704
10.189.4.8 Verify()	704
10.189.5 Friends And Related Function Documentation	704
10.189.5.1 operator<<	705
10.190 gdcmm::ModuleEntry Class Reference	705
10.190.1 Detailed Description	707
10.190.2 Member Typedef Documentation	707
10.190.2.1 Description	707
10.190.3 Constructor & Destructor Documentation	707
10.190.3.1 ModuleEntry()	707
10.190.3.2 ~ModuleEntry()	707
10.190.4 Member Function Documentation	707
10.190.4.1 GetDescription()	708
10.190.4.2 GetName()	708
10.190.4.3 GetType()	708
10.190.4.4 SetDescription()	708

---

10.190.4.5 SetName()	708
10.190.4.6 SetType()	708
10.190.5 Friends And Related Function Documentation	708
10.190.5.1 operator<<	709
10.190.6 Member Data Documentation	709
10.190.6.1 DataElementType	709
10.190.6.2 DescriptionField	709
10.190.6.3 Name	709
10.191 gdcM::Modules Class Reference	709
10.191.1 Detailed Description	710
10.191.2 Member Typedef Documentation	710
10.191.2.1 ModuleMapType	710
10.191.3 Constructor & Destructor Documentation	710
10.191.3.1 Modules()	710
10.191.4 Member Function Documentation	711
10.191.4.1 AddModule()	711
10.191.4.2 Clear()	711
10.191.4.3 GetModule()	711
10.191.4.4 IsEmpty()	711
10.191.5 Friends And Related Function Documentation	711
10.191.5.1 operator<<	711
10.192 gdcM::MovePatientRootQuery Class Reference	712
10.192.1 Detailed Description	713
10.192.2 Constructor & Destructor Documentation	713
10.192.2.1 MovePatientRootQuery()	713
10.192.3 Member Function Documentation	713
10.192.3.1 GetAbstractSyntaxUID()	713
10.192.3.2 GetTagListByLevel()	713
10.192.3.3 InitializeDataSet()	714
10.192.3.4 ValidateQuery()	714
10.192.4 Friends And Related Function Documentation	714
10.192.4.1 QueryFactory	714
10.193 gdcM::MoveStudyRootQuery Class Reference	715
10.193.1 Detailed Description	716
10.193.2 Constructor & Destructor Documentation	716
10.193.2.1 MoveStudyRootQuery()	716
10.193.3 Member Function Documentation	716
10.193.3.1 GetAbstractSyntaxUID()	716
10.193.3.2 GetTagListByLevel()	716

10.193.3.3 InitializeDataSet()	717
10.193.3.4 ValidateQuery()	717
10.193.4 Friends And Related Function Documentation	717
10.193.4.1 QueryFactory	717
10.194 gdcM::MrProtocol Class Reference	717
10.194.1 Detailed Description	718
10.194.2 Constructor & Destructor Documentation	718
10.194.2.1 MrProtocol()	718
10.194.2.2 ~MrProtocol()	718
10.194.3 Member Function Documentation	719
10.194.3.1 FindMrProtocolByName()	719
10.194.3.2 GetMrProtocolByName()	719
10.194.3.3 GetSliceArray()	719
10.194.3.4 GetVersion()	719
10.194.3.5 Load()	719
10.194.3.6 Print()	719
10.194.4 Friends And Related Function Documentation	720
10.194.4.1 operator<<	720
10.195 gdcM::network::NActionRQ Class Reference	720
10.195.1 Detailed Description	721
10.195.2 Member Function Documentation	721
10.195.2.1 ConstructPDV()	721
10.196 gdcM::network::NActionRSP Class Reference	721
10.196.1 Detailed Description	722
10.196.2 Member Function Documentation	722
10.196.2.1 ConstructPDVByDataSet()	722
10.197 gdcM::network::NCreateRQ Class Reference	723
10.197.1 Detailed Description	723
10.197.2 Member Function Documentation	724
10.197.2.1 ConstructPDV()	724
10.198 gdcM::network::NCreateRSP Class Reference	724
10.198.1 Detailed Description	725
10.198.2 Member Function Documentation	725
10.198.2.1 ConstructPDVByDataSet()	725
10.199 gdcM::network::NDeleteRQ Class Reference	726
10.199.1 Detailed Description	726
10.199.2 Member Function Documentation	727
10.199.2.1 ConstructPDV()	727
10.200 gdcM::network::NDeleteRSP Class Reference	727

10.200.1 Detailed Description . . . . .	728
10.200.2 Member Function Documentation . . . . .	728
10.200.2.1 ConstructPDVByDataSet() . . . . .	728
10.201 gdcmm::NestedModuleEntries Class Reference . . . . .	729
10.201.1 Detailed Description . . . . .	730
10.201.2 Member Typedef Documentation . . . . .	730
10.201.2.1 SizeType . . . . .	730
10.201.3 Constructor & Destructor Documentation . . . . .	730
10.201.3.1 NestedModuleEntries() . . . . .	731
10.201.4 Member Function Documentation . . . . .	731
10.201.4.1 AddModuleEntry() . . . . .	731
10.201.4.2 GetModuleEntry() [1/2] . . . . .	731
10.201.4.3 GetModuleEntry() [2/2] . . . . .	731
10.201.4.4 GetNumberOfModuleEntries() . . . . .	731
10.201.5 Friends And Related Function Documentation . . . . .	731
10.201.5.1 operator<< . . . . .	732
10.202 gdcmm::network::NEventReportRQ Class Reference . . . . .	732
10.202.1 Detailed Description . . . . .	733
10.202.2 Member Function Documentation . . . . .	733
10.202.2.1 ConstructPDV() . . . . .	733
10.203 gdcmm::network::NEventReportRSP Class Reference . . . . .	733
10.203.1 Detailed Description . . . . .	734
10.203.2 Member Function Documentation . . . . .	734
10.203.2.1 ConstructPDVByDataSet() . . . . .	734
10.204 gdcmm::network::NGetRQ Class Reference . . . . .	735
10.204.1 Detailed Description . . . . .	735
10.204.2 Member Function Documentation . . . . .	736
10.204.2.1 ConstructPDV() . . . . .	736
10.205 gdcmm::network::NGetRSP Class Reference . . . . .	736
10.205.1 Detailed Description . . . . .	737
10.205.2 Member Function Documentation . . . . .	737
10.205.2.1 ConstructPDVByDataSet() . . . . .	737
10.206 gdcmm::NoEvent Class Reference . . . . .	738
10.206.1 Detailed Description . . . . .	738
10.207 gdcmm::network::NormalizedMessageFactory Class Reference . . . . .	739
10.207.1 Member Function Documentation . . . . .	739
10.207.1.1 ConstructNAction() . . . . .	739
10.207.1.2 ConstructNCreate() . . . . .	739
10.207.1.3 ConstructNDelete() . . . . .	740

10.207.1.4 ConstructNEventReport()	740
10.207.1.5 ConstructNGet()	740
10.207.1.6 ConstructNSet()	740
10.208 gdcmm::NormalizedNetworkFunctions Class Reference	740
10.208.1 Detailed Description	741
10.208.2 Member Function Documentation	741
10.208.2.1 ConstructQuery()	741
10.208.2.2 NAction()	742
10.208.2.3 NCreate()	742
10.208.2.4 NDelete()	742
10.208.2.5 NEventReport()	742
10.208.2.6 NGet()	743
10.208.2.7 NSet()	743
10.209 gdcmm::network::NSetRQ Class Reference	743
10.209.1 Detailed Description	744
10.209.2 Member Function Documentation	744
10.209.2.1 ConstructPDV()	744
10.210 gdcmm::network::NSetRSP Class Reference	745
10.210.1 Detailed Description	745
10.210.2 Member Function Documentation	746
10.210.2.1 ConstructPDVByDataSet()	746
10.211 gdcmm::Object Class Reference	746
10.211.1 Detailed Description	748
10.211.2 Constructor & Destructor Documentation	748
10.211.2.1 Object() [1/2]	748
10.211.2.2 ~Object()	748
10.211.2.3 Object() [2/2]	748
10.211.3 Member Function Documentation	749
10.211.3.1 operator=()	749
10.211.3.2 Print()	749
10.211.3.3 Register()	749
10.211.3.4 UnRegister()	749
10.211.4 Friends And Related Function Documentation	749
10.211.4.1 operator<<	750
10.211.4.2 SmartPointer	750
10.212 gdcmm::OpenSSLCryptoFactory Class Reference	750
10.212.1 Constructor & Destructor Documentation	751
10.212.1.1 OpenSSLCryptoFactory()	751
10.212.2 Member Function Documentation	751

10.212.2.1 CreateCMSProvider()	751
10.212.2.2 InitOpenSSL()	751
10.213 gdcmm::OpenSSLCryptographicMessageSyntax Class Reference	752
10.213.1 Constructor & Destructor Documentation	753
10.213.1.1 OpenSSLCryptographicMessageSyntax()	753
10.213.1.2 ~OpenSSLCryptographicMessageSyntax()	753
10.213.2 Member Function Documentation	753
10.213.2.1 Decrypt()	753
10.213.2.2 Encrypt()	754
10.213.2.3 GetCipherType()	754
10.213.2.4 ParseCertificateFile()	754
10.213.2.5 ParseKeyFile()	754
10.213.2.6 SetCipherType()	754
10.213.2.7 SetPassword()	755
10.214 gdcmm::OpenSSLP7CryptoFactory Class Reference	755
10.214.1 Constructor & Destructor Documentation	756
10.214.1.1 OpenSSLP7CryptoFactory()	756
10.214.2 Member Function Documentation	756
10.214.2.1 CreateCMSProvider()	756
10.215 gdcmm::OpenSSLP7CryptographicMessageSyntax Class Reference	757
10.215.1 Detailed Description	758
10.215.2 Constructor & Destructor Documentation	758
10.215.2.1 OpenSSLP7CryptographicMessageSyntax()	758
10.215.2.2 ~OpenSSLP7CryptographicMessageSyntax()	758
10.215.3 Member Function Documentation	758
10.215.3.1 Decrypt()	758
10.215.3.2 Encrypt()	759
10.215.3.3 GetCipherType()	759
10.215.3.4 ParseCertificateFile()	759
10.215.3.5 ParseKeyFile()	759
10.215.3.6 SetCipherType()	759
10.215.3.7 SetPassword()	760
10.216 gdcmm::Orientation Class Reference	760
10.216.1 Detailed Description	761
10.216.2 Member Enumeration Documentation	761
10.216.2.1 OrientationType	761
10.216.3 Constructor & Destructor Documentation	761
10.216.3.1 Orientation()	761
10.216.3.2 ~Orientation()	761

10.216.4 Member Function Documentation	762
10.216.4.1 GetLabel()	762
10.216.4.2 GetMajorAxisFromPatientRelativeDirectionCosine()	762
10.216.4.3 GetObliquityThresholdCosineValue()	762
10.216.4.4 GetType()	762
10.216.4.5 Print()	763
10.216.4.6 SetObliquityThresholdCosineValue()	763
10.216.5 Friends And Related Function Documentation	763
10.216.5.1 operator<<	763
10.217 gdcm::Overlay Class Reference	763
10.217.1 Detailed Description	766
10.217.2 Member Enumeration Documentation	766
10.217.2.1 OverlayType	766
10.217.3 Constructor & Destructor Documentation	767
10.217.3.1 Overlay() [1/2]	767
10.217.3.2 ~Overlay()	767
10.217.3.3 Overlay() [2/2]	767
10.217.4 Member Function Documentation	767
10.217.4.1 Decompress()	767
10.217.4.2 GetBitPosition()	767
10.217.4.3 GetBitsAllocated()	768
10.217.4.4 GetColumns()	768
10.217.4.5 GetDescription()	768
10.217.4.6 GetGroup()	768
10.217.4.7 GetOrigin()	768
10.217.4.8 GetOverlayData()	768
10.217.4.9 GetOverlayTypeAsString()	769
10.217.4.10 GetOverlayTypeFromString()	769
10.217.4.11 GetRows()	769
10.217.4.12 GetType()	769
10.217.4.13 GetTypeAsEnum()	769
10.217.4.14 GetUnpackBuffer()	769
10.217.4.15 GetUnpackBufferLength()	770
10.217.4.16 GrabOverlayFromPixelData()	770
10.217.4.17 IsEmpty()	770
10.217.4.18 IsInPixelData() [1/2]	770
10.217.4.19 IsInPixelData() [2/2]	770
10.217.4.20 IsZero()	770
10.217.4.21 operator=()	771



10.217.4.22 Print()	771
10.217.4.23 SetBitPosition()	771
10.217.4.24 SetBitsAllocated()	771
10.217.4.25 SetColumns()	771
10.217.4.26 SetDescription()	772
10.217.4.27 setFrameOrigin()	772
10.217.4.28 SetGroup()	772
10.217.4.29 SetNumberOfFrames()	772
10.217.4.30 SetOrigin()	772
10.217.4.31 SetOverlay()	773
10.217.4.32 SetRows()	773
10.217.4.33 SetType()	773
10.217.4.34 Update()	773
10.218 gdcm::ParseException Class Reference	774
10.218.1 Detailed Description	775
10.218.2 Constructor & Destructor Documentation	775
10.218.2.1 ParseException()	775
10.218.2.2 ~ParseException()	775
10.218.3 Member Function Documentation	775
10.218.3.1 GetLastElement()	775
10.218.3.2 operator=()	775
10.218.3.3 SetLastElement()	776
10.219 gdcm::Parser Class Reference	776
10.219.1 Detailed Description	777
10.219.2 Member Typedef Documentation	777
10.219.2.1 EndElementHandler	777
10.219.2.2 StartElementHandler	777
10.219.3 Member Enumeration Documentation	777
10.219.3.1 ErrorType	777
10.219.4 Constructor & Destructor Documentation	778
10.219.4.1 Parser()	778
10.219.4.2 ~Parser()	778
10.219.5 Member Function Documentation	778
10.219.5.1 GetBuffer()	778
10.219.5.2 GetCurrentByteIndex()	778
10.219.5.3 GetErrorCode()	779
10.219.5.4 GetErrorString()	779
10.219.5.5 GetUserData()	779
10.219.5.6 Parse()	779

10.219.5.7 ParseBuffer()	779
10.219.5.8 Process()	779
10.219.5.9 SetElementHandler()	780
10.219.5.10 SetUserData()	780
10.220 gdcm::Patient Class Reference	780
10.220.1 Detailed Description	780
10.220.2 Constructor & Destructor Documentation	780
10.220.2.1 Patient()	780
10.221 gdcm::network::PDataTFPDU Class Reference	781
10.221.1 Detailed Description	782
10.221.2 Member Typedef Documentation	782
10.221.2.1 SizeType	782
10.221.3 Constructor & Destructor Documentation	782
10.221.3.1 PDataTFPDU()	782
10.221.4 Member Function Documentation	782
10.221.4.1 AddPresentationDataValue()	782
10.221.4.2 GetNumberOfPresentationDataValues()	783
10.221.4.3 GetPresentationDataValue()	783
10.221.4.4 IsLastFragment()	783
10.221.4.5 Print()	783
10.221.4.6 Read()	783
10.221.4.7 ReadInto()	783
10.221.4.8 Size()	784
10.221.4.9 Write()	784
10.222 gdcm::PDBElement Class Reference	784
10.222.1 Detailed Description	785
10.222.2 Constructor & Destructor Documentation	785
10.222.2.1 PDBElement()	785
10.222.3 Member Function Documentation	785
10.222.3.1 GetName()	786
10.222.3.2 GetValue()	786
10.222.3.3 operator==()	786
10.222.3.4 SetName()	786
10.222.3.5 SetValue()	786
10.222.4 Friends And Related Function Documentation	786
10.222.4.1 operator<<	787
10.222.5 Member Data Documentation	787
10.222.5.1 NameField	787
10.222.5.2 ValueField	787

10.223 gdcm::PDBHeader Class Reference . . . . .	787
10.223.1 Detailed Description . . . . .	788
10.223.2 Constructor & Destructor Documentation . . . . .	788
10.223.2.1 PDBHeader() . . . . .	788
10.223.2.2 ~PDBHeader() . . . . .	788
10.223.3 Member Function Documentation . . . . .	789
10.223.3.1 FindPDBElementByName() . . . . .	789
10.223.3.2 GetPDBEEnd() . . . . .	789
10.223.3.3 GetPDBElementByName() . . . . .	789
10.223.3.4 GetPDBInfoTag() . . . . .	789
10.223.3.5 LoadFromDataElement() . . . . .	790
10.223.3.6 Print() . . . . .	790
10.223.4 Friends And Related Function Documentation . . . . .	790
10.223.4.1 operator<< . . . . .	790
10.224 gdcm::PDFCodec Class Reference . . . . .	790
10.224.1 Detailed Description . . . . .	792
10.224.2 Constructor & Destructor Documentation . . . . .	792
10.224.2.1 PDFCodec() . . . . .	792
10.224.2.2 ~PDFCodec() . . . . .	792
10.224.3 Member Function Documentation . . . . .	792
10.224.3.1 CanCode() . . . . .	792
10.224.3.2 CanDecode() . . . . .	792
10.224.3.3 Decode() . . . . .	793
10.225 gdcm::network::PDUFactory Class Reference . . . . .	793
10.225.1 Detailed Description . . . . .	794
10.225.2 Member Function Documentation . . . . .	794
10.225.2.1 ConstructAbortPDU() . . . . .	794
10.225.2.2 ConstructPDU() . . . . .	794
10.225.2.3 ConstructReleasePDU() . . . . .	794
10.225.2.4 CreateCEchoPDU() . . . . .	794
10.225.2.5 CreateCFindPDU() . . . . .	794
10.225.2.6 CreateCMovePDU() . . . . .	795
10.225.2.7 CreateCStoreRQPDU() . . . . .	795
10.225.2.8 CreateCStoreRSPDU() . . . . .	795
10.225.2.9 CreateNActionPDU() . . . . .	795
10.225.2.10 CreateNCreatePDU() . . . . .	795
10.225.2.11 CreateNDeletePDU() . . . . .	795
10.225.2.12 CreateNEventReportPDU() . . . . .	796
10.225.2.13 CreateNGetPDU() . . . . .	796

10.225.2.14 CreateNSetPDU()	796
10.225.2.15 DetermineEventByPDU()	796
10.225.2.16 GetPDVs()	796
10.226 gdcmm::PersonName Class Reference	796
10.226.1 Detailed Description	797
10.226.2 Member Function Documentation	797
10.226.2.1 GetMaxLength()	797
10.226.2.2 GetNumberOfComponents()	797
10.226.2.3 Print()	798
10.226.2.4 SetBlob()	798
10.226.2.5 SetComponents() [1/2]	798
10.226.2.6 SetComponents() [2/2]	798
10.226.3 Member Data Documentation	798
10.226.3.1 Component	798
10.226.3.2 MaxLength	798
10.226.3.3 MaxNumberOfComponents	799
10.226.3.4 Padding	799
10.226.3.5 Separator	799
10.227 gdcmm::PGXCodec Class Reference	799
10.227.1 Detailed Description	800
10.227.2 Constructor & Destructor Documentation	800
10.227.2.1 PGXCodec()	800
10.227.2.2 ~PGXCodec()	801
10.227.3 Member Function Documentation	801
10.227.3.1 CanCode()	801
10.227.3.2 CanDecode()	801
10.227.3.3 Clone()	801
10.227.3.4 GetHeaderInfo()	801
10.227.3.5 Read()	802
10.227.3.6 Write()	802
10.228 gdcmm::PhotometricInterpretation Class Reference	802
10.228.1 Detailed Description	803
10.228.2 Member Enumeration Documentation	803
10.228.2.1 PType	804
10.228.3 Constructor & Destructor Documentation	804
10.228.3.1 PhotometricInterpretation()	804
10.228.4 Member Function Documentation	804
10.228.4.1 GetPString()	804
10.228.4.2 GetPType()	805

10.228.4.3 GetSamplesPerPixel()	805
10.228.4.4 GetString()	805
10.228.4.5 GetType()	805
10.228.4.6 IsLossless()	805
10.228.4.7 IsLossy()	805
10.228.4.8 IsRetired()	805
10.228.4.9 IsSameColorSpace()	806
10.228.4.10 operator PType()	806
10.228.5 Friends And Related Function Documentation	806
10.228.5.1 operator<<	806
10.229 gdcm::PixelFormat Class Reference	806
10.229.1 Detailed Description	808
10.229.2 Member Enumeration Documentation	808
10.229.2.1 ScalarType	808
10.229.3 Constructor & Destructor Documentation	809
10.229.3.1 PixelFormat() [1/3]	809
10.229.3.2 PixelFormat() [2/3]	809
10.229.3.3 PixelFormat() [3/3]	809
10.229.4 Member Function Documentation	809
10.229.4.1 GetBitsAllocated()	809
10.229.4.2 GetBitsStored()	810
10.229.4.3 GetHighBit()	810
10.229.4.4 GetMax()	810
10.229.4.5 GetMin()	810
10.229.4.6 GetPixelRepresentation()	810
10.229.4.7 GetPixelSize()	811
10.229.4.8 GetSamplesPerPixel()	811
10.229.4.9 GetScalarType()	811
10.229.4.10 GetScalarTypeAsString()	811
10.229.4.11 IsCompatible()	812
10.229.4.12 IsValid()	812
10.229.4.13 operator ScalarType()	812
10.229.4.14 operator!=( ) [1/2]	812
10.229.4.15 operator!=( ) [2/2]	812
10.229.4.16 operator==( ) [1/2]	812
10.229.4.17 operator==( ) [2/2]	813
10.229.4.18 Print()	813
10.229.4.19 SetBitsAllocated()	813
10.229.4.20 SetBitsStored()	813

10.229.4.21 SetHighBit()	813
10.229.4.22 SetPixelRepresentation()	813
10.229.4.23 SetSamplesPerPixel()	814
10.229.4.24 SetScalarType()	814
10.229.4.25 Validate()	814
10.229.5 Friends And Related Function Documentation	814
10.229.5.1 Bitmap	814
10.229.5.2 operator<<	815
10.230 gdcm::Pixmap Class Reference	815
10.230.1 Detailed Description	817
10.230.2 Constructor & Destructor Documentation	817
10.230.2.1 Pixmap()	817
10.230.2.2 ~Pixmap()	817
10.230.3 Member Function Documentation	817
10.230.3.1 AreOverlaysInPixelData()	817
10.230.3.2 GetCurve() [1/2]	818
10.230.3.3 GetCurve() [2/2]	818
10.230.3.4 GetIconImage() [1/2]	818
10.230.3.5 GetIconImage() [2/2]	818
10.230.3.6 GetNumberOfCurves()	818
10.230.3.7 GetNumberOfOverlays()	818
10.230.3.8 GetOverlay() [1/2]	819
10.230.3.9 GetOverlay() [2/2]	819
10.230.3.10 Print()	819
10.230.3.11 RemoveOverlay()	819
10.230.3.12 SetIconImage()	819
10.230.3.13 SetNumberOfCurves()	819
10.230.3.14 SetNumberOfOverlays()	820
10.230.3.15 UnusedBitsPresentInPixelData()	820
10.230.4 Member Data Documentation	820
10.230.4.1 Curves	820
10.230.4.2 Icon	820
10.230.4.3 Overlays	820
10.231 gdcm::PixmapReader Class Reference	821
10.231.1 Detailed Description	823
10.231.2 Constructor & Destructor Documentation	823
10.231.2.1 PixmapReader()	823
10.231.2.2 ~PixmapReader()	823
10.231.3 Member Function Documentation	823

10.231.3.1 GetPixmap() [1/2]	823
10.231.3.2 GetPixmap() [2/2]	824
10.231.3.3 Read()	824
10.231.3.4 ReadACRNEMAIImage()	824
10.231.3.5 ReadImage()	824
10.231.3.6 ReadImageInternal()	824
10.231.4 Member Data Documentation	825
10.231.4.1 PixelData	825
10.232 gdcm::PixmapToPixmapFilter Class Reference	825
10.232.1 Detailed Description	826
10.232.2 Constructor & Destructor Documentation	827
10.232.2.1 PixmapToPixmapFilter()	827
10.232.2.2 ~PixmapToPixmapFilter()	827
10.232.3 Member Function Documentation	827
10.232.3.1 GetInput()	827
10.232.3.2 GetOutput()	827
10.232.3.3 GetOutputAsPixmap()	827
10.233 gdcm::PixmapWriter Class Reference	828
10.233.1 Detailed Description	830
10.233.2 Constructor & Destructor Documentation	830
10.233.2.1 PixmapWriter()	830
10.233.2.2 ~PixmapWriter()	830
10.233.3 Member Function Documentation	830
10.233.3.1 DoConImage()	831
10.233.3.2 GetImage() [1/2]	831
10.233.3.3 GetImage() [2/2]	831
10.233.3.4 GetPixmap() [1/2]	831
10.233.3.5 GetPixmap() [2/2]	831
10.233.3.6 PrepareWrite()	831
10.233.3.7 SetImage()	832
10.233.3.8 SetPixmap()	832
10.233.3.9 Write()	832
10.233.4 Member Data Documentation	832
10.233.4.1 PixelData	832
10.234 gdcm::PNMCodec Class Reference	833
10.234.1 Detailed Description	834
10.234.2 Constructor & Destructor Documentation	834
10.234.2.1 PNMCodec()	834
10.234.2.2 ~PNMCodec()	834

10.234.3 Member Function Documentation	834
10.234.3.1 CanCode()	835
10.234.3.2 CanDecode()	835
10.234.3.3 Clone()	835
10.234.3.4 GetBufferLength()	835
10.234.3.5 GetHeaderInfo()	835
10.234.3.6 Read()	836
10.234.3.7 SetBufferLength()	836
10.234.3.8 Write()	836
10.235 gdcmm::Preamble Class Reference	836
10.235.1 Detailed Description	837
10.235.2 Constructor & Destructor Documentation	837
10.235.2.1 Preamble() [1/2]	838
10.235.2.2 ~Preamble()	838
10.235.2.3 Preamble() [2/2]	838
10.235.3 Member Function Documentation	838
10.235.3.1 Clear()	838
10.235.3.2 Create()	838
10.235.3.3 GetInternal()	838
10.235.3.4 GetLength()	839
10.235.3.5 IsEmpty()	839
10.235.3.6 IsValid()	839
10.235.3.7 operator=()	839
10.235.3.8 Print()	839
10.235.3.9 Read()	839
10.235.3.10 Remove()	840
10.235.3.11 Valid()	840
10.235.3.12 Write()	840
10.235.4 Friends And Related Function Documentation	840
10.235.4.1 operator<<	840
10.236 gdcmm::PresentationContext Class Reference	841
10.236.1 Detailed Description	842
10.236.2 Member Typedef Documentation	842
10.236.2.1 SizeType	842
10.236.2.2 TransferSyntaxArrayType	842
10.236.3 Constructor & Destructor Documentation	842
10.236.3.1 PresentationContext() [1/2]	842
10.236.3.2 PresentationContext() [2/2]	843
10.236.4 Member Function Documentation	843



10.236.4.1 AddTransferSyntax()	843
10.236.4.2 GetAbstractSyntax()	843
10.236.4.3 GetNumberOfTransferSyntaxes()	843
10.236.4.4 GetPresentationContextID()	843
10.236.4.5 GetTransferSyntax()	843
10.236.4.6 operator==( )	844
10.236.4.7 Print()	844
10.236.4.8 SetAbstractSyntax()	844
10.236.4.9 SetPresentationContextID()	844
10.236.5 Member Data Documentation	844
10.236.5.1 AbstractSyntax	844
10.236.5.2 ID	844
10.236.5.3 TransferSyntaxes	845
10.237 gdcmm::network::PresentationContextAC Class Reference	845
10.237.1 Detailed Description	845
10.237.2 Constructor & Destructor Documentation	845
10.237.2.1 PresentationContextAC()	846
10.237.3 Member Function Documentation	846
10.237.3.1 GetPresentationContextID()	846
10.237.3.2 GetReason()	846
10.237.3.3 GetTransferSyntax()	846
10.237.3.4 Print()	846
10.237.3.5 Read()	846
10.237.3.6 SetPresentationContextID()	847
10.237.3.7 SetReason()	847
10.237.3.8 SetTransferSyntax()	847
10.237.3.9 Size()	847
10.237.3.10 Write()	847
10.238 gdcmm::PresentationContextGenerator Class Reference	847
10.238.1 Detailed Description	848
10.238.2 Member Typedef Documentation	849
10.238.2.1 PresentationContextArrayType	849
10.238.2.2 SizeType	849
10.238.3 Constructor & Destructor Documentation	849
10.238.3.1 PresentationContextGenerator()	849
10.238.4 Member Function Documentation	849
10.238.4.1 AddFromFile()	849
10.238.4.2 AddPresentationContext()	849
10.238.4.3 GenerateFromFilenames()	850

10.238.4.4 GenerateFromUID()	850
10.238.4.5 GetDefaultTransferSyntax()	850
10.238.4.6 GetPresentationContexts()	850
10.238.4.7 SetDefaultTransferSyntax()	850
10.238.4.8 SetMergeModeToAbstractSyntax()	851
10.238.4.9 SetMergeModeToTransferSyntax()	851
10.239 gdcmm::network::PresentationContextRQ Class Reference	851
10.239.1 Detailed Description	852
10.239.2 Member Typedef Documentation	852
10.239.2.1 SizeType	852
10.239.3 Constructor & Destructor Documentation	852
10.239.3.1 PresentationContextRQ() [1/3]	852
10.239.3.2 PresentationContextRQ() [2/3]	852
10.239.3.3 PresentationContextRQ() [3/3]	852
10.239.4 Member Function Documentation	853
10.239.4.1 AddTransferSyntax()	853
10.239.4.2 GetAbstractSyntax() [1/2]	853
10.239.4.3 GetAbstractSyntax() [2/2]	853
10.239.4.4 GetNumberOfTransferSyntaxes()	853
10.239.4.5 GetPresentationContextID()	853
10.239.4.6 GetTransferSyntax() [1/2]	853
10.239.4.7 GetTransferSyntax() [2/2]	854
10.239.4.8 GetTransferSyntaxes()	854
10.239.4.9 operator==( )	854
10.239.4.10 Print()	854
10.239.4.11 Read()	854
10.239.4.12 SetAbstractSyntax()	854
10.239.4.13 SetPresentationContextID()	855
10.239.4.14 Size()	855
10.239.4.15 Write()	855
10.240 gdcmm::network::PresentationDataValue Class Reference	855
10.240.1 Detailed Description	856
10.240.2 Constructor & Destructor Documentation	856
10.240.2.1 PresentationDataValue()	856
10.240.3 Member Function Documentation	856
10.240.3.1 ConcatenatePDVBlobs()	856
10.240.3.2 ConcatenatePDVBlobsAsExplicit()	856
10.240.3.3 GetBlob()	857
10.240.3.4 GetIsCommand()	857

10.240.3.5 GetIsLastFragment()	857
10.240.3.6 GetMessageHeader()	857
10.240.3.7 GetPresentationContextID()	857
10.240.3.8 Print()	857
10.240.3.9 Read()	857
10.240.3.10 ReadInto()	858
10.240.3.11 SetBlob()	858
10.240.3.12 SetCommand()	858
10.240.3.13 SetDataSet()	858
10.240.3.14 SetLastFragment()	858
10.240.3.15 SetMessageHeader()	859
10.240.3.16 SetPresentationContextID()	859
10.240.3.17 Size()	859
10.240.3.18 Write()	859
10.241 gdcmm::Printer Class Reference	859
10.241.1 Detailed Description	861
10.241.2 Member Enumeration Documentation	861
10.241.2.1 PrintStyles	861
10.241.3 Constructor & Destructor Documentation	861
10.241.3.1 Printer()	861
10.241.3.2 ~Printer()	862
10.241.4 Member Function Documentation	862
10.241.4.1 GetPrintStyle()	862
10.241.4.2 Print()	862
10.241.4.3 PrintDataElement()	862
10.241.4.4 PrintDataSet()	863
10.241.4.5 PrintSQ()	863
10.241.4.6 SetColor()	863
10.241.4.7 SetFile()	863
10.241.4.8 SetStyle()	863
10.241.5 Member Data Documentation	864
10.241.5.1 F	864
10.241.5.2 MaxPrintLength	864
10.241.5.3 PrintStyle	864
10.242 gdcmm::PrivateDict Class Reference	864
10.242.1 Detailed Description	865
10.242.2 Constructor & Destructor Documentation	865
10.242.2.1 PrivateDict()	865
10.242.2.2 ~PrivateDict()	865

10.242.3 Member Function Documentation	865
10.242.3.1 AddDictEntry()	865
10.242.3.2 FindDictEntry()	866
10.242.3.3 GetDictEntry()	866
10.242.3.4 IsEmpty()	866
10.242.3.5 LoadDefault()	866
10.242.3.6 PrintXML()	866
10.242.3.7 RemoveDictEntry()	866
10.242.4 Friends And Related Function Documentation	867
10.242.4.1 Dicts	867
10.242.4.2 operator<<	867
10.243 gdcM::PrivateTag Class Reference	867
10.243.1 Detailed Description	868
10.243.2 Constructor & Destructor Documentation	868
10.243.2.1 PrivateTag() [1/2]	869
10.243.2.2 PrivateTag() [2/2]	869
10.243.3 Member Function Documentation	869
10.243.3.1 GetAsDataElement()	869
10.243.3.2 GetOwner()	869
10.243.3.3 operator<()	869
10.243.3.4 ReadFromCommaSeparatedString()	870
10.243.3.5 SetOwner()	870
10.243.4 Friends And Related Function Documentation	870
10.243.4.1 operator<<	870
10.244 gdcM::ProgressEvent Class Reference	870
10.244.1 Detailed Description	871
10.244.2 Member Typedef Documentation	872
10.244.2.1 Self	872
10.244.2.2 Superclass	872
10.244.3 Constructor & Destructor Documentation	872
10.244.3.1 ProgressEvent() [1/2]	872
10.244.3.2 ~ProgressEvent()	872
10.244.3.3 ProgressEvent() [2/2]	872
10.244.4 Member Function Documentation	872
10.244.4.1 CheckEvent()	873
10.244.4.2 GetEventName()	873
10.244.4.3 GetProgress()	873
10.244.4.4 MakeObject()	873
10.244.4.5 operator=()	873

10.244.4.6 SetProgress()	873
10.245 gdcm::PVRGCodec Class Reference	874
10.245.1 Detailed Description	875
10.245.2 Constructor & Destructor Documentation	875
10.245.2.1 PVRGCodec()	875
10.245.2.2 ~PVRGCodec()	875
10.245.3 Member Function Documentation	875
10.245.3.1 CanCode()	876
10.245.3.2 CanDecode()	876
10.245.3.3 Clone()	876
10.245.3.4 Code()	876
10.245.3.5 Decode()	877
10.245.3.6 SetLossyFlag()	877
10.246 gdcm::PythonFilter Class Reference	877
10.246.1 Detailed Description	877
10.246.2 Constructor & Destructor Documentation	878
10.246.2.1 PythonFilter()	878
10.246.2.2 ~PythonFilter()	878
10.246.3 Member Function Documentation	878
10.246.3.1 GetFile() [1/2]	878
10.246.3.2 GetFile() [2/2]	878
10.246.3.3 SetDicts()	878
10.246.3.4 SetFile()	878
10.246.3.5 ToPyObject()	879
10.246.3.6 UseDictAlways()	879
10.247 gdcm::QueryBase Class Reference	879
10.247.1 Detailed Description	880
10.247.2 Constructor & Destructor Documentation	880
10.247.2.1 ~QueryBase()	880
10.247.3 Member Function Documentation	880
10.247.3.1 GetAllRequiredTags()	880
10.247.3.2 GetAllTags()	881
10.247.3.3 GetHierarchicalSearchTags()	881
10.247.3.4 GetName()	881
10.247.3.5 GetOptionalTags()	881
10.247.3.6 GetQueryLevel()	881
10.247.3.7 GetRequiredTags()	882
10.247.3.8 GetUniqueTags()	882
10.248 gdcm::QueryFactory Class Reference	882

10.248.1 Detailed Description . . . . .	882
10.248.2 Member Function Documentation . . . . .	883
10.248.2.1 GetCharacterFromCurrentLocale() . . . . .	883
10.248.2.2 ListCharSets() . . . . .	883
10.248.2.3 ProduceCharacterSetDataElement() . . . . .	883
10.248.2.4 ProduceQuery() [1/2] . . . . .	883
10.248.2.5 ProduceQuery() [2/2] . . . . .	884
10.249 gdcmm::QueryImage Class Reference . . . . .	884
10.249.1 Detailed Description . . . . .	885
10.249.2 Member Function Documentation . . . . .	885
10.249.2.1 GetHierarchicalSearchTags() . . . . .	885
10.249.2.2 GetName() . . . . .	885
10.249.2.3 GetOptionalTags() . . . . .	885
10.249.2.4 GetQueryLevel() . . . . .	886
10.249.2.5 GetRequiredTags() . . . . .	886
10.249.2.6 GetUniqueTags() . . . . .	886
10.250 gdcmm::QueryPatient Class Reference . . . . .	886
10.250.1 Detailed Description . . . . .	887
10.250.2 Member Function Documentation . . . . .	887
10.250.2.1 GetHierarchicalSearchTags() . . . . .	887
10.250.2.2 GetName() . . . . .	888
10.250.2.3 GetOptionalTags() . . . . .	888
10.250.2.4 GetQueryLevel() . . . . .	888
10.250.2.5 GetRequiredTags() . . . . .	888
10.250.2.6 GetUniqueTags() . . . . .	888
10.251 gdcmm::QuerySeries Class Reference . . . . .	889
10.251.1 Detailed Description . . . . .	890
10.251.2 Member Function Documentation . . . . .	890
10.251.2.1 GetHierarchicalSearchTags() . . . . .	890
10.251.2.2 GetName() . . . . .	890
10.251.2.3 GetOptionalTags() . . . . .	890
10.251.2.4 GetQueryLevel() . . . . .	890
10.251.2.5 GetRequiredTags() . . . . .	891
10.251.2.6 GetUniqueTags() . . . . .	891
10.252 gdcmm::QueryStudy Class Reference . . . . .	891
10.252.1 Detailed Description . . . . .	892
10.252.2 Member Function Documentation . . . . .	892
10.252.2.1 GetHierarchicalSearchTags() . . . . .	892
10.252.2.2 GetName() . . . . .	893

10.252.2.3 GetOptionalTags()	893
10.252.2.4 GetQueryLevel()	893
10.252.2.5 GetRequiredTags()	893
10.252.2.6 GetUniqueTags()	893
10.253 gdcm::RAWCodec Class Reference	894
10.253.1 Detailed Description	895
10.253.2 Constructor & Destructor Documentation	895
10.253.2.1 RAWCodec()	895
10.253.2.2 ~RAWCodec()	895
10.253.3 Member Function Documentation	895
10.253.3.1 CanCode()	896
10.253.3.2 CanDecode()	896
10.253.3.3 Clone()	896
10.253.3.4 Code()	896
10.253.3.5 Decode()	897
10.253.3.6 DecodeByStreams()	897
10.253.3.7 DecodeBytes()	897
10.253.3.8 GetHeaderInfo()	897
10.254 gdcm::Reader Class Reference	898
10.254.1 Detailed Description	900
10.254.2 Constructor & Destructor Documentation	901
10.254.2.1 Reader()	901
10.254.2.2 ~Reader()	901
10.254.3 Member Function Documentation	901
10.254.3.1 CanRead()	901
10.254.3.2 GetFile() [1/2]	901
10.254.3.3 GetFile() [2/2]	902
10.254.3.4 GetStreamCurrentPosition()	902
10.254.3.5 GetStreamPtr()	902
10.254.3.6 Read()	902
10.254.3.7 ReadDataSet()	903
10.254.3.8 ReadMetaInformation()	903
10.254.3.9 ReadPreamble()	903
10.254.3.10 ReadSelectedPrivateTags()	903
10.254.3.11 ReadSelectedTags()	903
10.254.3.12 ReadUpToTag()	903
10.254.3.13 SetFile()	904
10.254.3.14 SetFileName()	904
10.254.3.15 SetStream()	905

10.254.4 Friends And Related Function Documentation	905
10.254.4.1 StreamImageReader	905
10.254.5 Member Data Documentation	905
10.254.5.1 F	905
10.255 gdcm::RealWorldValueMappingContent Struct Reference	906
10.255.1 Member Data Documentation	906
10.255.1.1 CodeMeaning	906
10.255.1.2 CodeValue	907
10.255.1.3 RealWorldValueIntercept	907
10.255.1.4 RealWorldValueSlope	907
10.256 gdcm::Region Class Reference	907
10.256.1 Detailed Description	908
10.256.2 Constructor & Destructor Documentation	908
10.256.2.1 Region()	908
10.256.2.2 ~Region()	908
10.256.3 Member Function Documentation	908
10.256.3.1 Area()	909
10.256.3.2 Clone()	909
10.256.3.3 ComputeBoundingBox()	909
10.256.3.4 Empty()	909
10.256.3.5 IsValid()	909
10.256.3.6 Print()	910
10.257 gdcm::Rescaler Class Reference	910
10.257.1 Detailed Description	911
10.257.2 Constructor & Destructor Documentation	911
10.257.2.1 Rescaler()	912
10.257.2.2 ~Rescaler()	912
10.257.3 Member Function Documentation	912
10.257.3.1 ComputeInterceptSlopePixelType()	912
10.257.3.2 ComputePixelTypeFromMinMax()	912
10.257.3.3 GetIntercept()	912
10.257.3.4 GetSlope()	912
10.257.3.5 InverseRescale()	913
10.257.3.6 InverseRescaleFunctionIntoBestFit()	913
10.257.3.7 Rescale()	913
10.257.3.8 RescaleFunctionIntoBestFit()	913
10.257.3.9 SetIntercept()	913
10.257.3.10 SetMinMaxForPixelType()	914
10.257.3.11 SetPixelFormat()	914



---

10.257.3.12 SetSlope()	914
10.257.3.13 SetTargetPixelType()	914
10.257.3.14 SetUseTargetPixelType()	914
10.258 gdcmm::RLECodec Class Reference	915
10.258.1 Detailed Description	916
10.258.2 Constructor & Destructor Documentation	916
10.258.2.1 RLECodec()	917
10.258.2.2 ~RLECodec()	917
10.258.3 Member Function Documentation	917
10.258.3.1 AppendFrameEncode()	917
10.258.3.2 AppendRowEncode()	917
10.258.3.3 CanCode()	917
10.258.3.4 CanDecode()	918
10.258.3.5 Clone()	918
10.258.3.6 Code()	918
10.258.3.7 Decode()	918
10.258.3.8 DecodeByStreams()	919
10.258.3.9 DecodeExtent()	919
10.258.3.10 GetBufferLength()	919
10.258.3.11 GetHeaderInfo()	919
10.258.3.12 IsFrameEncoder()	919
10.258.3.13 IsRowEncoder()	920
10.258.3.14 SetBufferLength()	920
10.258.3.15 SetLength()	920
10.258.3.16 StartEncode()	920
10.258.3.17 StopEncode()	920
10.258.4 Friends And Related Function Documentation	920
10.258.4.1 ImageRegionReader	921
10.259 gdcmm::network::RoleSelectionSub Class Reference	921
10.259.1 Detailed Description	921
10.259.2 Constructor & Destructor Documentation	921
10.259.2.1 RoleSelectionSub()	921
10.259.3 Member Function Documentation	921
10.259.3.1 Print()	922
10.259.3.2 Read()	922
10.259.3.3 SetTuple()	922
10.259.3.4 Size()	922
10.259.3.5 Write()	922
10.260 gdcmm::Scanner Class Reference	923

10.260.1 Detailed Description . . . . .	925
10.260.2 Member Typedef Documentation . . . . .	925
10.260.2.1 ConstIterator . . . . .	925
10.260.2.2 MappingType . . . . .	926
10.260.2.3 TagToValue . . . . .	926
10.260.2.4 TagToValueValueType . . . . .	926
10.260.2.5 ValuesType . . . . .	926
10.260.3 Constructor & Destructor Documentation . . . . .	926
10.260.3.1 Scanner() . . . . .	926
10.260.3.2 ~Scanner() . . . . .	926
10.260.4 Member Function Documentation . . . . .	926
10.260.4.1 AddPrivateTag() . . . . .	927
10.260.4.2 AddSkipTag() . . . . .	927
10.260.4.3 AddTag() . . . . .	927
10.260.4.4 Begin() . . . . .	927
10.260.4.5 ClearSkipTags() . . . . .	927
10.260.4.6 ClearTags() . . . . .	927
10.260.4.7 End() . . . . .	928
10.260.4.8 GetAllFileNamesFromTagToValue() . . . . .	928
10.260.4.9 GetFilenameFromTagToValue() . . . . .	928
10.260.4.10 GetFileNames() . . . . .	928
10.260.4.11 GetKeys() . . . . .	928
10.260.4.12 GetMapping() . . . . .	929
10.260.4.13 GetMappingFromTagToValue() . . . . .	929
10.260.4.14 GetMappings() . . . . .	929
10.260.4.15 GetOrderedValues() . . . . .	929
10.260.4.16 GetValue() . . . . .	929
10.260.4.17 GetValues() [ 1 / 2 ] . . . . .	930
10.260.4.18 GetValues() [ 2 / 2 ] . . . . .	930
10.260.4.19 IsKey() . . . . .	930
10.260.4.20 New() . . . . .	930
10.260.4.21 Print() . . . . .	931
10.260.4.22 PrintTable() . . . . .	931
10.260.4.23 ProcessPublicTag() . . . . .	931
10.260.4.24 Scan() . . . . .	931
10.260.5 Friends And Related Function Documentation . . . . .	931
10.260.5.1 operator<< . . . . .	932
10.261 gdcm::Segment Class Reference . . . . .	932
10.261.1 Detailed Description . . . . .	934

10.261.2 Member Typedef Documentation	934
10.261.2.1 BasicCodedEntryVector	934
10.261.2.2 SurfaceVector	934
10.261.3 Member Enumeration Documentation	934
10.261.3.1 ALGOType	934
10.261.4 Constructor & Destructor Documentation	935
10.261.4.1 Segment()	935
10.261.4.2 ~Segment()	935
10.261.5 Member Function Documentation	935
10.261.5.1 AddSurface()	935
10.261.5.2 GetALGOType()	935
10.261.5.3 GetALGOTypeString()	936
10.261.5.4 GetAnatomicRegion() [1/2]	936
10.261.5.5 GetAnatomicRegion() [2/2]	936
10.261.5.6 GetAnatomicRegionModifiers() [1/2]	936
10.261.5.7 GetAnatomicRegionModifiers() [2/2]	936
10.261.5.8 GetPropertyCategory() [1/2]	936
10.261.5.9 GetPropertyCategory() [2/2]	936
10.261.5.10 GetPropertyType() [1/2]	937
10.261.5.11 GetPropertyType() [2/2]	937
10.261.5.12 GetPropertyTypeModifiers() [1/2]	937
10.261.5.13 GetPropertyTypeModifiers() [2/2]	937
10.261.5.14 GetSegmentAlgorithmName()	937
10.261.5.15 GetSegmentAlgorithmType()	937
10.261.5.16 GetSegmentDescription()	937
10.261.5.17 GetSegmentLabel()	938
10.261.5.18 GetSegmentNumber()	938
10.261.5.19 GetSurface()	938
10.261.5.20 GetSurfaceCount()	938
10.261.5.21 GetSurfaces() [1/2]	938
10.261.5.22 GetSurfaces() [2/2]	938
10.261.5.23 SetAnatomicRegion()	938
10.261.5.24 SetAnatomicRegionModifiers()	939
10.261.5.25 SetPropertyCategory()	939
10.261.5.26 SetPropertyType()	939
10.261.5.27 SetPropertyTypeModifiers()	939
10.261.5.28 SetSegmentAlgorithmName()	939
10.261.5.29 SetSegmentAlgorithmType() [1/2]	939
10.261.5.30 SetSegmentAlgorithmType() [2/2]	940

10.261.5.31 SetSegmentDescription()	940
10.261.5.32 SetSegmentLabel()	940
10.261.5.33 SetSegmentNumber()	940
10.261.5.34 SetSurfaceCount()	940
10.261.6 Member Data Documentation	940
10.261.6.1 AnatomicRegion	940
10.261.6.2 AnatomicRegionModifiers	941
10.261.6.3 PropertyCategory	941
10.261.6.4 PropertyType	941
10.261.6.5 PropertyTypeModifiers	941
10.261.6.6 SegmentAlgorithmName	941
10.261.6.7 SegmentAlgorithmType	941
10.261.6.8 SegmentDescription	941
10.261.6.9 SegmentLabel	942
10.261.6.10 SegmentNumber	942
10.261.6.11 SurfaceCount	942
10.261.6.12 Surfaces	942
10.262 gdcm::SegmentedPaletteColorLookupTable Class Reference	942
10.262.1 Detailed Description	943
10.262.2 Constructor & Destructor Documentation	943
10.262.2.1 SegmentedPaletteColorLookupTable()	943
10.262.2.2 ~SegmentedPaletteColorLookupTable()	944
10.262.3 Member Function Documentation	944
10.262.3.1 Print()	944
10.262.3.2 SetLUT()	944
10.263 gdcm::SegmentReader Class Reference	945
10.263.1 Detailed Description	946
10.263.2 Member Typedef Documentation	946
10.263.2.1 SegmentMap	947
10.263.2.2 SegmentVector	947
10.263.3 Constructor & Destructor Documentation	947
10.263.3.1 SegmentReader()	947
10.263.3.2 ~SegmentReader()	947
10.263.4 Member Function Documentation	947
10.263.4.1 GetSegments() [1/2]	947
10.263.4.2 GetSegments() [2/2]	947
10.263.4.3 Read()	948
10.263.4.4 ReadSegment()	948
10.263.4.5 ReadSegments()	948

10.263.5 Member Data Documentation . . . . .	948
10.263.5.1 Segments . . . . .	948
10.264 gdcmm::SegmentWriter Class Reference . . . . .	949
10.264.1 Detailed Description . . . . .	950
10.264.2 Member Typedef Documentation . . . . .	950
10.264.2.1 SegmentVector . . . . .	950
10.264.3 Constructor & Destructor Documentation . . . . .	950
10.264.3.1 SegmentWriter() . . . . .	950
10.264.3.2 ~SegmentWriter() . . . . .	950
10.264.4 Member Function Documentation . . . . .	951
10.264.4.1 AddSegment() . . . . .	951
10.264.4.2 GetNumberOfSegments() . . . . .	951
10.264.4.3 GetSegment() . . . . .	951
10.264.4.4 GetSegments() [1/2] . . . . .	951
10.264.4.5 GetSegments() [2/2] . . . . .	951
10.264.4.6 PrepareWrite() . . . . .	951
10.264.4.7 SetNumberOfSegments() . . . . .	952
10.264.4.8 SetSegments() . . . . .	952
10.264.4.9 Write() . . . . .	952
10.264.5 Member Data Documentation . . . . .	952
10.264.5.1 Segments . . . . .	952
10.265 gdcmm::SequenceOfFragments Class Reference . . . . .	953
10.265.1 Detailed Description . . . . .	954
10.265.2 Member Typedef Documentation . . . . .	955
10.265.2.1 ConstIterator . . . . .	955
10.265.2.2 FragmentVector . . . . .	955
10.265.2.3 Iterator . . . . .	955
10.265.2.4 SizeType . . . . .	955
10.265.3 Constructor & Destructor Documentation . . . . .	955
10.265.3.1 SequenceOfFragments() . . . . .	955
10.265.4 Member Function Documentation . . . . .	955
10.265.4.1 AddFragment() . . . . .	956
10.265.4.2 Begin() [1/2] . . . . .	956
10.265.4.3 Begin() [2/2] . . . . .	956
10.265.4.4 Clear() . . . . .	956
10.265.4.5 ComputeByteLength() . . . . .	956
10.265.4.6 ComputeLength() . . . . .	956
10.265.4.7 End() [1/2] . . . . .	957
10.265.4.8 End() [2/2] . . . . .	957

10.265.4.9	GetBuffer()	957
10.265.4.10	GetFragBuffer()	957
10.265.4.11	GetFragment()	957
10.265.4.12	GetLength()	958
10.265.4.13	GetNumberOfFragments()	958
10.265.4.14	GetTable() [1/2]	958
10.265.4.15	GetTable() [2/2]	958
10.265.4.16	New()	958
10.265.4.17	operator==()	958
10.265.4.18	Print()	959
10.265.4.19	Read()	959
10.265.4.20	ReadPreValue()	959
10.265.4.21	ReadValue()	959
10.265.4.22	SetLength()	959
10.265.4.23	Write()	960
10.265.4.24	WriteBuffer()	960
10.266	gdcmm::SequenceOfItems Class Reference	960
10.266.1	Detailed Description	962
10.266.2	Member Typedef Documentation	963
10.266.2.1	ConstIterator	963
10.266.2.2	ItemVector	963
10.266.2.3	Iterator	963
10.266.2.4	SizeType	963
10.266.3	Constructor & Destructor Documentation	963
10.266.3.1	SequenceOfItems()	964
10.266.4	Member Function Documentation	964
10.266.4.1	AddItem()	964
10.266.4.2	AddNewUndefinedLengthItem()	964
10.266.4.3	Begin() [1/2]	964
10.266.4.4	Begin() [2/2]	964
10.266.4.5	Clear()	965
10.266.4.6	ComputeLength()	965
10.266.4.7	End() [1/2]	965
10.266.4.8	End() [2/2]	965
10.266.4.9	FindDataElement()	965
10.266.4.10	GetItem() [1/2]	965
10.266.4.11	GetItem() [2/2]	966
10.266.4.12	GetLength()	966
10.266.4.13	GetNumberOfItems()	966

10.266.4.14 IsEmpty()	966
10.266.4.15 IsUndefinedLength()	966
10.266.4.16 New()	967
10.266.4.17 operator=()	967
10.266.4.18 operator==()	967
10.266.4.19 Print()	967
10.266.4.20 Read()	967
10.266.4.21 RemoveItemByIndex()	968
10.266.4.22 SetLength()	968
10.266.4.23 SetLengthToUndefined()	968
10.266.4.24 SetNumberOfItems()	968
10.266.4.25 Write()	968
10.266.5 Member Data Documentation	968
10.266.5.1 Items	969
10.266.5.2 SequenceLengthField	969
10.267 gdcm::SerieHelper Class Reference	969
10.267.1 Detailed Description	970
10.267.2 Member Typedef Documentation	971
10.267.2.1 Rule	971
10.267.2.2 SerieRestrictions	971
10.267.2.3 SingleSerieUIDFileSetmap	971
10.267.3 Constructor & Destructor Documentation	971
10.267.3.1 SerieHelper()	971
10.267.3.2 ~SerieHelper()	971
10.267.4 Member Function Documentation	971
10.267.4.1 AddFile()	972
10.267.4.2 AddFileName()	972
10.267.4.3 AddRestriction() [1/3]	972
10.267.4.4 AddRestriction() [2/3]	972
10.267.4.5 AddRestriction() [3/3]	972
10.267.4.6 Clear()	972
10.267.4.7 CreateDefaultUniqueSeriesIdentifier()	973
10.267.4.8 CreateUniqueSeriesIdentifier()	973
10.267.4.9 FileNameOrdering()	973
10.267.4.10 GetFirstSingleSerieUIDFileSet()	973
10.267.4.11 GetNextSingleSerieUIDFileSet()	973
10.267.4.12 ImageNumberOrdering()	973
10.267.4.13 ImagePositionPatientOrdering()	973
10.267.4.14 OrderFileList()	974

10.267.4.15 SetDirectory()	974
10.267.4.16 SetLoadMode()	974
10.267.4.17 SetUseSeriesDetails()	974
10.267.4.18 UserOrdering()	974
10.267.5 Member Data Documentation	974
10.267.5.1 elem	974
10.267.5.2 ItFileSetHt	975
10.267.5.3 op	975
10.267.5.4 SingleSerieUIDFileSetHT	975
10.267.5.5 value	975
10.268 gdcm::Series Class Reference	975
10.268.1 Detailed Description	975
10.268.2 Constructor & Destructor Documentation	976
10.268.2.1 Series()	976
10.269 gdcm::network::ServiceClassApplicationInformation Class Reference	976
10.269.1 Detailed Description	976
10.269.2 Constructor & Destructor Documentation	976
10.269.2.1 ServiceClassApplicationInformation()	976
10.269.3 Member Function Documentation	977
10.269.3.1 Print()	977
10.269.3.2 Read()	977
10.269.3.3 SetTuple()	977
10.269.3.4 Size()	977
10.269.3.5 Write()	977
10.270 gdcm::ServiceClassUser Class Reference	978
10.270.1 Detailed Description	980
10.270.2 Constructor & Destructor Documentation	980
10.270.2.1 ServiceClassUser() [1/2]	980
10.270.2.2 ~ServiceClassUser()	980
10.270.2.3 ServiceClassUser() [2/2]	980
10.270.3 Member Function Documentation	981
10.270.3.1 GetAETitle()	981
10.270.3.2 GetCalledAETitle()	981
10.270.3.3 GetTimeout()	981
10.270.3.4 InitializeConnection()	981
10.270.3.5 IsPresentationContextAccepted()	981
10.270.3.6 New()	982
10.270.3.7 operator=()	982
10.270.3.8 SendEcho()	982



10.270.3.9 SendFind()	982
10.270.3.10 SendMove() [1/3]	982
10.270.3.11 SendMove() [2/3]	983
10.270.3.12 SendMove() [3/3]	983
10.270.3.13 SendStore() [1/3]	983
10.270.3.14 SendStore() [2/3]	983
10.270.3.15 SendStore() [3/3]	983
10.270.3.16 SetAETitle()	984
10.270.3.17 SetCalledAETitle()	984
10.270.3.18 SetHostname()	984
10.270.3.19 SetPort()	984
10.270.3.20 SetPortSCP()	985
10.270.3.21 SetPresentationContexts()	985
10.270.3.22 SetTimeout()	985
10.270.3.23 StartAssociation()	985
10.270.3.24 StopAssociation()	986
10.271 gdcmm::SHA1 Class Reference	986
10.271.1 Detailed Description	986
10.271.2 Constructor & Destructor Documentation	987
10.271.2.1 SHA1() [1/2]	987
10.271.2.2 ~SHA1()	987
10.271.2.3 SHA1() [2/2]	987
10.271.3 Member Function Documentation	987
10.271.3.1 Compute()	987
10.271.3.2 ComputeFile()	987
10.271.3.3 operator=()	988
10.272 gdcmm::SimpleMemberCommand< T > Class Template Reference	988
10.272.1 Detailed Description	990
10.272.2 Member Typedef Documentation	990
10.272.2.1 Self	990
10.272.2.2 TMemberFunctionPointer	990
10.272.3 Constructor & Destructor Documentation	990
10.272.3.1 SimpleMemberCommand() [1/2]	990
10.272.3.2 SimpleMemberCommand() [2/2]	991
10.272.3.3 ~SimpleMemberCommand()	991
10.272.4 Member Function Documentation	991
10.272.4.1 Execute() [1/2]	991
10.272.4.2 Execute() [2/2]	991
10.272.4.3 New()	992

10.272.4.4 operator=()	992
10.272.4.5 SetCallbackFunction()	992
10.272.5 Member Data Documentation	992
10.272.5.1 m_MemberFunction	992
10.272.5.2 m_This	993
10.273 gdcm::SimpleSubjectWatcher Class Reference	993
10.273.1 Detailed Description	993
10.273.2 Constructor & Destructor Documentation	994
10.273.2.1 SimpleSubjectWatcher() [1/2]	994
10.273.2.2 ~SimpleSubjectWatcher()	994
10.273.2.3 SimpleSubjectWatcher() [2/2]	994
10.273.3 Member Function Documentation	994
10.273.3.1 EndFilter()	994
10.273.3.2 operator=()	994
10.273.3.3 ShowAbort()	995
10.273.3.4 ShowAnonymization()	995
10.273.3.5 ShowData()	995
10.273.3.6 ShowDataSet()	995
10.273.3.7 ShowFileName()	995
10.273.3.8 ShowIteration()	996
10.273.3.9 ShowProgress()	996
10.273.3.10 StartFilter()	996
10.273.3.11 TestAbortOff()	996
10.273.3.12 TestAbortOn()	996
10.274 gdcm::MrProtocol::Slice Struct Reference	997
10.274.1 Member Data Documentation	997
10.274.1.1 Normal	997
10.274.1.2 Position	997
10.275 gdcm::MrProtocol::SliceArray Struct Reference	998
10.275.1 Member Data Documentation	998
10.275.1.1 Slices	998
10.276 gdcm::SmartPointer< ObjectType > Class Template Reference	999
10.276.1 Detailed Description	1000
10.276.2 Constructor & Destructor Documentation	1000
10.276.2.1 SmartPointer() [1/4]	1001
10.276.2.2 SmartPointer() [2/4]	1001
10.276.2.3 SmartPointer() [3/4]	1001
10.276.2.4 SmartPointer() [4/4]	1001
10.276.2.5 ~SmartPointer()	1001

10.276.3 Member Function Documentation	1001
10.276.3.1 GetPointer()	1002
10.276.3.2 operator ObjectType *()	1002
10.276.3.3 operator*()	1002
10.276.3.4 operator->()	1002
10.276.3.5 operator=() [1/3]	1002
10.276.3.6 operator=() [2/3]	1003
10.276.3.7 operator=() [3/3]	1003
10.277 gdcm::network::SOPClassExtendedNegociationSub Class Reference	1003
10.277.1 Detailed Description	1003
10.277.2 Constructor & Destructor Documentation	1004
10.277.2.1 SOPClassExtendedNegociationSub()	1004
10.277.3 Member Function Documentation	1004
10.277.3.1 Print()	1004
10.277.3.2 Read()	1004
10.277.3.3 SetTuple()	1004
10.277.3.4 Size()	1004
10.277.3.5 Write()	1005
10.278 gdcm::SOPClassUIDToIOD Class Reference	1005
10.278.1 Detailed Description	1005
10.278.2 Member Typedef Documentation	1005
10.278.2.1 const	1005
10.278.3 Member Function Documentation	1006
10.278.3.1 GetIOD()	1006
10.278.3.2 GetIODFromSOPClassUID()	1006
10.278.3.3 GetNumberOfSOPClassToIOD()	1006
10.278.3.4 GetSOPClassUIDFromIOD()	1006
10.278.3.5 GetSOPClassUIDToIOD()	1006
10.278.3.6 GetSOPClassUIDToIODs()	1007
10.279 gdcm::Sorter Class Reference	1007
10.279.1 Detailed Description	1008
10.279.2 Member Typedef Documentation	1009
10.279.2.1 SelectionMap	1009
10.279.2.2 SortFunction	1009
10.279.3 Constructor & Destructor Documentation	1009
10.279.3.1 Sorter()	1009
10.279.3.2 ~Sorter()	1009
10.279.4 Member Function Documentation	1009
10.279.4.1 AddSelect()	1009

10.279.4.2 GetFileNames()	1010
10.279.4.3 Print()	1010
10.279.4.4 SetSortFunction()	1010
10.279.4.5 SetTagsToRead()	1010
10.279.4.6 Sort()	1011
10.279.4.7 StableSort()	1011
10.279.5 Friends And Related Function Documentation	1011
10.279.5.1 operator<<	1011
10.279.6 Member Data Documentation	1011
10.279.6.1 Filenames	1011
10.279.6.2 Selection	1012
10.279.6.3 SortFunc	1012
10.279.6.4 TagsToRead	1012
10.280 gdcm::Spacing Class Reference	1012
10.280.1 Detailed Description	1013
10.280.2 Member Enumeration Documentation	1013
10.280.2.1 SpacingType	1013
10.280.3 Constructor & Destructor Documentation	1014
10.280.3.1 Spacing()	1014
10.280.3.2 ~Spacing()	1014
10.280.4 Member Function Documentation	1014
10.280.4.1 ComputePixelAspectRatioFromPixelSpacing()	1014
10.281 gdcm::Spectroscopy Class Reference	1014
10.281.1 Detailed Description	1015
10.281.2 Constructor & Destructor Documentation	1015
10.281.2.1 Spectroscopy()	1015
10.282 gdcm::SplitMosaicFilter Class Reference	1015
10.282.1 Detailed Description	1016
10.282.2 Constructor & Destructor Documentation	1016
10.282.2.1 SplitMosaicFilter()	1016
10.282.2.2 ~SplitMosaicFilter()	1016
10.282.3 Member Function Documentation	1016
10.282.3.1 ComputeMOSAICDimensions()	1016
10.282.3.2 ComputeMOSAICSliceNormal()	1017
10.282.3.3 ComputeMOSAICSlicePosition()	1017
10.282.3.4 GetAcquisitionSize()	1017
10.282.3.5 GetFile() [1/2]	1017
10.282.3.6 GetFile() [2/2]	1017
10.282.3.7 GetImage() [1/2]	1017

10.282.3.8 GetImage() [2/2]	1018
10.282.3.9 GetNumberOfImagesInMosaic()	1018
10.282.3.10 SetFile()	1018
10.282.3.11 SetImage()	1018
10.282.3.12 Split()	1018
10.283 gdcm::StartEvent Class Reference	1019
10.284 gdcm::static_assert_test< x > Struct Template Reference	1020
10.285 gdcm::STATIC_ASSERTION_FAILURE< x > Struct Template Reference	1020
10.286 gdcm::STATIC_ASSERTION_FAILURE< true > Struct Reference	1020
10.286.1 Member Enumeration Documentation	1020
10.286.1.1 anonymous enum	1020
10.287 gdcm::StreamImageReader Class Reference	1021
10.287.1 Detailed Description	1021
10.287.2 Constructor & Destructor Documentation	1021
10.287.2.1 StreamImageReader()	1022
10.287.2.2 ~StreamImageReader()	1022
10.287.3 Member Function Documentation	1022
10.287.3.1 CanReadImage()	1022
10.287.3.2 DefinePixelExtent()	1022
10.287.3.3 DefineProperBufferLength()	1023
10.287.3.4 GetDimensionsValueForResolution()	1023
10.287.3.5 GetFile()	1023
10.287.3.6 Read()	1023
10.287.3.7 ReadImageInformation()	1024
10.287.3.8 SetFileName()	1024
10.287.3.9 SetStream()	1024
10.288 gdcm::StreamImageWriter Class Reference	1025
10.288.1 Detailed Description	1026
10.288.2 Constructor & Destructor Documentation	1027
10.288.2.1 StreamImageWriter()	1027
10.288.2.2 ~StreamImageWriter()	1027
10.288.3 Member Function Documentation	1027
10.288.3.1 CanWriteFile()	1027
10.288.3.2 DefinePixelExtent()	1027
10.288.3.3 DefineProperBufferLength()	1028
10.288.3.4 SetFile()	1028
10.288.3.5 SetFileName()	1028
10.288.3.6 SetStream()	1028
10.288.3.7 Write()	1029

10.288.3.8 WriteImageInformation()	1029
10.288.3.9 WriteImageSubregionRAW()	1029
10.288.3.10 WriteRawHeader()	1030
10.288.4 Member Data Documentation	1030
10.288.4.1 mElementOffsets	1030
10.288.4.2 mElementOffsets1	1030
10.288.4.3 mspFile	1030
10.288.4.4 mWriter	1030
10.288.4.5 mXMax	1031
10.288.4.6 mXMin	1031
10.288.4.7 mYMax	1031
10.288.4.8 mYMin	1031
10.288.4.9 mZMax	1031
10.288.4.10 mZMin	1031
10.289 gdcm::StrictScanner Class Reference	1032
10.289.1 Detailed Description	1034
10.289.2 Member Typedef Documentation	1034
10.289.2.1 ConstIterator	1034
10.289.2.2 MappingType	1035
10.289.2.3 TagToValue	1035
10.289.2.4 TagToValueValueType	1035
10.289.2.5 ValuesType	1035
10.289.3 Constructor & Destructor Documentation	1035
10.289.3.1 StrictScanner()	1035
10.289.3.2 ~StrictScanner()	1035
10.289.4 Member Function Documentation	1035
10.289.4.1 AddPrivateTag()	1036
10.289.4.2 AddSkipTag()	1036
10.289.4.3 AddTag()	1036
10.289.4.4 Begin()	1036
10.289.4.5 ClearSkipTags()	1036
10.289.4.6 ClearTags()	1036
10.289.4.7 End()	1037
10.289.4.8 GetAllFilenamesFromTagToValue()	1037
10.289.4.9 GetFilenameFromTagToValue()	1037
10.289.4.10 GetFilenames()	1037
10.289.4.11 GetKeys()	1037
10.289.4.12 GetMapping()	1037
10.289.4.13 GetMappingFromTagToValue()	1038

10.289.4.14 GetMappings()	1038
10.289.4.15 GetOrderedValues()	1038
10.289.4.16 GetValue()	1038
10.289.4.17 GetValues() [1/2]	1038
10.289.4.18 GetValues() [2/2]	1039
10.289.4.19 IsKey()	1039
10.289.4.20 New()	1039
10.289.4.21 Print()	1039
10.289.4.22 PrintTable()	1039
10.289.4.23 ProcessPublicTag()	1040
10.289.4.24 Scan()	1040
10.289.5 Friends And Related Function Documentation	1040
10.289.5.1 operator<<	1040
10.290 gdcm::String< TDelimiter, TMaxLength, TPadChar > Class Template Reference	1041
10.290.1 Detailed Description	1042
10.290.2 Member Typedef Documentation	1043
10.290.2.1 const_iterator	1043
10.290.2.2 const_reference	1043
10.290.2.3 const_reverse_iterator	1043
10.290.2.4 difference_type	1043
10.290.2.5 iterator	1043
10.290.2.6 pointer	1044
10.290.2.7 reference	1044
10.290.2.8 reverse_iterator	1044
10.290.2.9 size_type	1044
10.290.2.10 value_type	1044
10.290.3 Constructor & Destructor Documentation	1044
10.290.3.1 String() [1/4]	1044
10.290.3.2 String() [2/4]	1045
10.290.3.3 String() [3/4]	1045
10.290.3.4 String() [4/4]	1045
10.290.4 Member Function Documentation	1045
10.290.4.1 IsValid()	1045
10.290.4.2 operator const char *()	1045
10.290.4.3 Trim() [1/2]	1046
10.290.4.4 Trim() [2/2]	1046
10.290.4.5 Truncate()	1046
10.291 gdcm::StringFilter Class Reference	1046
10.291.1 Detailed Description	1047

10.291.2 Constructor & Destructor Documentation	1047
10.291.2.1 StringFilter()	1047
10.291.2.2 ~StringFilter()	1048
10.291.3 Member Function Documentation	1048
10.291.3.1 ExecuteQuery() [1/2]	1048
10.291.3.2 ExecuteQuery() [2/2]	1048
10.291.3.3 FromString()	1048
10.291.3.4 GetFile() [1/2]	1048
10.291.3.5 GetFile() [2/2]	1049
10.291.3.6 SetDicts()	1049
10.291.3.7 SetFile()	1049
10.291.3.8 ToString() [1/3]	1049
10.291.3.9 ToString() [2/3]	1049
10.291.3.10 ToString() [3/3]	1050
10.291.3.11 ToStringPair() [1/3]	1050
10.291.3.12 ToStringPair() [2/3]	1050
10.291.3.13 ToStringPair() [3/3]	1050
10.291.3.14 UseDictAlways()	1050
10.292 gdcM::Study Class Reference	1051
10.292.1 Detailed Description	1051
10.292.2 Constructor & Destructor Documentation	1051
10.292.2.1 Study()	1051
10.293 gdcM::Subject Class Reference	1051
10.293.1 Detailed Description	1053
10.293.2 Constructor & Destructor Documentation	1053
10.293.2.1 Subject()	1053
10.293.2.2 ~Subject()	1053
10.293.3 Member Function Documentation	1053
10.293.3.1 AddObserver() [1/2]	1053
10.293.3.2 AddObserver() [2/2]	1054
10.293.3.3 GetCommand()	1054
10.293.3.4 HasObserver()	1054
10.293.3.5 InvokeEvent() [1/2]	1054
10.293.3.6 InvokeEvent() [2/2]	1054
10.293.3.7 RemoveAllObservers()	1054
10.293.3.8 RemoveObserver()	1055
10.294 gdcM::Surface Class Reference	1055
10.294.1 Detailed Description	1057
10.294.2 Member Enumeration Documentation	1058



10.294.2.1 STATES	1058
10.294.2.2 VIEWType	1058
10.294.3 Constructor & Destructor Documentation	1058
10.294.3.1 Surface()	1058
10.294.3.2 ~Surface()	1059
10.294.4 Member Function Documentation	1059
10.294.4.1 GetAlgorithmFamily() [1/2]	1059
10.294.4.2 GetAlgorithmFamily() [2/2]	1059
10.294.4.3 GetAlgorithmName()	1059
10.294.4.4 GetAlgorithmVersion()	1059
10.294.4.5 GetAxisOfRotation()	1059
10.294.4.6 GetCenterOfRotation()	1060
10.294.4.7 GetFiniteVolume()	1060
10.294.4.8 GetManifold()	1060
10.294.4.9 GetMaximumPointDistance()	1060
10.294.4.10 GetMeanPointDistance()	1060
10.294.4.11 GetMeshPrimitive() [1/2]	1060
10.294.4.12 GetMeshPrimitive() [2/2]	1061
10.294.4.13 GetNumberOfSurfacePoints()	1061
10.294.4.14 GetNumberOfVectors()	1061
10.294.4.15 GetPointCoordinatesData() [1/2]	1061
10.294.4.16 GetPointCoordinatesData() [2/2]	1061
10.294.4.17 GetPointPositionAccuracy()	1061
10.294.4.18 GetPointsBoundingBoxCoordinates()	1062
10.294.4.19 GetProcessingAlgorithm() [1/2]	1062
10.294.4.20 GetProcessingAlgorithm() [2/2]	1062
10.294.4.21 GetRecommendedDisplayCIELabValue() [1/2]	1062
10.294.4.22 GetRecommendedDisplayCIELabValue() [2/2]	1062
10.294.4.23 GetRecommendedDisplayGrayscaleValue()	1062
10.294.4.24 GetRecommendedPresentationOpacity()	1063
10.294.4.25 GetRecommendedPresentationType()	1063
10.294.4.26 GetSTATES()	1063
10.294.4.27 GetSTATESString()	1063
10.294.4.28 GetSurfaceComments()	1063
10.294.4.29 GetSurfaceNumber()	1063
10.294.4.30 GetSurfaceProcessing()	1063
10.294.4.31 GetSurfaceProcessingDescription()	1064
10.294.4.32 GetSurfaceProcessingRatio()	1064
10.294.4.33 GetVectorAccuracy()	1064

10.294.4.34	GetVectorCoordinateData() [1/2]	1064
10.294.4.35	GetVectorCoordinateData() [2/2]	1064
10.294.4.36	GetVectorDimensionality()	1064
10.294.4.37	GetVIEWType()	1064
10.294.4.38	GetVIEWTypeString()	1065
10.294.4.39	SetAlgorithmFamily()	1065
10.294.4.40	SetAlgorithmName()	1065
10.294.4.41	SetAlgorithmVersion()	1065
10.294.4.42	SetAxisOfRotation()	1065
10.294.4.43	SetCenterOfRotation()	1065
10.294.4.44	SetFiniteVolume()	1066
10.294.4.45	SetManifold()	1066
10.294.4.46	SetMaximumPointDistance()	1066
10.294.4.47	SetMeanPointDistance()	1066
10.294.4.48	SetMeshPrimitive()	1066
10.294.4.49	SetNumberOfSurfacePoints()	1066
10.294.4.50	SetNumberOfVectors()	1067
10.294.4.51	SetPointCoordinatesData()	1067
10.294.4.52	SetPointPositionAccuracy()	1067
10.294.4.53	SetPointsBoundingBoxCoordinates()	1067
10.294.4.54	SetProcessingAlgorithm()	1067
10.294.4.55	SetRecommendedDisplayCIELabValue() [1/3]	1067
10.294.4.56	SetRecommendedDisplayCIELabValue() [2/3]	1068
10.294.4.57	SetRecommendedDisplayCIELabValue() [3/3]	1068
10.294.4.58	SetRecommendedDisplayGrayscaleValue()	1068
10.294.4.59	SetRecommendedPresentationOpacity()	1068
10.294.4.60	SetRecommendedPresentationType()	1068
10.294.4.61	SetSurfaceComments()	1068
10.294.4.62	SetSurfaceNumber()	1069
10.294.4.63	SetSurfaceProcessing()	1069
10.294.4.64	SetSurfaceProcessingDescription()	1069
10.294.4.65	SetSurfaceProcessingRatio()	1069
10.294.4.66	SetVectorAccuracy()	1069
10.294.4.67	SetVectorCoordinateData()	1069
10.294.4.68	SetVectorDimensionality()	1070
10.295	gdcm::SurfaceHelper Class Reference	1070
10.295.1	Detailed Description	1070
10.295.2	Member Typedef Documentation	1071
10.295.2.1	ColorArray	1071

10.295.3 Member Function Documentation	1071
10.295.3.1 RecommendedDisplayCIELabToRGB() [1/2]	1071
10.295.3.2 RecommendedDisplayCIELabToRGB() [2/2]	1071
10.295.3.3 RGBToRecommendedDisplayCIELab()	1072
10.295.3.4 RGBToRecommendedDisplayGrayscale()	1073
10.296 gdcmm::SurfaceReader Class Reference	1073
10.296.1 Detailed Description	1075
10.296.2 Constructor & Destructor Documentation	1075
10.296.2.1 SurfaceReader()	1075
10.296.2.2 ~SurfaceReader()	1075
10.296.3 Member Function Documentation	1075
10.296.3.1 GetNumberOfSurfaces()	1076
10.296.3.2 Read()	1076
10.296.3.3 ReadPointMacro()	1076
10.296.3.4 ReadSurface()	1076
10.296.3.5 ReadSurfaces()	1076
10.297 gdcmm::SurfaceWriter Class Reference	1077
10.297.1 Detailed Description	1078
10.297.2 Constructor & Destructor Documentation	1078
10.297.2.1 SurfaceWriter()	1078
10.297.2.2 ~SurfaceWriter()	1078
10.297.3 Member Function Documentation	1078
10.297.3.1 ComputeNumberOfSurfaces()	1078
10.297.3.2 GetNumberOfSurfaces()	1079
10.297.3.3 PrepareWrite()	1079
10.297.3.4 PrepareWritePointMacro()	1079
10.297.3.5 SetNumberOfSurfaces()	1079
10.297.3.6 Write()	1079
10.297.4 Member Data Documentation	1079
10.297.4.1 NumberOfSurfaces	1080
10.298 gdcmm::SwapCode Class Reference	1080
10.298.1 Detailed Description	1081
10.298.2 Member Enumeration Documentation	1081
10.298.2.1 SwapCodeType	1081
10.298.3 Constructor & Destructor Documentation	1081
10.298.3.1 SwapCode()	1081
10.298.4 Member Function Documentation	1081
10.298.4.1 GetIndex()	1081
10.298.4.2 GetSwapCodeString()	1082

10.298.4.3 operator SwapCode::SwapCodeType()	1082
10.298.5 Friends And Related Function Documentation	1082
10.298.5.1 operator<<	1082
10.299 gdcmm::SwapperDoOp Class Reference	1082
10.299.1 Member Function Documentation	1082
10.299.1.1 Swap()	1083
10.299.1.2 SwapArray()	1083
10.300 gdcmm::SwapperNoOp Class Reference	1083
10.300.1 Detailed Description	1083
10.300.2 Member Function Documentation	1083
10.300.2.1 Swap()	1084
10.300.2.2 SwapArray()	1084
10.301 gdcmm::System Class Reference	1084
10.301.1 Detailed Description	1085
10.301.2 Member Function Documentation	1085
10.301.2.1 ConvertToUNC()	1085
10.301.2.2 DeleteDirectory()	1086
10.301.2.3 EncodeBytes()	1086
10.301.2.4 FileExists()	1086
10.301.2.5 FileIsDirectory()	1086
10.301.2.6 FileIsSymlink()	1087
10.301.2.7 FileSize()	1087
10.301.2.8 FileTime()	1087
10.301.2.9 FormatDateTime()	1087
10.301.2.10 GetCurrentDateTime()	1088
10.301.2.11 GetCurrentModuleFileName()	1088
10.301.2.12 GetCurrentProcessFileName()	1088
10.301.2.13 GetCurrentResourcesDirectory()	1088
10.301.2.14 GetCWD()	1088
10.301.2.15 GetHostName()	1088
10.301.2.16 GetLastSystemError()	1089
10.301.2.17 GetLocaleCharset()	1089
10.301.2.18 GetPermissions()	1089
10.301.2.19 GetTimezoneOffsetFromUTC()	1089
10.301.2.20 MakeDirectory()	1089
10.301.2.21 ParseDateTime() [1/2]	1090
10.301.2.22 ParseDateTime() [2/2]	1090
10.301.2.23 RemoveFile()	1090
10.301.2.24 SetPermissions()	1090

---

10.301.2.25 StrCaseCmp()	1091
10.301.2.26 StrNCaseCmp()	1091
10.301.2.27 StrSep()	1091
10.301.2.28 StrTokR()	1091
10.302 gdcM::Table Class Reference	1092
10.302.1 Detailed Description	1093
10.302.2 Member Typedef Documentation	1093
10.302.2.1 MapTableEntry	1093
10.302.3 Constructor & Destructor Documentation	1093
10.302.3.1 Table() [1/2]	1093
10.302.3.2 ~Table()	1093
10.302.3.3 Table() [2/2]	1093
10.302.4 Member Function Documentation	1093
10.302.4.1 GetTableEntry()	1094
10.302.4.2 InsertEntry()	1094
10.302.4.3 operator=()	1094
10.302.5 Friends And Related Function Documentation	1094
10.302.5.1 operator<<	1094
10.302.6 Member Data Documentation	1094
10.302.6.1 TableInternal	1094
10.303 gdcM::TableEntry Class Reference	1095
10.303.1 Detailed Description	1095
10.303.2 Constructor & Destructor Documentation	1095
10.303.2.1 TableEntry()	1095
10.303.2.2 ~TableEntry()	1095
10.304 gdcM::TableReader Class Reference	1096
10.304.1 Detailed Description	1096
10.304.2 Constructor & Destructor Documentation	1097
10.304.2.1 TableReader()	1097
10.304.2.2 ~TableReader()	1097
10.304.3 Member Function Documentation	1097
10.304.3.1 CharacterDataHandler()	1097
10.304.3.2 EndElement()	1097
10.304.3.3 GetDefs()	1097
10.304.3.4 GetFilename()	1098
10.304.3.5 HandleOD()	1098
10.304.3.6 HandleODEntry()	1098
10.304.3.7 HandleMacro()	1098
10.304.3.8 HandleMacroEntry()	1098

10.304.3.9 HandleMacroEntryDescription()	1098
10.304.3.10 HandleModule()	1099
10.304.3.11 HandleModuleEntry()	1099
10.304.3.12 HandleModuleEntryDescription()	1099
10.304.3.13 HandleModuleInclude()	1099
10.304.3.14 Read()	1099
10.304.3.15 SetFilename()	1099
10.304.3.16 StartElement()	1100
10.305 gdcmm::network::TableRow Class Reference	1100
10.305.1 Constructor & Destructor Documentation	1101
10.305.1.1 TableRow()	1101
10.305.1.2 ~TableRow()	1101
10.305.2 Member Data Documentation	1101
10.305.2.1 transitions	1101
10.306 gdcmm::Tag Class Reference	1101
10.306.1 Detailed Description	1103
10.306.2 Constructor & Destructor Documentation	1103
10.306.2.1 Tag() [1/3]	1104
10.306.2.2 Tag() [2/3]	1104
10.306.2.3 Tag() [3/3]	1104
10.306.3 Member Function Documentation	1104
10.306.3.1 GetElement()	1104
10.306.3.2 GetElementTag()	1105
10.306.3.3 GetGroup()	1105
10.306.3.4 GetLength()	1105
10.306.3.5 GetPrivateCreator()	1105
10.306.3.6 IsGroupLength()	1105
10.306.3.7 IsGroupXX()	1106
10.306.3.8 IsIllegal()	1106
10.306.3.9 IsPrivate()	1106
10.306.3.10 IsPrivateCreator()	1106
10.306.3.11 IsPublic()	1107
10.306.3.12 operator!=(())	1107
10.306.3.13 operator<()	1107
10.306.3.14 operator<=()	1107
10.306.3.15 operator=()	1107
10.306.3.16 operator==(())	1108
10.306.3.17 operator[]() [1/2]	1108
10.306.3.18 operator[]() [2/2]	1108

10.306.3.19 PrintAsContinuousString()	1108
10.306.3.20 PrintAsContinuousUpperCaseString()	1108
10.306.3.21 PrintAsPipeSeparatedString()	1109
10.306.3.22 Read()	1109
10.306.3.23 ReadFromCommaSeparatedString()	1109
10.306.3.24 ReadFromContinuousString()	1109
10.306.3.25 ReadFromPipeSeparatedString()	1109
10.306.3.26 SetElement()	1110
10.306.3.27 SetElementTag() [1/2]	1110
10.306.3.28 SetElementTag() [2/2]	1110
10.306.3.29 SetGroup()	1110
10.306.3.30 SetPrivateCreator()	1111
10.306.3.31 Write()	1111
10.306.4 Friends And Related Function Documentation	1111
10.306.4.1 operator<<	1111
10.306.4.2 operator>>	1111
10.306.5 Member Data Documentation	1112
10.306.5.1 bytes	1112
10.306.5.2 tag	1112
10.306.5.3 tags	1112
10.307 gdcm::TagPath Class Reference	1112
10.307.1 Detailed Description	1113
10.307.2 Constructor & Destructor Documentation	1113
10.307.2.1 TagPath()	1113
10.307.2.2 ~TagPath()	1113
10.307.3 Member Function Documentation	1113
10.307.3.1 ConstructFromString()	1113
10.307.3.2 ConstructFromTagList()	1114
10.307.3.3 IsValid()	1114
10.307.3.4 Print()	1114
10.307.3.5 Push() [1/2]	1114
10.307.3.6 Push() [2/2]	1114
10.308 gdcm::Testing Class Reference	1115
10.308.1 Detailed Description	1116
10.308.2 Member Typedef Documentation	1116
10.308.2.1 MD5DataImagesType	1116
10.308.2.2 MediaStorageDataFilesType	1116
10.308.3 Constructor & Destructor Documentation	1116
10.308.3.1 Testing()	1116

10.308.3.2 ~Testing()	1117
10.308.4 Member Function Documentation	1117
10.308.4.1 ComputeFileMD5()	1117
10.308.4.2 ComputeMD5()	1117
10.308.4.3 GetDataExtraRoot()	1117
10.308.4.4 GetDataRoot()	1118
10.308.4.5 GetFileName()	1118
10.308.4.6 GetFileNames()	1118
10.308.4.7 GetLossyFlagFromFile()	1118
10.308.4.8 GetMD5DataImage()	1119
10.308.4.9 GetMD5DataImages()	1119
10.308.4.10 GetMD5FromBrokenFile()	1119
10.308.4.11 GetMD5FromFile()	1119
10.308.4.12 GetMediaStorageDataFile()	1119
10.308.4.13 GetMediaStorageDataFiles()	1119
10.308.4.14 GetMediaStorageFromFile()	1120
10.308.4.15 GetNumberOfFileNames()	1120
10.308.4.16 GetNumberOfMD5DataImages()	1120
10.308.4.17 GetNumberOfMediaStorageDataFiles()	1120
10.308.4.18 GetPixelSpacingDataRoot()	1120
10.308.4.19 GetSelectedPrivateGroupOffsetFromFile()	1121
10.308.4.20 GetSelectedTagsOffsetFromFile()	1121
10.308.4.21 GetSourceDirectory()	1121
10.308.4.22 GetStreamOffsetFromFile()	1121
10.308.4.23 GetTempDirectory()	1121
10.308.4.24 GetTempDirectoryW()	1122
10.308.4.25 GetTempFilename()	1122
10.308.4.26 GetTempFilenameW()	1122
10.308.4.27 Print()	1122
10.309 gdcmm::Trace Class Reference	1123
10.309.1 Detailed Description	1124
10.309.2 Constructor & Destructor Documentation	1124
10.309.2.1 Trace()	1124
10.309.2.2 ~Trace()	1124
10.309.3 Member Function Documentation	1124
10.309.3.1 DebugOff()	1124
10.309.3.2 DebugOn()	1125
10.309.3.3 ErrorOff()	1125
10.309.3.4 ErrorOn()	1125



10.309.3.5 GetDebugFlag()	1125
10.309.3.6 GetDebugStream()	1125
10.309.3.7 GetErrorFlag()	1125
10.309.3.8 GetErrorStream()	1126
10.309.3.9 GetStream()	1126
10.309.3.10 GetWarningFlag()	1126
10.309.3.11 GetWarningStream()	1126
10.309.3.12 SetDebug()	1126
10.309.3.13 SetDebugStream()	1126
10.309.3.14 SetError()	1127
10.309.3.15 SetErrorStream()	1127
10.309.3.16 SetStream()	1127
10.309.3.17 SetStreamToFile()	1127
10.309.3.18 SetWarning()	1127
10.309.3.19 SetWarningStream()	1128
10.309.3.20 WarningOff()	1128
10.309.3.21 WarningOn()	1128
10.310 gdcm::TransferSyntax Class Reference	1128
10.310.1 Detailed Description	1130
10.310.2 Member Enumeration Documentation	1130
10.310.2.1 NegotiatedType	1130
10.310.2.2 TSType	1131
10.310.3 Constructor & Destructor Documentation	1131
10.310.3.1 TransferSyntax()	1132
10.310.4 Member Function Documentation	1132
10.310.4.1 CanStoreLossy()	1132
10.310.4.2 GetNegotiatedType()	1132
10.310.4.3 GetString()	1132
10.310.4.4 GetSwapCode()	1132
10.310.4.5 GetTSString()	1133
10.310.4.6 GetTSType()	1133
10.310.4.7 IsEncapsulated()	1133
10.310.4.8 IsEncoded()	1133
10.310.4.9 IsExplicit()	1133
10.310.4.10 IsImplicit()	1134
10.310.4.11 IsLossless()	1134
10.310.4.12 IsLossy()	1134
10.310.4.13 IsValid()	1134
10.310.4.14 operator TSType()	1134

10.310.5 Friends And Related Function Documentation . . . . .	1134
10.310.5.1 operator<< . . . . .	1134
10.311 gdcmm::network::TransferSyntaxSub Class Reference . . . . .	1135
10.311.1 Detailed Description . . . . .	1135
10.311.2 Constructor & Destructor Documentation . . . . .	1135
10.311.2.1 TransferSyntaxSub() . . . . .	1135
10.311.3 Member Function Documentation . . . . .	1135
10.311.3.1 GetName() . . . . .	1136
10.311.3.2 operator==( ) . . . . .	1136
10.311.3.3 Print() . . . . .	1136
10.311.3.4 Read() . . . . .	1136
10.311.3.5 SetName() . . . . .	1136
10.311.3.6 SetNameFromUID() . . . . .	1136
10.311.3.7 Size() . . . . .	1137
10.311.3.8 Write() . . . . .	1137
10.312 gdcmm::network::Transition Struct Reference . . . . .	1137
10.312.1 Constructor & Destructor Documentation . . . . .	1138
10.312.1.1 Transition() [1/2] . . . . .	1138
10.312.1.2 ~Transition() . . . . .	1138
10.312.1.3 Transition() [2/2] . . . . .	1138
10.312.2 Member Function Documentation . . . . .	1138
10.312.2.1 MakeNew() . . . . .	1138
10.312.3 Member Data Documentation . . . . .	1139
10.312.3.1 mAction . . . . .	1139
10.312.3.2 mEnd . . . . .	1139
10.313 gdcmm::Type Class Reference . . . . .	1139
10.313.1 Detailed Description . . . . .	1140
10.313.2 Member Enumeration Documentation . . . . .	1140
10.313.2.1 TypeType . . . . .	1140
10.313.3 Constructor & Destructor Documentation . . . . .	1140
10.313.3.1 Type() . . . . .	1141
10.313.4 Member Function Documentation . . . . .	1141
10.313.4.1 GetTypeString() . . . . .	1141
10.313.4.2 GetTypeType() . . . . .	1141
10.313.4.3 operator TypeType() . . . . .	1141
10.313.5 Friends And Related Function Documentation . . . . .	1141
10.313.5.1 operator<< . . . . .	1141
10.314 gdcmm::UI Struct Reference . . . . .	1142
10.314.1 Friends And Related Function Documentation . . . . .	1142

10.314.1.1 operator<< . . . . .	1142
10.314.2 Member Data Documentation . . . . .	1142
10.314.2.1 Internal . . . . .	1142
10.315 gdcmm::UIDGenerator Class Reference . . . . .	1142
10.315.1 Detailed Description . . . . .	1143
10.315.2 Constructor & Destructor Documentation . . . . .	1143
10.315.2.1 UIDGenerator() . . . . .	1143
10.315.3 Member Function Documentation . . . . .	1144
10.315.3.1 Generate() . . . . .	1144
10.315.3.2 GenerateUUID() . . . . .	1144
10.315.3.3 GetGDCMUID() . . . . .	1144
10.315.3.4 GetRoot() . . . . .	1144
10.315.3.5 IsValid() . . . . .	1145
10.315.3.6 SetRoot() . . . . .	1145
10.316 gdcmm::UIDs Class Reference . . . . .	1145
10.316.1 Detailed Description . . . . .	1161
10.316.2 Member Typedef Documentation . . . . .	1161
10.316.2.1 TransferSyntaxStringsType . . . . .	1161
10.316.3 Member Enumeration Documentation . . . . .	1162
10.316.3.1 TSName . . . . .	1162
10.316.3.2 TSType . . . . .	1171
10.316.4 Member Function Documentation . . . . .	1181
10.316.4.1 GetName() . . . . .	1181
10.316.4.2 GetNumberOfTransferSyntaxStrings() . . . . .	1181
10.316.4.3 GetString() . . . . .	1181
10.316.4.4 GetTransferSyntaxString() . . . . .	1181
10.316.4.5 GetTransferSyntaxStrings() . . . . .	1182
10.316.4.6 GetUIDName() . . . . .	1182
10.316.4.7 GetUIDString() . . . . .	1182
10.316.4.8 operator TSType() . . . . .	1182
10.316.4.9 SetFromUID() . . . . .	1182
10.317 gdcmm::network::ULAction Class Reference . . . . .	1183
10.317.1 Detailed Description . . . . .	1184
10.317.2 Constructor & Destructor Documentation . . . . .	1184
10.317.2.1 ULAction() [1/2] . . . . .	1184
10.317.2.2 ~ULAction() . . . . .	1185
10.317.2.3 ULAction() [2/2] . . . . .	1185
10.317.3 Member Function Documentation . . . . .	1185
10.317.3.1 operator=() . . . . .	1185

10.317.3.2 PerformAction()	1185
10.318 gdcmm::network::ULActionAA1 Class Reference	1186
10.318.1 Member Function Documentation	1186
10.318.1.1 PerformAction()	1187
10.319 gdcmm::network::ULActionAA2 Class Reference	1187
10.319.1 Member Function Documentation	1188
10.319.1.1 PerformAction()	1188
10.320 gdcmm::network::ULActionAA3 Class Reference	1188
10.320.1 Member Function Documentation	1189
10.320.1.1 PerformAction()	1189
10.321 gdcmm::network::ULActionAA4 Class Reference	1190
10.321.1 Member Function Documentation	1190
10.321.1.1 PerformAction()	1191
10.322 gdcmm::network::ULActionAA5 Class Reference	1191
10.322.1 Member Function Documentation	1192
10.322.1.1 PerformAction()	1192
10.323 gdcmm::network::ULActionAA6 Class Reference	1192
10.323.1 Member Function Documentation	1193
10.323.1.1 PerformAction()	1193
10.324 gdcmm::network::ULActionAA7 Class Reference	1194
10.324.1 Member Function Documentation	1194
10.324.1.1 PerformAction()	1195
10.325 gdcmm::network::ULActionAA8 Class Reference	1195
10.325.1 Member Function Documentation	1196
10.325.1.1 PerformAction()	1196
10.326 gdcmm::network::ULActionAE1 Class Reference	1196
10.326.1 Member Function Documentation	1197
10.326.1.1 PerformAction()	1197
10.327 gdcmm::network::ULActionAE2 Class Reference	1198
10.327.1 Member Function Documentation	1198
10.327.1.1 PerformAction()	1199
10.328 gdcmm::network::ULActionAE3 Class Reference	1199
10.328.1 Member Function Documentation	1200
10.328.1.1 PerformAction()	1200
10.329 gdcmm::network::ULActionAE4 Class Reference	1200
10.329.1 Member Function Documentation	1201
10.329.1.1 PerformAction()	1201
10.330 gdcmm::network::ULActionAE5 Class Reference	1202
10.330.1 Member Function Documentation	1202

10.330.1.1 PerformAction()	1203
10.331 gdcn::network::ULActionAE6 Class Reference	1203
10.331.1 Member Function Documentation	1204
10.331.1.1 PerformAction()	1204
10.332 gdcn::network::ULActionAE7 Class Reference	1204
10.332.1 Member Function Documentation	1205
10.332.1.1 PerformAction()	1205
10.333 gdcn::network::ULActionAE8 Class Reference	1206
10.333.1 Member Function Documentation	1206
10.333.1.1 PerformAction()	1207
10.334 gdcn::network::ULActionAR1 Class Reference	1207
10.334.1 Member Function Documentation	1208
10.334.1.1 PerformAction()	1208
10.335 gdcn::network::ULActionAR10 Class Reference	1208
10.335.1 Member Function Documentation	1209
10.335.1.1 PerformAction()	1209
10.336 gdcn::network::ULActionAR2 Class Reference	1210
10.336.1 Member Function Documentation	1210
10.336.1.1 PerformAction()	1211
10.337 gdcn::network::ULActionAR3 Class Reference	1211
10.337.1 Member Function Documentation	1212
10.337.1.1 PerformAction()	1212
10.338 gdcn::network::ULActionAR4 Class Reference	1212
10.338.1 Member Function Documentation	1213
10.338.1.1 PerformAction()	1213
10.339 gdcn::network::ULActionAR5 Class Reference	1214
10.339.1 Member Function Documentation	1214
10.339.1.1 PerformAction()	1215
10.340 gdcn::network::ULActionAR6 Class Reference	1215
10.340.1 Member Function Documentation	1216
10.340.1.1 PerformAction()	1216
10.341 gdcn::network::ULActionAR7 Class Reference	1216
10.341.1 Member Function Documentation	1217
10.341.1.1 PerformAction()	1217
10.342 gdcn::network::ULActionAR8 Class Reference	1218
10.342.1 Member Function Documentation	1218
10.342.1.1 PerformAction()	1219
10.343 gdcn::network::ULActionAR9 Class Reference	1219
10.343.1 Member Function Documentation	1220

10.343.1.1 PerformAction()	1220
10.344 gdcn::network::ULActionDT1 Class Reference	1220
10.344.1 Member Function Documentation	1221
10.344.1.1 PerformAction()	1221
10.345 gdcn::network::ULActionDT2 Class Reference	1222
10.345.1 Member Function Documentation	1222
10.345.1.1 PerformAction()	1223
10.346 gdcn::network::ULBasicCallback Class Reference	1223
10.346.1 Detailed Description	1224
10.346.2 Constructor & Destructor Documentation	1224
10.346.2.1 ULBasicCallback()	1224
10.346.2.2 ~ULBasicCallback()	1224
10.346.3 Member Function Documentation	1224
10.346.3.1 GetDataSets()	1224
10.346.3.2 GetResponses()	1225
10.346.3.3 HandleDataSet()	1225
10.346.3.4 HandleResponse()	1225
10.347 gdcn::network::ULConnection Class Reference	1225
10.347.1 Detailed Description	1226
10.347.2 Constructor & Destructor Documentation	1227
10.347.2.1 ULConnection() [1/2]	1227
10.347.2.2 ~ULConnection()	1227
10.347.2.3 ULConnection() [2/2]	1227
10.347.3 Member Function Documentation	1227
10.347.3.1 AddAcceptedPresentationContext()	1227
10.347.3.2 FindContext()	1227
10.347.3.3 GetAcceptedPresentationContexts() [1/2]	1228
10.347.3.4 GetAcceptedPresentationContexts() [2/2]	1228
10.347.3.5 GetConnectionInfo()	1228
10.347.3.6 GetMaxPDUSize()	1228
10.347.3.7 GetPresentationContextACByID()	1228
10.347.3.8 GetPresentationContextIDFromPresentationContext()	1228
10.347.3.9 GetPresentationContextRQByID()	1229
10.347.3.10 GetPresentationContexts()	1229
10.347.3.11 GetProtocol()	1229
10.347.3.12 GetState()	1229
10.347.3.13 GetTimer()	1229
10.347.3.14 InitializeConnection()	1229
10.347.3.15 InitializeIncomingConnection()	1230

10.347.3.16 operator=()	1230
10.347.3.17 SetMaxPDUSize()	1230
10.347.3.18 SetPresentationContexts() [1/2]	1230
10.347.3.19 SetPresentationContexts() [2/2]	1230
10.347.3.20 SetState()	1230
10.347.3.21 StopProtocol()	1231
10.347.4 Friends And Related Function Documentation	1231
10.347.4.1 ULActionAE6	1231
10.347.4.2 ULConnectionManager	1231
10.348 gdcmm::network::ULConnectionCallback Class Reference	1231
10.348.1 Detailed Description	1232
10.348.2 Constructor & Destructor Documentation	1232
10.348.2.1 ULConnectionCallback()	1232
10.348.2.2 ~ULConnectionCallback()	1232
10.348.3 Member Function Documentation	1233
10.348.3.1 DataSetHandled()	1233
10.348.3.2 DataSetHandles()	1233
10.348.3.3 HandleDataSet()	1233
10.348.3.4 HandleResponse()	1233
10.348.3.5 ResetHandledDataSet()	1233
10.348.3.6 SetImplicitFlag()	1233
10.348.4 Member Data Documentation	1234
10.348.4.1 mImplicit	1234
10.349 gdcmm::network::ULConnectionInfo Class Reference	1234
10.349.1 Detailed Description	1234
10.349.2 Constructor & Destructor Documentation	1234
10.349.2.1 ULConnectionInfo()	1235
10.349.3 Member Function Documentation	1235
10.349.3.1 GetCalledAETitle()	1235
10.349.3.2 GetCalledComputerName()	1235
10.349.3.3 GetCalledIPAddress()	1235
10.349.3.4 GetCalledIPPort()	1235
10.349.3.5 GetCallingAETitle()	1235
10.349.3.6 GetMaxPDULength()	1235
10.349.3.7 Initialize()	1236
10.349.3.8 SetMaxPDULength()	1236
10.350 gdcmm::network::ULConnectionManager Class Reference	1236
10.350.1 Detailed Description	1238
10.350.2 Constructor & Destructor Documentation	1238

10.350.2.1 ULConnectionManager() [1/2]	1238
10.350.2.2 ULConnectionManager() [2/2]	1238
10.350.2.3 ~ULConnectionManager()	1238
10.350.3 Member Function Documentation	1239
10.350.3.1 BreakConnection()	1239
10.350.3.2 BreakConnectionNow()	1239
10.350.3.3 EstablishConnection()	1239
10.350.3.4 EstablishConnectionMove()	1239
10.350.3.5 RunEventLoop()	1240
10.350.3.6 RunMoveEventLoop()	1240
10.350.3.7 SendEcho()	1240
10.350.3.8 SendFind() [1/2]	1240
10.350.3.9 SendFind() [2/2]	1240
10.350.3.10 SendMove() [1/2]	1240
10.350.3.11 SendMove() [2/2]	1241
10.350.3.12 SendNAction() [1/2]	1241
10.350.3.13 SendNAction() [2/2]	1241
10.350.3.14 SendNCreate() [1/2]	1241
10.350.3.15 SendNCreate() [2/2]	1241
10.350.3.16 SendNDelete() [1/2]	1241
10.350.3.17 SendNDelete() [2/2]	1242
10.350.3.18 SendNEventReport() [1/2]	1242
10.350.3.19 SendNEventReport() [2/2]	1242
10.350.3.20 SendNGet() [1/2]	1242
10.350.3.21 SendNGet() [2/2]	1242
10.350.3.22 SendNSet() [1/2]	1242
10.350.3.23 SendNSet() [2/2]	1243
10.350.3.24 SendStore() [1/2]	1243
10.350.3.25 SendStore() [2/2]	1243
10.350.4 Member Data Documentation	1243
10.350.4.1 mConnection	1243
10.350.4.2 mSecondaryConnection	1243
10.350.4.3 mTransitions	1244
10.351 gdcmm::network::ULEvent Class Reference	1244
10.351.1 Detailed Description	1244
10.351.2 Constructor & Destructor Documentation	1244
10.351.2.1 ULEvent() [1/2]	1245
10.351.2.2 ULEvent() [2/2]	1245
10.351.2.3 ~ULEvent()	1245



10.351.3 Member Function Documentation . . . . .	1245
10.351.3.1 GetDataSetPos() . . . . .	1245
10.351.3.2 GetEvent() . . . . .	1245
10.351.3.3 GetIStream() . . . . .	1245
10.351.3.4 GetPDUs() . . . . .	1246
10.351.3.5 SetEvent() . . . . .	1246
10.351.3.6 SetPDU() . . . . .	1246
10.352 gdcmm::network::ULTransitionTable Class Reference . . . . .	1246
10.352.1 Detailed Description . . . . .	1246
10.352.2 Constructor & Destructor Documentation . . . . .	1247
10.352.2.1 ULTransitionTable() . . . . .	1247
10.352.3 Member Function Documentation . . . . .	1247
10.352.3.1 HandleEvent() . . . . .	1247
10.352.3.2 PrintTable() . . . . .	1247
10.353 gdcmm::network::ULWritingCallback Class Reference . . . . .	1248
10.353.1 Constructor & Destructor Documentation . . . . .	1249
10.353.1.1 ULWritingCallback() . . . . .	1249
10.353.1.2 ~ULWritingCallback() . . . . .	1249
10.353.2 Member Function Documentation . . . . .	1249
10.353.2.1 HandleDataSet() . . . . .	1249
10.353.2.2 HandleResponse() . . . . .	1249
10.353.2.3 SetDirectory() . . . . .	1250
10.354 gdcmm::UNExplicitDataElement Class Reference . . . . .	1250
10.354.1 Detailed Description . . . . .	1251
10.354.2 Member Function Documentation . . . . .	1251
10.354.2.1 GetLength() . . . . .	1252
10.354.2.2 Read() . . . . .	1252
10.354.2.3 ReadPreValue() . . . . .	1252
10.354.2.4 ReadValue() . . . . .	1252
10.354.2.5 ReadWithLength() . . . . .	1252
10.355 gdcmm::UNExplicitImplicitDataElement Class Reference . . . . .	1253
10.355.1 Detailed Description . . . . .	1254
10.355.2 Member Function Documentation . . . . .	1254
10.355.2.1 GetLength() . . . . .	1254
10.355.2.2 Read() . . . . .	1254
10.355.2.3 ReadPreValue() . . . . .	1254
10.355.2.4 ReadValue() . . . . .	1255
10.356 gdcmm::Unpacker12Bits Class Reference . . . . .	1255
10.356.1 Detailed Description . . . . .	1255

10.356.2 Member Function Documentation . . . . .	1255
10.356.2.1 Pack() . . . . .	1256
10.356.2.2 Unpack() . . . . .	1256
10.357 gdcm::Usage Class Reference . . . . .	1256
10.357.1 Detailed Description . . . . .	1257
10.357.2 Member Enumeration Documentation . . . . .	1257
10.357.2.1 UsageType . . . . .	1257
10.357.3 Constructor & Destructor Documentation . . . . .	1258
10.357.3.1 Usage() . . . . .	1258
10.357.4 Member Function Documentation . . . . .	1258
10.357.4.1 GetUsageString() . . . . .	1258
10.357.4.2 GetUsageType() . . . . .	1258
10.357.4.3 operator UsageType() . . . . .	1258
10.357.5 Friends And Related Function Documentation . . . . .	1258
10.357.5.1 operator<< . . . . .	1258
10.358 gdcm::UserEvent Class Reference . . . . .	1259
10.359 gdcm::network::UserInformation Class Reference . . . . .	1260
10.359.1 Detailed Description . . . . .	1260
10.359.2 Constructor & Destructor Documentation . . . . .	1260
10.359.2.1 UserInformation() [1/2] . . . . .	1260
10.359.2.2 ~UserInformation() . . . . .	1260
10.359.2.3 UserInformation() [2/2] . . . . .	1261
10.359.3 Member Function Documentation . . . . .	1261
10.359.3.1 AddRoleSelectionSub() . . . . .	1261
10.359.3.2 AddSOPClassExtendedNegociationSub() . . . . .	1261
10.359.3.3 GetMaximumLengthSub() [1/2] . . . . .	1261
10.359.3.4 GetMaximumLengthSub() [2/2] . . . . .	1261
10.359.3.5 operator=() . . . . .	1261
10.359.3.6 Print() . . . . .	1262
10.359.3.7 Read() . . . . .	1262
10.359.3.8 Size() . . . . .	1262
10.359.3.9 Write() . . . . .	1262
10.360 gdcm::UUIDGenerator Class Reference . . . . .	1262
10.360.1 Detailed Description . . . . .	1263
10.360.2 Member Function Documentation . . . . .	1263
10.360.2.1 Generate() . . . . .	1263
10.360.2.2 IsValid() . . . . .	1263
10.361 gdcm::Validate Class Reference . . . . .	1263
10.361.1 Detailed Description . . . . .	1264

10.361.2 Constructor & Destructor Documentation	1264
10.361.2.1 Validate()	1264
10.361.2.2 ~Validate()	1264
10.361.3 Member Function Documentation	1264
10.361.3.1 GetValidatedFile()	1264
10.361.3.2 SetFile()	1265
10.361.3.3 Validation()	1265
10.361.4 Member Data Documentation	1265
10.361.4.1 F	1265
10.361.4.2 V	1265
10.362 gdcM::Value Class Reference	1266
10.362.1 Detailed Description	1267
10.362.2 Constructor & Destructor Documentation	1267
10.362.2.1 Value()	1267
10.362.2.2 ~Value()	1267
10.362.3 Member Function Documentation	1267
10.362.3.1 Clear()	1267
10.362.3.2 GetLength()	1268
10.362.3.3 operator==()	1268
10.362.3.4 SetLength()	1268
10.362.3.5 SetLengthOnly()	1268
10.362.4 Friends And Related Function Documentation	1268
10.362.4.1 DataElement	1268
10.363 gdcM::ValueIO< TDE, TSwap, TType > Class Template Reference	1269
10.363.1 Detailed Description	1269
10.363.2 Member Function Documentation	1269
10.363.2.1 Read()	1269
10.363.2.2 Write()	1269
10.364 gdcM::MrProtocol::Vector3 Struct Reference	1270
10.364.1 Member Data Documentation	1270
10.364.1.1 dCor	1270
10.364.1.2 dSag	1270
10.364.1.3 dTra	1270
10.365 gdcM::Version Class Reference	1270
10.365.1 Detailed Description	1271
10.365.2 Constructor & Destructor Documentation	1271
10.365.2.1 Version()	1271
10.365.2.2 ~Version()	1271
10.365.3 Member Function Documentation	1271

10.365.3.1 GetBuildVersion()	1272
10.365.3.2 GetMajorVersion()	1272
10.365.3.3 GetMinorVersion()	1272
10.365.3.4 GetVersion()	1272
10.365.3.5 Print()	1272
10.365.4 Friends And Related Function Documentation	1272
10.365.4.1 operator<<	1272
10.366 gdcM::VL Class Reference	1273
10.366.1 Detailed Description	1274
10.366.2 Member Typedef Documentation	1274
10.366.2.1 Type	1274
10.366.3 Constructor & Destructor Documentation	1274
10.366.3.1 VL()	1274
10.366.4 Member Function Documentation	1274
10.366.4.1 GetLength()	1274
10.366.4.2 GetVL16Max()	1275
10.366.4.3 GetVL32Max()	1275
10.366.4.4 IsOdd()	1275
10.366.4.5 IsUndefined()	1275
10.366.4.6 operator uint32_t()	1275
10.366.4.7 operator++() [1/2]	1275
10.366.4.8 operator++() [2/2]	1275
10.366.4.9 operator+=( )	1276
10.366.4.10 Read()	1276
10.366.4.11 Read16()	1276
10.366.4.12 SetToUndefined()	1276
10.366.4.13 Write()	1276
10.366.4.14 Write16()	1276
10.366.5 Friends And Related Function Documentation	1277
10.366.5.1 operator<<	1277
10.367 gdcM::VM Class Reference	1277
10.367.1 Detailed Description	1278
10.367.2 Member Enumeration Documentation	1279
10.367.2.1 VMType	1279
10.367.3 Constructor & Destructor Documentation	1280
10.367.3.1 VM()	1280
10.367.4 Member Function Documentation	1280
10.367.4.1 Compatible()	1280
10.367.4.2 GetIndex()	1280

10.367.4.3 GetLength()	1280
10.367.4.4 GetNumberOfElementsFromArray()	1280
10.367.4.5 GetVMString()	1281
10.367.4.6 GetVMType()	1281
10.367.4.7 GetVMTypeFromLength()	1281
10.367.4.8 IsValid()	1281
10.367.4.9 operator VMType()	1281
10.367.5 Friends And Related Function Documentation	1281
10.367.5.1 operator<<	1281
10.368 gdcmm::VMToLength< T > Struct Template Reference	1282
10.369 gdcmm::VR Class Reference	1282
10.369.1 Detailed Description	1284
10.369.2 Member Enumeration Documentation	1284
10.369.2.1 VRType	1284
10.369.3 Constructor & Destructor Documentation	1285
10.369.3.1 VR()	1285
10.369.4 Member Function Documentation	1285
10.369.4.1 CanDisplay()	1286
10.369.4.2 Compatible()	1286
10.369.4.3 GetLength() [1/2]	1286
10.369.4.4 GetLength() [2/2]	1286
10.369.4.5 GetSize()	1286
10.369.4.6 GetSizeof()	1286
10.369.4.7 GetVRString()	1287
10.369.4.8 GetVRStringFromFile()	1287
10.369.4.9 GetVRType()	1287
10.369.4.10 GetVRTypeFromFile()	1287
10.369.4.11 IsASCII()	1287
10.369.4.12 IsASCII2()	1287
10.369.4.13 IsBinary()	1288
10.369.4.14 IsBinary2()	1288
10.369.4.15 IsDual()	1288
10.369.4.16 IsSwap()	1288
10.369.4.17 IsValid() [1/2]	1288
10.369.4.18 IsValid() [2/2]	1288
10.369.4.19 IsVRFile()	1289
10.369.4.20 operator VRType()	1289
10.369.4.21 Read()	1289
10.369.4.22 Write()	1289

10.369.5 Friends And Related Function Documentation	1289
10.369.5.1 operator<<	1289
10.370 gdcm::VR16ExplicitDataElement Class Reference	1290
10.370.1 Detailed Description	1291
10.370.2 Member Function Documentation	1291
10.370.2.1 GetLength()	1291
10.370.2.2 Read()	1291
10.370.2.3 ReadPreValue()	1291
10.370.2.4 ReadValue()	1292
10.370.2.5 ReadWithLength()	1292
10.371 gdcm::VRToEncoding< T > Struct Template Reference	1292
10.372 gdcm::VRToType< T > Struct Template Reference	1292
10.373 gdcm::VRVLSize< T > Class Template Reference	1293
10.374 gdcm::VRVLSize< 0 > Class Reference	1293
10.374.1 Member Function Documentation	1293
10.374.1.1 Read()	1293
10.374.1.2 Write()	1293
10.375 gdcm::VRVLSize< 1 > Class Reference	1293
10.375.1 Member Function Documentation	1294
10.375.1.1 Read()	1294
10.375.1.2 Write()	1294
10.376 vtkGDCMImageReader Class Reference	1294
10.376.1 Detailed Description	1297
10.376.2 Constructor & Destructor Documentation	1297
10.376.2.1 vtkGDCMImageReader()	1297
10.376.2.2 ~vtkGDCMImageReader()	1297
10.376.3 Member Function Documentation	1297
10.376.3.1 CanReadFile()	1297
10.376.3.2 ExecuteData()	1298
10.376.3.3 ExecuteInformation()	1298
10.376.3.4 FillMedicalImageInformation()	1298
10.376.3.5 GetDescriptiveName()	1298
10.376.3.6 GetFileExtensions()	1298
10.376.3.7 GetIconImage()	1298
10.376.3.8 GetOverlay()	1298
10.376.3.9 LoadSingleFile()	1299
10.376.3.10 New()	1299
10.376.3.11 PrintSelf()	1299
10.376.3.12 RequestDataCompat()	1299

---

10.376.3.13 RequestInformationCompat()	1299
10.376.3.14 SetCurve()	1300
10.376.3.15 SetFileNames()	1300
10.376.3.16 SetFilePattern()	1300
10.376.3.17 SetFilePrefix()	1300
10.376.3.18 SetMedicalImageProperties()	1300
10.376.3.19 vtkBooleanMacro() [1/5]	1300
10.376.3.20 vtkBooleanMacro() [2/5]	1301
10.376.3.21 vtkBooleanMacro() [3/5]	1301
10.376.3.22 vtkBooleanMacro() [4/5]	1301
10.376.3.23 vtkBooleanMacro() [5/5]	1301
10.376.3.24 vtkGetMacro() [1/11]	1301
10.376.3.25 vtkGetMacro() [2/11]	1301
10.376.3.26 vtkGetMacro() [3/11]	1302
10.376.3.27 vtkGetMacro() [4/11]	1302
10.376.3.28 vtkGetMacro() [5/11]	1302
10.376.3.29 vtkGetMacro() [6/11]	1302
10.376.3.30 vtkGetMacro() [7/11]	1302
10.376.3.31 vtkGetMacro() [8/11]	1302
10.376.3.32 vtkGetMacro() [9/11]	1303
10.376.3.33 vtkGetMacro() [10/11]	1303
10.376.3.34 vtkGetMacro() [11/11]	1303
10.376.3.35 vtkGetObjectMacro() [1/4]	1303
10.376.3.36 vtkGetObjectMacro() [2/4]	1303
10.376.3.37 vtkGetObjectMacro() [3/4]	1303
10.376.3.38 vtkGetObjectMacro() [4/4]	1304
10.376.3.39 vtkGetStringMacro() [1/2]	1304
10.376.3.40 vtkGetStringMacro() [2/2]	1304
10.376.3.41 vtkGetVector3Macro()	1304
10.376.3.42 vtkGetVector6Macro()	1304
10.376.3.43 vtkSetMacro() [1/4]	1304
10.376.3.44 vtkSetMacro() [2/4]	1305
10.376.3.45 vtkSetMacro() [3/4]	1305
10.376.3.46 vtkSetMacro() [4/4]	1305
10.376.3.47 vtkSetVector6Macro()	1305
10.376.3.48 vtkTypeMacro()	1305
10.376.4 Member Data Documentation	1305
10.376.4.1 ApplyInverseVideo	1306
10.376.4.2 ApplyLookupTable	1306

10.376.4.3 ApplyPlanarConfiguration . . . . .	1306
10.376.4.4 ApplyShiftScale . . . . .	1306
10.376.4.5 ApplyYBRToRGB . . . . .	1306
10.376.4.6 Curve . . . . .	1306
10.376.4.7 DirectionCosines . . . . .	1306
10.376.4.8 FileNames . . . . .	1307
10.376.4.9 ForceRescale . . . . .	1307
10.376.4.10 IconDataScalarType . . . . .	1307
10.376.4.11 IconImageDataExtent . . . . .	1307
10.376.4.12 IconNumberOfScalarComponents . . . . .	1307
10.376.4.13 ImageFormat . . . . .	1307
10.376.4.14 ImageOrientationPatient . . . . .	1307
10.376.4.15 ImagePositionPatient . . . . .	1308
10.376.4.16 LoadIconImage . . . . .	1308
10.376.4.17 LoadOverlays . . . . .	1308
10.376.4.18 LossyFlag . . . . .	1308
10.376.4.19 MedicalImageProperties . . . . .	1308
10.376.4.20 NumberOfIconImages . . . . .	1308
10.376.4.21 NumberOfOverlays . . . . .	1308
10.376.4.22 PlanarConfiguration . . . . .	1309
10.376.4.23 Scale . . . . .	1309
10.376.4.24 Shift . . . . .	1309
10.377 vtkGDCMImageReader2 Class Reference . . . . .	1309
10.377.1 Detailed Description . . . . .	1312
10.377.2 Constructor & Destructor Documentation . . . . .	1312
10.377.2.1 vtkGDCMImageReader2() . . . . .	1312
10.377.2.2 ~vtkGDCMImageReader2() . . . . .	1312
10.377.3 Member Function Documentation . . . . .	1312
10.377.3.1 CanReadFile() . . . . .	1312
10.377.3.2 FillMedicalImageInformation() . . . . .	1312
10.377.3.3 GetDescriptiveName() . . . . .	1312
10.377.3.4 GetFileExtensions() . . . . .	1313
10.377.3.5 GetIconImage() . . . . .	1313
10.377.3.6 GetIconImagePort() . . . . .	1313
10.377.3.7 GetOverlay() . . . . .	1313
10.377.3.8 GetOverlayPort() . . . . .	1313
10.377.3.9 LoadSingleFile() . . . . .	1313
10.377.3.10 New() . . . . .	1314
10.377.3.11 PrintSelf() . . . . .	1314



10.377.3.12 ProcessRequest()	1314
10.377.3.13 RequestData()	1314
10.377.3.14 RequestDataCompat()	1314
10.377.3.15 RequestInformation()	1315
10.377.3.16 RequestInformationCompat()	1315
10.377.3.17 SetCurve()	1315
10.377.3.18 SetFilePattern()	1315
10.377.3.19 SetFilePrefix()	1315
10.377.3.20 SetMedicalImageProperties()	1315
10.377.3.21 vtkBooleanMacro() [1/5]	1316
10.377.3.22 vtkBooleanMacro() [2/5]	1316
10.377.3.23 vtkBooleanMacro() [3/5]	1316
10.377.3.24 vtkBooleanMacro() [4/5]	1316
10.377.3.25 vtkBooleanMacro() [5/5]	1316
10.377.3.26 vtkGetMacro() [1/11]	1316
10.377.3.27 vtkGetMacro() [2/11]	1317
10.377.3.28 vtkGetMacro() [3/11]	1317
10.377.3.29 vtkGetMacro() [4/11]	1317
10.377.3.30 vtkGetMacro() [5/11]	1317
10.377.3.31 vtkGetMacro() [6/11]	1317
10.377.3.32 vtkGetMacro() [7/11]	1317
10.377.3.33 vtkGetMacro() [8/11]	1318
10.377.3.34 vtkGetMacro() [9/11]	1318
10.377.3.35 vtkGetMacro() [10/11]	1318
10.377.3.36 vtkGetMacro() [11/11]	1318
10.377.3.37 vtkGetObjectMacro() [1/2]	1318
10.377.3.38 vtkGetObjectMacro() [2/2]	1318
10.377.3.39 vtkGetStringMacro() [1/2]	1319
10.377.3.40 vtkGetStringMacro() [2/2]	1319
10.377.3.41 vtkGetVector3Macro()	1319
10.377.3.42 vtkGetVector6Macro()	1319
10.377.3.43 vtkSetMacro() [1/4]	1319
10.377.3.44 vtkSetMacro() [2/4]	1319
10.377.3.45 vtkSetMacro() [3/4]	1320
10.377.3.46 vtkSetMacro() [4/4]	1320
10.377.3.47 vtkSetVector6Macro()	1320
10.377.3.48 vtkTypeMacro()	1320
10.377.4 Member Data Documentation	1320
10.377.4.1 ApplyInverseVideo	1320

10.377.4.2 ApplyLookupTable . . . . .	1320
10.377.4.3 ApplyPlanarConfiguration . . . . .	1321
10.377.4.4 ApplyShiftScale . . . . .	1321
10.377.4.5 ApplyYBRToRGB . . . . .	1321
10.377.4.6 Curve . . . . .	1321
10.377.4.7 DirectionCosines . . . . .	1321
10.377.4.8 ForceRescale . . . . .	1321
10.377.4.9 IconDataScalarType . . . . .	1321
10.377.4.10 IconImageDataExtent . . . . .	1322
10.377.4.11 IconNumberOfScalarComponents . . . . .	1322
10.377.4.12 ImageFormat . . . . .	1322
10.377.4.13 ImageOrientationPatient . . . . .	1322
10.377.4.14 ImagePositionPatient . . . . .	1322
10.377.4.15 LoadIconImage . . . . .	1322
10.377.4.16 LoadOverlays . . . . .	1322
10.377.4.17 LossyFlag . . . . .	1323
10.377.4.18 NumberOfIconImages . . . . .	1323
10.377.4.19 NumberOfOverlays . . . . .	1323
10.377.4.20 PlanarConfiguration . . . . .	1323
10.377.4.21 Scale . . . . .	1323
10.377.4.22 Shift . . . . .	1323
10.378 vtkGDCMImageWriter Class Reference . . . . .	1324
10.378.1 Detailed Description . . . . .	1326
10.378.2 Member Enumeration Documentation . . . . .	1326
10.378.2.1 CompressionTypes . . . . .	1326
10.378.3 Constructor & Destructor Documentation . . . . .	1326
10.378.3.1 vtkGDCMImageWriter() . . . . .	1326
10.378.3.2 ~vtkGDCMImageWriter() . . . . .	1326
10.378.4 Member Function Documentation . . . . .	1326
10.378.4.1 GetDescriptiveName() . . . . .	1327
10.378.4.2 GetFileExtensions() . . . . .	1327
10.378.4.3 GetFileName() . . . . .	1327
10.378.4.4 New() . . . . .	1327
10.378.4.5 PrintSelf() . . . . .	1327
10.378.4.6 SetDirectionCosines() . . . . .	1327
10.378.4.7 SetDirectionCosinesFromImageOrientationPatient() . . . . .	1328
10.378.4.8 SetFileNames() . . . . .	1328
10.378.4.9 SetMedicalImageProperties() . . . . .	1328
10.378.4.10 vtkBooleanMacro() [1/2] . . . . .	1328

10.378.4.11	<a href="#">vtkBooleanMacro()</a> [2/2]	1328
10.378.4.12	<a href="#">vtkGetMacro()</a> [1/7]	1329
10.378.4.13	<a href="#">vtkGetMacro()</a> [2/7]	1329
10.378.4.14	<a href="#">vtkGetMacro()</a> [3/7]	1329
10.378.4.15	<a href="#">vtkGetMacro()</a> [4/7]	1329
10.378.4.16	<a href="#">vtkGetMacro()</a> [5/7]	1329
10.378.4.17	<a href="#">vtkGetMacro()</a> [6/7]	1329
10.378.4.18	<a href="#">vtkGetMacro()</a> [7/7]	1330
10.378.4.19	<a href="#">vtkGetObjectMacro()</a> [1/3]	1330
10.378.4.20	<a href="#">vtkGetObjectMacro()</a> [2/3]	1330
10.378.4.21	<a href="#">vtkGetObjectMacro()</a> [3/3]	1330
10.378.4.22	<a href="#">vtkGetStringMacro()</a> [1/2]	1330
10.378.4.23	<a href="#">vtkGetStringMacro()</a> [2/2]	1330
10.378.4.24	<a href="#">vtkSetMacro()</a> [1/7]	1331
10.378.4.25	<a href="#">vtkSetMacro()</a> [2/7]	1331
10.378.4.26	<a href="#">vtkSetMacro()</a> [3/7]	1331
10.378.4.27	<a href="#">vtkSetMacro()</a> [4/7]	1331
10.378.4.28	<a href="#">vtkSetMacro()</a> [5/7]	1331
10.378.4.29	<a href="#">vtkSetMacro()</a> [6/7]	1331
10.378.4.30	<a href="#">vtkSetMacro()</a> [7/7]	1332
10.378.4.31	<a href="#">vtkSetStringMacro()</a> [1/2]	1332
10.378.4.32	<a href="#">vtkSetStringMacro()</a> [2/2]	1332
10.378.4.33	<a href="#">vtkTypeMacro()</a>	1332
10.378.4.34	<a href="#">Write()</a>	1332
10.378.4.35	<a href="#">WriteGDCMData()</a>	1333
10.378.4.36	<a href="#">WriteSlice()</a>	1333
10.379	<a href="#">vtkGDCMMedicalImageProperties Class Reference</a>	1333
10.379.1	<a href="#">Constructor &amp; Destructor Documentation</a>	1334
10.379.1.1	<a href="#">vtkGDCMMedicalImageProperties()</a>	1334
10.379.1.2	<a href="#">~vtkGDCMMedicalImageProperties()</a>	1334
10.379.2	<a href="#">Member Function Documentation</a>	1334
10.379.2.1	<a href="#">Clear()</a>	1335
10.379.2.2	<a href="#">GetFile()</a>	1335
10.379.2.3	<a href="#">New()</a>	1335
10.379.2.4	<a href="#">PrintSelf()</a>	1335
10.379.2.5	<a href="#">PushBackFile()</a>	1335
10.379.2.6	<a href="#">vtkTypeMacro()</a>	1335
10.379.3	<a href="#">Friends And Related Function Documentation</a>	1335
10.379.3.1	<a href="#">vtkGDCMImageReader</a>	1336

10.379.3.2 vtkGDCMImageReader2 . . . . .	1336
10.379.3.3 vtkGDCMImageWriter . . . . .	1336
10.380 vtkGDCMPolyDataReader Class Reference . . . . .	1336
10.380.1 Detailed Description . . . . .	1338
10.380.2 Constructor & Destructor Documentation . . . . .	1338
10.380.2.1 vtkGDCMPolyDataReader() . . . . .	1338
10.380.2.2 ~vtkGDCMPolyDataReader() . . . . .	1338
10.380.3 Member Function Documentation . . . . .	1338
10.380.3.1 FillMedicalImageInformation() . . . . .	1338
10.380.3.2 New() . . . . .	1338
10.380.3.3 PrintSelf() . . . . .	1339
10.380.3.4 RequestData() . . . . .	1339
10.380.3.5 RequestData_HemodynamicWaveformStorage() . . . . .	1339
10.380.3.6 RequestData_RTStructureSetStorage() . . . . .	1339
10.380.3.7 RequestInformation() . . . . .	1339
10.380.3.8 RequestInformation_HemodynamicWaveformStorage() . . . . .	1339
10.380.3.9 RequestInformation_RTStructureSetStorage() . . . . .	1340
10.380.3.10 vtkGetObjectMacro() [1/2] . . . . .	1340
10.380.3.11 vtkGetObjectMacro() [2/2] . . . . .	1340
10.380.3.12 vtkGetStringMacro() . . . . .	1340
10.380.3.13 vtkSetStringMacro() . . . . .	1340
10.380.3.14 vtkTypeMacro() . . . . .	1340
10.380.4 Member Data Documentation . . . . .	1341
10.380.4.1 FileName . . . . .	1341
10.380.4.2 MedicalImageProperties . . . . .	1341
10.380.4.3 RTStructSetProperties . . . . .	1341
10.381 vtkGDCMPolyDataWriter Class Reference . . . . .	1341
10.381.1 Detailed Description . . . . .	1343
10.381.2 Constructor & Destructor Documentation . . . . .	1343
10.381.2.1 vtkGDCMPolyDataWriter() . . . . .	1343
10.381.2.2 ~vtkGDCMPolyDataWriter() . . . . .	1343
10.381.3 Member Function Documentation . . . . .	1343
10.381.3.1 InitializeRTStructSet() . . . . .	1343
10.381.3.2 New() . . . . .	1344
10.381.3.3 PrintSelf() . . . . .	1344
10.381.3.4 SetMedicalImageProperties() . . . . .	1344
10.381.3.5 SetNumberOfInputPorts() . . . . .	1344
10.381.3.6 SetRTStructSetProperties() . . . . .	1345
10.381.3.7 vtkTypeMacro() . . . . .	1345

10.381.3.8 WriteData()	1345
10.381.3.9 WriteRTSTRUCTData()	1345
10.381.3.10 WriteRTSTRUCTInfo()	1345
10.381.4 Member Data Documentation	1345
10.381.4.1 MedicalImageProperties	1346
10.381.4.2 RTStructSetProperties	1346
10.382 vtkGDCMTesting Class Reference	1346
10.382.1 Detailed Description	1347
10.382.2 Member Typedef Documentation	1347
10.382.2.1 MD5MetaImagesType	1347
10.382.3 Constructor & Destructor Documentation	1347
10.382.3.1 vtkGDCMTesting()	1348
10.382.3.2 ~vtkGDCMTesting()	1348
10.382.4 Member Function Documentation	1348
10.382.4.1 GetGDCMDataRoot()	1348
10.382.4.2 GetMD5MetaImage()	1348
10.382.4.3 GetMHDMD5FromFile()	1348
10.382.4.4 GetNumberOfMD5MetaImages()	1349
10.382.4.5 GetRAWMD5FromFile()	1349
10.382.4.6 GetVTKDataRoot()	1349
10.382.4.7 New()	1349
10.382.4.8 PrintSelf()	1349
10.382.4.9 vtkTypeMacro()	1350
10.383 vtkGDCMThreadedImageReader Class Reference	1350
10.383.1 Constructor & Destructor Documentation	1351
10.383.1.1 vtkGDCMThreadedImageReader()	1352
10.383.1.2 ~vtkGDCMThreadedImageReader()	1352
10.383.2 Member Function Documentation	1352
10.383.2.1 ExecuteData()	1352
10.383.2.2 ExecuteInformation()	1352
10.383.2.3 New()	1352
10.383.2.4 PrintSelf()	1352
10.383.2.5 ReadFiles()	1353
10.383.2.6 RequestDataCompat()	1353
10.383.2.7 vtkBooleanMacro()	1353
10.383.2.8 vtkGetMacro()	1353
10.383.2.9 vtkSetMacro() [1/3]	1353
10.383.2.10 vtkSetMacro() [2/3]	1353
10.383.2.11 vtkSetMacro() [3/3]	1354

---

10.383.2.12 vtkTypeMacro()	1354
10.384 vtkGDCMThreadedImageReader2 Class Reference	1354
10.384.1 Constructor & Destructor Documentation	1356
10.384.1.1 vtkGDCMThreadedImageReader2()	1356
10.384.1.2 ~vtkGDCMThreadedImageReader2()	1356
10.384.2 Member Function Documentation	1356
10.384.2.1 GetFileName()	1356
10.384.2.2 New()	1356
10.384.2.3 PrintSelf()	1357
10.384.2.4 RequestInformation()	1357
10.384.2.5 SetFileName()	1357
10.384.2.6 SetFileNames()	1357
10.384.2.7 SplitExtent()	1357
10.384.2.8 ThreadedRequestData()	1358
10.384.2.9 vtkBooleanMacro() [1/3]	1358
10.384.2.10 vtkBooleanMacro() [2/3]	1358
10.384.2.11 vtkBooleanMacro() [3/3]	1358
10.384.2.12 vtkGetMacro() [1/8]	1358
10.384.2.13 vtkGetMacro() [2/8]	1359
10.384.2.14 vtkGetMacro() [3/8]	1359
10.384.2.15 vtkGetMacro() [4/8]	1359
10.384.2.16 vtkGetMacro() [5/8]	1359
10.384.2.17 vtkGetMacro() [6/8]	1359
10.384.2.18 vtkGetMacro() [7/8]	1359
10.384.2.19 vtkGetMacro() [8/8]	1360
10.384.2.20 vtkGetObjectMacro()	1360
10.384.2.21 vtkGetVector3Macro() [1/2]	1360
10.384.2.22 vtkGetVector3Macro() [2/2]	1360
10.384.2.23 vtkGetVector6Macro()	1360
10.384.2.24 vtkSetMacro() [1/7]	1360
10.384.2.25 vtkSetMacro() [2/7]	1361
10.384.2.26 vtkSetMacro() [3/7]	1361
10.384.2.27 vtkSetMacro() [4/7]	1361
10.384.2.28 vtkSetMacro() [5/7]	1361
10.384.2.29 vtkSetMacro() [6/7]	1361
10.384.2.30 vtkSetMacro() [7/7]	1361
10.384.2.31 vtkSetVector3Macro() [1/2]	1362
10.384.2.32 vtkSetVector3Macro() [2/2]	1362
10.384.2.33 vtkSetVector6Macro()	1362

---

10.384.2.34 vtkTypeMacro()	1362
10.385 vtkImageColorViewer Class Reference	1363
10.385.1 Detailed Description	1365
10.385.2 Member Enumeration Documentation	1365
10.385.2.1 anonymous enum	1365
10.385.3 Constructor & Destructor Documentation	1366
10.385.3.1 vtkImageColorViewer()	1366
10.385.3.2 ~vtkImageColorViewer()	1366
10.385.4 Member Function Documentation	1366
10.385.4.1 AddInput()	1366
10.385.4.2 AddInputConnection()	1366
10.385.4.3 GetColorLevel()	1367
10.385.4.4 GetColorWindow()	1367
10.385.4.5 GetInput()	1367
10.385.4.6 GetOffScreenRendering()	1367
10.385.4.7 GetOverlayVisibility()	1367
10.385.4.8 GetPosition()	1367
10.385.4.9 GetSize()	1367
10.385.4.10 GetSliceMax()	1368
10.385.4.11 GetSliceMin()	1368
10.385.4.12 GetSliceRange() [1/3]	1368
10.385.4.13 GetSliceRange() [2/3]	1368
10.385.4.14 GetSliceRange() [3/3]	1368
10.385.4.15 GetWindowName()	1368
10.385.4.16 InstallPipeline()	1368
10.385.4.17 New()	1369
10.385.4.18 PrintSelf()	1369
10.385.4.19 Render()	1369
10.385.4.20 SetColorLevel()	1369
10.385.4.21 SetColorWindow()	1369
10.385.4.22 SetDisplayId()	1370
10.385.4.23 SetInput()	1370
10.385.4.24 SetInputConnection()	1370
10.385.4.25 SetOffScreenRendering()	1370
10.385.4.26 SetOverlayVisibility()	1370
10.385.4.27 SetParentId()	1370
10.385.4.28 SetPosition() [1/2]	1371
10.385.4.29 SetPosition() [2/2]	1371
10.385.4.30 SetRenderer()	1371

10.385.4.31 SetRenderWindow()	1371
10.385.4.32 SetSize() [1/2]	1371
10.385.4.33 SetSize() [2/2]	1372
10.385.4.34 SetSlice()	1372
10.385.4.35 SetSliceOrientation()	1372
10.385.4.36 SetSliceOrientationToXY()	1372
10.385.4.37 SetSliceOrientationToXZ()	1372
10.385.4.38 SetSliceOrientationToYZ()	1372
10.385.4.39 SetupInteractor()	1373
10.385.4.40 SetWindowId()	1373
10.385.4.41 UnInstallPipeline()	1373
10.385.4.42 UpdateDisplayExtent()	1373
10.385.4.43 UpdateOrientation()	1373
10.385.4.44 VTK_LEGACY() [1/4]	1373
10.385.4.45 VTK_LEGACY() [2/4]	1374
10.385.4.46 VTK_LEGACY() [3/4]	1374
10.385.4.47 VTK_LEGACY() [4/4]	1374
10.385.4.48 vtkBooleanMacro()	1374
10.385.4.49 vtkGetMacro() [1/2]	1374
10.385.4.50 vtkGetMacro() [2/2]	1374
10.385.4.51 vtkGetObjectMacro() [1/5]	1375
10.385.4.52 vtkGetObjectMacro() [2/5]	1375
10.385.4.53 vtkGetObjectMacro() [3/5]	1375
10.385.4.54 vtkGetObjectMacro() [4/5]	1375
10.385.4.55 vtkGetObjectMacro() [5/5]	1375
10.385.4.56 vtkTypeMacro()	1375
10.385.5 Friends And Related Function Documentation	1376
10.385.5.1 vtkImageColorViewerCallback	1376
10.385.6 Member Data Documentation	1376
10.385.6.1 FirstRender	1376
10.385.6.2 ImageActor	1376
10.385.6.3 Interactor	1376
10.385.6.4 InteractorStyle	1376
10.385.6.5 OverlayImageActor	1376
10.385.6.6 Renderer	1377
10.385.6.7 RenderWindow	1377
10.385.6.8 Slice	1377
10.385.6.9 SliceOrientation	1377
10.385.6.10 WindowLevel	1377



10.386 vtkImageMapToColors16 Class Reference . . . . .	1378
10.386.1 Constructor & Destructor Documentation . . . . .	1379
10.386.1.1 vtkImageMapToColors16() . . . . .	1379
10.386.1.2 ~vtkImageMapToColors16() . . . . .	1379
10.386.2 Member Function Documentation . . . . .	1380
10.386.2.1 GetMTime() . . . . .	1380
10.386.2.2 New() . . . . .	1380
10.386.2.3 PrintSelf() . . . . .	1380
10.386.2.4 RequestData() . . . . .	1380
10.386.2.5 RequestInformation() . . . . .	1380
10.386.2.6 SetLookupTable() . . . . .	1381
10.386.2.7 SetOutputFormatToLuminance() . . . . .	1381
10.386.2.8 SetOutputFormatToLuminanceAlpha() . . . . .	1381
10.386.2.9 SetOutputFormatToRGB() . . . . .	1381
10.386.2.10 SetOutputFormatToRGBA() . . . . .	1381
10.386.2.11 ThreadedRequestData() . . . . .	1381
10.386.2.12 vtkBooleanMacro() . . . . .	1382
10.386.2.13 vtkGetMacro() [1/3] . . . . .	1382
10.386.2.14 vtkGetMacro() [2/3] . . . . .	1382
10.386.2.15 vtkGetMacro() [3/3] . . . . .	1382
10.386.2.16 vtkGetObjectMacro() . . . . .	1382
10.386.2.17 vtkSetMacro() [1/3] . . . . .	1382
10.386.2.18 vtkSetMacro() [2/3] . . . . .	1383
10.386.2.19 vtkSetMacro() [3/3] . . . . .	1383
10.386.2.20 vtkTypeMacro() . . . . .	1383
10.386.3 Member Data Documentation . . . . .	1383
10.386.3.1 ActiveComponent . . . . .	1383
10.386.3.2 DataWasPassed . . . . .	1383
10.386.3.3 LookupTable . . . . .	1383
10.386.3.4 OutputFormat . . . . .	1384
10.386.3.5 PassAlphaToOutput . . . . .	1384
10.387 vtkImageMapToWindowLevelColors2 Class Reference . . . . .	1384
10.387.1 Constructor & Destructor Documentation . . . . .	1385
10.387.1.1 vtkImageMapToWindowLevelColors2() . . . . .	1385
10.387.1.2 ~vtkImageMapToWindowLevelColors2() . . . . .	1385
10.387.2 Member Function Documentation . . . . .	1386
10.387.2.1 New() . . . . .	1386
10.387.2.2 PrintSelf() . . . . .	1386
10.387.2.3 RequestData() . . . . .	1386

10.387.2.4 RequestInformation()	1386
10.387.2.5 ThreadedRequestData()	1386
10.387.2.6 vtkGetMacro() [1/2]	1387
10.387.2.7 vtkGetMacro() [2/2]	1387
10.387.2.8 vtkSetMacro() [1/2]	1387
10.387.2.9 vtkSetMacro() [2/2]	1387
10.387.2.10 vtkTypeMacro()	1387
10.387.3 Member Data Documentation	1387
10.387.3.1 Level	1388
10.387.3.2 Window	1388
10.388 vtkImagePlanarComponentsToComponents Class Reference	1388
10.388.1 Constructor & Destructor Documentation	1389
10.388.1.1 vtkImagePlanarComponentsToComponents()	1389
10.388.1.2 ~vtkImagePlanarComponentsToComponents()	1389
10.388.2 Member Function Documentation	1389
10.388.2.1 New()	1389
10.388.2.2 PrintSelf()	1390
10.388.2.3 RequestData()	1390
10.388.2.4 vtkTypeMacro()	1390
10.389 vtkImageRGBToYBR Class Reference	1390
10.389.1 Constructor & Destructor Documentation	1391
10.389.1.1 vtkImageRGBToYBR()	1391
10.389.1.2 ~vtkImageRGBToYBR()	1391
10.389.2 Member Function Documentation	1392
10.389.2.1 New()	1392
10.389.2.2 PrintSelf()	1392
10.389.2.3 ThreadedExecute()	1392
10.389.2.4 vtkTypeMacro()	1392
10.390 vtkImageYBRToRGB Class Reference	1393
10.390.1 Constructor & Destructor Documentation	1394
10.390.1.1 vtkImageYBRToRGB()	1394
10.390.1.2 ~vtkImageYBRToRGB()	1394
10.390.2 Member Function Documentation	1394
10.390.2.1 New()	1394
10.390.2.2 PrintSelf()	1394
10.390.2.3 ThreadedExecute()	1395
10.390.2.4 vtkTypeMacro()	1395
10.391 vtkLookupTable16 Class Reference	1395
10.391.1 Constructor & Destructor Documentation	1396

10.391.1.1	<a href="#">vtkLookupTable16()</a>	1397
10.391.1.2	<a href="#">~vtkLookupTable16()</a>	1397
10.391.2	<a href="#">Member Function Documentation</a>	1397
10.391.2.1	<a href="#">Build()</a>	1397
10.391.2.2	<a href="#">GetPointer()</a>	1397
10.391.2.3	<a href="#">MapScalarsThroughTable2()</a>	1397
10.391.2.4	<a href="#">New()</a>	1398
10.391.2.5	<a href="#">PrintSelf()</a>	1398
10.391.2.6	<a href="#">SetNumberOfTableValues()</a>	1398
10.391.2.7	<a href="#">vtkTypeMacro()</a>	1398
10.391.2.8	<a href="#">WritePointer()</a>	1398
10.391.3	<a href="#">Member Data Documentation</a>	1398
10.391.3.1	<a href="#">Table16</a>	1399
10.392	<a href="#">vtkRTStructSetProperties Class Reference</a>	1399
10.392.1	<a href="#">Detailed Description</a>	1401
10.392.2	<a href="#">Constructor &amp; Destructor Documentation</a>	1401
10.392.2.1	<a href="#">vtkRTStructSetProperties()</a>	1401
10.392.2.2	<a href="#">~vtkRTStructSetProperties()</a>	1401
10.392.3	<a href="#">Member Function Documentation</a>	1401
10.392.3.1	<a href="#">AddContourReferencedFrameOfReference()</a>	1402
10.392.3.2	<a href="#">AddReferencedFrameOfReference()</a>	1402
10.392.3.3	<a href="#">AddStructureSetROI()</a>	1402
10.392.3.4	<a href="#">AddStructureSetROIObservation()</a>	1402
10.392.3.5	<a href="#">Clear()</a>	1402
10.392.3.6	<a href="#">DeepCopy()</a>	1403
10.392.3.7	<a href="#">GetContourReferencedFrameOfReferenceClassUID()</a>	1403
10.392.3.8	<a href="#">GetContourReferencedFrameOfReferenceInstanceUID()</a>	1403
10.392.3.9	<a href="#">GetNumberOfContourReferencedFrameOfReferences()</a> [1/2]	1403
10.392.3.10	<a href="#">GetNumberOfContourReferencedFrameOfReferences()</a> [2/2]	1403
10.392.3.11	<a href="#">GetNumberOfReferencedFrameOfReferences()</a>	1403
10.392.3.12	<a href="#">GetNumberOfStructureSetROIs()</a>	1404
10.392.3.13	<a href="#">GetReferencedFrameOfReferenceClassUID()</a>	1404
10.392.3.14	<a href="#">GetReferencedFrameOfReferenceInstanceUID()</a>	1404
10.392.3.15	<a href="#">GetStructureSetObservationNumber()</a>	1404
10.392.3.16	<a href="#">GetStructureSetROIDescription()</a>	1404
10.392.3.17	<a href="#">GetStructureSetROIGenerationAlgorithm()</a>	1404
10.392.3.18	<a href="#">GetStructureSetROIName()</a>	1405
10.392.3.19	<a href="#">GetStructureSetROINumber()</a>	1405
10.392.3.20	<a href="#">GetStructureSetROIObservationLabel()</a>	1405

10.392.3.21 GetStructureSetROIRefFrameRefUID()	1405
10.392.3.22 GetStructureSetRTROIInterpretedType()	1405
10.392.3.23 New()	1405
10.392.3.24 PrintSelf()	1406
10.392.3.25 vtkGetStringMacro() [1/9]	1406
10.392.3.26 vtkGetStringMacro() [2/9]	1406
10.392.3.27 vtkGetStringMacro() [3/9]	1406
10.392.3.28 vtkGetStringMacro() [4/9]	1406
10.392.3.29 vtkGetStringMacro() [5/9]	1406
10.392.3.30 vtkGetStringMacro() [6/9]	1407
10.392.3.31 vtkGetStringMacro() [7/9]	1407
10.392.3.32 vtkGetStringMacro() [8/9]	1407
10.392.3.33 vtkGetStringMacro() [9/9]	1407
10.392.3.34 vtkSetStringMacro() [1/9]	1407
10.392.3.35 vtkSetStringMacro() [2/9]	1407
10.392.3.36 vtkSetStringMacro() [3/9]	1408
10.392.3.37 vtkSetStringMacro() [4/9]	1408
10.392.3.38 vtkSetStringMacro() [5/9]	1408
10.392.3.39 vtkSetStringMacro() [6/9]	1408
10.392.3.40 vtkSetStringMacro() [7/9]	1408
10.392.3.41 vtkSetStringMacro() [8/9]	1408
10.392.3.42 vtkSetStringMacro() [9/9]	1409
10.392.3.43 vtkTypeMacro()	1409
10.392.4 Member Data Documentation	1409
10.392.4.1 Internals	1409
10.392.4.2 ReferenceFrameOfReferenceUID	1409
10.392.4.3 ReferenceSeriesInstanceUID	1409
10.392.4.4 SeriesInstanceUID	1409
10.392.4.5 SOPInstanceUID	1410
10.392.4.6 StructureSetDate	1410
10.392.4.7 StructureSetLabel	1410
10.392.4.8 StructureSetName	1410
10.392.4.9 StructureSetTime	1410
10.392.4.10 StudyInstanceUID	1410
10.393 gdcmm::Waveform Class Reference	1410
10.393.1 Detailed Description	1411
10.393.2 Constructor & Destructor Documentation	1411
10.393.2.1 Waveform()	1411
10.394 gdcmm::WLMFindQuery Class Reference	1411

10.394.1 Detailed Description . . . . .	1412
10.394.2 Constructor & Destructor Documentation . . . . .	1412
10.394.2.1 WLMFindQuery() . . . . .	1413
10.394.3 Member Function Documentation . . . . .	1413
10.394.3.1 GetAbstractSyntaxUID() . . . . .	1413
10.394.3.2 GetTagListByLevel() . . . . .	1413
10.394.3.3 GetValidDataSet() . . . . .	1413
10.394.3.4 InitializeDataSet() . . . . .	1413
10.394.3.5 ValidateQuery() . . . . .	1414
10.394.4 Friends And Related Function Documentation . . . . .	1414
10.394.4.1 QueryFactory . . . . .	1414
10.395 gdcmm::Writer Class Reference . . . . .	1414
10.395.1 Detailed Description . . . . .	1416
10.395.2 Constructor & Destructor Documentation . . . . .	1417
10.395.2.1 Writer() . . . . .	1417
10.395.2.2 ~Writer() . . . . .	1417
10.395.3 Member Function Documentation . . . . .	1417
10.395.3.1 CheckFileMetaInformationOff() . . . . .	1417
10.395.3.2 CheckFileMetaInformationOn() . . . . .	1417
10.395.3.3 GetCheckFileMetaInformation() . . . . .	1417
10.395.3.4 GetFile() . . . . .	1418
10.395.3.5 GetStreamPtr() . . . . .	1418
10.395.3.6 SetCheckFileMetaInformation() . . . . .	1418
10.395.3.7 SetFile() . . . . .	1418
10.395.3.8 SetFileName() . . . . .	1419
10.395.3.9 SetStream() . . . . .	1419
10.395.3.10 SetWriteDataSetOnly() . . . . .	1419
10.395.3.11 Write() . . . . .	1419
10.395.4 Friends And Related Function Documentation . . . . .	1420
10.395.4.1 StreamImageWriter . . . . .	1420
10.395.5 Member Data Documentation . . . . .	1420
10.395.5.1 Ofstream . . . . .	1420
10.395.5.2 Stream . . . . .	1420
10.396 gdcmm::XMLDictReader Class Reference . . . . .	1420
10.396.1 Detailed Description . . . . .	1421
10.396.2 Constructor & Destructor Documentation . . . . .	1421
10.396.2.1 XMLDictReader() . . . . .	1421
10.396.2.2 ~XMLDictReader() . . . . .	1422
10.396.3 Member Function Documentation . . . . .	1422

10.396.3.1 CharacterDataHandler()	1422
10.396.3.2 EndElement()	1422
10.396.3.3 GetDict()	1422
10.396.3.4 HandleDescription()	1422
10.396.3.5 HandleEntry()	1423
10.396.3.6 StartElement()	1423
10.397 gdcmm::XMLPrinter Class Reference	1423
10.397.1 Member Enumeration Documentation	1424
10.397.1.1 PrintStyles	1424
10.397.2 Constructor & Destructor Documentation	1425
10.397.2.1 XMLPrinter()	1425
10.397.2.2 ~XMLPrinter()	1425
10.397.3 Member Function Documentation	1425
10.397.3.1 GetPrintStyle()	1425
10.397.3.2 HandleBulkData()	1425
10.397.3.3 Print()	1425
10.397.3.4 PrintDataElement()	1426
10.397.3.5 PrintDataSet()	1426
10.397.3.6 PrintSQ()	1426
10.397.3.7 SetFile()	1426
10.397.3.8 SetStyle()	1426
10.397.4 Member Data Documentation	1426
10.397.4.1 F	1427
10.397.4.2 PrintStyle	1427
10.398 gdcmm::XMLPrivateDictReader Class Reference	1427
10.398.1 Detailed Description	1428
10.398.2 Constructor & Destructor Documentation	1428
10.398.2.1 XMLPrivateDictReader()	1428
10.398.2.2 ~XMLPrivateDictReader()	1428
10.398.3 Member Function Documentation	1428
10.398.3.1 CharacterDataHandler()	1429
10.398.3.2 EndElement()	1429
10.398.3.3 GetPrivateDict()	1429
10.398.3.4 HandleDescription()	1429
10.398.3.5 HandleEntry()	1429
10.398.3.6 StartElement()	1429
<b>11 File Documentation</b>	<b>1431</b>
11.1 gdcmmAAbortPDU.h File Reference	1431

---

11.2 gdcmAAssociateACPDU.h File Reference . . . . .	1432
11.3 gdcmAAssociateRJPDU.h File Reference . . . . .	1432
11.4 gdcmAAssociateRQPDU.h File Reference . . . . .	1433
11.5 gdcmAbstractSyntax.h File Reference . . . . .	1434
11.6 gdcmAnonymizeEvent.h File Reference . . . . .	1435
11.7 gdcmAnonymizer.h File Reference . . . . .	1436
11.8 gdcmApplicationContext.h File Reference . . . . .	1437
11.9 gdcmApplicationEntity.h File Reference . . . . .	1438
11.10 gdcmAReleaseRPPDU.h File Reference . . . . .	1438
11.11 gdcmAReleaseRQPDU.h File Reference . . . . .	1439
11.12 gdcmARTIMTimer.h File Reference . . . . .	1440
11.13 gdcmASN1.h File Reference . . . . .	1441
11.14 gdcmAsynchronousOperationsWindowSub.h File Reference . . . . .	1442
11.15 gdcmAttribute.h File Reference . . . . .	1442
11.16 gdcmAudioCodec.h File Reference . . . . .	1444
11.17 gdcmBase64.h File Reference . . . . .	1445
11.18 gdcmBaseCompositeMessage.h File Reference . . . . .	1445
11.19 gdcmBaseNormalizedMessage.h File Reference . . . . .	1446
11.20 gdcmBasePDU.h File Reference . . . . .	1447
11.21 gdcmBaseQuery.h File Reference . . . . .	1448
11.22 gdcmBaseRootQuery.h File Reference . . . . .	1449
11.23 gdcmBasicOffsetTable.h File Reference . . . . .	1450
11.24 gdcmBitmap.h File Reference . . . . .	1451
11.25 gdcmBitmapToBitmapFilter.h File Reference . . . . .	1452
11.26 gdcmBoxRegion.h File Reference . . . . .	1453
11.27 gdcmByteBuffer.h File Reference . . . . .	1453
11.28 gdcmByteSwap.h File Reference . . . . .	1455
11.29 gdcmByteSwapFilter.h File Reference . . . . .	1455
11.30 gdcmByteValue.h File Reference . . . . .	1456
11.31 gdcmCAPICryptoFactory.h File Reference . . . . .	1457
11.32 gdcmCAPICryptographicMessageSyntax.h File Reference . . . . .	1458
11.33 gdcmCEchoMessages.h File Reference . . . . .	1458
11.34 gdcmCFindMessages.h File Reference . . . . .	1459
11.35 gdcmCMoveMessages.h File Reference . . . . .	1460
11.36 gdcmCodec.h File Reference . . . . .	1461
11.37 gdcmCoder.h File Reference . . . . .	1462
11.38 gdcmCodeString.h File Reference . . . . .	1464
11.39 gdcmCommand.h File Reference . . . . .	1465
11.40 gdcmCommandDataSet.h File Reference . . . . .	1466

11.41 gdcCompositeMessageFactory.h File Reference . . . . .	1467
11.42 gdcCompositeNetworkFunctions.h File Reference . . . . .	1467
11.43 gdcConstCharWrapper.h File Reference . . . . .	1468
11.44 gdcCP246ExplicitDataElement.h File Reference . . . . .	1468
11.45 gdcCryptoFactory.h File Reference . . . . .	1469
11.46 gdcCryptographicMessageSyntax.h File Reference . . . . .	1470
11.47 gdcCSAElement.h File Reference . . . . .	1471
11.48 gdcCSAHeader.h File Reference . . . . .	1472
11.49 gdcCSAHeaderDict.h File Reference . . . . .	1473
11.50 gdcCSAHeaderDictEntry.h File Reference . . . . .	1474
11.51 gdcCStoreMessages.h File Reference . . . . .	1475
11.52 gdcCurve.h File Reference . . . . .	1476
11.53 gdcDataElement.h File Reference . . . . .	1477
11.54 gdcDataEvent.h File Reference . . . . .	1479
11.55 gdcDataSet.h File Reference . . . . .	1480
11.56 gdcDataSetEvent.h File Reference . . . . .	1481
11.57 gdcDataSetHelper.h File Reference . . . . .	1481
11.58 gdcDecoder.h File Reference . . . . .	1482
11.59 gdcDefinedTerms.h File Reference . . . . .	1484
11.60 gdcDeflateStream.h File Reference . . . . .	1484
11.61 gdcDefs.h File Reference . . . . .	1485
11.62 gdcDeltaEncodingCodec.h File Reference . . . . .	1486
11.63 gdcDICOMDIR.h File Reference . . . . .	1487
11.64 gdcDICOMDIRGenerator.h File Reference . . . . .	1488
11.65 gdcDict.h File Reference . . . . .	1488
11.66 gdcDictConverter.h File Reference . . . . .	1490
11.67 gdcDictEntry.h File Reference . . . . .	1490
11.68 gdcDictPrinter.h File Reference . . . . .	1492
11.69 gdcDicts.h File Reference . . . . .	1492
11.70 gdcDIMSE.h File Reference . . . . .	1494
11.71 gdcDirectionCosines.h File Reference . . . . .	1494
11.72 gdcDirectory.h File Reference . . . . .	1495
11.73 gdcDirectoryHelper.h File Reference . . . . .	1496
11.74 gdcDummyValueGenerator.h File Reference . . . . .	1497
11.75 gdcDumper.h File Reference . . . . .	1498
11.76 gdcElement.h File Reference . . . . .	1498
11.77 gdcEmptyMaskGenerator.h File Reference . . . . .	1500
11.78 gdcEncapsulatedDocument.h File Reference . . . . .	1501
11.79 gdcEnumeratedValues.h File Reference . . . . .	1501



11.80 gdcEquipmentManufacturer.h File Reference . . . . .	1502
11.81 gdcEvent.h File Reference . . . . .	1503
11.81.1 Macro Definition Documentation . . . . .	1504
11.81.1.1 gdcEventMacro . . . . .	1504
11.82 gdcException.h File Reference . . . . .	1504
11.83 gdcExplicitDataElement.h File Reference . . . . .	1505
11.84 gdcExplicitImplicitDataElement.h File Reference . . . . .	1506
11.85 gdcFiducials.h File Reference . . . . .	1507
11.86 gdcFile.h File Reference . . . . .	1507
11.87 gdcFileAnonymizer.h File Reference . . . . .	1509
11.88 gdcFileChangeTransferSyntax.h File Reference . . . . .	1509
11.89 gdcFileDecompressLookupTable.h File Reference . . . . .	1510
11.90 gdcFileDerivation.h File Reference . . . . .	1511
11.91 gdcFileExplicitFilter.h File Reference . . . . .	1512
11.92 gdcFileMetaInformation.h File Reference . . . . .	1513
11.93 gdcFilename.h File Reference . . . . .	1514
11.94 gdcFileNameEvent.h File Reference . . . . .	1514
11.95 gdcFilenameGenerator.h File Reference . . . . .	1515
11.96 gdcFileSet.h File Reference . . . . .	1516
11.97 gdcFileStreamer.h File Reference . . . . .	1517
11.98 gdcFindPatientRootQuery.h File Reference . . . . .	1518
11.99 gdcFindStudyRootQuery.h File Reference . . . . .	1519
11.100 gdcFragment.h File Reference . . . . .	1519
11.101 gdcGlobal.h File Reference . . . . .	1521
11.102 gdcGroupDict.h File Reference . . . . .	1522
11.103 gdcIconImage.h File Reference . . . . .	1522
11.104 gdcIconImageFilter.h File Reference . . . . .	1523
11.105 gdcIconImageGenerator.h File Reference . . . . .	1524
11.106 gdcImage.h File Reference . . . . .	1525
11.107 gdcImageApplyLookupTable.h File Reference . . . . .	1526
11.108 gdcImageChangePhotometricInterpretation.h File Reference . . . . .	1527
11.109 gdcImageChangePlanarConfiguration.h File Reference . . . . .	1528
11.110 gdcImageChangeTransferSyntax.h File Reference . . . . .	1528
11.111 gdcImageCodec.h File Reference . . . . .	1529
11.112 gdcImageConverter.h File Reference . . . . .	1530
11.113 gdcImageFragmentSplitter.h File Reference . . . . .	1531
11.114 gdcImageHelper.h File Reference . . . . .	1532
11.115 gdcImageReader.h File Reference . . . . .	1533
11.116 gdcImageRegionReader.h File Reference . . . . .	1534

11.117 gdcmlImageToImageFilter.h File Reference . . . . .	1534
11.118 gdcmlImageWriter.h File Reference . . . . .	1535
11.119 gdcmlImplementationClassUIDSub.h File Reference . . . . .	1536
11.120 gdcmlImplementationUIDSub.h File Reference . . . . .	1537
11.121 gdcmlImplementationVersionNameSub.h File Reference . . . . .	1538
11.122 gdcmlImplicitDataElement.h File Reference . . . . .	1539
11.123 gdcmlIOD.h File Reference . . . . .	1540
11.124 gdcmlIODEntry.h File Reference . . . . .	1541
11.125 gdcmlIODs.h File Reference . . . . .	1543
11.126 gdcmlPPSorter.h File Reference . . . . .	1544
11.127 gdcmlItem.h File Reference . . . . .	1545
11.128 gdcmlJPEG12Codec.h File Reference . . . . .	1547
11.129 gdcmlJPEG16Codec.h File Reference . . . . .	1547
11.130 gdcmlJPEG2000Codec.h File Reference . . . . .	1548
11.131 gdcmlJPEG8Codec.h File Reference . . . . .	1549
11.132 gdcmlJPEGCodec.h File Reference . . . . .	1550
11.133 gdcmlJPEGLSCodec.h File Reference . . . . .	1551
11.134 gdcmlJSON.h File Reference . . . . .	1551
11.135 gdcmlKAKADUCodec.h File Reference . . . . .	1552
11.136 gdcmlLegacyMacro.h File Reference . . . . .	1553
11.136.1 Macro Definition Documentation . . . . .	1553
11.136.1.1 GDCM_LEGACY . . . . .	1554
11.136.1.2 GDCM_LEGACY_BODY . . . . .	1554
11.136.1.3 GDCM_LEGACY_REPLACED_BODY . . . . .	1554
11.137 gdcmlLO.h File Reference . . . . .	1554
11.138 gdcmlLookupTable.h File Reference . . . . .	1555
11.139 gdcmlMacro.h File Reference . . . . .	1556
11.140 gdcmlMacroEntry.h File Reference . . . . .	1558
11.140.1 Macro Definition Documentation . . . . .	1559
11.140.1.1 GDCMMACROENTRY_H . . . . .	1559
11.141 gdcmlMacros.h File Reference . . . . .	1560
11.142 gdcmlMaximumLengthSub.h File Reference . . . . .	1561
11.143 gdcmlMD5.h File Reference . . . . .	1563
11.144 gdcmlMediaStorage.h File Reference . . . . .	1563
11.145 gdcmlMeshPrimitive.h File Reference . . . . .	1565
11.146 gdcmlModalityPerformedProcedureStepCreateQuery.h File Reference . . . . .	1566
11.147 gdcmlModalityPerformedProcedureStepSetQuery.h File Reference . . . . .	1567
11.148 gdcmlModule.h File Reference . . . . .	1567
11.149 gdcmlModuleEntry.h File Reference . . . . .	1569

---

11.150 gdcModules.h File Reference . . . . .	1571
11.151 gdcMovePatientRootQuery.h File Reference . . . . .	1572
11.152 gdcMoveStudyRootQuery.h File Reference . . . . .	1573
11.153 gdcMrProtocol.h File Reference . . . . .	1573
11.154 gdcNActionMessages.h File Reference . . . . .	1575
11.155 gdcNCreateMessages.h File Reference . . . . .	1575
11.156 gdcNDeleteMessages.h File Reference . . . . .	1576
11.157 gdcNestedModuleEntries.h File Reference . . . . .	1577
11.158 gdcNetworkEvents.h File Reference . . . . .	1578
11.159 gdcNetworkStateID.h File Reference . . . . .	1579
11.160 gdcNEventReportMessages.h File Reference . . . . .	1580
11.161 gdcNGetMessages.h File Reference . . . . .	1581
11.162 gdcNormalizedMessageFactory.h File Reference . . . . .	1581
11.163 gdcNormalizedNetworkFunctions.h File Reference . . . . .	1582
11.164 gdcNSetMessages.h File Reference . . . . .	1583
11.165 gdcObject.h File Reference . . . . .	1583
11.166 gdcOpenSSLCryptoFactory.h File Reference . . . . .	1585
11.167 gdcOpenSSLCryptographicMessageSyntax.h File Reference . . . . .	1585
11.168 gdcOpenSSLP7CryptoFactory.h File Reference . . . . .	1587
11.169 gdcOpenSSLP7CryptographicMessageSyntax.h File Reference . . . . .	1587
11.170 gdcOrientation.h File Reference . . . . .	1589
11.171 gdcOverlay.h File Reference . . . . .	1589
11.172 gdcParseException.h File Reference . . . . .	1590
11.173 gdcParser.h File Reference . . . . .	1592
11.174 gdcPatient.h File Reference . . . . .	1592
11.175 gdcPDataTFPDU.h File Reference . . . . .	1593
11.176 gdcPDBelement.h File Reference . . . . .	1594
11.177 gdcPDBHeader.h File Reference . . . . .	1595
11.178 gdcPDFCodec.h File Reference . . . . .	1596
11.179 gdcPDUFactory.h File Reference . . . . .	1597
11.180 gdcPersonName.h File Reference . . . . .	1597
11.181 gdcPGXCodec.h File Reference . . . . .	1598
11.182 gdcPhotometricInterpretation.h File Reference . . . . .	1599
11.183 gdcPixelFormat.h File Reference . . . . .	1600
11.184 gdcPixmap.h File Reference . . . . .	1601
11.185 gdcPixmapReader.h File Reference . . . . .	1602
11.186 gdcPixmapToPixmapFilter.h File Reference . . . . .	1603
11.187 gdcPixmapWriter.h File Reference . . . . .	1603
11.188 gdcPNMCodec.h File Reference . . . . .	1605

11.189 gdcnPreamble.h File Reference . . . . .	1605
11.190 gdcmPresentationContext.h File Reference . . . . .	1607
11.191 gdcmPresentationContextAC.h File Reference . . . . .	1608
11.192 gdcmPresentationContextGenerator.h File Reference . . . . .	1609
11.193 gdcmPresentationContextRQ.h File Reference . . . . .	1609
11.194 gdcmPresentationDataValue.h File Reference . . . . .	1610
11.195 gdcmPrinter.h File Reference . . . . .	1611
11.196 gdcmPrivateTag.h File Reference . . . . .	1613
11.197 gdcmProgressEvent.h File Reference . . . . .	1614
11.198 gdcmPVRGCodec.h File Reference . . . . .	1615
11.199 gdcmPythonFilter.h File Reference . . . . .	1615
11.200 gdcmQueryBase.h File Reference . . . . .	1616
11.201 gdcmQueryFactory.h File Reference . . . . .	1617
11.202 gdcmQueryImage.h File Reference . . . . .	1618
11.203 gdcmQueryPatient.h File Reference . . . . .	1619
11.204 gdcmQuerySeries.h File Reference . . . . .	1620
11.205 gdcmQueryStudy.h File Reference . . . . .	1621
11.206 gdcmRAWCodec.h File Reference . . . . .	1622
11.207 gdcmReader.h File Reference . . . . .	1622
11.208 gdcmRegion.h File Reference . . . . .	1624
11.209 gdcmRescaler.h File Reference . . . . .	1625
11.210 gdcmRLECodec.h File Reference . . . . .	1626
11.211 gdcmRoleSelectionSub.h File Reference . . . . .	1626
11.212 gdcmScanner.h File Reference . . . . .	1627
11.213 gdcmSegment.h File Reference . . . . .	1628
11.214 gdcmSegmentedPaletteColorLookupTable.h File Reference . . . . .	1629
11.215 gdcmSegmentHelper.h File Reference . . . . .	1630
11.216 gdcmSegmentReader.h File Reference . . . . .	1631
11.217 gdcmSegmentWriter.h File Reference . . . . .	1632
11.218 gdcmSequenceOfFragments.h File Reference . . . . .	1633
11.219 gdcmSequenceOfItems.h File Reference . . . . .	1634
11.220 gdcmSerieHelper.h File Reference . . . . .	1634
11.221 gdcmSeries.h File Reference . . . . .	1636
11.222 gdcmServiceClassApplicationInformation.h File Reference . . . . .	1637
11.223 gdcmServiceClassUser.h File Reference . . . . .	1638
11.224 gdcmSHA1.h File Reference . . . . .	1638
11.225 gdcmSimpleSubjectWatcher.h File Reference . . . . .	1639
11.226 gdcmSmartPointer.h File Reference . . . . .	1640
11.227 gdcmSOPClassExtendedNegociationSub.h File Reference . . . . .	1641

---

11.228 gdcmsOPClassUIDToIOD.h File Reference . . . . .	1642
11.229 gdcmsorter.h File Reference . . . . .	1643
11.230 gdcmspacing.h File Reference . . . . .	1645
11.231 gdcmspectroscopy.h File Reference . . . . .	1645
11.232 gdcmsplitmosaicfilter.h File Reference . . . . .	1646
11.233 gdcmsstaticassert.h File Reference . . . . .	1647
11.233.1 Macro Definition Documentation . . . . .	1647
11.233.1.1 GDCM_DO_JOIN . . . . .	1648
11.233.1.2 GDCM_DO_JOIN2 . . . . .	1648
11.233.1.3 GDCM_JOIN . . . . .	1648
11.233.1.4 GDCM_STATIC_ASSERT . . . . .	1648
11.234 gdcmsstreamimageReader.h File Reference . . . . .	1649
11.235 gdcmsstreamimageWriter.h File Reference . . . . .	1649
11.236 gdcmsstrictScanner.h File Reference . . . . .	1650
11.237 gdcmsstring.h File Reference . . . . .	1651
11.238 gdcmsstringFilter.h File Reference . . . . .	1652
11.239 gdcmsstudy.h File Reference . . . . .	1653
11.240 gdcmssubject.h File Reference . . . . .	1654
11.241 gdcmsurface.h File Reference . . . . .	1655
11.242 gdcmsurfaceHelper.h File Reference . . . . .	1656
11.243 gdcmsurfaceReader.h File Reference . . . . .	1657
11.244 gdcmsurfaceWriter.h File Reference . . . . .	1658
11.245 gdcmswapCode.h File Reference . . . . .	1658
11.246 gdcmswapper.h File Reference . . . . .	1660
11.247 gdcmsystem.h File Reference . . . . .	1661
11.248 gdcmsTable.h File Reference . . . . .	1661
11.249 gdcmsTableEntry.h File Reference . . . . .	1662
11.250 gdcmsTableReader.h File Reference . . . . .	1664
11.251 gdcmsTag.h File Reference . . . . .	1665
11.252 gdcmsTagPath.h File Reference . . . . .	1666
11.253 gdcmsTagToVR.h File Reference . . . . .	1666
11.254 gdcmsTerminal.h File Reference . . . . .	1667
11.255 gdcmsTestDriver.h File Reference . . . . .	1668
11.256 gdcmsTesting.h File Reference . . . . .	1669
11.257 gdcmsTrace.h File Reference . . . . .	1670
11.257.1 Macro Definition Documentation . . . . .	1671
11.257.1.1 GDCM_FUNCTION . . . . .	1671
11.257.1.2 gdcmsAssertAlwaysMacro . . . . .	1671
11.257.1.3 gdcmsAssertMacro . . . . .	1671

11.257.1.4 gdcDebugMacro . . . . .	1673
11.257.1.5 gdcErrorMacro . . . . .	1673
11.257.1.6 gdcWarningMacro . . . . .	1674
11.258 gdcTransferSyntax.h File Reference . . . . .	1674
11.259 gdcTransferSyntaxSub.h File Reference . . . . .	1675
11.260 gdcType.h File Reference . . . . .	1676
11.261 gdcTypes.h File Reference . . . . .	1677
11.262 gdcUIDGenerator.h File Reference . . . . .	1678
11.263 gdcUIDs.h File Reference . . . . .	1679
11.264 gdcULAction.h File Reference . . . . .	1680
11.265 gdcULActionAA.h File Reference . . . . .	1681
11.266 gdcULActionAE.h File Reference . . . . .	1681
11.267 gdcULActionAR.h File Reference . . . . .	1682
11.268 gdcULActionDT.h File Reference . . . . .	1683
11.269 gdcULBasicCallback.h File Reference . . . . .	1683
11.270 gdcULConnection.h File Reference . . . . .	1684
11.271 gdcULConnectionCallback.h File Reference . . . . .	1685
11.272 gdcULConnectionInfo.h File Reference . . . . .	1686
11.273 gdcULConnectionManager.h File Reference . . . . .	1687
11.274 gdcULEvent.h File Reference . . . . .	1688
11.275 gdcULTransitionTable.h File Reference . . . . .	1689
11.276 gdcULWritingCallback.h File Reference . . . . .	1690
11.277 gdcUNExplicitDataElement.h File Reference . . . . .	1691
11.278 gdcUNExplicitImplicitDataElement.h File Reference . . . . .	1691
11.279 gdcUnpacker12Bits.h File Reference . . . . .	1692
11.280 gdcUsage.h File Reference . . . . .	1693
11.281 gdcUserInformation.h File Reference . . . . .	1695
11.282 gdcUUIDGenerator.h File Reference . . . . .	1696
11.283 gdcValidate.h File Reference . . . . .	1696
11.284 gdcValue.h File Reference . . . . .	1697
11.285 gdcValueIO.h File Reference . . . . .	1698
11.286 gdcVersion.h File Reference . . . . .	1698
11.287 gdcVL.h File Reference . . . . .	1699
11.288 gdcVM.h File Reference . . . . .	1700
11.288.1 Macro Definition Documentation . . . . .	1701
11.288.1.1 TYPETOLENGTH . . . . .	1701
11.289 gdcVR.h File Reference . . . . .	1701
11.289.1 Macro Definition Documentation . . . . .	1703
11.289.1.1 TYPETOENCODING . . . . .	1703

11.289.1.2 VRTypeTemplateCase . . . . .	1703
11.290 gdcVR16ExplicitDataElement.h File Reference . . . . .	1703
11.291 gdcWaveform.h File Reference . . . . .	1704
11.292 gdcWin32.h File Reference . . . . .	1704
11.292.1 Macro Definition Documentation . . . . .	1705
11.292.1.1 GDCM_EXPORT . . . . .	1705
11.293 gdcWLMFindQuery.h File Reference . . . . .	1705
11.294 gdcWriter.h File Reference . . . . .	1706
11.295 gdcXMLDictReader.h File Reference . . . . .	1707
11.296 gdcXMLPrinter.h File Reference . . . . .	1707
11.297 gdcXMLPrivateDictReader.h File Reference . . . . .	1708
11.298 README.txt File Reference . . . . .	1709
11.299 TestsList.txt File Reference . . . . .	1709
11.300 vtkGDCMImageReader.h File Reference . . . . .	1709
11.300.1 Macro Definition Documentation . . . . .	1710
11.300.1.1 VTK_CMYK . . . . .	1710
11.300.1.2 VTK_INVERSE_LUMINANCE . . . . .	1710
11.300.1.3 VTK_LOOKUP_TABLE . . . . .	1710
11.300.1.4 VTK_YBR . . . . .	1710
11.301 vtkGDCMImageReader2.h File Reference . . . . .	1711
11.301.1 Macro Definition Documentation . . . . .	1711
11.301.1.1 VTK_CMYK . . . . .	1711
11.301.1.2 VTK_INVERSE_LUMINANCE . . . . .	1712
11.301.1.3 VTK_LOOKUP_TABLE . . . . .	1712
11.301.1.4 VTK_YBR . . . . .	1712
11.302 vtkGDCMImageWriter.h File Reference . . . . .	1712
11.303 vtkGDCMMedicalImageProperties.h File Reference . . . . .	1713
11.304 vtkGDCMPolyDataReader.h File Reference . . . . .	1713
11.305 vtkGDCMPolyDataWriter.h File Reference . . . . .	1714
11.306 vtkGDCMTesting.h File Reference . . . . .	1715
11.307 vtkGDCMThreadedImageReader.h File Reference . . . . .	1715
11.308 vtkGDCMThreadedImageReader2.h File Reference . . . . .	1716
11.309 vtkImageColorViewer.h File Reference . . . . .	1716
11.310 vtkImageMapToColors16.h File Reference . . . . .	1717
11.311 vtkImageMapToWindowLevelColors2.h File Reference . . . . .	1717
11.312 vtkImagePlanarComponentsToComponents.h File Reference . . . . .	1718
11.313 vtkImageRGBToYBR.h File Reference . . . . .	1718
11.314 vtkImageYBRToRGB.h File Reference . . . . .	1719
11.315 vtkLookupTable16.h File Reference . . . . .	1719

11.316 vtkRTStructSetProperties.h File Reference . . . . .	1720
<b>12 Example Documentation</b>	<b>1721</b>
12.1 TestByteSwap.cxx . . . . .	1721
12.2 PatchFile.cxx . . . . .	1723
12.3 SimplePrint.cs . . . . .	1724
12.4 TestReader.cxx . . . . .	1725
12.5 TestReader.py . . . . .	1726
12.6 DecompressJPEGFile.cs . . . . .	1727
12.7 ManipulateFile.cs . . . . .	1728
12.8 ClinicalTrialIdentificationWorkflow.cs . . . . .	1729
12.9 GenerateDICOMDIR.cs . . . . .	1731
12.10 GenFakeImage.cxx . . . . .	1732
12.11 ReformatFile.cs . . . . .	1733
12.12 DecompressImage.cs . . . . .	1734
12.13 StandardizeFiles.cs . . . . .	1736
12.14 ScanDirectory.cs . . . . .	1737
12.15 BasicAnonymizer.cs . . . . .	1738
12.16 BasicImageAnonymizer.cs . . . . .	1739
12.17 CompressLossyJPEG.cs . . . . .	1740
12.18 DecompressImageMultiframe.cs . . . . .	1741
12.19 DumpCSA.cs . . . . .	1743
12.20 ExtractEncapsulatedFile.cs . . . . .	1744
12.21 ExtractImageRegion.cs . . . . .	1745
12.22 ExtractImageRegionWithLUT.cs . . . . .	1746
12.23 ExtractOneFrame.cs . . . . .	1747
12.24 FileAnonymize.cs . . . . .	1748
12.25 FileChangeTS.cs . . . . .	1749
12.26 FileChangeTSLossy.cs . . . . .	1751
12.27 FileStreaming.cs . . . . .	1753
12.28 GetArray.cs . . . . .	1754
12.29 MpegVideoInfo.cs . . . . .	1755
12.30 NewSequence.cs . . . . .	1759
12.31 RescaleImage.cs . . . . .	1760
12.32 SendFileSCU.cs . . . . .	1761
12.33 SimplePrintPatientName.cs . . . . .	1761
12.34 SortImage2.cs . . . . .	1762
12.35 CStoreQtProgress.cxx . . . . .	1762
12.36 ChangePrivateTags.cxx . . . . .	1764



12.37 ChangeSequenceUltrasound.cxx . . . . .	1765
12.38 CheckBigEndianBug.cxx . . . . .	1766
12.39 ClinicalTrialAnnotate.cxx . . . . .	1768
12.40 CompressImage.cxx . . . . .	1769
12.41 ConvertToQImage.cxx . . . . .	1770
12.42 CreateARGBImage.cxx . . . . .	1771
12.43 CreateCMYKImage.cxx . . . . .	1772
12.44 CreateJPIPDataSet.cxx . . . . .	1773
12.45 DeriveSeries.cxx . . . . .	1774
12.46 DiffFile.cxx . . . . .	1775
12.47 DiscriminateVolume.cxx . . . . .	1776
12.48 DumpADAC.cxx . . . . .	1779
12.49 DumpExamCard.cxx . . . . .	1783
12.50 DumpGEMSMovieGroup.cxx . . . . .	1791
12.51 DumpImageHeaderInfo.cxx . . . . .	1797
12.52 DumpPhilipsECHO.cxx . . . . .	1799
12.53 DumpSiemensBase64.cxx . . . . .	1804
12.54 DumpToSQLITE3.cxx . . . . .	1805
12.55 DumpToshibaDTI.cxx . . . . .	1806
12.56 DumpVisusChange.cxx . . . . .	1808
12.57 DuplicatePCDE.cxx . . . . .	1809
12.58 ELSCINT1WaveToText.cxx . . . . .	1812
12.59 EmptyMask.cxx . . . . .	1813
12.60 EncapsulateFileInRawData.cxx . . . . .	1814
12.61 ExtractEncryptedContent.cxx . . . . .	1815
12.62 ExtractIconFromFile.cxx . . . . .	1816
12.63 Extracting_All_Resolution.cxx . . . . .	1817
12.64 Fake_Image_Using_Stream_Image_Writer.cxx . . . . .	1822
12.65 FixBrokenJ2K.cxx . . . . .	1824
12.66 FixJAIBugJPEGLS.cxx . . . . .	1826
12.67 FixOrientation.cxx . . . . .	1828
12.68 GenAllVR.cxx . . . . .	1829
12.69 GenFakeIdentifyFile.cxx . . . . .	1831
12.70 GenLongSeqs.cxx . . . . .	1833
12.71 GenSeqs.cxx . . . . .	1834
12.72 GenerateStandardSOPClasses.cxx . . . . .	1836
12.73 GetJPEGSamplePrecision.cxx . . . . .	1837
12.74 GetSequenceUltrasound.cxx . . . . .	1838
12.75 GetSubSequenceData.cxx . . . . .	1839

12.76 HelloVizWorld.cxx . . . . .	1842
12.77 HelloWorld.cxx . . . . .	1843
12.78 LargeVRDSExplicit.cxx . . . . .	1844
12.79 MakeTemplate.cxx . . . . .	1845
12.80 MergeTwoFiles.cxx . . . . .	1846
12.81 MrProtocol.cxx . . . . .	1847
12.82 PrintLUT.cxx . . . . .	1854
12.83 PublicDict.cxx . . . . .	1855
12.84 QIDO-RS.cxx . . . . .	1855
12.85 ReadAndDumpDICOMDIR.cxx . . . . .	1856
12.86 ReadAndDumpDICOMDIR2.cxx . . . . .	1858
12.87 ReadAndPrintAttributes.cxx . . . . .	1862
12.88 ReadExplicitLengthSQIVR.cxx . . . . .	1863
12.89 ReadGEMSSDO.cxx . . . . .	1864
12.90 ReadMultiTimesException.cxx . . . . .	1866
12.91 ReadUTF8QtDir.cxx . . . . .	1867
12.92 SimpleScanner.cxx . . . . .	1868
12.93 SortImage.cxx . . . . .	1870
12.94 StreamImageReaderTest.cxx . . . . .	1871
12.95 TemplateEmptyImage.cxx . . . . .	1874
12.96 TraverseModules.cxx . . . . .	1875
12.97 VolumeSorter.cxx . . . . .	1876
12.98 csa2img.cxx . . . . .	1879
12.99 iU22tomultisc.cxx . . . . .	1880
12.100 pmsct_rgb1.cxx . . . . .	1881
12.101 rle2img.cxx . . . . .	1885
12.102 uid_unique.cxx . . . . .	1887
12.103 DecompressImage.java . . . . .	1888
12.104 DecompressPixmap.java . . . . .	1889
12.105 ExtractImageRegion.java . . . . .	1890
12.106 FileAnonymize.java . . . . .	1890
12.107 HelloSimple.java . . . . .	1891
12.108 ReadFiles.java . . . . .	1892
12.109 ScanDirectory.java . . . . .	1893
12.110 SimplePrint.java . . . . .	1897
12.111 AddPrivateAttribute.py . . . . .	1897
12.112 ConvertMPL.py . . . . .	1898
12.113 ConvertNumpy.py . . . . .	1899
12.114 ConvertPIL.py . . . . .	1900

12.115 CreateRAWStorage.py	1901
12.116 DecompressImage.py	1903
12.117 DumbAnonymizer.py	1904
12.118 ExtractImageRegion.py	1905
12.119 FindAllPatientName.py	1906
12.120 FixCommaBug.py	1907
12.121 GetPortionCSAHeader.py	1908
12.122 HelloWorld.py	1908
12.123 ManipulateFile.py	1909
12.124 ManipulateSequence.py	1910
12.125 MergeFile.py	1911
12.126 NewSequence.py	1912
12.127 PhilipsPrivateRescaleInterceptSlope.py	1913
12.128 PlaySound.py	1913
12.129 PrivateDict.py	1914
12.130 ReWriteSCAsMR.py	1915
12.131 ReadAndDumpDICOMDIR.py	1916
12.132 RemovePrivateTags.py	1918
12.133 ScanDirectory.py	1918
12.134 SortImage.py	1919
12.135 WriteBuffer.py	1920
12.136 HelloActiviz.cs	1921
12.137 HelloActiviz2.cs	1922
12.138 HelloActiviz3.cs	1923
12.139 HelloActiviz4.cs	1924
12.140 HelloActiviz5.cs	1924
12.141 HelloVTKWorld.cs	1925
12.142 HelloVTKWorld2.cs	1926
12.143 MetalmageMD5Activiz.cs	1927
12.144 RefCounting.cs	1928
12.145 Compute3DSpacing.cxx	1929
12.146 Convert16BitsTo8Bits.cxx	1930
12.147 ConvertMultiFrameToSingleFrame.cxx	1930
12.148 ConvertRGBToLuminance.cxx	1931
12.149 ConvertSingleBitTo8Bits.cxx	1932
12.150 CreateFakePET.cxx	1933
12.151 CreateFakeRTDOSE.cxx	1935
12.152 GenerateRTSTRUCT.cxx	1936
12.153 MagnifyFile.cxx	1938

12.154 gdcmmorthoplanes.cxx . . . . .	1939
12.155 gdcmmreslice.cxx . . . . .	1945
12.156 gdcmmrtionplan.cxx . . . . .	1947
12.157 gdcmmrtplan.cxx . . . . .	1950
12.158 gdcmmscene.cxx . . . . .	1954
12.159 gdcmmtexture.cxx . . . . .	1955
12.160 gdcmmvolume.cxx . . . . .	1957
12.161 offscreenimage.cxx . . . . .	1958
12.162 reslicesphere.cxx . . . . .	1959
12.163 rtstructapp.cxx . . . . .	1966
12.164 threadgdcmm.cxx . . . . .	1967
12.165 AWTMedical3.java . . . . .	1970
12.166 HelloVTKWorld.java . . . . .	1974
12.167 MIPViewer.java . . . . .	1975
12.168 MPRViewer.java . . . . .	1977
12.169 MPRViewer2.java . . . . .	1979
12.170 ReadSeriesIntoVTK.java . . . . .	1983
12.171 CastConvertPhilips.py . . . . .	1984
12.172 headsq2dcm.py . . . . .	1986

<b>Index</b>	<b>1987</b>
--------------	-------------

## Chapter 1

# GDCM Documentation

This is the developpers documentation.

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

`http://gdcm.sourceforge.net/3.0/gdcm-3.0.8.pdf`

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

`http://gdcm.sourceforge.net/3.0/gdcm-3.0.8-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](#)

### 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>	77
<a href="#">gdc::SegmentHelper</a>	84
<a href="#">gdc::terminal</a>	84
Class for Terminal	84



## Chapter 6

# Hierarchical Index

### 6.1 Class Hierarchy

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

gdcn::network::AbstractSyntax . . . . .	104
gdcn::network::ApplicationContext . . . . .	119
gdcn::ApplicationEntity . . . . .	121
gdcn::network::ARTIMTimer . . . . .	129
gdcn::ASN1 . . . . .	130
gdcn::network::AsynchronousOperationsWindowSub . . . . .	132
gdcn::Attribute< Group, Element, TVR, TVM > . . . . .	134
gdcn::Attribute< Group, Element, TVR, VM::VM1 > . . . . .	144
gdcn::Attribute< Group, Element, TVR, VM::VM1_n > . . . . .	153
gdcn::Attribute< Group, Element, TVR, VM::VM1_3 > . . . . .	151
gdcn::Attribute< Group, Element, TVR, VM::VM1_8 > . . . . .	152
gdcn::Attribute< Group, Element, TVR, VM::VM2_n > . . . . .	161
gdcn::Attribute< Group, Element, TVR, VM::VM2_2n > . . . . .	160
gdcn::Attribute< Group, Element, TVR, VM::VM3_n > . . . . .	164
gdcn::Attribute< Group, Element, TVR, VM::VM3_3n > . . . . .	163
gdcn::Base64 . . . . .	168
gdcn::network::BaseCompositeMessage . . . . .	171
gdcn::network::CEchoRQ . . . . .	237
gdcn::network::CEchoRSP . . . . .	239
gdcn::network::CFindCancelRQ . . . . .	240
gdcn::network::CFindRQ . . . . .	242
gdcn::network::CFindRSP . . . . .	243
gdcn::network::CMoveCancelRq . . . . .	245
gdcn::network::CMoveRQ . . . . .	246
gdcn::network::CMoveRSP . . . . .	247
gdcn::network::CStoreRQ . . . . .	300
gdcn::network::CStoreRSP . . . . .	301
gdcn::network::BaseNormalizedMessage . . . . .	173
gdcn::network::NActionRQ . . . . .	720
gdcn::network::NActionRSP . . . . .	721

gdcmm::network::NCreateRQ . . . . .	723
gdcmm::network::NCreateRSP . . . . .	724
gdcmm::network::NDeleteRQ . . . . .	726
gdcmm::network::NDeleteRSP . . . . .	727
gdcmm::network::NEventReportRQ . . . . .	732
gdcmm::network::NEventReportRSP . . . . .	733
gdcmm::network::NGetRQ . . . . .	735
gdcmm::network::NGetRSP . . . . .	736
gdcmm::network::NSetRQ . . . . .	743
gdcmm::network::NSetRSP . . . . .	745
gdcmm::network::BasePDU . . . . .	175
gdcmm::network::AAabortPDU . . . . .	87
gdcmm::network::AAAssociateACPDU . . . . .	90
gdcmm::network::AAAssociateRJPDU . . . . .	94
gdcmm::network::AAAssociateRQPDU . . . . .	97
gdcmm::network::AReleaseRPPDU . . . . .	124
gdcmm::network::AReleaseRQPDU . . . . .	126
gdcmm::network::PDataTFPDU . . . . .	781
std::basic_string< Char >	
std::string	
gdcmm::String< TDelimiter, TMaxLength, TPadChar > . . . . .	1041
gdcmm::SegmentHelper::BasicCodedEntry . . . . .	188
gdcmm::BitmapToBitmapFilter . . . . .	209
gdcmm::PixmapToPixmapFilter . . . . .	825
gdcmm::ImageToImageFilter . . . . .	579
gdcmm::ImageApplyLookupTable . . . . .	529
gdcmm::ImageChangePhotometricInterpretation . . . . .	532
gdcmm::ImageChangePlanarConfiguration . . . . .	536
gdcmm::ImageChangeTransferSyntax . . . . .	540
gdcmm::ImageFragmentSplitter . . . . .	561
gdcmm::ByteBuffer . . . . .	217
gdcmm::ByteSwap< T > . . . . .	219
gdcmm::ByteSwapFilter . . . . .	221
gdcmm::network::CFind . . . . .	240
gdcmm::Coder . . . . .	250
gdcmm::Codec . . . . .	249
gdcmm::AudioCodec . . . . .	166
gdcmm::ImageCodec . . . . .	545
gdcmm::DeltaEncodingCodec . . . . .	352
gdcmm::JPEG2000Codec . . . . .	619
gdcmm::JPEGCodec . . . . .	630
gdcmm::JPEG12Codec . . . . .	613
gdcmm::JPEG16Codec . . . . .	616
gdcmm::JPEG8Codec . . . . .	626
gdcmm::JPEGLSCCodec . . . . .	638
gdcmm::KAKADUCCodec . . . . .	647
gdcmm::PGXCodec . . . . .	799
gdcmm::PNMCodec . . . . .	833
gdcmm::PVRGCodec . . . . .	874
gdcmm::RAWCodec . . . . .	894
gdcmm::RLECodec . . . . .	915
gdcmm::PDFCodec . . . . .	790
gdcmm::CodeString . . . . .	252



gdcm::network::CompositeMessageFactory . . . . .	263
gdcm::CompositeNetworkFunctions . . . . .	264
gdcm::ConstCharWrapper . . . . .	269
gdcm::CryptoFactory . . . . .	272
gdcm::CAPICryptoFactory . . . . .	232
gdcm::OpenSSLCryptoFactory . . . . .	750
gdcm::OpenSSLP7CryptoFactory . . . . .	755
gdcm::CryptographicMessageSyntax . . . . .	275
gdcm::CAPICryptographicMessageSyntax . . . . .	233
gdcm::OpenSSLCryptographicMessageSyntax . . . . .	752
gdcm::OpenSSLP7CryptographicMessageSyntax . . . . .	757
gdcm::CSAElement . . . . .	278
gdcm::CSAHeader . . . . .	286
gdcm::CSAHeaderDict . . . . .	292
gdcm::CSAHeaderDictEntry . . . . .	295
gdcm::DataElement . . . . .	309
gdcm::CP246ExplicitDataElement . . . . .	270
gdcm::ExplicitDataElement . . . . .	439
gdcm::ExplicitImplicitDataElement . . . . .	442
gdcm::Fragment . . . . .	504
gdcm::BasicOffsetTable . . . . .	191
gdcm::ImplicitDataElement . . . . .	590
gdcm::Item . . . . .	607
gdcm::UNExplicitDataElement . . . . .	1250
gdcm::UNExplicitImplicitDataElement . . . . .	1253
gdcm::VR16ExplicitDataElement . . . . .	1290
gdcm::DataSet . . . . .	327
gdcm::CommandDataSet . . . . .	260
gdcm::FileMetaInformation . . . . .	468
gdcm::DataSetHelper . . . . .	344
gdcm::Decoder . . . . .	345
gdcm::Codec . . . . .	249
gdcm::DefinedTerms . . . . .	347
gdcm::Defs . . . . .	347
gdcm::DICOMDIR . . . . .	354
gdcm::DICOMDIRGenerator . . . . .	355
gdcm::Dict . . . . .	359
gdcm::DictConverter . . . . .	363
gdcm::DictEntry . . . . .	368
gdcm::Dicts . . . . .	376
gdcm::network::DIMSE . . . . .	380
gdcm::DirectionCosines . . . . .	382
gdcm::Directory . . . . .	385
gdcm::DirectoryHelper . . . . .	390
gdcm::DummyValueGenerator . . . . .	392
gdcm::Element< TVR, TVM > . . . . .	395
gdcm::Element< TVR, VM::VM1_n > . . . . .	403
gdcm::Element< TVR, VM::VM1_2 > . . . . .	401
gdcm::Element< TVR, VM::VM2_n > . . . . .	410
gdcm::Element< TVR, VM::VM2_2n > . . . . .	408
gdcm::Element< TVR, VM::VM3_n > . . . . .	414
gdcm::Element< TVR, VM::VM3_3n > . . . . .	412

gdcm::Element< VR::AS, VM::VM5 > . . . . .	416
gdcm::Element< VR::OB, VM::VM1_n > . . . . .	395
gdcm::Element< VR::OB, VM::VM1 > . . . . .	417
gdcm::Element< VR::OW, VM::VM1_n > . . . . .	395
gdcm::Element< VR::OW, VM::VM1 > . . . . .	418
gdcm::ElementDisableCombinations< TVR, TVM > . . . . .	420
gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n > . . . . .	421
gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n > . . . . .	421
gdcm::EmptyMaskGenerator . . . . .	421
gdcm::EncapsulatedDocument . . . . .	424
gdcm::EncodingImplementation< T > . . . . .	425
gdcm::EncodingImplementation< VR::VRASCII > . . . . .	426
gdcm::EncodingImplementation< VR::VRBINARY > . . . . .	427
gdcm::EnumeratedValues . . . . .	430
gdcm::EquipmentManufacturer . . . . .	430
gdcm::Event . . . . .	432
gdcm::AnyEvent . . . . .	118
gdcm::AbortEvent . . . . .	103
gdcm::AnonymizeEvent . . . . .	107
gdcm::DataEvent . . . . .	323
gdcm::DataSetEvent . . . . .	340
gdcm::EndEvent . . . . .	429
gdcm::ExitEvent . . . . .	438
gdcm::FileNameEvent . . . . .	481
gdcm::InitializeEvent . . . . .	593
gdcm::IterationEvent . . . . .	612
gdcm::ModifiedEvent . . . . .	701
gdcm::ProgressEvent . . . . .	870
gdcm::StartEvent . . . . .	1019
gdcm::UserEvent . . . . .	1259
gdcm::NoEvent . . . . .	738
std::exception	
gdcm::CSAHeaderDictException . . . . .	299
gdcm::DataElementException . . . . .	323
gdcm::Exception . . . . .	436
gdcm::ParseException . . . . .	774
gdcm::Fiducials . . . . .	445
gdcm::FileDerivation . . . . .	461
gdcm::FileExplicitFilter . . . . .	465
gdcm::Filename . . . . .	477
gdcm::FilenameGenerator . . . . .	485
gdcm::FileSet . . . . .	489
gdcm::Global . . . . .	507
gdcm::GroupDict . . . . .	511
gdcm::IconImageFilter . . . . .	514
gdcm::IconImageGenerator . . . . .	517
gdcm::ignore_char . . . . .	521
gdcm::ImageConverter . . . . .	559
gdcm::ImageHelper . . . . .	564
gdcm::network::ImplementationClassUIDSub . . . . .	586
gdcm::network::ImplementationUIDSub . . . . .	587
gdcm::network::ImplementationVersionNameSub . . . . .	588
gdcm::IOD . . . . .	594

gdcm::IODEntry	596
gdcm::IODs	599
gdcm::JSON	645
gdcm::Scanner::Itstr	663
gdcm::StrictScanner::Itstr	664
gdcm::Macro	664
gdcm::Macros	667
gdcm::network::MaximumLengthSub	669
gdcm::MD5	671
gdcm::MediaStorage	672
gdcm::Module	702
gdcm::ModuleEntry	705
gdcm::NestedModuleEntries	729
gdcm::Modules	709
gdcm::MrProtocol	717
gdcm::network::NormalizedMessageFactory	739
gdcm::NormalizedNetworkFunctions	740
gdcm::Object	746
gdcm::BaseQuery	177
gdcm::BaseRootQuery	183
gdcm::FindPatientRootQuery	498
gdcm::FindStudyRootQuery	501
gdcm::MovePatientRootQuery	712
gdcm::MoveStudyRootQuery	715
gdcm::WLMFindQuery	1411
gdcm::ModalityPerformedProcedureStepCreateQuery	695
gdcm::ModalityPerformedProcedureStepSetQuery	698
gdcm::Bitmap	194
gdcm::Pixmap	815
gdcm::Image	522
gdcm::Curve	303
gdcm::File	445
gdcm::FileWithName	496
gdcm::LookupTable	655
gdcm::SegmentedPaletteColorLookupTable	942
gdcm::MeshPrimitive	689
gdcm::Overlay	763
gdcm::Segment	932
gdcm::Subject	1051
gdcm::Anonymizer	110
gdcm::Command	257
gdcm::MemberCommand< T >	683
gdcm::SimpleMemberCommand< T >	988
gdcm::FileAnonymizer	450
gdcm::FileChangeTransferSyntax	454
gdcm::FileDecompressLookupTable	458
gdcm::FileStreamer	491
gdcm::Scanner	923
gdcm::ServiceClassUser	978
gdcm::StrictScanner	1032
gdcm::network::ULConnectionManager	1236
gdcm::Surface	1055
gdcm::Value	1266

gdcmm::ByteValue . . . . .	223
gdcmm::SequenceOfFragments . . . . .	953
gdcmm::SequenceOfItems . . . . .	960
gdcmm::Orientation . . . . .	760
gdcmm::Parser . . . . .	776
gdcmm::Patient . . . . .	780
gdcmm::PDBelement . . . . .	784
gdcmm::PDBHeader . . . . .	787
gdcmm::network::PDUFactory . . . . .	793
gdcmm::PersonName . . . . .	796
gdcmm::PhotometricInterpretation . . . . .	802
gdcmm::PixelFormat . . . . .	806
gdcmm::Preamble . . . . .	836
gdcmm::PresentationContext . . . . .	841
gdcmm::network::PresentationContextAC . . . . .	845
gdcmm::PresentationContextGenerator . . . . .	847
gdcmm::network::PresentationContextRQ . . . . .	851
gdcmm::network::PresentationDataValue . . . . .	855
gdcmm::Printer . . . . .	859
gdcmm::DictPrinter . . . . .	373
gdcmm::Dumper . . . . .	393
gdcmm::PrivateDict . . . . .	864
gdcmm::PythonFilter . . . . .	877
gdcmm::QueryBase . . . . .	879
gdcmm::QueryImage . . . . .	884
gdcmm::QueryPatient . . . . .	886
gdcmm::QuerySeries . . . . .	889
gdcmm::QueryStudy . . . . .	891
gdcmm::QueryFactory . . . . .	882
gdcmm::Reader . . . . .	898
gdcmm::PixmapReader . . . . .	821
gdcmm::ImageReader . . . . .	571
gdcmm::ImageRegionReader . . . . .	575
gdcmm::SegmentReader . . . . .	945
gdcmm::SurfaceReader . . . . .	1073
gdcmm::RealWorldValueMappingContent . . . . .	906
gdcmm::Region . . . . .	907
gdcmm::BoxRegion . . . . .	212
gdcmm::Rescaler . . . . .	910
gdcmm::network::RoleSelectionSub . . . . .	921
gdcmm::SerieHelper . . . . .	969
gdcmm::Series . . . . .	975
gdcmm::network::ServiceClassApplicationInformation . . . . .	976
gdcmm::SHA1 . . . . .	986
gdcmm::SimpleSubjectWatcher . . . . .	993
gdcmm::MrProtocol::Slice . . . . .	997
gdcmm::MrProtocol::SliceArray . . . . .	998
gdcmm::SmartPointer< ObjectType > . . . . .	999
gdcmm::SmartPointer< gdcmm::Bitmap > . . . . .	999
gdcmm::SmartPointer< gdcmm::File > . . . . .	999
gdcmm::SmartPointer< gdcmm::Image > . . . . .	999
gdcmm::SmartPointer< gdcmm::MemberCommand > . . . . .	999
gdcmm::SmartPointer< gdcmm::MeshPrimitive > . . . . .	999

gdcm::SmartPointer< gdcm::Pixmap > . . . . .	999
gdcm::SmartPointer< gdcm::SimpleMemberCommand > . . . . .	999
gdcm::SmartPointer< gdcm::Subject > . . . . .	999
gdcm::SmartPointer< LookupTable > . . . . .	999
gdcm::SmartPointer< Segment > . . . . .	999
gdcm::SmartPointer< Surface > . . . . .	999
gdcm::SmartPointer< Value > . . . . .	999
gdcm::network::SOPClassExtendedNegociationSub . . . . .	1003
gdcm::SOPClassUIDToIOD . . . . .	1005
gdcm::Sorter . . . . .	1007
gdcm::IPPSorter . . . . .	602
gdcm::Spacing . . . . .	1012
gdcm::Spectroscopy . . . . .	1014
gdcm::SplitMosaicFilter . . . . .	1015
gdcm::static_assert_test< x > . . . . .	1020
gdcm::STATIC_ASSERTION_FAILURE< x > . . . . .	1020
gdcm::STATIC_ASSERTION_FAILURE< true > . . . . .	1020
gdcm::StreamImageReader . . . . .	1021
gdcm::StreamImageWriter . . . . .	1025
String<'\', 64 > . . . . .	
gdcm::LO . . . . .	651
gdcm::StringFilter . . . . .	1046
gdcm::Study . . . . .	1051
gdcm::SurfaceHelper . . . . .	1070
gdcm::SwapCode . . . . .	1080
gdcm::SwapperDoOp . . . . .	1082
gdcm::SwapperNoOp . . . . .	1083
gdcm::System . . . . .	1084
gdcm::Table . . . . .	1092
gdcm::TableEntry . . . . .	1095
gdcm::TableReader . . . . .	1096
gdcm::XMLDictReader . . . . .	1420
gdcm::XMLPrivateDictReader . . . . .	1427
gdcm::network::TableRow . . . . .	1100
gdcm::Tag . . . . .	1101
gdcm::PrivateTag . . . . .	867
gdcm::TagPath . . . . .	1112
gdcm::Testing . . . . .	1115
gdcm::Trace . . . . .	1123
gdcm::TransferSyntax . . . . .	1128
gdcm::network::TransferSyntaxSub . . . . .	1135
gdcm::network::Transition . . . . .	1137
gdcm::Type . . . . .	1139
gdcm::UI . . . . .	1142
gdcm::UIDGenerator . . . . .	1142
gdcm::UIDs . . . . .	1145
gdcm::network::ULAction . . . . .	1183
gdcm::network::ULActionAA1 . . . . .	1186
gdcm::network::ULActionAA2 . . . . .	1187
gdcm::network::ULActionAA3 . . . . .	1188
gdcm::network::ULActionAA4 . . . . .	1190
gdcm::network::ULActionAA5 . . . . .	1191
gdcm::network::ULActionAA6 . . . . .	1192

gdcmm::network::ULActionAA7	1194
gdcmm::network::ULActionAA8	1195
gdcmm::network::ULActionAE1	1196
gdcmm::network::ULActionAE2	1198
gdcmm::network::ULActionAE3	1199
gdcmm::network::ULActionAE4	1200
gdcmm::network::ULActionAE5	1202
gdcmm::network::ULActionAE6	1203
gdcmm::network::ULActionAE7	1204
gdcmm::network::ULActionAE8	1206
gdcmm::network::ULActionAR1	1207
gdcmm::network::ULActionAR10	1208
gdcmm::network::ULActionAR2	1210
gdcmm::network::ULActionAR3	1211
gdcmm::network::ULActionAR4	1212
gdcmm::network::ULActionAR5	1214
gdcmm::network::ULActionAR6	1215
gdcmm::network::ULActionAR7	1216
gdcmm::network::ULActionAR8	1218
gdcmm::network::ULActionAR9	1219
gdcmm::network::ULActionDT1	1220
gdcmm::network::ULActionDT2	1222
gdcmm::network::ULConnection	1225
gdcmm::network::ULConnectionCallback	1231
gdcmm::network::ULBasicCallback	1223
gdcmm::network::ULWritingCallback	1248
gdcmm::network::ULConnectionInfo	1234
gdcmm::network::ULEvent	1244
gdcmm::network::ULTransitionTable	1246
gdcmm::Unpacker12Bits	1255
gdcmm::Usage	1256
gdcmm::network::UserInformation	1260
gdcmm::UUIDGenerator	1262
gdcmm::Validate	1263
gdcmm::ValueIO< TDE, TSwap, TType >	1269
gdcmm::MrProtocol::Vector3	1270
gdcmm::Version	1270
gdcmm::VL	1273
gdcmm::VM	1277
gdcmm::VMToLength< T >	1282
gdcmm::VR	1282
gdcmm::VRToEncoding< T >	1292
gdcmm::VRToType< T >	1292
gdcmm::VRToType< TagToType< Group, Element >::VRType >	1292
gdcmm::VRToType< TVR >	1292
gdcmm::VRVLSIZE< T >	1293
gdcmm::VRVLSIZE< 0 >	1293
gdcmm::VRVLSIZE< 1 >	1293
vtkImageAlgorithm	
vtkImagePlanarComponentsToComponents	1388
vtkImageMapToColors	
vtkImageMapToWindowLevelColors2	1384
vtkImageWriter	
vtkGDCMImageWriter	1324

vtkLookupTable	
vtkLookupTable16 . . . . .	.1395
vtkMedicalImageProperties	
vtkGDCMMedicalImageProperties . . . . .	.1333
vtkMedicalImageReader2	
vtkGDCMImageReader . . . . .	.1294
vtkGDCMThreadedImageReader . . . . .	.1350
vtkGDCMImageReader2 . . . . .	.1309
vtkObject	
vtkGDCMTesting . . . . .	.1346
vtkImageColorViewer . . . . .	.1363
vtkRTStructSetProperties . . . . .	.1399
vtkPolyDataAlgorithm	
vtkGDCMPolyDataReader . . . . .	.1336
vtkPolyDataWriter	
vtkGDCMPolyDataWriter . . . . .	.1341
vtkThreadedImageAlgorithm	
vtkGDCMThreadedImageReader2 . . . . .	.1354
vtkImageMapToColors16 . . . . .	.1378
vtkImageRGBToYBR . . . . .	.1390
vtkImageYBRToRGB . . . . .	.1393
gdcm::Waveform . . . . .	.1410
gdcm::Writer . . . . .	.1414
gdcm::PixmapWriter . . . . .	.828
gdcm::ImageWriter . . . . .	.582
gdcm::SegmentWriter . . . . .	.949
gdcm::SurfaceWriter . . . . .	.1077
gdcm::XMLPrinter . . . . .	.1423





## Chapter 7

# Class Index

### 7.1 Class List

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

<a href="#">gdcmm::network::AAabortPDU</a>	
<a href="#">AAabortPDU</a>	87
<a href="#">gdcmm::network::AAssociateACPDU</a>	
<a href="#">AAssociateACPDU</a>	90
<a href="#">gdcmm::network::AAssociateRJPDU</a>	
<a href="#">AAssociateRJPDU</a>	94
<a href="#">gdcmm::network::AAssociateRQPDU</a>	
<a href="#">AAssociateRQPDU</a>	97
<a href="#">gdcmm::AbortEvent</a>	103
<a href="#">gdcmm::network::AbstractSyntax</a>	
<a href="#">AbstractSyntax</a>	104
<a href="#">gdcmm::AnonymizeEvent</a>	
<a href="#">AnonymizeEvent</a>	107
<a href="#">gdcmm::Anonymizer</a>	
<a href="#">Anonymizer</a>	110
<a href="#">gdcmm::AnyEvent</a>	118
<a href="#">gdcmm::network::ApplicationContext</a>	
<a href="#">ApplicationContext</a>	119
<a href="#">gdcmm::ApplicationEntity</a>	
<a href="#">ApplicationEntity</a>	121
<a href="#">gdcmm::network::AReleaseRPPDU</a>	
<a href="#">AReleaseRPPDU</a>	124
<a href="#">gdcmm::network::AReleaseRQPDU</a>	
<a href="#">AReleaseRQPDU</a>	126
<a href="#">gdcmm::network::ARTIMTimer</a>	
<a href="#">ARTIMTimer</a>	129
<a href="#">gdcmm::ASN1</a>	
Class for <a href="#">ASN1</a>	130
<a href="#">gdcmm::network::AsynchronousOperationsWindowSub</a>	
<a href="#">AsynchronousOperationsWindowSub</a>	132

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	134
gdcmm::Attribute< Group, Element, TVR, VM::VM1 >	144
gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >	151
gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >	152
gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >	153
gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >	160
gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >	161
gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >	163
gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >	164
gdcmm::AudioCodec	
AudioCodec	166
gdcmm::Base64	
Class for Base64	168
gdcmm::network::BaseCompositeMessage	
BaseCompositeMessage	171
gdcmm::network::BaseNormalizedMessage	
BaseNormalizedMessage	173
gdcmm::network::BasePDU	
BasePDU	175
gdcmm::BaseQuery	
BaseQuery	177
gdcmm::BaseRootQuery	
BaseRootQuery	183
gdcmm::SegmentHelper::BasicCodedEntry	
This structure defines a basic coded entry with all of its attributes	188
gdcmm::BasicOffsetTable	
Class to represent a BasicOffsetTable	191
gdcmm::Bitmap	
Bitmap class	194
gdcmm::BitmapToBitmapFilter	
BitmapToBitmapFilter class	209
gdcmm::BoxRegion	
Class for manipulation box region	212
gdcmm::ByteBuffer	
ByteBuffer	217
gdcmm::ByteSwap< T >	
ByteSwap	219
gdcmm::ByteSwapFilter	
ByteSwapFilter	221
gdcmm::ByteValue	
Class to represent binary value (array of bytes)	223
gdcmm::CAPICryptoFactory	232
gdcmm::CAPICryptographicMessageSyntax	233
gdcmm::network::CEchoRQ	
CEchoRQ	237
gdcmm::network::CEchoRSP	
CEchoRSP this file defines the messages for the cecho action	239
gdcmm::network::CFind	240
gdcmm::network::CFindCancelRQ	
CFindCancelRQ this file defines the messages for the cfind action	240
gdcmm::network::CFindRQ	
CFindRQ	242

gdcm::network::CFindRSP	
CFindRSP this file defines the messages for the cfind action	243
gdcm::network::CMoveCancelRq	245
gdcm::network::CMoveRQ	
CMoveRQ	246
gdcm::network::CMoveRSP	
CMoveRSP this file defines the messages for the cmove action	247
gdcm::Codec	
Codec class	249
gdcm::Coder	
Coder	250
gdcm::CodeString	
CodeString	252
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	263
gdcm::CompositeNetworkFunctions	
Composite Network Functions	264
gdcm::ConstCharWrapper	
Do not use me	269
gdcm::CP246ExplicitDataElement	
Class to read/write a <a href="#">DataElement</a> as CP246Explicit Data <a href="#">Element</a>	270
gdcm::CryptoFactory	
Class to do handle the crypto factory	272
gdcm::CryptographicMessageSyntax	275
gdcm::CSAElement	
Class to represent a CSA <a href="#">Element</a>	278
gdcm::CSAHeader	
Class for <a href="#">CSAHeader</a>	286
gdcm::CSAHeaderDict	
Class to represent a map of <a href="#">CSAHeaderDictEntry</a>	292
gdcm::CSAHeaderDictEntry	
Class to represent an Entry in the <a href="#">Dict</a>	295
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	309
gdcm::DataElementException	323
gdcm::DataEvent	
DataEvent	323
gdcm::DataSet	
Class to represent a Data Set (which contains Data Elements)	327
gdcm::DataSetEvent	
DataSetEvent	340
gdcm::DataSetHelper	
DataSetHelper (internal class, not intended for user level)	344

gdcm::Decoder	
Decoder	345
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 <a href="#">Element</a> with Defined Terms that does not contain a <a href="#">Value</a> equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation <a href="#">Type</a> ID (4008,0210) is an example of a Data <a href="#">Element</a> having Defined Terms. It is defined to have a <a href="#">Value</a> that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data <a href="#">Element</a> has Defined Terms other Interpretation <a href="#">Type</a> IDs may be defined by the implementor	347
gdcm::Defs	
FIXME I do not like the name 'Defs'	347
gdcm::DeltaEncodingCodec	
DeltaEncodingCodec compression used by some private vendor	352
gdcm::DICOMDIR	
DICOMDIR class	354
gdcm::DICOMDIRGenerator	
DICOMDIRGenerator class	355
gdcm::Dict	
Class to represent a map of <a href="#">DictEntry</a>	359
gdcm::DictConverter	
Class to convert a .dic file into something else:	363
gdcm::DictEntry	
Class to represent an Entry in the <a href="#">Dict</a>	368
gdcm::DictPrinter	
DictPrinter class	373
gdcm::Dicts	
Class to manipulate the sum of knowledge (all the dict user load)	376
gdcm::network::DIMSE	
DIMSE	380
gdcm::DirectionCosines	
Class to handle <a href="#">DirectionCosines</a>	382
gdcm::Directory	
Class for manipulation directories	385
gdcm::DirectoryHelper	
DirectoryHelper	390
gdcm::DummyValueGenerator	
Class for generating dummy value	392
gdcm::Dumper	
Codec class	393
gdcm::Element< TVR, TVM >	
Element class	395
gdcm::Element< TVR, VM::VM1_2 >	401
gdcm::Element< TVR, VM::VM1_n >	403
gdcm::Element< TVR, VM::VM2_2n >	408
gdcm::Element< TVR, VM::VM2_n >	410
gdcm::Element< TVR, VM::VM3_3n >	412
gdcm::Element< TVR, VM::VM3_n >	414
gdcm::Element< VR::AS, VM::VM5 >	416
gdcm::Element< VR::OB, VM::VM1 >	417
gdcm::Element< VR::OW, VM::VM1 >	418

gdcm::ElementDisableCombinations< TVR, TVM >	
A class which is used to produce compile errors for an invalid combination of template parameters	420
gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n >	421
gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n >	421
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	421
gdcm::EncapsulatedDocument	
EncapsulatedDocument	424
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	430
gdcm::Event	
Superclass for callback/observer methods	432
gdcm::Exception	
Exception	436
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>	442
gdcm::Fiducials	
Fiducials	445
gdcm::File	
DICOM File	445
gdcm::FileAnonymizer	
FileAnonymizer	450
gdcm::FileChangeTransferSyntax	
FileChangeTransferSyntax	454
gdcm::FileDecompressLookupTable	
FileDecompressLookupTable class	458
gdcm::FileDerivation	
FileDerivation class	461
gdcm::FileExplicitFilter	
FileExplicitFilter class	465
gdcm::FileMetaInformation	
Class to represent a <a href="#">File</a> Meta Information	468
gdcm::Filename	
Class to manipulate file name's	477
gdcm::FileNameEvent	
FileNameEvent	481
gdcm::FilenameGenerator	
FilenameGenerator	485
gdcm::FileSet	489
gdcm::FileStreamer	
FileStreamer	491

gdcm::FileWithName	
FileWithName	496
gdcm::FindPatientRootQuery	
PatientRootQuery	498
gdcm::FindStudyRootQuery	
FindStudyRootQuery	501
gdcm::Fragment	
Class to represent a <a href="#">Fragment</a>	504
gdcm::Global	
Global	507
gdcm::GroupDict	
Class to represent the mapping from group number to its abbreviation and name	511
gdcm::IconImageFilter	
IconImageFilter	514
gdcm::IconImageGenerator	
IconImageGenerator	517
gdcm::ignore_char	521
gdcm::Image	
Image	522
gdcm::ImageApplyLookupTable	
ImageApplyLookupTable class	529
gdcm::ImageChangePhotometricInterpretation	
ImageChangePhotometricInterpretation class	532
gdcm::ImageChangePlanarConfiguration	
ImageChangePlanarConfiguration class	536
gdcm::ImageChangeTransferSyntax	
ImageChangeTransferSyntax class	540
gdcm::ImageCodec	
ImageCodec	545
gdcm::ImageConverter	
Image Converter	559
gdcm::ImageFragmentSplitter	
ImageFragmentSplitter class	561
gdcm::ImageHelper	
ImageHelper (internal class, not intended for user level)	564
gdcm::ImageReader	
ImageReader	571
gdcm::ImageRegionReader	
ImageRegionReader	575
gdcm::ImageToImageFilter	
ImageToImageFilter class	579
gdcm::ImageWriter	
ImageWriter	582
gdcm::network::ImplementationClassUIDSub	
ImplementationClassUIDSub	586
gdcm::network::ImplementationUIDSub	
ImplementationUIDSub	587
gdcm::network::ImplementationVersionNameSub	
ImplementationVersionNameSub	588
gdcm::ImplicitDataElement	
Class to represent an <i>Implicit VR</i> <a href="#">Data Element</a>	590
gdcm::InitializeEvent	593
gdcm::IOD	
Class for representing a <a href="#">IOD</a>	594

<a href="#">gdcm::IODEntry</a>	
Class for representing a <a href="#">IODEntry</a>	596
<a href="#">gdcm::IODs</a>	
Class for representing a <a href="#">IODs</a>	599
<a href="#">gdcm::IPPSorter</a>	
<a href="#">IPPSorter</a>	602
<a href="#">gdcm::Item</a>	
Class to represent an <a href="#">Item</a>	607
<a href="#">gdcm::IterationEvent</a>	612
<a href="#">gdcm::JPEG12Codec</a>	
Class to do JPEG 12bits (lossy & lossless)	613
<a href="#">gdcm::JPEG16Codec</a>	
Class to do JPEG 16bits (lossless)	616
<a href="#">gdcm::JPEG2000Codec</a>	
Class to do JPEG 2000	619
<a href="#">gdcm::JPEG8Codec</a>	
Class to do JPEG 8bits (lossy & lossless)	626
<a href="#">gdcm::JPEGCodec</a>	
JPEG codec	630
<a href="#">gdcm::JPEGLSCodec</a>	
JPEG-LS	638
<a href="#">gdcm::JSON</a>	645
<a href="#">gdcm::KAKADUCodec</a>	
<a href="#">KAKADUCodec</a>	647
<a href="#">gdcm::LO</a>	
LO	651
<a href="#">gdcm::LookupTable</a>	
LookupTable class	655
<a href="#">gdcm::Scanner::ltstr</a>	663
<a href="#">gdcm::StrictScanner::ltstr</a>	664
<a href="#">gdcm::Macro</a>	
Class for representing a <a href="#">Macro</a>	664
<a href="#">gdcm::Macros</a>	
Class for representing a <a href="#">Modules</a>	667
<a href="#">gdcm::network::MaximumLengthSub</a>	
MaximumLengthSub	669
<a href="#">gdcm::MD5</a>	
Class for MD5	671
<a href="#">gdcm::MediaStorage</a>	
MediaStorage	672
<a href="#">gdcm::MemberCommand&lt; T &gt;</a>	
Command subclass that calls a pointer to a member function	683
<a href="#">gdcm::MeshPrimitive</a>	
This class defines surface mesh primitives	689
<a href="#">gdcm::ModalityPerformedProcedureStepCreateQuery</a>	
ModalityPerformedProcedureStepCreateQuery	695
<a href="#">gdcm::ModalityPerformedProcedureStepSetQuery</a>	
ModalityPerformedProcedureStepSetQuery	698
<a href="#">gdcm::ModifiedEvent</a>	701
<a href="#">gdcm::Module</a>	
Class for representing a <a href="#">Module</a>	702
<a href="#">gdcm::ModuleEntry</a>	
Class for representing a <a href="#">ModuleEntry</a>	705

gdcmm::Modules	
Class for representing a <a href="#">Modules</a>	709
gdcmm::MovePatientRootQuery	
MovePatientRootQuery	712
gdcmm::MoveStudyRootQuery	
MoveStudyRootQuery	715
gdcmm::MrProtocol	
Class for <a href="#">MrProtocol</a>	717
gdcmm::network::NActionRQ	
NActionRQ	720
gdcmm::network::NActionRSP	
NActionRSP this file defines the messages for the NAction action	721
gdcmm::network::NCreateRQ	
NCreateRQ	723
gdcmm::network::NCreateRSP	
NCreateRSP this file defines the messages for the ncreate action	724
gdcmm::network::NDeleteRQ	
NDeleteRQ	726
gdcmm::network::NDeleteRSP	
NDeleteRSP this file defines the messages for the ndelete action	727
gdcmm::NestedModuleEntries	
Class for representing a <a href="#">NestedModuleEntries</a>	729
gdcmm::network::NEventReportRQ	
NEventReportRQ	732
gdcmm::network::NEventReportRSP	
NEventReportRSP this file defines the messages for the neventreport action	733
gdcmm::network::NGetRQ	
NGetRQ	735
gdcmm::network::NGetRSP	
NGetRSP this file defines the messages for the nget action	736
gdcmm::NoEvent	738
gdcmm::network::NormalizedMessageFactory	739
gdcmm::NormalizedNetworkFunctions	
Normalized Network Functions	740
gdcmm::network::NSetRQ	
NSetRQ	743
gdcmm::network::NSetRSP	
NSetRSP this file defines the messages for the nset action	745
gdcmm::Object	
Object	746
gdcmm::OpenSSLCryptoFactory	750
gdcmm::OpenSSLCryptographicMessageSyntax	752
gdcmm::OpenSSLP7CryptoFactory	755
gdcmm::OpenSSLP7CryptographicMessageSyntax	757
gdcmm::Orientation	
Class to handle <a href="#">Orientation</a>	760
gdcmm::Overlay	
Overlay class	763
gdcmm::ParseException	
ParseException Standard exception handling object	774
gdcmm::Parser	
Parser ala XML_Parser from expat (SAX)	776
gdcmm::Patient	
See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54	780



gdcm::network::PDataTFPDU	
PDataTFPDU	781
gdcm::PDBElement	
Class to represent a PDB <a href="#">Element</a>	784
gdcm::PDBHeader	
Class for <a href="#">PDBHeader</a>	787
gdcm::PDFCodec	
PDFCodec class	790
gdcm::network::PDUFactory	
PDUFactory basically, given an initial byte, construct the	793
gdcm::PersonName	
PersonName class	796
gdcm::PGXCodec	
Class to do PGX	799
gdcm::PhotometricInterpretation	
Class to represent an <a href="#">PhotometricInterpretation</a>	802
gdcm::PixelFormat	
PixelFormat	806
gdcm::Pixmap	
Pixmap class	815
gdcm::PixmapReader	
PixmapReader	821
gdcm::PixmapToPixmapFilter	
PixmapToPixmapFilter class	825
gdcm::PixmapWriter	
PixmapWriter	828
gdcm::PNMCodec	
Class to do PNM	833
gdcm::Preamble	
DICOM <a href="#">Preamble</a> (Part 10)	836
gdcm::PresentationContext	
PresentationContext	841
gdcm::network::PresentationContextAC	
PresentationContextAC	845
gdcm::PresentationContextGenerator	
PresentationContextGenerator	847
gdcm::network::PresentationContextRQ	
PresentationContextRQ	851
gdcm::network::PresentationDataValue	
PresentationDataValue	855
gdcm::Printer	
Printer class	859
gdcm::PrivateDict	
Private <a href="#">Dict</a>	864
gdcm::PrivateTag	
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)	867
gdcm::ProgressEvent	
ProgressEvent	870
gdcm::PVRGCodec	
PVRGCodec	874
gdcm::PythonFilter	
PythonFilter <a href="#">PythonFilter</a> is the class that make gdcm2.x looks more like gdcm1 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	877

gdcmm::QueryBase	
QueryBase	879
gdcmm::QueryFactory	
QueryFactory.h	882
gdcmm::QueryImage	
QueryImage	884
gdcmm::QueryPatient	
QueryPatient	886
gdcmm::QuerySeries	
QuerySeries	889
gdcmm::QueryStudy	
QueryStudy.h	891
gdcmm::RAWCodec	
RAWCodec class	894
gdcmm::Reader	
Reader ala DOM (Document Object Model)	898
gdcmm::RealWorldValueMappingContent	906
gdcmm::Region	
Class for manipulation region	907
gdcmm::Rescaler	
Rescale class	910
gdcmm::RLECodec	
Class to do RLE	915
gdcmm::network::RoleSelectionSub	
RoleSelectionSub	921
gdcmm::Scanner	
Scanner	923
gdcmm::Segment	
This class defines a segment	932
gdcmm::SegmentedPaletteColorLookupTable	
SegmentedPaletteColorLookupTable class	942
gdcmm::SegmentReader	
This class defines a segment reader	945
gdcmm::SegmentWriter	
This class defines a segment writer	949
gdcmm::SequenceOfFragments	
Class to represent a Sequence Of Fragments	953
gdcmm::SequenceOfItems	
Class to represent a Sequence Of Items	960
gdcmm::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 disapear soon, you've been warned	969
gdcmm::Series	
Series	975
gdcmm::network::ServiceClassApplicationInformation	976
gdcmm::ServiceClassUser	
ServiceClassUser	978
gdcmm::SHA1	
Class for SHA1	986
gdcmm::SimpleMemberCommand< T >	
Command subclass that calls a pointer to a member function	988
gdcmm::SimpleSubjectWatcher	
SimpleSubjectWatcher	993
gdcmm::MrProtocol::Slice	997

gdcm::MrProtocol::SliceArray	998
gdcm::SmartPointer< ObjectType >	
Class for Smart Pointer	999
gdcm::network::SOPClassExtendedNegociationSub	
SOPClassExtendedNegociationSub	1003
gdcm::SOPClassUIDToIOD	
Class convert a class SOP Class UID into IOD	1005
gdcm::Sorter	
Sorter	1007
gdcm::Spacing	
Class for Spacing	1012
gdcm::Spectroscopy	
Spectroscopy class	1014
gdcm::SplitMosaicFilter	
SplitMosaicFilter class	1015
gdcm::StartEvent	1019
gdcm::static_assert_test< x >	1020
gdcm::STATIC_ASSERTION_FAILURE< x >	1020
gdcm::STATIC_ASSERTION_FAILURE< true >	1020
gdcm::StreamImageReader	
StreamImageReader	1021
gdcm::StreamImageWriter	
StreamImageReader	1025
gdcm::StrictScanner	
StrictScanner	1032
gdcm::String< TDelimiter, TMaxLength, TPadChar >	
String	1041
gdcm::StringFilter	
StringFilter	1046
gdcm::Study	
Study	1051
gdcm::Subject	
Subject	1051
gdcm::Surface	
This class defines a SURFACE IE	1055
gdcm::SurfaceHelper	
SurfaceHelper	1070
gdcm::SurfaceReader	
This class defines a SURFACE IE reader	1073
gdcm::SurfaceWriter	
This class defines a SURFACE IE writer	1077
gdcm::SwapCode	
SwapCode representation	1080
gdcm::SwapperDoOp	1082
gdcm::SwapperNoOp	1083
gdcm::System	
Class to do system operation	1084
gdcm::Table	
Table	1092
gdcm::TableEntry	
TableEntry	1095
gdcm::TableReader	
Class for representing a TableReader	1096
gdcm::network::TableRow	1100

<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> )	1101
<a href="#">gdcmm::TagPath</a>	
Class to handle a path of tag	1112
<a href="#">gdcmm::Testing</a>	
Class for testing	1115
<a href="#">gdcmm::Trace</a>	
Trace	1123
<a href="#">gdcmm::TransferSyntax</a>	
Class to manipulate Transfer Syntax	1128
<a href="#">gdcmm::network::TransferSyntaxSub</a>	
TransferSyntaxSub	1135
<a href="#">gdcmm::network::Transition</a>	1137
<a href="#">gdcmm::Type</a>	
Type	1139
<a href="#">gdcmm::UI</a>	1142
<a href="#">gdcmm::UIDGenerator</a>	
Class for generating unique UID	1142
<a href="#">gdcmm::UIDs</a>	
All known uids	1145
<a href="#">gdcmm::network::ULAction</a>	
ULAction	1183
<a href="#">gdcmm::network::ULActionAA1</a>	1186
<a href="#">gdcmm::network::ULActionAA2</a>	1187
<a href="#">gdcmm::network::ULActionAA3</a>	1188
<a href="#">gdcmm::network::ULActionAA4</a>	1190
<a href="#">gdcmm::network::ULActionAA5</a>	1191
<a href="#">gdcmm::network::ULActionAA6</a>	1192
<a href="#">gdcmm::network::ULActionAA7</a>	1194
<a href="#">gdcmm::network::ULActionAA8</a>	1195
<a href="#">gdcmm::network::ULActionAE1</a>	1196
<a href="#">gdcmm::network::ULActionAE2</a>	1198
<a href="#">gdcmm::network::ULActionAE3</a>	1199
<a href="#">gdcmm::network::ULActionAE4</a>	1200
<a href="#">gdcmm::network::ULActionAE5</a>	1202
<a href="#">gdcmm::network::ULActionAE6</a>	1203
<a href="#">gdcmm::network::ULActionAE7</a>	1204
<a href="#">gdcmm::network::ULActionAE8</a>	1206
<a href="#">gdcmm::network::ULActionAR1</a>	1207
<a href="#">gdcmm::network::ULActionAR10</a>	1208
<a href="#">gdcmm::network::ULActionAR2</a>	1210
<a href="#">gdcmm::network::ULActionAR3</a>	1211
<a href="#">gdcmm::network::ULActionAR4</a>	1212
<a href="#">gdcmm::network::ULActionAR5</a>	1214
<a href="#">gdcmm::network::ULActionAR6</a>	1215
<a href="#">gdcmm::network::ULActionAR7</a>	1216
<a href="#">gdcmm::network::ULActionAR8</a>	1218
<a href="#">gdcmm::network::ULActionAR9</a>	1219
<a href="#">gdcmm::network::ULActionDT1</a>	1220
<a href="#">gdcmm::network::ULActionDT2</a>	1222
<a href="#">gdcmm::network::ULBasicCallback</a>	
ULBasicCallback	1223
<a href="#">gdcmm::network::ULConnection</a>	
ULConnection	1225

gdcm::network::ULConnectionCallback	1231
gdcm::network::ULConnectionInfo	
ULConnectionInfo	1234
gdcm::network::ULConnectionManager	
ULConnectionManager	1236
gdcm::network::ULEvent	
ULEvent	1244
gdcm::network::ULTransitionTable	
ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates	1246
gdcm::network::ULWritingCallback	1248
gdcm::UNExplicitDataElement	
Class to read/write a <a href="#">DataElement</a> as UNExplicit Data <a href="#">Element</a>	1250
gdcm::UNExplicitImplicitDataElement	
Class to read/write a <a href="#">DataElement</a> as ExplicitImplicit Data <a href="#">Element</a>	1253
gdcm::Unpacker12Bits	
Pack/Unpack 12 bits pixel into 16bits	1255
gdcm::Usage	
Usage	1256
gdcm::UserEvent	1259
gdcm::network::UserInformation	
UserInformation	1260
gdcm::UUIDGenerator	
Class for generating unique UUID	1262
gdcm::Validate	
Validate class	1263
gdcm::Value	
Class to represent the value of a Data <a href="#">Element</a>	1266
gdcm::ValueIO< TDE, TSwap, TType >	
Class to dispatch template calls	1269
gdcm::MrProtocol::Vector3	1270
gdcm::Version	
Major/minor and build version	1270
gdcm::VL	
Value Length	1273
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	1277
gdcm::VMToLength< T >	1282
gdcm::VR	
VR class	1282
gdcm::VR16ExplicitDataElement	
Class to read/write a <a href="#">DataElement</a> as Explicit Data <a href="#">Element</a>	1290
gdcm::VRToEncoding< T >	1292
gdcm::VRToType< T >	1292
gdcm::VRVLSize< T >	1293
gdcm::VRVLSize< 0 >	1293
gdcm::VRVLSize< 1 >	1293
vtkGDCMImageReader	1294
vtkGDCMImageReader2	1309
vtkGDCMImageWriter	1324
vtkGDCMMedicalImageProperties	1333
vtkGDCMPolyDataReader	1336
vtkGDCMPolyDataWriter	1341
vtkGDCMTesting	1346

<a href="#">vtkGDCMThreadedImageReader</a>	1350
<a href="#">vtkGDCMThreadedImageReader2</a>	1354
<a href="#">vtkImageColorViewer</a>	1363
<a href="#">vtkImageMapToColors16</a>	1378
<a href="#">vtkImageMapToWindowLevelColors2</a>	1384
<a href="#">vtkImagePlanarComponentsToComponents</a>	1388
<a href="#">vtkImageRGBToYBR</a>	1390
<a href="#">vtkImageYBRToRGB</a>	1393
<a href="#">vtkLookupTable16</a>	1395
<a href="#">vtkRTStructSetProperties</a>	1399
<a href="#">gdcm::Waveform</a>	
<a href="#">Waveform</a> class	1410
<a href="#">gdcm::WLMFindQuery</a>	
<a href="#">PatientRootQuery</a>	1411
<a href="#">gdcm::Writer</a>	
<a href="#">Writer</a> ala DOM (Document <a href="#">Object</a> Model)	1414
<a href="#">gdcm::XMLDictReader</a>	
Class for representing a <a href="#">XMLDictReader</a>	1420
<a href="#">gdcm::XMLPrinter</a>	1423
<a href="#">gdcm::XMLPrivateDictReader</a>	
Class for representing a <a href="#">XMLPrivateDictReader</a>	1427

## Chapter 8

# File Index

### 8.1 File List

Here is a list of all files with brief descriptions:

<a href="#">gdcmAAbortPDU.h</a>	1431
<a href="#">gdcmAAssociateACPDU.h</a>	1432
<a href="#">gdcmAAssociateRJPDU.h</a>	1432
<a href="#">gdcmAAssociateRQPDU.h</a>	1433
<a href="#">gdcmAbstractSyntax.h</a>	1434
<a href="#">gdcmAnonymizeEvent.h</a>	1435
<a href="#">gdcmAnonymizer.h</a>	1436
<a href="#">gdcmApplicationContext.h</a>	1437
<a href="#">gdcmApplicationEntity.h</a>	1438
<a href="#">gdcmAReleaseRPPDU.h</a>	1438
<a href="#">gdcmAReleaseRQPDU.h</a>	1439
<a href="#">gdcmARTIMTimer.h</a>	1440
<a href="#">gdcmASN1.h</a>	1441
<a href="#">gdcmAsynchronousOperationsWindowSub.h</a>	1442
<a href="#">gdcmAttribute.h</a>	1442
<a href="#">gdcmAudioCodec.h</a>	1444
<a href="#">gdcmBase64.h</a>	1445
<a href="#">gdcmBaseCompositeMessage.h</a>	1445
<a href="#">gdcmBaseNormalizedMessage.h</a>	1446
<a href="#">gdcmBasePDU.h</a>	1447
<a href="#">gdcmBaseQuery.h</a>	1448
<a href="#">gdcmBaseRootQuery.h</a>	1449
<a href="#">gdcmBasicOffsetTable.h</a>	1450
<a href="#">gdcmBitmap.h</a>	1451
<a href="#">gdcmBitmapToBitmapFilter.h</a>	1452
<a href="#">gdcmBoxRegion.h</a>	1453
<a href="#">gdcmByteBuffer.h</a>	1453
<a href="#">gdcmByteSwap.h</a>	1455
<a href="#">gdcmByteSwapFilter.h</a>	1455
<a href="#">gdcmByteValue.h</a>	1456
<a href="#">gdcmCAPICryptoFactory.h</a>	1457

gdcmCAPICryptographicMessageSyntax.h	1458
gdcmCEchoMessages.h	1458
gdcmCFindMessages.h	1459
gdcmCMoveMessages.h	1460
gdcmCodec.h	1461
gdcmCoder.h	1462
gdcmCodeString.h	1464
gdcmCommand.h	1465
gdcmCommandDataSet.h	1466
gdcmCompositeMessageFactory.h	1467
gdcmCompositeNetworkFunctions.h	1467
gdcmConstCharWrapper.h	1468
gdcmCP246ExplicitDataElement.h	1468
gdcmCryptoFactory.h	1469
gdcmCryptographicMessageSyntax.h	1470
gdcmCSAElement.h	1471
gdcmCSAHeader.h	1472
gdcmCSAHeaderDict.h	1473
gdcmCSAHeaderDictEntry.h	1474
gdcmCStoreMessages.h	1475
gdcmCurve.h	1476
gdcmDataElement.h	1477
gdcmDataEvent.h	1479
gdcmDataSet.h	1480
gdcmDataSetEvent.h	1481
gdcmDataSetHelper.h	1481
gdcmDecoder.h	1482
gdcmDefinedTerms.h	1484
gdcmDeflateStream.h	1484
gdcmDefs.h	1485
gdcmDeltaEncodingCodec.h	1486
gdcmDICOMDIR.h	1487
gdcmDICOMDIRGenerator.h	1488
gdcmDict.h	1488
gdcmDictConverter.h	1490
gdcmDictEntry.h	1490
gdcmDictPrinter.h	1492
gdcmDicts.h	1492
gdcmDIMSE.h	1494
gdcmDirectionCosines.h	1494
gdcmDirectory.h	1495
gdcmDirectoryHelper.h	1496
gdcmDummyValueGenerator.h	1497
gdcmDumper.h	1498
gdcmElement.h	1498
gdcmEmptyMaskGenerator.h	1500
gdcmEncapsulatedDocument.h	1501
gdcmEnumeratedValues.h	1501
gdcmEquipmentManufacturer.h	1502
gdcmEvent.h	1503
gdcmException.h	1504
gdcmExplicitDataElement.h	1505
gdcmExplicitImplicitDataElement.h	1506
gdcmFiducials.h	1507



gdcmFile.h	1507
gdcmFileAnonymizer.h	1509
gdcmFileChangeTransferSyntax.h	1509
gdcmFileDecompressLookupTable.h	1510
gdcmFileDerivation.h	1511
gdcmFileExplicitFilter.h	1512
gdcmFileMetaInformation.h	1513
gdcmFilename.h	1514
gdcmFileNameEvent.h	1514
gdcmFilenameGenerator.h	1515
gdcmFileSet.h	1516
gdcmFileStreamer.h	1517
gdcmFindPatientRootQuery.h	1518
gdcmFindStudyRootQuery.h	1519
gdcmFragment.h	1519
gdcmGlobal.h	1521
gdcmGroupDict.h	1522
gdcmIconImage.h	1522
gdcmIconImageFilter.h	1523
gdcmIconImageGenerator.h	1524
gdcmImage.h	1525
gdcmImageApplyLookupTable.h	1526
gdcmImageChangePhotometricInterpretation.h	1527
gdcmImageChangePlanarConfiguration.h	1528
gdcmImageChangeTransferSyntax.h	1528
gdcmImageCodec.h	1529
gdcmImageConverter.h	1530
gdcmImageFragmentSplitter.h	1531
gdcmImageHelper.h	1532
gdcmImageReader.h	1533
gdcmImageRegionReader.h	1534
gdcmImageToImageFilter.h	1534
gdcmImageWriter.h	1535
gdcmImplementationClassUIDSub.h	1536
gdcmImplementationUIDSub.h	1537
gdcmImplementationVersionNameSub.h	1538
gdcmImplicitDataElement.h	1539
gdcmIOD.h	1540
gdcmIODEntry.h	1541
gdcmIODs.h	1543
gdcmIPPSorter.h	1544
gdcmItem.h	1545
gdcmJPEG12Codec.h	1547
gdcmJPEG16Codec.h	1547
gdcmJPEG2000Codec.h	1548
gdcmJPEG8Codec.h	1549
gdcmJPEGCodec.h	1550
gdcmJPEGLSCodec.h	1551
gdcmJSON.h	1551
gdcmKAKADUCodec.h	1552
gdcmLegacyMacro.h	1553
gdcmLO.h	1554
gdcmLookupTable.h	1555
gdcmMacro.h	1556

gdcmMacroEntry.h	1558
gdcmMacros.h	1560
gdcmMaximumLengthSub.h	1561
gdcmMD5.h	1563
gdcmMediaStorage.h	1563
gdcmMeshPrimitive.h	1565
gdcmModalityPerformedProcedureStepCreateQuery.h	1566
gdcmModalityPerformedProcedureStepSetQuery.h	1567
gdcmModule.h	1567
gdcmModuleEntry.h	1569
gdcmModules.h	1571
gdcmMovePatientRootQuery.h	1572
gdcmMoveStudyRootQuery.h	1573
gdcmMrProtocol.h	1573
gdcmNActionMessages.h	1575
gdcmNCreateMessages.h	1575
gdcmNDeleteMessages.h	1576
gdcmNestedModuleEntries.h	1577
gdcmNetworkEvents.h	1578
gdcmNetworkStateID.h	1579
gdcmNEventReportMessages.h	1580
gdcmNGetMessages.h	1581
gdcmNormalizedMessageFactory.h	1581
gdcmNormalizedNetworkFunctions.h	1582
gdcmNSetMessages.h	1583
gdcmObject.h	1583
gdcmOpenSSLCryptoFactory.h	1585
gdcmOpenSSLCryptographicMessageSyntax.h	1585
gdcmOpenSSLP7CryptoFactory.h	1587
gdcmOpenSSLP7CryptographicMessageSyntax.h	1587
gdcmOrientation.h	1589
gdcmOverlay.h	1589
gdcmParseException.h	1590
gdcmParser.h	1592
gdcmPatient.h	1592
gdcmPDataTFPDU.h	1593
gdcmPDBElement.h	1594
gdcmPDBHeader.h	1595
gdcmPDFCodec.h	1596
gdcmPDUFactory.h	1597
gdcmPersonName.h	1597
gdcmPGXCodec.h	1598
gdcmPhotometricInterpretation.h	1599
gdcmPixelFormat.h	1600
gdcmPixmap.h	1601
gdcmPixmapReader.h	1602
gdcmPixmapToPixmapFilter.h	1603
gdcmPixmapWriter.h	1603
gdcmPNMCodec.h	1605
gdcmPreamble.h	1605
gdcmPresentationContext.h	1607
gdcmPresentationContextAC.h	1608
gdcmPresentationContextGenerator.h	1609
gdcmPresentationContextRQ.h	1609

gdcmPresentationDataValue.h	1610
gdcmPrinter.h	1611
gdcmPrivateTag.h	1613
gdcmProgressEvent.h	1614
gdcmPVRGCodec.h	1615
gdcmPythonFilter.h	1615
gdcmQueryBase.h	1616
gdcmQueryFactory.h	1617
gdcmQueryImage.h	1618
gdcmQueryPatient.h	1619
gdcmQuerySeries.h	1620
gdcmQueryStudy.h	1621
gdcmRAWCodec.h	1622
gdcmReader.h	1622
gdcmRegion.h	1624
gdcmRescaler.h	1625
gdcmRLECodec.h	1626
gdcmRoleSelectionSub.h	1626
gdcmScanner.h	1627
gdcmSegment.h	1628
gdcmSegmentedPaletteColorLookupTable.h	1629
gdcmSegmentHelper.h	1630
gdcmSegmentReader.h	1631
gdcmSegmentWriter.h	1632
gdcmSequenceOfFragments.h	1633
gdcmSequenceOfItems.h	1634
gdcmSerieHelper.h	1634
gdcmSeries.h	1636
gdcmServiceClassApplicationInformation.h	1637
gdcmServiceClassUser.h	1638
gdcmSHA1.h	1638
gdcmSimpleSubjectWatcher.h	1639
gdcmSmartPointer.h	1640
gdcmSOPClassExtendedNegociationSub.h	1641
gdcmSOPClassUIDToIOD.h	1642
gdcmSorter.h	1643
gdcmSpacing.h	1645
gdcmSpectroscopy.h	1645
gdcmSplitMosaicFilter.h	1646
gdcmStaticAssert.h	1647
gdcmStreamImageReader.h	1649
gdcmStreamImageWriter.h	1649
gdcmStrictScanner.h	1650
gdcmString.h	1651
gdcmStringFilter.h	1652
gdcmStudy.h	1653
gdcmSubject.h	1654
gdcmSurface.h	1655
gdcmSurfaceHelper.h	1656
gdcmSurfaceReader.h	1657
gdcmSurfaceWriter.h	1658
gdcmSwapCode.h	1658
gdcmSwapper.h	1660
gdcmSystem.h	1661

gdcmTable.h	1661
gdcmTableEntry.h	1662
gdcmTableReader.h	1664
gdcmTag.h	1665
gdcmTagPath.h	1666
gdcmTagToVR.h	1666
gdcmTerminal.h	1667
gdcmTestDriver.h	1668
gdcmTesting.h	1669
gdcmTrace.h	1670
gdcmTransferSyntax.h	1674
gdcmTransferSyntaxSub.h	1675
gdcmType.h	1676
gdcmTypes.h	1677
gdcmUIDGenerator.h	1678
gdcmUIDs.h	1679
gdcmULAction.h	1680
gdcmULActionAA.h	1681
gdcmULActionAE.h	1681
gdcmULActionAR.h	1682
gdcmULActionDT.h	1683
gdcmULBasicCallback.h	1683
gdcmULConnection.h	1684
gdcmULConnectionCallback.h	1685
gdcmULConnectionInfo.h	1686
gdcmULConnectionManager.h	1687
gdcmULEvent.h	1688
gdcmULTransitionTable.h	1689
gdcmULWritingCallback.h	1690
gdcmUNExplicitDataElement.h	1691
gdcmUNExplicitImplicitDataElement.h	1691
gdcmUnpacker12Bits.h	1692
gdcmUsage.h	1693
gdcmUserInformation.h	1695
gdcmUUIDGenerator.h	1696
gdcmValidate.h	1696
gdcmValue.h	1697
gdcmValueIO.h	1698
gdcmVersion.h	1698
gdcmVL.h	1699
gdcmVM.h	1700
gdcmVR.h	1701
gdcmVR16ExplicitDataElement.h	1703
gdcmWaveform.h	1704
gdcmWin32.h	1704
gdcmWLMFindQuery.h	1705
gdcmWriter.h	1706
gdcmXMLDictReader.h	1707
gdcmXMLPrinter.h	1707
gdcmXMLPrivateDictReader.h	1708
vtkGDCMImageReader.h	1709
vtkGDCMImageReader2.h	1711
vtkGDCMImageWriter.h	1712
vtkGDCMMedicalImageProperties.h	1713

<a href="#">vtkGDCMPolyDataReader.h</a>	1713
<a href="#">vtkGDCMPolyDataWriter.h</a>	1714
<a href="#">vtkGDCMTesting.h</a>	1715
<a href="#">vtkGDCMThreadedImageReader.h</a>	1715
<a href="#">vtkGDCMThreadedImageReader2.h</a>	1716
<a href="#">vtkImageColorViewer.h</a>	1716
<a href="#">vtkImageMapToColors16.h</a>	1717
<a href="#">vtkImageMapToWindowLevelColors2.h</a>	1717
<a href="#">vtkImagePlanarComponentsToComponents.h</a>	1718
<a href="#">vtkImageRGBToYBR.h</a>	1718
<a href="#">vtkImageYBRToRGB.h</a>	1719
<a href="#">vtkLookupTable16.h</a>	1719
<a href="#">vtkRTStructSetProperties.h</a>	1720



## Chapter 9

# Namespace Documentation

### 9.1 gdcn Namespace Reference

#### Namespaces

- [network](#)
- [SegmentHelper](#)
- [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 [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 [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\_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 [JPEGLSCCodec](#)

- 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 [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 [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 [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 > [UCCComp](#)
- typedef [String](#)<"\", 64, 0 > [UICComp](#)
- typedef [String](#)<"\", 4294967294 > [URComp](#)
- typedef [String](#)<"\", 64 > [UTComp](#)

## Enumerations

- enum `CompOperators` {  
    `GDCM_EQUAL` = 0 ,  
    `GDCM_DIFFERENT` ,  
    `GDCM_GREATER` ,  
    `GDCM_GREATEROREQUAL` ,  
    `GDCM_LESS` ,  
    `GDCM_LESSCOREQUAL` }
- 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 [File](#) &val)
- std::ostream & [operator<<](#) (std::ostream &os, const [FileMetaInformation](#) &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 [PDSElement](#) &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 Sorter &s)`
- `std::ostream & operator<< (std::ostream &os, const StrictScanner &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)`
- `std::ostream & operator<< (std::ostream &os, Event &e)`  
*Generic inserter operator for Event and its subclasses.*
- `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, unsigned int i, int iexp)`
- `TYPETOENCODING (SQ, VRBINARY, unsigned char) TYPETOENCODING(UN`
- `template<typename Float >  
static void x16printf (char *buf, int size, Float f)`

## Variables

- static `Global GlobalInstance`
- `VRBINARY`

### 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 DAComp

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

### 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 TMComp

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

#### 9.1.2.17 UCComp

```
typedef String<'\\', 4294967294> gdcm::UCComp
```

#### 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 UTComp

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

### 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]
```

Referenced by `doround()`.

### 9.1.4.2 backslash()

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

Referenced by `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]
```

Referenced by x16printf().

#### 9.1.4.5 doround()

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

References add1().

Referenced by 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]
```

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

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

#### 9.1.4.9 operator<<() [1/56]

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

#### 9.1.4.10 operator<<() [2/56]

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

#### 9.1.4.11 operator<<() [3/56]

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

#### 9.1.4.12 operator<<() [4/56]

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

#### 9.1.4.13 operator<<() [5/56]

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

#### 9.1.4.14 operator<<() [6/56]

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

#### 9.1.4.15 operator<<() [7/56]

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

#### 9.1.4.16 operator<<() [8/56]

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

#### 9.1.4.17 operator<<() [9/56]

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

#### 9.1.4.18 operator<<() [10/56]

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

#### 9.1.4.19 operator<<() [11/56]

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

#### 9.1.4.20 operator<<() [12/56]

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

**9.1.4.21 operator<<() [13/56]**

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

**9.1.4.22 operator<<() [14/56]**

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

**9.1.4.23 operator<<() [15/56]**

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

**9.1.4.24 operator<<() [16/56]**

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

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

**9.1.4.25 operator<<() [17/56]**

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

#### 9.1.4.26 operator<<() [18/56]

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

#### 9.1.4.27 operator<<() [19/56]

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

#### 9.1.4.28 operator<<() [20/56]

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

#### 9.1.4.29 operator<<() [21/56]

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

#### 9.1.4.30 operator<<() [22/56]

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

#### 9.1.4.31 operator<<() [23/56]

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



**9.1.4.32 operator<<() [24/56]**

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

**9.1.4.33 operator<<() [25/56]**

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

**9.1.4.34 operator<<() [26/56]**

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

**9.1.4.35 operator<<() [27/56]**

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

**9.1.4.36 operator<<() [28/56]**

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

**9.1.4.37 operator<<() [29/56]**

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

**9.1.4.38 operator<<() [30/56]**

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

**9.1.4.39 operator<<() [31/56]**

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

**9.1.4.40 operator<<() [32/56]**

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

**9.1.4.41 operator<<() [33/56]**

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

**9.1.4.42 operator<<() [34/56]**

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

**9.1.4.43 operator<<() [35/56]**

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

**9.1.4.44 operator<<() [36/56]**

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

**9.1.4.45 operator<<() [37/56]**

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

**9.1.4.46 operator<<() [38/56]**

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

**9.1.4.47 operator<<() [39/56]**

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

**9.1.4.48 operator<<() [40/56]**

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

**9.1.4.49 operator<<() [41/56]**

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

**9.1.4.50 operator<<() [42/56]**

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

**9.1.4.51 operator<<() [43/56]**

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

**9.1.4.52 operator<<() [44/56]**

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

**9.1.4.53 operator<<() [45/56]**

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

**9.1.4.54 operator<<() [46/56]**

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

**9.1.4.55 operator<<() [47/56]**

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

**9.1.4.56 operator<<() [48/56]**

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

**9.1.4.57 operator<<() [49/56]**

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

References gdcm::Region::Print().

**9.1.4.58 operator<<() [50/56]**

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

**9.1.4.59 operator<<() [51/56]**

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

**9.1.4.60 operator<<() [52/56]**

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

#### 9.1.4.61 operator<<() [53/56]

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

#### 9.1.4.62 operator<<() [54/56]

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

#### 9.1.4.63 operator<<() [55/56]

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

#### 9.1.4.64 operator<<() [56/56]

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

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

References `gdcm::Event::Print()`.

#### 9.1.4.65 operator==(

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

#### Examples

[DumpPhilipsECHO.cxx](#).

#### 9.1.4.66 operator>>() [1/3]

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

#### 9.1.4.67 operator>>>() [2/3]

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

References gdcm::ignore\_char::m\_char.

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

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

#### 9.1.4.69 Round()

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

Referenced by gdcm::ImageChangePhotometricInterpretation::RGB2YBR(), and gdcm::ImageChangePhotometricInterpretation::YBR2RGB().

#### 9.1.4.70 roundat()

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

References doround().

Referenced by x16printf().

#### 9.1.4.71 TYPETOENCODING()

```
gdcmm::TYPETOENCODING (
    SQ ,
    VRBINARY ,
    unsigned char )
```

#### 9.1.4.72 x16printf()

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

References `clean()`, and `roundat()`.

Referenced by `gdcmm::EncodingImplementation< VR::VRASCII >::Write()`.

### 9.1.5 Variable Documentation

#### 9.1.5.1 GlobalInstance

```
Global gdcmm::GlobalInstance [static]
```

#### 9.1.5.2 VRBINARY

```
gdcmm::VRBINARY
```

Referenced by `gdcmm::Element< TVR, VM::VM1_n >::Set()`, and `gdcmm::Element< TVR, VM::VM1_n >::SetNoSwap()`.



## 9.2 gdcmm::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 gdcn::SegmentHelper Namespace Reference

### Classes

- struct [BasicCodedEntry](#)

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

## 9.4 gdcn::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 gdcmm::terminal::Attribute
```

Enumerator

reset	
bright	
dim	
underline	
blink	
reverse	
hidden	

#### 9.4.2.2 Color

```
enum gdcmm::terminal::Color
```

Enumerator

black	
red	
green	
yellow	
blue	
magenta	
cyan	
white	

### 9.4.2.3 Mode

```
enum gdcmm::terminal::Mode
```

Enumerator

CONSOLE	
VT100	

## 9.4.3 Function Documentation

### 9.4.3.1 setattribute()

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

### 9.4.3.2 setbgcolor()

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

### 9.4.3.3 setfgcolor()

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

### 9.4.3.4 setmode()

```
GDCM_EXPORT void gdcmm::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

### 10.1.1 Detailed Description

[AAbortPDU](#).

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

### 10.1.2 Constructor & Destructor Documentation

#### 10.1.2.1 AAbortPDU()

```
gdcn::network::AAbortPDU::AAbortPDU ( )
```

### 10.1.3 Member Function Documentation

#### 10.1.3.1 IsLastFragment()

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

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

#### 10.1.3.2 Print()

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

Implements [gdcn::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 [UserInfo](#) & [GetUserInfo](#) () 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

## 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 gdcmm::network::AAssociateACPDU::SizeType
```

### 10.2.3 Constructor & Destructor Documentation

### 10.2.3.1 AAssociateACPDU()

```
gdcmm::network::AAssociateACPDU::AAssociateACPDU ( )
```

## 10.2.4 Member Function Documentation

### 10.2.4.1 AddPresentationContextAC()

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

### 10.2.4.2 GetNumberOfPresentationContextAC()

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

### 10.2.4.3 GetPresentationContextAC()

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

### 10.2.4.4 GetUserInfoInformation()

```
const UserInfoInformation& gdcmm::network::AAssociateACPDU::GetUserInfoInformation ( ) const [inline]
```

### 10.2.4.5 InitFromRQ()

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



#### 10.2.4.6 IsLastFragment()

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

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

#### 10.2.4.7 Print()

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

Implements [gdcm::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 Function Documentation

#### 10.2.5.1 AAssociateRQPDU

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

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

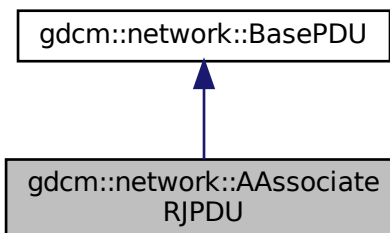
- [gdcmAAssociateACPDU.h](#)

## 10.3 gdcm::network::AAssociateRJPDU Class Reference

[AAssociateRJPDU](#).

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

Inheritance diagram for `gdcm::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

### 10.3.1 Detailed Description

[AAssociateRJPDU](#).

[Table 9-21](#) ASSOCIATE-RJ PDU FIELDS

### 10.3.2 Constructor & Destructor Documentation

#### 10.3.2.1 AAssociateRJPDU()

```
gdcmm::network::AAssociateRJPDU::AAssociateRJPDU ( )
```

### 10.3.3 Member Function Documentation

#### 10.3.3.1 IsLastFragment()

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

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

#### 10.3.3.2 Print()

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

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

#### 10.3.3.3 Read()

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

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

#### 10.3.3.4 Size()

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

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

#### 10.3.3.5 Write()

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

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

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

- [gdcmmAAssociateRJPDU.h](#)

## 10.4 gdcmm::network::AAssociateRQPDU Class Reference

[AAssociateRQPDU](#).

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

Inheritance diagram for gdcmm::network::AAssociateRQPDU:



Collaboration diagram for gdcmm::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

## 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> gdcm::network::AAssociateRQPDU::PresentationContextArrayType
```

### 10.4.2.2 SizeType

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

## 10.4.3 Constructor & Destructor Documentation

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

```
gdcm::network::AAssociateRQPDU::AAssociateRQPDU ( )
```

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

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

## 10.4.4 Member Function Documentation

### 10.4.4.1 AddPresentationContext()

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

### 10.4.4.2 GetCalledAETitle()

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

#### 10.4.4.3 GetCallingAETitle()

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

#### 10.4.4.4 GetNumberOfPresentationContext()

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

#### 10.4.4.5 GetPresentationContext()

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

#### 10.4.4.6 GetPresentationContextByAbstractSyntax()

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

#### 10.4.4.7 GetPresentationContextByID()

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

#### 10.4.4.8 GetPresentationContexts()

```
PresentationContextArrayType const& gdcn::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 UserInfoInformation& 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 SetUserInfoInformation()

```
void gdcm::network::AAssociateRQPDU::SetUserInfoInformation (
    UserInfoInformation 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 Function Documentation

#### 10.4.5.1 AAssociateACPDU

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

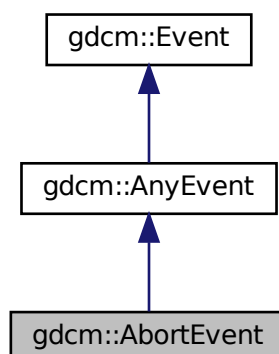
The documentation for this class was generated from the following file:

- [gdcmAAssociateRQPDU.h](#)

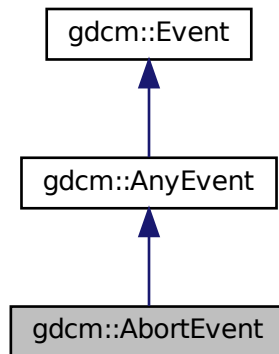
## 10.5 gdcm::AbortEvent Class Reference

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

Inheritance diagram for gdcm::AbortEvent:



Collaboration diagram for `gdcm::AbortEvent`:



### Additional Inherited Members

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

#### 10.6.3.5 Read()

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

#### 10.6.3.6 SetName()

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

#### 10.6.3.7 SetNameFromUID()

```
void gdcM::network::AbstractSyntax::SetNameFromUID (
    UIDs::TSName tsname )
```

#### 10.6.3.8 Size()

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

#### 10.6.3.9 Write()

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

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

- [gdcMAbstractSyntax.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 [gdcmm::Event](#) \*e) const override
- const char \* [GetEventName](#) () const override
- [Tag](#) const & [GetTag](#) () const
- [gdcmm::Event](#) \* [MakeObject](#) () const override
- void [operator=](#) (const [Self](#) &)=delete
- void [SetTag](#) (const [Tag](#) &t)

### 10.7.1 Detailed Description

[AnonymizeEvent](#).

Special type of event triggered during the Anonymization process

See also

[Anonymizer](#)

### 10.7.2 Member Typedef Documentation

#### 10.7.2.1 Self

```
typedef AnonymizeEvent gdcmm::AnonymizeEvent::Self
```

#### 10.7.2.2 Superclass

```
typedef AnyEvent gdcmm::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]
```

#### 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:

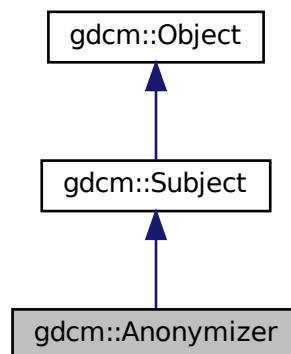
- [gdcMAnonymizeEvent.h](#)

## 10.8 gdcM::Anonymizer Class Reference

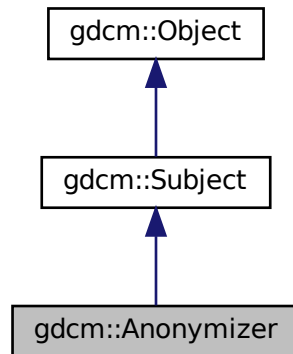
[Anonymizer](#).

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

Inheritance diagram for gdcM::Anonymizer:



Collaboration diagram for gdcmm::Anonymizer:



## Public Member Functions

- [Anonymizer](#) ()
- [~Anonymizer](#) () override
- bool [BasicApplicationLevelConfidentialityProfile](#) (bool deidentify=true)
- bool [Empty](#) (Tag const &t)
- const [CryptographicMessageSyntax](#) \* [GetCryptographicMessageSyntax](#) () const
- [File](#) & [GetFile](#) ()
- bool [Remove](#) (Tag const &t)
- 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](#) (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.*

## 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)

### 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  $O(m*\log(n))$  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

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

## 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

### 10.8.3.3 CanEmptyTag()

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

### 10.8.3.4 ClearInternalUIDs()

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

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

#### Warning

the mapping is definitely lost

### 10.8.3.5 Empty()

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

Make [Tag](#) t empty (if not found tag will be created) Warning: does not handle SQ element

#### Examples

[CreateJPIPDataSet.cxx](#).

### 10.8.3.6 GetBasicApplicationLevelConfidentialityProfileAttributes()

```
static std::vector<Tag> gdcm::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes  
( ) [static]
```

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

#### Examples

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

### 10.8.3.7 GetCryptographicMessageSyntax()

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

### 10.8.3.8 GetFile()

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

### 10.8.3.9 New()

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

for wrapped language: instantiate a reference counted object

### 10.8.3.10 RecurseDataSet()

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

### 10.8.3.11 Remove()

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

remove a tag (even a SQ can be removed) Return code is false when tag t cannot be found

### 10.8.3.12 RemoveGroupLength()

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

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

#### Examples

[ClinicalTrialAnnotate.cxx](#).

### 10.8.3.13 RemovePrivateTags()

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

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

#### Examples

[ClinicalTrialAnnotate.cxx](#).

### 10.8.3.14 RemoveRetired()

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

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

### 10.8.3.15 Replace() [1/2]

```
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](#), and [EncapsulateFileInRawData.cxx](#).



### 10.8.3.16 Replace() [2/2]

```
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

### 10.8.3.17 SetCryptographicMessageSyntax()

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

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

### 10.8.3.18 SetFile()

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

Set/Get [File](#).

#### Examples

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

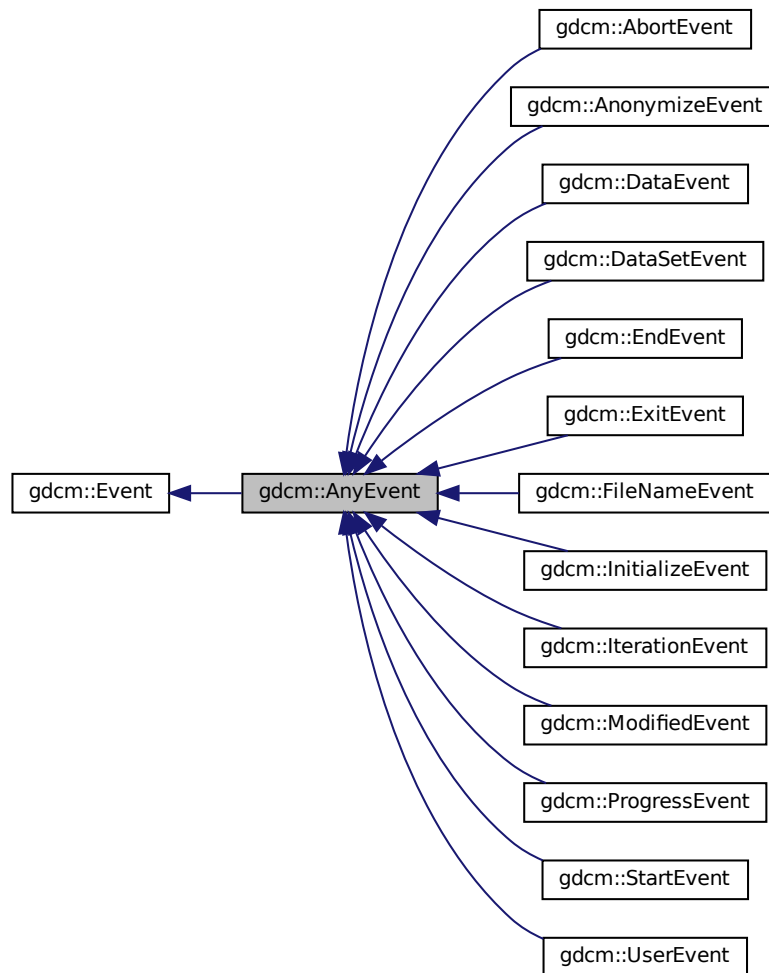
The documentation for this class was generated from the following file:

- [gdcmAnonymizer.h](#)

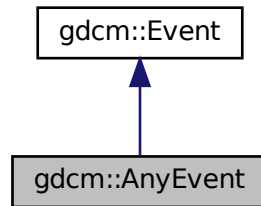
## 10.9 gdcm::AnyEvent Class Reference

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

Inheritance diagram for gdcm::AnyEvent:



Collaboration diagram for gdcm::AnyEvent:



### Additional Inherited Members

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()

```
gdcmm::network::ApplicationContext::ApplicationContext ( )
```

## 10.10.3 Member Function Documentation

### 10.10.3.1 GetName()

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

### 10.10.3.2 Print()

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

### 10.10.3.3 Read()

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

### 10.10.3.4 SetName()

```
void gdcmm::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:

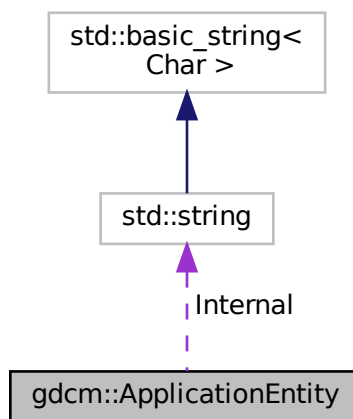
- [gdcmApplicationContext.h](#)

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

### 10.11.2.4 Squeeze()

```
void gdcm::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:

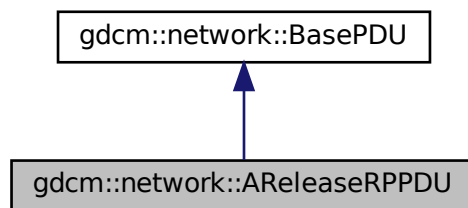
- [gdcMApplicationEntity.h](#)

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

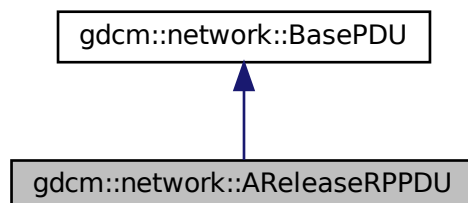
[AReleaseRPPDU](#).

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

Inheritance diagram for gdcM::network::AReleaseRPPDU:



Collaboration diagram for gdcM::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

### 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()

```
gdcm::network::AReleaseRPPDU::AReleaseRPPDU ( )
```

### 10.12.3 Member Function Documentation

#### 10.12.3.1 IsLastFragment()

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

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

#### 10.12.3.2 Print()

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

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

### 10.12.3.3 Read()

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

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

### 10.12.3.4 Size()

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

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

### 10.12.3.5 Write()

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

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

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

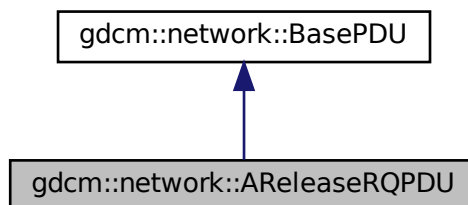
- [gdcmAReleaseRPPDU.h](#)

## 10.13 gdcm::network::AReleaseRQPDU Class Reference

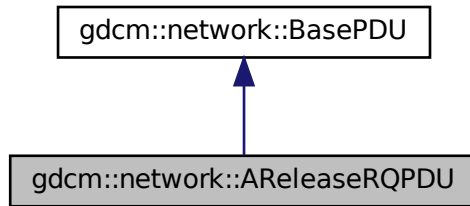
[AReleaseRQPDU](#).

```
#include <gdcmAReleaseRQPDU.h>
```

Inheritance diagram for gdcm::network::AReleaseRQPDU:



Collaboration diagram for gdcm::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

### 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]

```
gdcm::ASN1::ASN1 ( )
```

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

```
gdcm::ASN1::~~ASN1 ( )
```

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

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

### 10.15.3 Member Function Documentation

#### 10.15.3.1 operator=()

```
void gdcm::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 gdcn::network::AsynchronousOperationsWindowSub::Size ( ) const
```

### 10.16.3.4 Write()

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

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

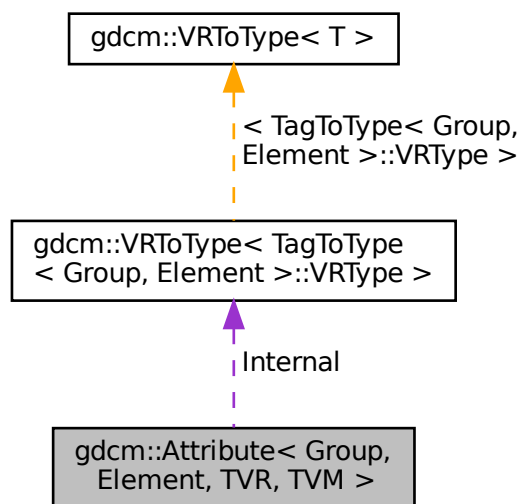
- [gdcnAsynchronousOperationsWindowSub.h](#)

## 10.17 gdcn::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 <gdcnAttribute.h>
```

Collaboration diagram for gdcn::Attribute< Group, Element, TVR, TVM >:



## Public Types

- enum { [VMType](#) = VMToLength<TVM>::Length }
- typedef [VRTType](#)< TVR >::Type [ArrayType](#)

## Public Member Functions

- [GDCM\\_STATIC\\_ASSERT](#) (((((VR::VRTType) TVR &VR::VR\_VM1) &&((VM::VMType) TVM==VM::VM1))||!((VR::VRTType) TVR &VR::VR\_VM1)))
- [GDCM\\_STATIC\\_ASSERT](#) (((VM::VMType) TVM &(VM::VMType)(TagToType< Group, [Element](#) >::VMType)))
- [GDCM\\_STATIC\\_ASSERT](#) (((VR::VRTType) TVR &(VR::VRTType)(TagToType< Group, [Element](#) >::VRTType)))
- [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 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.

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](#), [CreateJPIPDataSet.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](#), [SortImage.cxx](#), [StreamImageReaderTest.cxx](#), [VolumeSorter.cxx](#), [gdcmmrtionplan.cxx](#), [gdcmmrtplan.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 gdcmm::Attribute< Group, Element, TVR, TVM >::ArrayType
```

#### Examples

[ReadAndPrintAttributes.cxx](#).

### 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 gdcmm::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 gdcmm::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 gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues ( ) const [inline]
```

#### Examples

[LargeVRDSExplicit.cxx](#).

Referenced by `gdcmm::Attribute< Group, Element, TVR, TVM >::operator<()>`, and `gdcmm::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 gdcmm::Attribute< Group, Element, TVR, TVM >::GetTag ( ) [inline], [static]
```

#### Examples

[PatchFile.cxx](#), [gdcmmrtionplan.cxx](#), and [gdcmmrtplan.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](#), [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 gdcM::Attribute< Group, Element, TVR, TVM >::operator!= (
    const Attribute< Group, Element, TVR, TVM > & att ) const [inline]
```

References gdcM::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 gdcM::Attribute< Group, Element, TVR, TVM >::operator< (
    const Attribute< Group, Element, TVR, TVM > & att ) const [inline]
```

References gdcM::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), and gdcM::Attribute< Group, Element, TVR, TVM >::GetValues().

**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 gdcM::Attribute< Group, Element, TVR, TVM >::operator==(
    const Attribute< Group, Element, TVR, TVM > & att ) const [inline]
```

References gdcM::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& gdcM::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& gdcm::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 gdcm::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](#), [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](#), [LargeVRDSExplicit.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 [LargeVRDSExplicit.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]
```

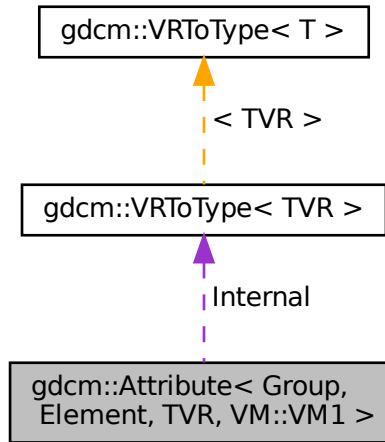
The documentation for this class was generated from the following file:

- [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 `VRToType< TVR >::Type` `ArrayType`

### Public Member Functions

- `GDCM_STATIC_ASSERT` (((((VR::VRType) TVR & VR::VR\_VM1) && ((VM::VMType) VM::VM1 == VM::VM1)) || !((VR::VRType) TVR & VR::VR\_VM1)))
- `GDCM_STATIC_ASSERT` (((VM::VMType) VM::VM1 & (VM::VMType)(TagToType< Group, Element >::VMType)))
- `GDCM_STATIC_ASSERT` (((VR::VRType) TVR & (VR::VRType)(TagToType< Group, Element >::VRType)))
- `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 gdcmm::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

VMType	
--------	--

### 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>
gdcmm::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 gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement ( ) const [inline]
```

References gdcm::DataElement::GetVR(), gdcm::DataElement::SetByteValue(), and gdcm::DataElement::SetVR().

### 10.18.3.6 GetDictVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VM gdcm::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 gdcm::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 gdcm::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 gdcm::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& gdcm::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& gdcM::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* gdcM::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 gdcM::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 gdcM::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 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().

**10.18.3.16 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 >::GetValues().



**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
```

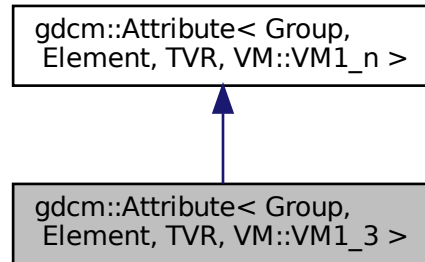
The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

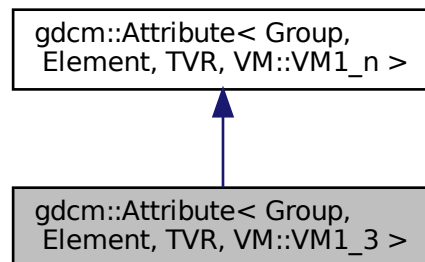
## 10.19 gdcm::Attribute< Group, Element, TVR, VM::VM1\_3 > Class Template Reference

```
#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

### Additional Inherited Members

#### 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]
```

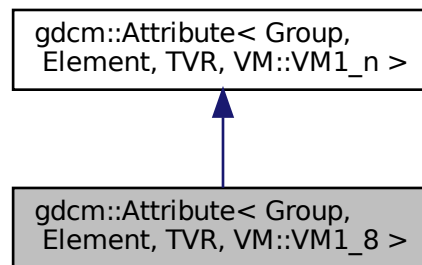
The documentation for this class was generated from the following file:

- [gdcMAttribute.h](#)

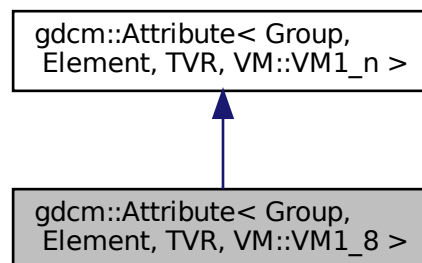
## 10.20 gdcM::Attribute< Group, Element, TVR, VM::VM1\_8 > Class Template Reference

```
#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

## Additional Inherited Members

### 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]
```

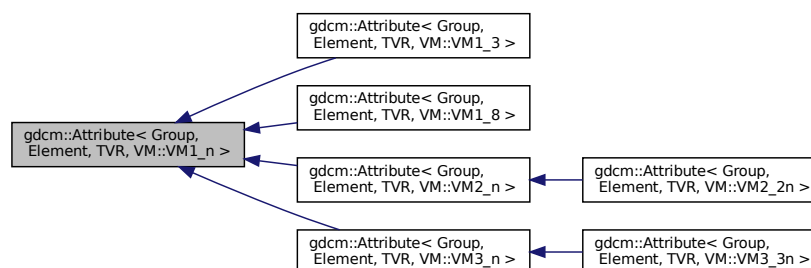
The documentation for this class was generated from the following file:

- [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>
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GDCM_STATIC_ASSERT (
    (((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TagToType< Group, Element >::VMType==VM::VM1))||((V
TVR &VR::VR_VM1)) )
```

### 10.21.3.2 GDCM\_STATIC\_ASSERT() [2/3]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::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>
gdcm::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 gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement ( ) const [inline]
```

References gdcm::DataElement::GetVR(), gdcm::DataElement::SetByteValue(), and gdcm::DataElement::SetVR().

#### 10.21.3.5 GetDictVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VM gdcm::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 gdcm::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 gdcm::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]
```

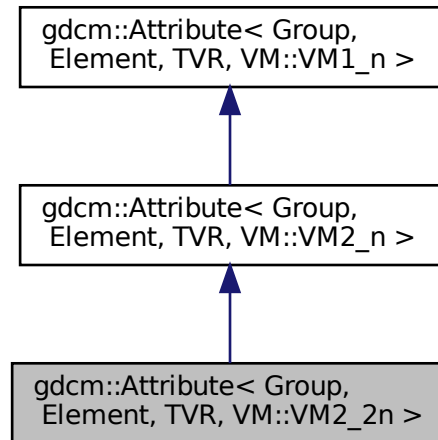
The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

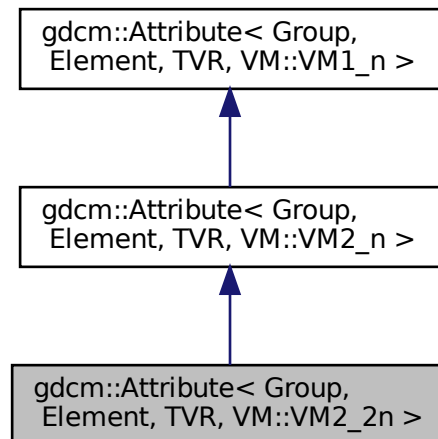
## 10.22 gdcM::Attribute< Group, Element, TVR, VM::VM2\_2n > Class Template Reference

```
#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](#) ()

## Additional Inherited Members

### 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]
```

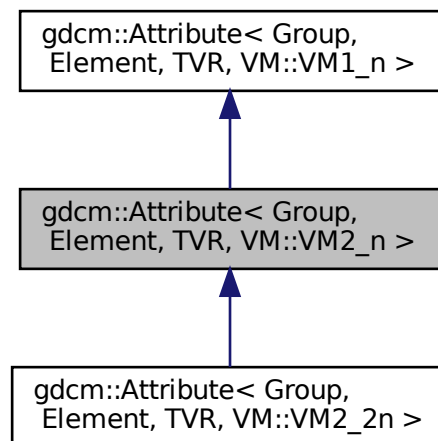
The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

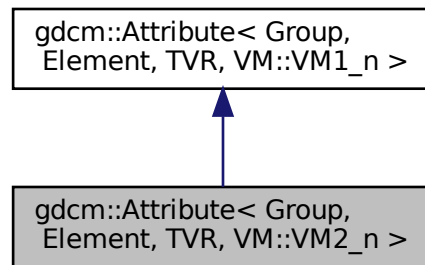
## 10.23 gdcm::Attribute< Group, Element, TVR, VM::VM2\_n > Class Template Reference

```
#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

## Additional Inherited Members

### 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]
```

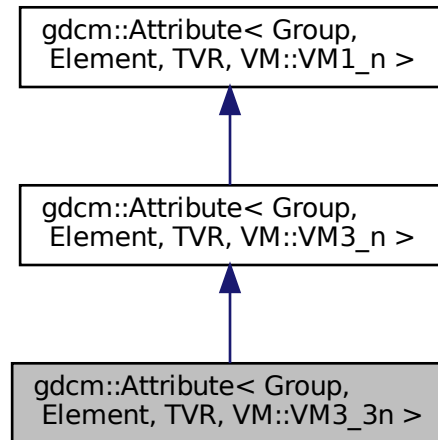
The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

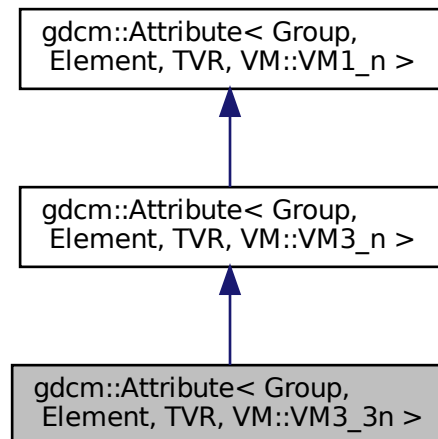
## 10.24 gdcm::Attribute< Group, Element, TVR, VM::VM3\_3n > Class Template Reference

```
#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](#) ()

## Additional Inherited Members

### 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]
```

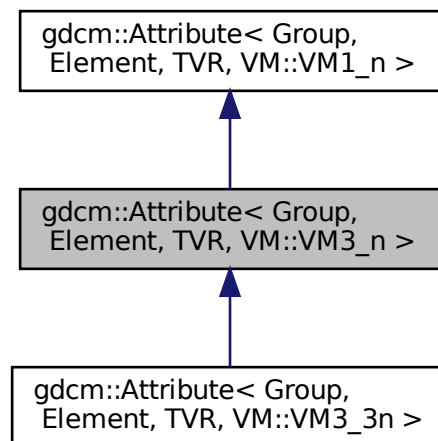
The documentation for this class was generated from the following file:

- [gdcMAttribute.h](#)

## 10.25 [gdcM::Attribute](#)< Group, Element, TVR, VM::VM3\_n > Class Template Reference

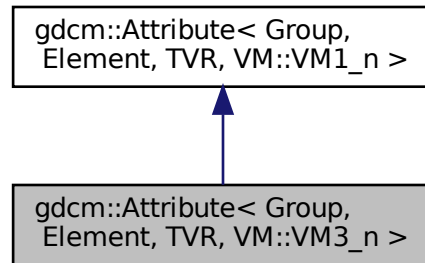
```
#include <gdcMAttribute.h>
```

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](#) ()

## Additional Inherited Members

### 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:

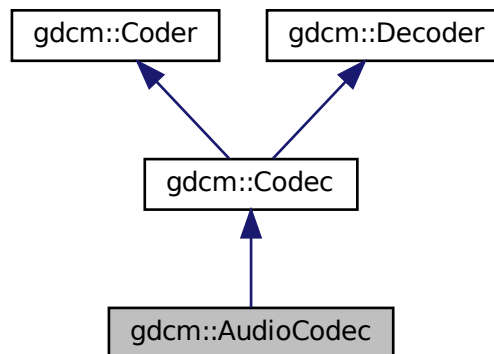
- [gdcmAttribute.h](#)

## 10.26 gdcm::AudioCodec Class Reference

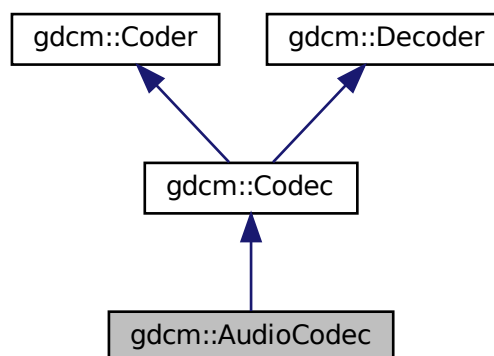
[AudioCodec.](#)

```
#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.*

## Additional Inherited Members

### 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 gdcM::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:

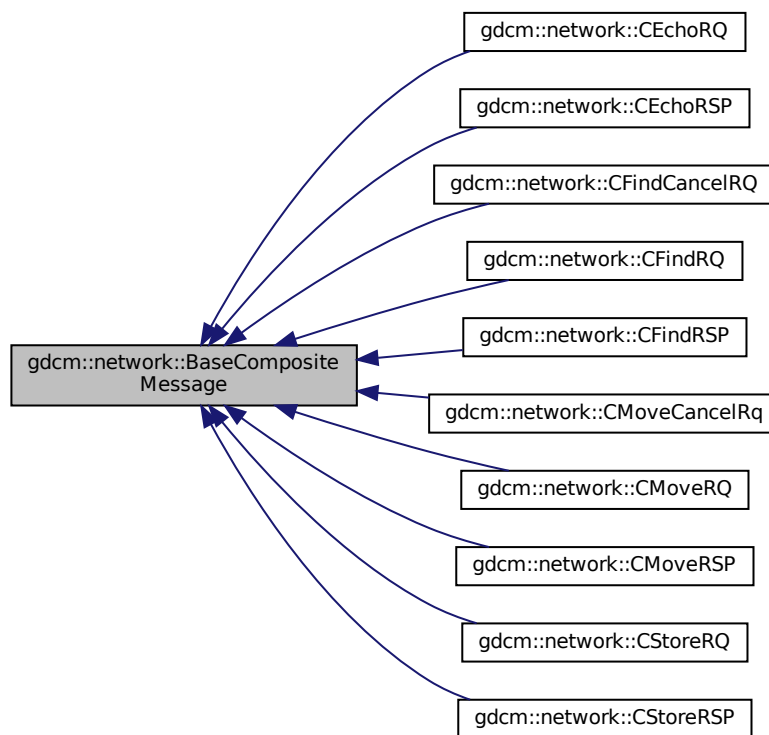
- [gdcmBase64.h](#)

## 10.28 gdcm::network::BaseCompositeMessage Class Reference

[BaseCompositeMessage](#).

```
#include <gdcmBaseCompositeMessage.h>
```

Inheritance diagram for gdcm::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::CMoveRQ`, `gdcm::network::CFindRQ`, and `gdcm::network::CEchoRQ`.

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

- [gdcmBaseCompositeMessage.h](#)

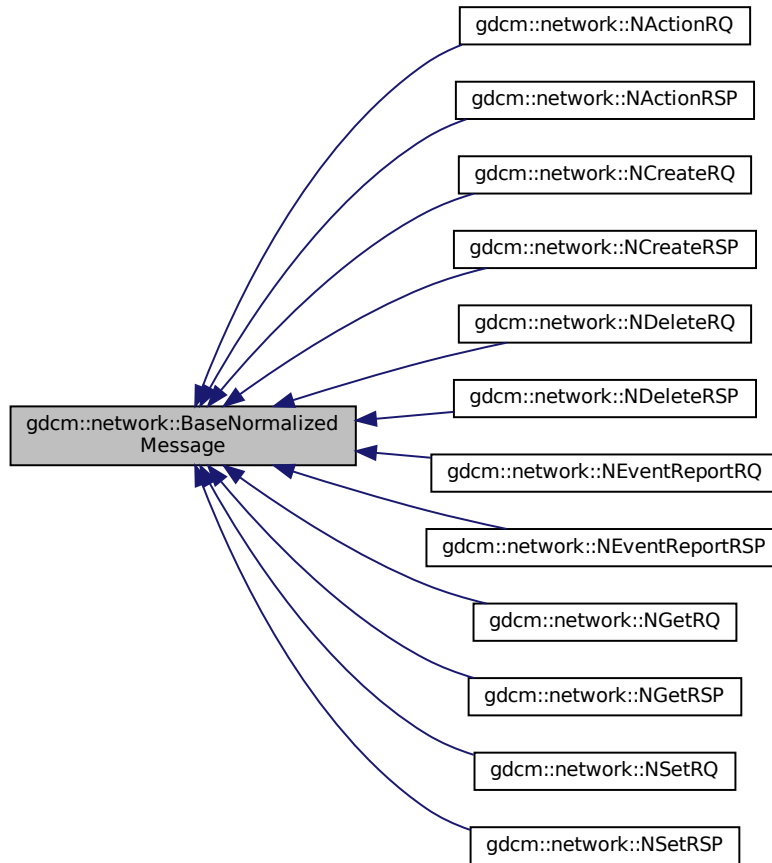


## 10.29 gdcm::network::BaseNormalizedMessage Class Reference

[BaseNormalizedMessage](#).

```
#include <gdcmBaseNormalizedMessage.h>
```

Inheritance diagram for gdcm::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::NSetRQ](#), [gdcm::network::NGetRQ](#), [gdcm::network::NEventReportRQ](#), [gdcm::network::NDeleteRQ](#), [gdcm::network::NCreateRQ](#), and [gdcm::network::NActionRQ](#).

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

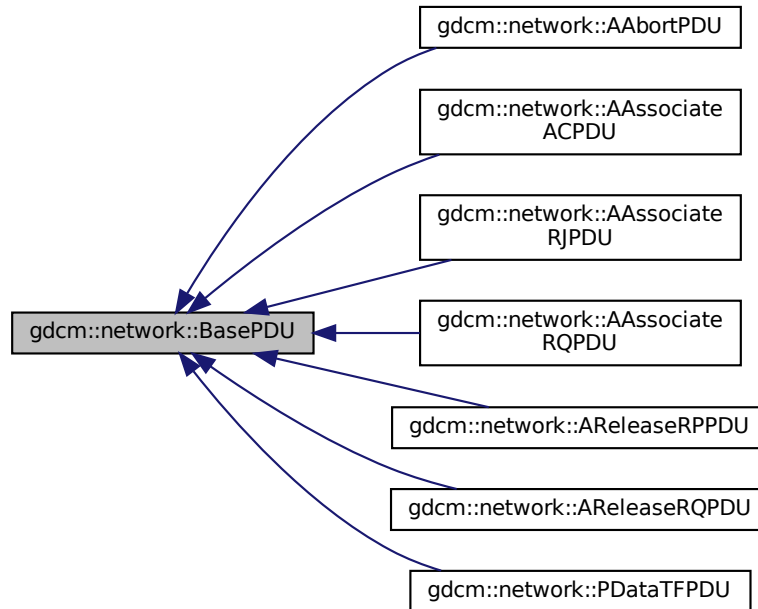
- [gdcmBaseNormalizedMessage.h](#)

## 10.30 gdcm::network::BasePDU Class Reference

[BasePDU](#).

```
#include <gdcmBasePDU.h>
```

Inheritance diagram for gdcm::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 gdcn::network::BasePDU::~BasePDU ( ) [virtual], [default]
```

## 10.30.3 Member Function Documentation

### 10.30.3.1 IsLastFragment()

```
virtual bool gdcn::network::BasePDU::IsLastFragment ( ) const [pure virtual]
```

Implemented in [gdcn::network::PDataTFPDU](#), [gdcn::network::AReleaseRQPDU](#), [gdcn::network::AReleaseRPPDU](#), [gdcn::network::AAssociateRQPDU](#), [gdcn::network::AAssociateRJPDU](#), [gdcn::network::AAssociateACPDU](#), and [gdcn::network::AAbortPDU](#).

### 10.30.3.2 Print()

```
virtual void gdcn::network::BasePDU::Print (
    std::ostream & os ) const [pure virtual]
```

Implemented in [gdcn::network::PDataTFPDU](#), [gdcn::network::AReleaseRQPDU](#), [gdcn::network::AReleaseRPPDU](#), [gdcn::network::AAssociateRQPDU](#), [gdcn::network::AAssociateRJPDU](#), [gdcn::network::AAssociateACPDU](#), and [gdcn::network::AAbortPDU](#).

### 10.30.3.3 Read()

```
virtual std::istream& gdcm::network::BasePDU::Read (
    std::istream & is ) [pure virtual]
```

Implemented in [gdcm::network::PDataTFPDU](#), [gdcm::network::AReleaseRQPDU](#), [gdcm::network::AReleaseRPPDU](#), [gdcm::network::AAssociateRQPDU](#), [gdcm::network::AAssociateRJPDU](#), [gdcm::network::AAssociateACPDU](#), and [gdcm::network::AAabortPDU](#).

### 10.30.3.4 Size()

```
virtual size_t gdcm::network::BasePDU::Size ( ) const [pure virtual]
```

Implemented in [gdcm::network::PDataTFPDU](#), [gdcm::network::AReleaseRQPDU](#), [gdcm::network::AReleaseRPPDU](#), [gdcm::network::AAssociateRQPDU](#), [gdcm::network::AAssociateRJPDU](#), [gdcm::network::AAssociateACPDU](#), and [gdcm::network::AAabortPDU](#).

### 10.30.3.5 Write()

```
virtual const std::ostream& gdcm::network::BasePDU::Write (
    std::ostream & os ) const [pure virtual]
```

Implemented in [gdcm::network::PDataTFPDU](#), [gdcm::network::AReleaseRQPDU](#), [gdcm::network::AReleaseRPPDU](#), [gdcm::network::AAssociateRQPDU](#), [gdcm::network::AAssociateRJPDU](#), [gdcm::network::AAssociateACPDU](#), and [gdcm::network::AAabortPDU](#).

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

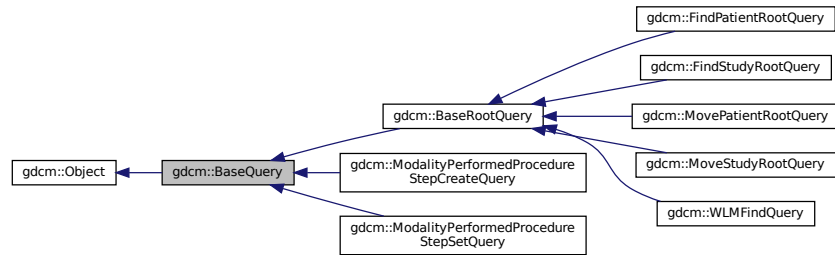
- [gdcmBasePDU.h](#)

## 10.31 gdcm::BaseQuery Class Reference

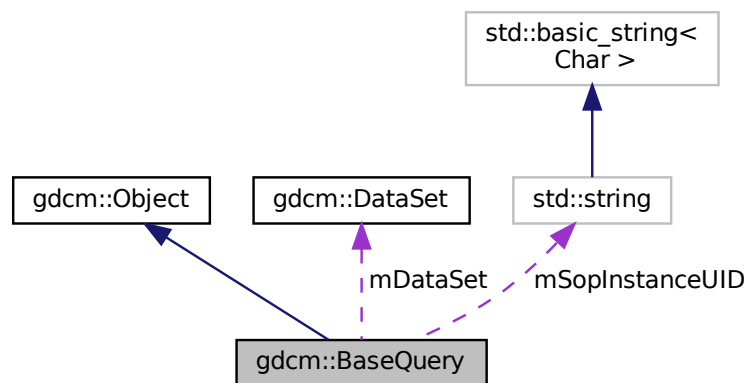
[BaseQuery](#).

```
#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)

## 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 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()

```
gdcmm::BaseQuery::BaseQuery ( ) [protected]
```

#### 10.31.2.2 ~BaseQuery()

```
gdcmm::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::WLMFindQuery](#), [gdcM::MoveStudyRootQuery](#), [gdcM::MovePatientRootQuery](#), [gdcM::ModalityPerformedProcedureStepCreateQuery](#), [gdcM::FindStudyRootQuery](#), and [gdcM::FindPatientRootQuery](#).

### 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::BaseRootQuery](#), [gdcm::WLMFindQuery](#), [gdcm::MoveStudyRootQuery](#), [gdcm::MovePatientRootQuery](#), [gdcm::ModalityPerformedProcedureStepSetQuery](#), [gdcm::ModalityPerformedProcedureStepCreateQuery](#), [gdcm::FindStudyRootQuery](#), and [gdcm::FindPatientRootQuery](#).

#### 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 Function 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:

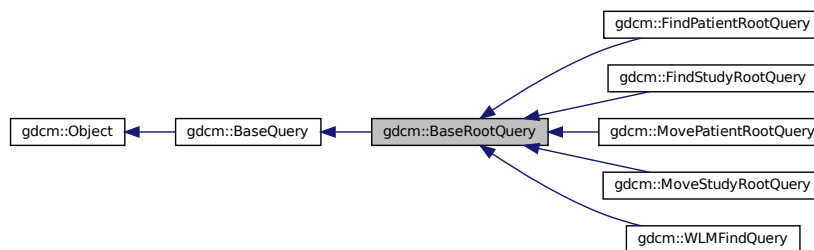
- [gdcmBaseQuery.h](#)

## 10.32 gdcm::BaseRootQuery Class Reference

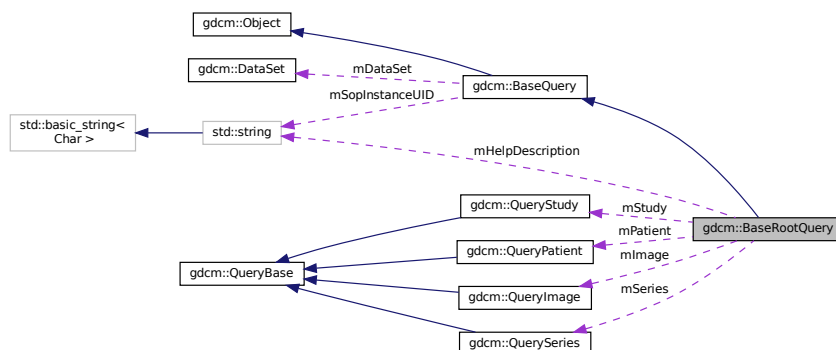
[BaseRootQuery](#).

```
#include <gdcmBaseRootQuery.h>
```

Inheritance diagram for gdcm::BaseRootQuery:



Collaboration diagram for gdcm::BaseRootQuery:



## Public Member Functions

- [~BaseRootQuery](#) () override
- [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

## 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 Attributes

- std::string [mHelpDescription](#)
- [QueryImage](#) [mImage](#)
- [QueryPatient](#) [mPatient](#)
- [ERootType](#) [mRootType](#)
- [QuerySeries](#) [mSeries](#)
- [QueryStudy](#) [mStudy](#)

## 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]
```

## 10.32.3 Member Function Documentation

### 10.32.3.1 Construct()

```
static QueryBase* gdcm::BaseRootQuery::Construct (
    ERootType inRootType,
    EQueryLevel qlevel ) [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 q1 ) [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::WLMFindQuery](#), [gdcM::MoveStudyRootQuery](#), [gdcM::MovePatientRootQuery](#), [gdcM::FindStudyRootQuery](#), and [gdcM::FindPatientRootQuery](#).

#### 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 dcm4k

Implemented in [gdcM::WLMFindQuery](#), [gdcM::MoveStudyRootQuery](#), [gdcM::MovePatientRootQuery](#), [gdcM::FindStudyRootQuery](#), and [gdcM::FindPatientRootQuery](#).

#### 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::WLMFindQuery](#), [gdcM::MoveStudyRootQuery](#), [gdcM::MovePatientRootQuery](#), [gdcM::FindStudyRootQuery](#), and [gdcM::FindPatientRootQuery](#).

## 10.32.4 Friends And Related Function 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:

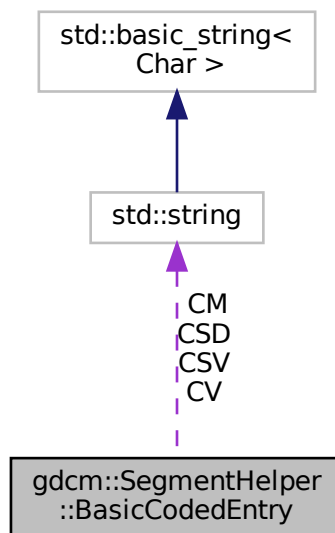
- [gdcmBaseRootQuery.h](#)

## 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:

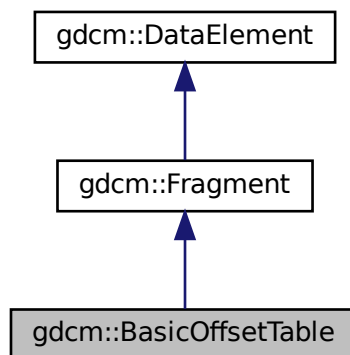
- [gdcmSegmentHelper.h](#)

## 10.34 gdcm::BasicOffsetTable Class Reference

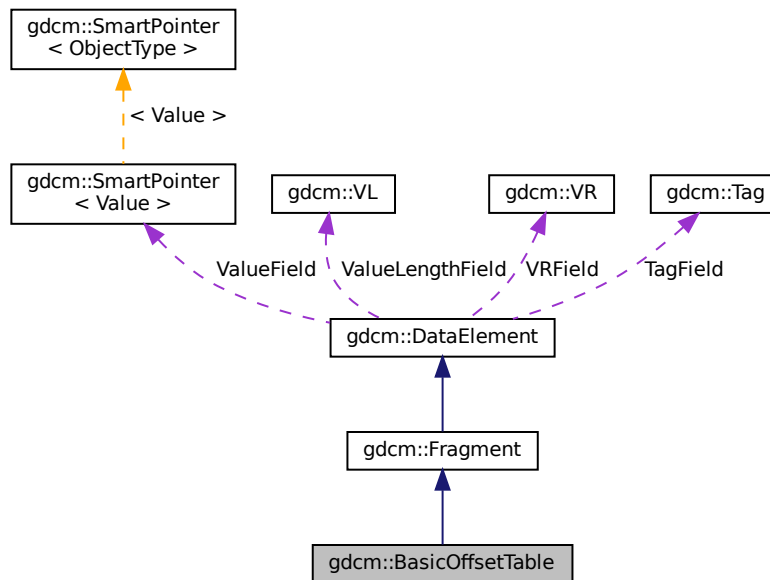
Class to represent a [BasicOffsetTable](#).

```
#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)`

## Friends

- `std::ostream & operator<< (std::ostream &os, const BasicOffsetTable &val)`

## Additional Inherited Members

### 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 Function 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:

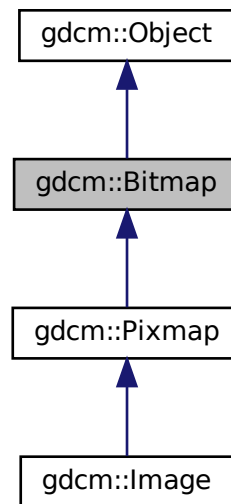
- [gdcmBasicOffsetTable.h](#)

## 10.35 gdcm::Bitmap Class Reference

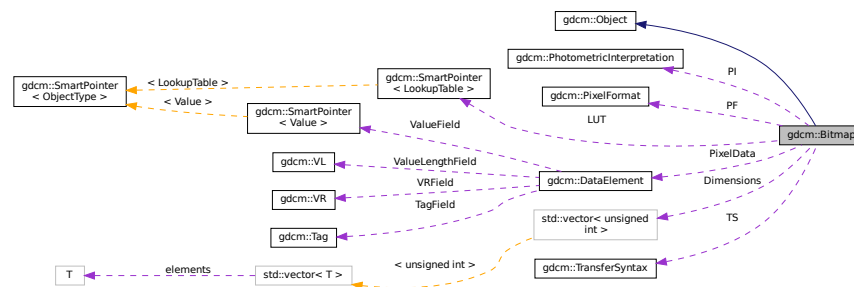
[Bitmap](#) class.

```
#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  
*Acces 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

## 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 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

[ConvertToQImage.cxx](#), [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

[ConvertToQImage.cxx](#), [GenFakelImage.cxx](#), [GetSubSequenceData.cxx](#), [PatchFile.cxx](#), [ReadMultiTimesException.cxx](#), 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
```

#### 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](#), 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

[HelloVizWorld.cxx](#), and [threadgdcm.cxx](#).

#### 10.35.4.16 GetPhotometricInterpretation()

```
const PhotometricInterpretation& gdcm::Bitmap::GetPhotometricInterpretation ( ) const
```

return the photometric interpretation

##### Examples

[ConvertToQImage.cxx](#), [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](#), [ExtractIconFromFile.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenFakeImage.cxx](#), [GetJPEGSamplePrecision.cxx](#), [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::Image](#), and [gdcm::Pixmap](#).

#### 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

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

#### 10.35.4.28 SetDimension()

```
void gdcm::Bitmap::SetDimension (
    unsigned int idx,
    unsigned int dim )
```

##### Examples

[GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

#### 10.35.4.29 SetDimensions()

```
void gdcm::Bitmap::SetDimensions (
    const unsigned int dims[3] )
```

##### Examples

[CreateARGBImage.cxx](#), and [CreateCMYKImage.cxx](#).

#### 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](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

#### 10.35.4.34 SetPhotometricInterpretation()

```
void gdcm::Bitmap::SetPhotometricInterpretation (
    PhotometricInterpretation const & pi )
```

##### Examples

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).



#### 10.35.4.35 SetPixelFormat()

```
void gdcm::Bitmap::SetPixelFormat (
    PixelFormat const & pf ) [inline]
```

##### Examples

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [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

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), and [MergeTwoFiles.cxx](#).

**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 gdcm::Bitmap::TryKAKADUCodec (
    char * buffer,
    bool & lossyflag ) const    [protected]
```

#### 10.35.4.45 TryPVRGCodec()

```
bool gdcm::Bitmap::TryPVRGCodec (
    char * buffer,
    bool & lossyflag ) const    [protected]
```

#### 10.35.4.46 TryRAWCodec()

```
bool gdcm::Bitmap::TryRAWCodec (
    char * buffer,
    bool & lossyflag ) const    [protected]
```

#### 10.35.4.47 TryRLECodec()

```
bool gdcm::Bitmap::TryRLECodec (
    char * buffer,
    bool & lossyflag ) const    [protected]
```

#### 10.35.4.48 UnusedBitsPresentInPixelData()

```
virtual bool gdcm::Bitmap::UnusedBitsPresentInPixelData ( ) const    [inline], [virtual]
```

Reimplemented in [gdcm::Pixmap](#).

### 10.35.5 Friends And Related Function 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

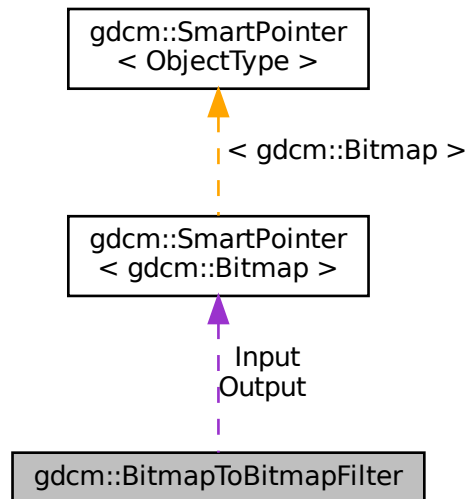
`BitmapToBitmapFilter` class.

```
#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

[CompressImage.cxx](#).

## 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:

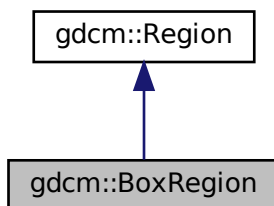
- [gdcmBitmapToBitmapFilter.h](#)

## 10.37 gdcm::BoxRegion Class Reference

Class for manipulation box region.

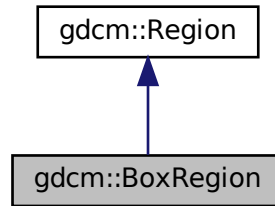
```
#include <gdcmBoxRegion.h>
```

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.*

## 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)

### 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.

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()

```
gdcmm::ByteBuffer::ByteBuffer ( ) [inline]
```

### 10.38.3 Member Function Documentation

#### 10.38.3.1 Get()

```
char* gdcmm::ByteBuffer::Get (
    int len ) [inline]
```

#### 10.38.3.2 GetStart()

```
const char* gdcmm::ByteBuffer::GetStart ( ) const [inline]
```

#### 10.38.3.3 ShiftEnd()

```
void gdcmm::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](#) ()
- 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 ( )
```

#### 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:

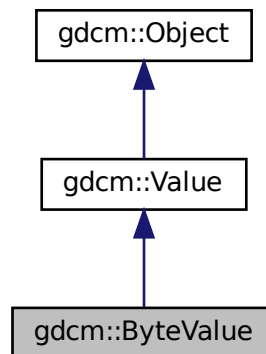
- [gdcmByteSwapFilter.h](#)

## 10.41 gdcm::ByteValue Class Reference

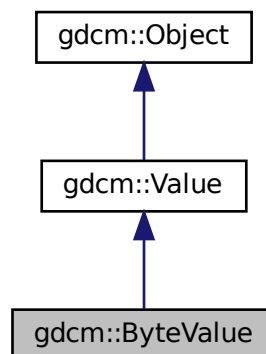
Class to represent binary value (array of bytes)

```
#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) I 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

## Protected Member Functions

- void [Print](#) (std::ostream &os) const override
- void [SetLengthOnly](#) ([VL](#) vl) override

### 10.41.1 Detailed Description

Class to represent binary value (array of bytes)

Note

#### Examples

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [MrProtocol.cxx](#), [PatchFile.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 gdcM::ByteValue::Append (
    ByteValue const & bv )
```

#### 10.41.3.2 Clear()

```
void gdcM::ByteValue::Clear ( ) [inline], [override], [virtual]
```

Implements [gdcM::Value](#).

#### 10.41.3.3 ComputeLength()

```
VL gdcM::ByteValue::ComputeLength ( ) const [inline]
```

Referenced by [gdcM::Fragment::Write\(\)](#).

#### 10.41.3.4 Fill()

```
void gdcM::ByteValue::Fill (
    char c ) [inline]
```

##### Examples

[DuplicatePCDE.cxx](#).

#### 10.41.3.5 GetBuffer()

```
bool gdcM::ByteValue::GetBuffer (
    char * buffer,
    unsigned long length ) const
```

##### Examples

[FixJAIBugJPEGSL.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](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [MrProtocol.cxx](#), [PatchFile.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::Attribute< Group, Element, TVR, VM::VM1 >::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](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [MrProtocol.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::Attribute< Group, Element, TVR, VM::VM1 >::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 gdcm::ByteValue::IsEmpty ( ) const [inline]
```

#### 10.41.3.11 IsPrintable()

```
bool gdcm::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) I 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 > &()

```
gdcm::ByteValue::operator const std::vector< char > & ( ) const [inline]
```

#### 10.41.3.13 operator=()

```
ByteValue& gdcm::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 gdcmm::ByteValue::PrintASCII (
    std::ostream & os,
    VL maxlength ) const
```

**10.41.3.18 PrintASCIIXML()**

```
void gdcmm::ByteValue::PrintASCIIXML (
    std::ostream & os ) const
```

**10.41.3.19 PrintGroupLength()**

```
void gdcmm::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](#).

#### Examples

[ReadExplicitLengthSQIVR.cxx](#).

#### 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

#### 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](#).

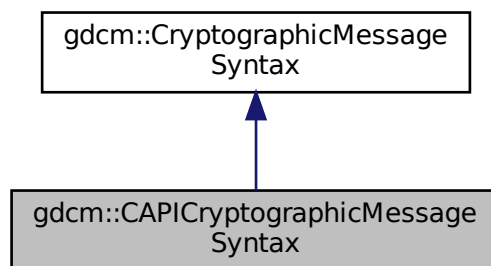
The documentation for this class was generated from the following file:

- [gdcmCAPICryptoFactory.h](#)

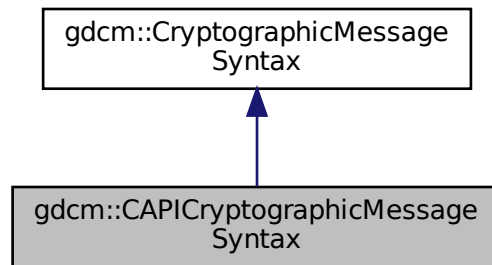
## 10.43 gdcm::CAPICryptographicMessageSyntax Class Reference

```
#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)

## Additional Inherited Members

### 10.43.1 Constructor & Destructor Documentation

#### 10.43.1.1 CAPICryptographicMessageSyntax()

```
gdcM::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 gdcM::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

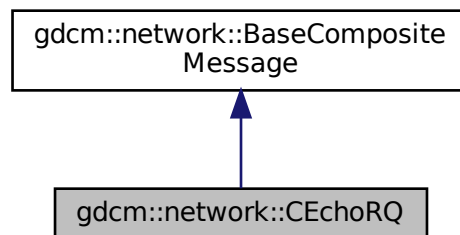
[CEchoRQ](#).

```
#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 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> gdcm::network::CEchoRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery ) [override], [virtual]
```

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

### 10.44.3 Member Data Documentation

#### 10.44.3.1 AffectedSOPClassUID

[UIComp](#) gdcm::network::CEchoRQ::AffectedSOPClassUID

#### 10.44.3.2 MessageID

uint16\_t gdcm::network::CEchoRQ::MessageID

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

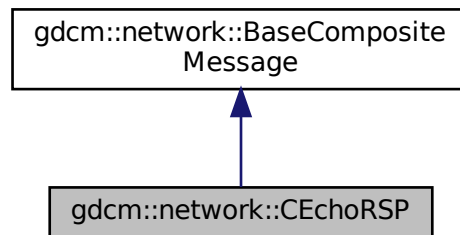
- [gdcmCEchoMessages.h](#)
- [gdcmDIMSE.h](#)

## 10.45 gdcm::network::CEchoRSP Class Reference

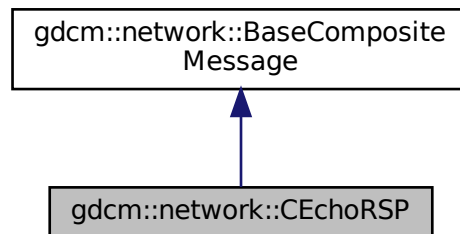
[CEchoRSP](#) this file defines the messages for the cecho action.

```
#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)`

### 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:

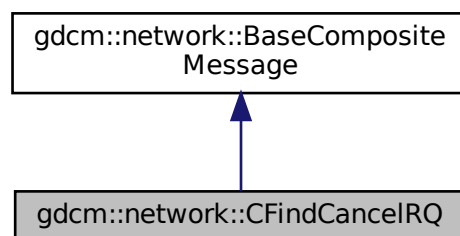
- [gdcMDIMSE.h](#)

## 10.47 gdcM::network::CFindCancelRQ Class Reference

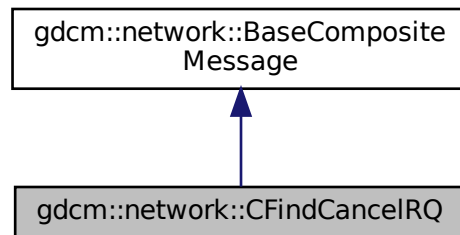
[CFindCancelRQ](#) this file defines the messages for the cfind action.

```
#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)

### 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

### 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:

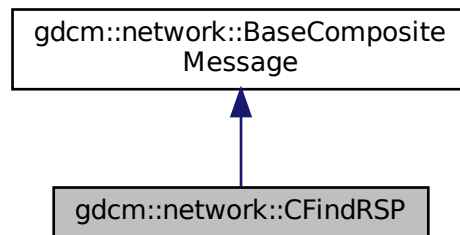
- [gdcmCFindMessages.h](#)

## 10.49 gdcm::network::CFindRSP Class Reference

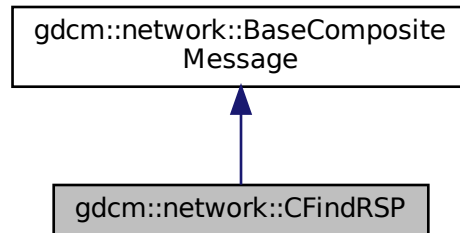
[CFindRSP](#) this file defines the messages for the cfind action.

```
#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`)

### 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:

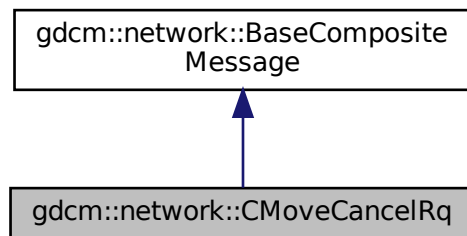
- [gdcmCFindMessages.h](#)



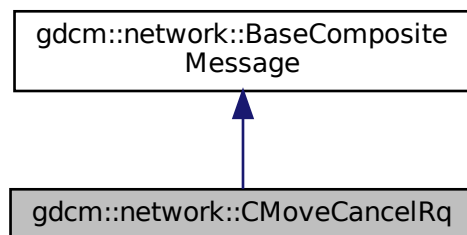
## 10.50 gdcm::network::CMoveCancelRq Class Reference

```
#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)

#### 10.50.1 Member Function Documentation

### 10.50.1.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcm::network::CMoveCancelRq::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

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

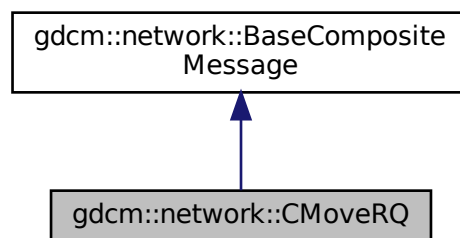
- [gdcmCMoveMessages.h](#)

## 10.51 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

### 10.51.1 Detailed Description

[CMoveRQ](#).

this file defines the messages for the cmove action

### 10.51.2 Member Function Documentation

#### 10.51.2.1 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:

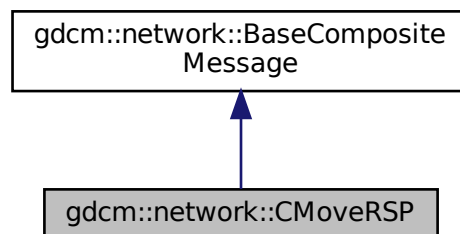
- [gdcmCMoveMessages.h](#)

## 10.52 gdcm::network::CMoveRSP Class Reference

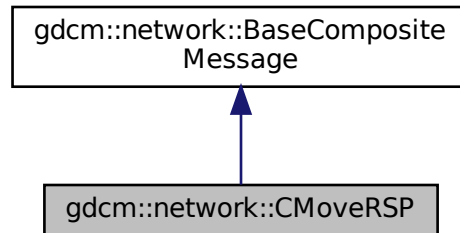
[CMoveRSP](#) this file defines the messages for the cmove action.

```
#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)

### 10.52.1 Detailed Description

[CMoveRSP](#) this file defines the messages for the cmove action.

### 10.52.2 Member Function Documentation

#### 10.52.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcm::network::CMoveRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

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

- [gdcmCMoveMessages.h](#)

## 10.53 gdcmm::Codec Class Reference

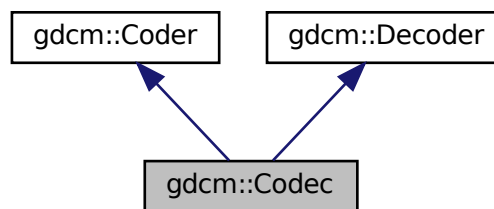
[Codec](#) class.

```
#include <gdcmmCodec.h>
```

Inheritance diagram for gdcmm::Codec:



Collaboration diagram for gdcmm::Codec:



### Additional Inherited Members

#### 10.53.1 Detailed Description

[Codec](#) class.

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

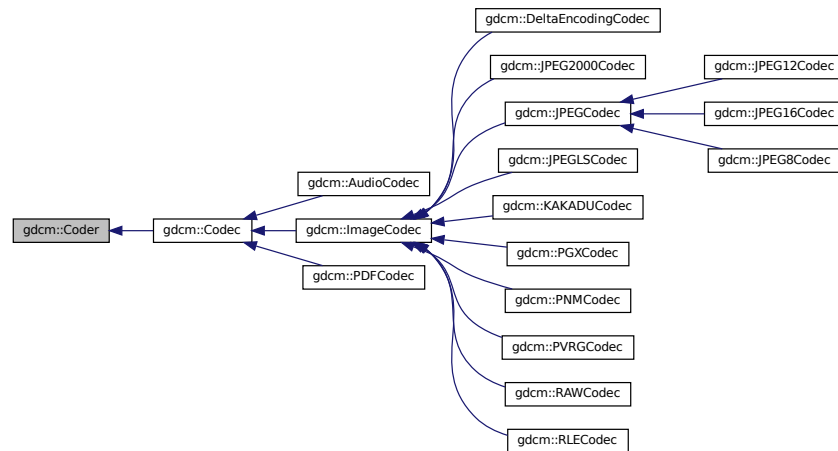
- [gdcmmCodec.h](#)

## 10.54 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)

#### 10.54.1 Detailed Description

[Coder.](#)

#### 10.54.2 Constructor & Destructor Documentation

### 10.54.2.1 ~Coder()

```
virtual gdcm::Coder::~~Coder ( ) [virtual], [default]
```

## 10.54.3 Member Function Documentation

### 10.54.3.1 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::RLECodec](#), [gdcm::RAWCodec](#), [gdcm::PVRGCodec](#), [gdcm::PNMCodec](#), [gdcm::PGXCodec](#), [gdcm::KAKADUCodec](#), [gdcm::JPEGLSCodec](#), [gdcm::JPEGCodec](#), [gdcm::JPEG2000Codec](#), [gdcm::PDFCodec](#), [gdcm::ImageCodec](#), and [gdcm::AudioCodec](#).

### 10.54.3.2 Code()

```
virtual bool gdcm::Coder::Code (
    DataElement const & in_,
    DataElement & out_ ) [inline], [virtual]
```

Code.

Reimplemented in [gdcm::RLECodec](#), [gdcm::RAWCodec](#), [gdcm::PVRGCodec](#), [gdcm::KAKADUCodec](#), [gdcm::JPEGLSCodec](#), [gdcm::JPEGCodec](#), and [gdcm::JPEG2000Codec](#).

### 10.54.3.3 InternalCode()

```
virtual bool gdcm::Coder::InternalCode (
    const char * bv,
    unsigned long len,
    std::ostream & os ) [inline], [protected], [virtual]
```

Reimplemented in [gdcm::JPEG8Codec](#), [gdcm::JPEG16Codec](#), and [gdcm::JPEG12Codec](#).

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

- [gdcmCoder.h](#)

## 10.55 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)



## 10.55.1 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.

## 10.55.2 Member Typedef Documentation

### 10.55.2.1 const\_iterator

```
typedef InternalClass::const_iterator gdcm::CodeString::const_iterator
```

### 10.55.2.2 const\_reference

```
typedef InternalClass::const_reference gdcm::CodeString::const_reference
```

### 10.55.2.3 const\_reverse\_iterator

```
typedef InternalClass::const_reverse_iterator gdcm::CodeString::const_reverse_iterator
```

### 10.55.2.4 difference\_type

```
typedef InternalClass::difference_type gdcm::CodeString::difference_type
```

#### 10.55.2.5 iterator

```
typedef InternalClass::iterator gdcM::CodeString::iterator
```

#### 10.55.2.6 pointer

```
typedef InternalClass::pointer gdcM::CodeString::pointer
```

#### 10.55.2.7 reference

```
typedef InternalClass::reference gdcM::CodeString::reference
```

#### 10.55.2.8 reverse\_iterator

```
typedef InternalClass::reverse_iterator gdcM::CodeString::reverse_iterator
```

#### 10.55.2.9 size\_type

```
typedef InternalClass::size_type gdcM::CodeString::size_type
```

#### 10.55.2.10 value\_type

```
typedef InternalClass::value_type gdcM::CodeString::value_type
```

### 10.55.3 Constructor & Destructor Documentation

### 10.55.3.1 CodeString() [1/4]

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

[CodeString](#) constructors.

### 10.55.3.2 CodeString() [2/4]

```
gdcm::CodeString::CodeString (
    const value\_type * s ) [inline]
```

### 10.55.3.3 CodeString() [3/4]

```
gdcm::CodeString::CodeString (
    const value\_type * s,
    size\_type n ) [inline]
```

### 10.55.3.4 CodeString() [4/4]

```
gdcm::CodeString::CodeString (
    const InternalClass & s,
    size\_type pos = 0,
    size\_type n = InternalClass::npos ) [inline]
```

## 10.55.4 Member Function Documentation

### 10.55.4.1 GetAsString()

```
std::string gdcm::CodeString::GetAsString ( ) const [inline]
```

Return the full code string as std::string.

#### 10.55.4.2 IsValid()

```
bool gdcm::CodeString::IsValid ( ) const
```

Check if [CodeString](#) obj is correct..

#### 10.55.4.3 Size()

```
size_type gdcm::CodeString::Size ( ) const [inline]
```

Return the size of the string.

#### 10.55.4.4 TrimInternal()

```
std::string gdcm::CodeString::TrimInternal ( ) const [inline], [protected]
```

### 10.55.5 Friends And Related Function Documentation

#### 10.55.5.1 operator"!=

```
bool operator!= (
    const CodeString & ref,
    const CodeString & cs ) [friend]
```

#### 10.55.5.2 operator<<

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

### 10.55.5.3 operator==

```
bool operator== (
    const CodeString & ref,
    const CodeString & cs ) [friend]
```

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

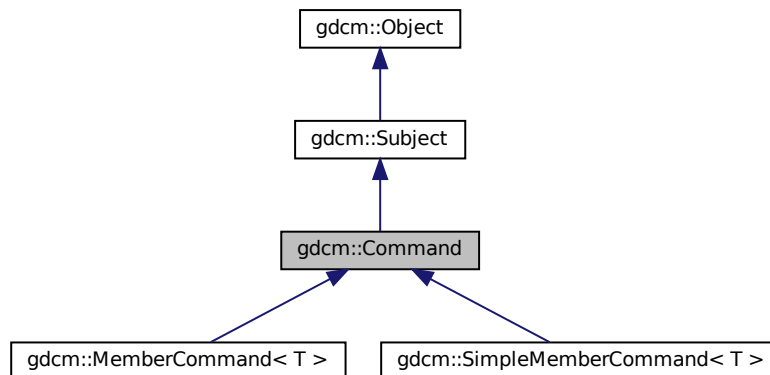
- [gdcmCodeString.h](#)

## 10.56 gdcm::Command Class Reference

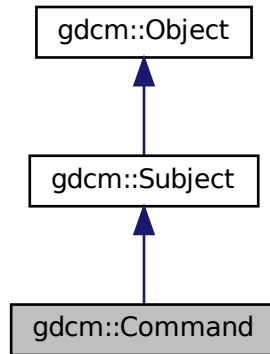
[Command](#) superclass for callback/observer methods.

```
#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

## Protected Member Functions

- `Command` ()
- `~Command` () override

### 10.56.1 Detailed Description

`Command` superclass for callback/observer methods.

See also

[Subject](#)

### 10.56.2 Constructor & Destructor Documentation

### 10.56.2.1 Command() [1/2]

```
gdcmm::Command::Command (
    const Command & ) [delete]
```

### 10.56.2.2 Command() [2/2]

```
gdcmm::Command::Command ( ) [protected]
```

### 10.56.2.3 ~Command()

```
gdcmm::Command::~~Command ( ) [override], [protected]
```

## 10.56.3 Member Function Documentation

### 10.56.3.1 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::MemberCommand< T >](#), and [gdcmm::SimpleMemberCommand< T >](#).

### 10.56.3.2 Execute() [2/2]

```
virtual void gdcmm::Command::Execute (
    Subject * caller,
    const Event & event ) [pure virtual]
```

Abstract method that defines the action to be taken by the command.

Implemented in [gdcmm::MemberCommand< T >](#), and [gdcmm::SimpleMemberCommand< T >](#).

### 10.56.3.3 operator=()

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

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

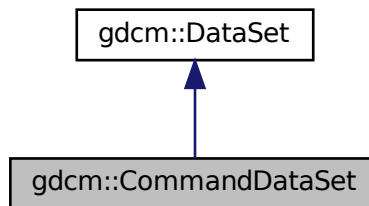
- [gdcMCommand.h](#)

## 10.57 gdcM::CommandDataSet Class Reference

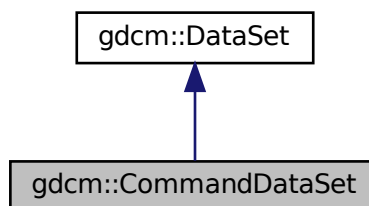
Class to represent a [Command DataSet](#).

```
#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.*

## Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [CommandDataSet](#) &\_val)

## Additional Inherited Members

### 10.57.1 Detailed Description

Class to represent a [Command DataSet](#).

See also

[DataSet](#)

### 10.57.2 Constructor & Destructor Documentation

#### 10.57.2.1 CommandDataSet()

```
gdcm::CommandDataSet::CommandDataSet ( ) [default]
```

#### 10.57.2.2 ~CommandDataSet()

```
gdcm::CommandDataSet::~~CommandDataSet ( ) [default]
```

### 10.57.3 Member Function Documentation

### 10.57.3.1 Insert()

```
void gdcM::CommandDataSet::Insert (
    const DataElement & de ) [inline]
```

References [gdcMErrorMacro](#), [gdcM::Tag::GetGroup\(\)](#), and [gdcM::DataElement::GetTag\(\)](#).

### 10.57.3.2 Read()

```
std::istream& gdcM::CommandDataSet::Read (
    std::istream & is )
```

Read.

### 10.57.3.3 Replace()

```
void gdcM::CommandDataSet::Replace (
    const DataElement & de ) [inline]
```

References [gdcM::DataElement::GetTag\(\)](#).

### 10.57.3.4 Write()

```
std::ostream& gdcM::CommandDataSet::Write (
    std::ostream & os ) const
```

Write.

## 10.57.4 Friends And Related Function Documentation

### 10.57.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const CommandDataSet & _val ) [friend]
```

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

- [gdcMCommandDataSet.h](#)

## 10.58 gdcmm::network::CompositeMessageFactory Class Reference

[CompositeMessageFactory](#).

```
#include <gdcmmCompositeMessageFactory.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)

### 10.58.1 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).

### 10.58.2 Member Function Documentation

#### 10.58.2.1 ConstructCEchoRQ()

```
static std::vector<PresentationDataValue> gdcmm::network::CompositeMessageFactory::ConstructCEchoRQ (
    const ULConnection & inConnection ) [static]
```

#### 10.58.2.2 ConstructCFindRQ()

```
static std::vector<PresentationDataValue> gdcmm::network::CompositeMessageFactory::ConstructCFindRQ (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery ) [static]
```

### 10.58.2.3 ConstructCMoveRQ()

```
static std::vector<PresentationDataValue> gdcM::network::CompositeMessageFactory::ConstructCMoveRQ (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery ) [static]
```

### 10.58.2.4 ConstructCStoreRQ()

```
static std::vector<PresentationDataValue> gdcM::network::CompositeMessageFactory::ConstructCStoreRQ (
    const ULConnection & inConnection,
    const File & file,
    bool writeDataSet = true ) [static]
```

### 10.58.2.5 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](#)

## 10.59 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)

### 10.59.1 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)

### 10.59.2 Member Typedef Documentation

#### 10.59.2.1 KeyValuePairArrayType

```
typedef std::vector< KeyValuePairType > gdcm::CompositeNetworkFunctions::KeyValuePairArrayType
```

#### 10.59.2.2 KeyValuePairType

```
typedef std::pair<Tag, std::string> gdcm::CompositeNetworkFunctions::KeyValuePairType
```

### 10.59.3 Member Function Documentation

#### 10.59.3.1 CEcho()

```
static bool gdcml::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.

#### 10.59.3.2 CFind()

```
static bool gdcml::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.

**10.59.3.3 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.

**10.59.3.4 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)
------------------	--

**10.59.3.5 ConstructQuery() [2/2]**

```
static BaseRootQuery* gdcmm::CompositeNetworkFunctions::ConstructQuery (
    ERootType inRootType,
    EQueryLevel inQueryLevel,
    const KeyValuePairArrayType & keys,
    EQueryType queryType = eFind ) [static]
```

**Deprecated****10.59.3.6 CStore()**

```
static bool gdcmm::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

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

- [gdcmCompositeNetworkFunctions.h](#)

## 10.60 gdcm::ConstCharWrapper Class Reference

Do not use me.

```
#include <gdcmConstCharWrapper.h>
```

### Public Member Functions

- [ConstCharWrapper](#) (const char \*i=0)
- [operator const char \\*](#) () const

#### 10.60.1 Detailed Description

Do not use me.

#### 10.60.2 Constructor & Destructor Documentation

##### 10.60.2.1 ConstCharWrapper()

```
gdcm::ConstCharWrapper::ConstCharWrapper (  
    const char * i = 0 ) [inline]
```

#### 10.60.3 Member Function Documentation

##### 10.60.3.1 operator const char \*()

```
gdcm::ConstCharWrapper::operator const char * ( ) const [inline]
```

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

- [gdcmConstCharWrapper.h](#)

## 10.61 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)`

## Additional Inherited Members

### 10.61.1 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

### 10.61.2 Member Function Documentation

#### 10.61.2.1 GetLength()

```
VL gdcm::CP246ExplicitDataElement::GetLength ( ) const
```

#### 10.61.2.2 Read()

```
template<typename TSwap >  
std::istream& gdcm::CP246ExplicitDataElement::Read (  
    std::istream & is )
```

#### 10.61.2.3 ReadPreValue()

```
template<typename TSwap >  
std::istream& gdcm::CP246ExplicitDataElement::ReadPreValue (  
    std::istream & is )
```

#### 10.61.2.4 ReadValue()

```
template<typename TSwap >
std::istream& gdcm::CP246ExplicitDataElement::ReadValue (
    std::istream & is,
    bool readvalues = true )
```

#### 10.61.2.5 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:

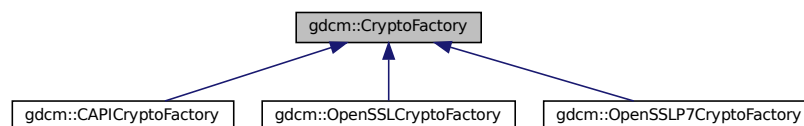
- [gdcmCP246ExplicitDataElement.h](#)

## 10.62 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

### 10.62.1 Detailed Description

Class to do handle the crypto factory.

GDCM needs to access in a platform independant 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.

### 10.62.2 Member Enumeration Documentation

#### 10.62.2.1 CryptoLib

enum [gdcmm::CryptoFactory::CryptoLib](#)

Enumerator

DEFAULT	
OPENSSL	
CAPI	
OPENSSL7	

## 10.62.3 Constructor & Destructor Documentation

### 10.62.3.1 CryptoFactory() [1/2]

```
gdcM::CryptoFactory::CryptoFactory (
    CryptoLib id ) [inline], [protected]
```

### 10.62.3.2 CryptoFactory() [2/2]

```
gdcM::CryptoFactory::CryptoFactory ( ) [protected], [default]
```

### 10.62.3.3 ~CryptoFactory()

```
gdcM::CryptoFactory::~~CryptoFactory ( ) [protected], [default]
```

## 10.62.4 Member Function Documentation

### 10.62.4.1 CreateCMSProvider()

```
virtual CryptographicMessageSyntax* gdcM::CryptoFactory::CreateCMSProvider ( ) [pure virtual]
```

Implemented in [gdcM::OpenSSLP7CryptoFactory](#), [gdcM::OpenSSLCryptoFactory](#), and [gdcM::CAPICryptoFactory](#).

### 10.62.4.2 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](#)

## 10.63 gdcmm::CryptographicMessageSyntax Class Reference

```
#include <gdcmmCryptographicMessageSyntax.h>
```

Inheritance diagram for gdcmm::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

### 10.63.1 Member Enumeration Documentation

#### 10.63.1.1 CipherTypes

```
enum gdcmm::CryptographicMessageSyntax::CipherTypes
```

## Enumerator

DES3_CIPHER	
AES128_CIPHER	
AES192_CIPHER	
AES256_CIPHER	

## 10.63.2 Constructor & Destructor Documentation

### 10.63.2.1 CryptographicMessageSyntax() [1/2]

```
gdcM::CryptographicMessageSyntax::CryptographicMessageSyntax ( ) [default]
```

### 10.63.2.2 ~CryptographicMessageSyntax()

```
virtual gdcM::CryptographicMessageSyntax::~~CryptographicMessageSyntax ( ) [virtual], [default]
```

### 10.63.2.3 CryptographicMessageSyntax() [2/2]

```
gdcM::CryptographicMessageSyntax::CryptographicMessageSyntax (
    const CryptographicMessageSyntax & ) [delete]
```

## 10.63.3 Member Function Documentation

### 10.63.3.1 Decrypt()

```
virtual bool gdcM::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 [gdcM::OpenSSLP7CryptographicMessageSyntax](#), [gdcM::OpenSSLCryptographicMessageSyntax](#), and [gdcM::CAPICryptographicMessageSyntax](#).



### 10.63.3.2 Encrypt()

```
virtual bool gdcmm::CryptographicMessageSyntax::Encrypt (
    char * output,
    size_t & outlen,
    const char * array,
    size_t len ) const [pure virtual]
```

create a CMS envelopedData structure

Implemented in [gdcmm::OpenSSLP7CryptographicMessageSyntax](#), [gdcmm::OpenSSLCryptographicMessageSyntax](#), and [gdcmm::CAPICryptographicMessageSyntax](#).

### 10.63.3.3 GetCipherType()

```
virtual CipherTypes gdcmm::CryptographicMessageSyntax::GetCipherType ( ) const [pure virtual]
```

Implemented in [gdcmm::OpenSSLP7CryptographicMessageSyntax](#), [gdcmm::OpenSSLCryptographicMessageSyntax](#), and [gdcmm::CAPICryptographicMessageSyntax](#).

### 10.63.3.4 operator=()

```
void gdcmm::CryptographicMessageSyntax::operator= (
    const CryptographicMessageSyntax & ) [delete]
```

### 10.63.3.5 ParseCertificateFile()

```
virtual bool gdcmm::CryptographicMessageSyntax::ParseCertificateFile (
    const char * filename ) [pure virtual]
```

Implemented in [gdcmm::OpenSSLP7CryptographicMessageSyntax](#), [gdcmm::OpenSSLCryptographicMessageSyntax](#), and [gdcmm::CAPICryptographicMessageSyntax](#).

### 10.63.3.6 ParseKeyFile()

```
virtual bool gdcmm::CryptographicMessageSyntax::ParseKeyFile (
    const char * filename ) [pure virtual]
```

Implemented in [gdcmm::OpenSSLP7CryptographicMessageSyntax](#), [gdcmm::OpenSSLCryptographicMessageSyntax](#), and [gdcmm::CAPICryptographicMessageSyntax](#).

### 10.63.3.7 SetCipherType()

```
virtual void gdcm::CryptographicMessageSyntax::SetCipherType (
    CipherTypes type ) [pure virtual]
```

Implemented in [gdcm::OpenSSLP7CryptographicMessageSyntax](#), [gdcm::OpenSSLCryptographicMessageSyntax](#), and [gdcm::CAPICryptographicMessageSyntax](#).

### 10.63.3.8 SetPassword()

```
virtual bool gdcm::CryptographicMessageSyntax::SetPassword (
    const char * pass,
    size_t passLen ) [pure virtual]
```

Implemented in [gdcm::OpenSSLCryptographicMessageSyntax](#), [gdcm::CAPICryptographicMessageSyntax](#), and [gdcm::OpenSSLP7CryptographicMessageSyntax](#).

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

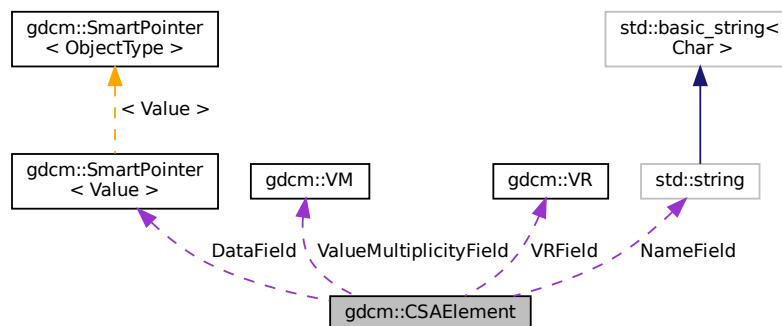
- [gdcmCryptographicMessageSyntax.h](#)

## 10.64 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)`

### 10.64.1 Detailed Description

Class to represent a CSA [Element](#).

See also

[CSAHeader](#)

Examples

[DumpSiemensBase64.cxx](#), [MrProtocol.cxx](#), and [csa2img.cxx](#).

### 10.64.2 Member Typedef Documentation

#### 10.64.2.1 DataPtr

```
typedef SmartPointer<Value> gdcm::CSAElement::DataPtr [protected]
```

### 10.64.3 Constructor & Destructor Documentation

#### 10.64.3.1 CSAElement() [1/2]

```
gdcm::CSAElement::CSAElement (  
    unsigned int kf = 0 ) [inline]
```

#### 10.64.3.2 CSAElement() [2/2]

```
gdcm::CSAElement::CSAElement (  
    const CSAElement & _val ) [inline]
```

## 10.64.4 Member Function Documentation

### 10.64.4.1 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](#).

### 10.64.4.2 GetKey()

```
unsigned int gdcm::CSAElement::GetKey ( ) const [inline]
```

Set/Get Key.

Referenced by operator<().

### 10.64.4.3 GetName()

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

Set/Get Name.

### 10.64.4.4 GetNoOfItems()

```
unsigned int gdcm::CSAElement::GetNoOfItems ( ) const [inline]
```

Set/Get NoOfItems.

#### 10.64.4.5 GetSyngoDT()

```
unsigned int gdcm::CSAElement::GetSyngoDT ( ) const [inline]
```

Set/Get SyngoDT.

#### 10.64.4.6 GetValue() [1/2]

```
Value& gdcm::CSAElement::GetValue ( ) [inline]
```

#### 10.64.4.7 GetValue() [2/2]

```
Value const& gdcm::CSAElement::GetValue ( ) const [inline]
```

Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):

#### Examples

[csa2img.cxx](#).

#### 10.64.4.8 GetVM()

```
const VM& gdcm::CSAElement::GetVM ( ) const [inline]
```

Set/Get [VM](#).

#### 10.64.4.9 GetVR()

```
VR const& gdcm::CSAElement::GetVR ( ) const [inline]
```

Set/Get [VR](#).

#### 10.64.4.10 IsEmpty()

```
bool gdcm::CSAElement::IsEmpty ( ) const [inline]
```

Check if CSA [Element](#) is empty.

##### Examples

[csa2img.cxx](#).

#### 10.64.4.11 operator<()

```
bool gdcm::CSAElement::operator< (
    const CSAElement & de ) const [inline]
```

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

#### 10.64.4.12 operator=()

```
CSAElement& gdcm::CSAElement::operator= (
    const CSAElement & de ) [default]
```

#### 10.64.4.13 operator==()

```
bool gdcm::CSAElement::operator== (
    const CSAElement & de ) const [inline]
```

References [KeyField](#), [NameField](#), [SyngoDTField](#), [ValueMultiplicityField](#), and [VRField](#).

#### 10.64.4.14 SetByteValue()

```
void gdcm::CSAElement::SetByteValue (
    const char * array,
    VL length ) [inline]
```

Set.

**10.64.4.15 SetKey()**

```
void gdcM::CSAElement::SetKey (
    unsigned int key ) [inline]
```

**10.64.4.16 SetName()**

```
void gdcM::CSAElement::SetName (
    const char * name ) [inline]
```

**10.64.4.17 SetNoOfItems()**

```
void gdcM::CSAElement::SetNoOfItems (
    unsigned int items ) [inline]
```

**10.64.4.18 SetSyngoDT()**

```
void gdcM::CSAElement::SetSyngoDT (
    unsigned int syngodt ) [inline]
```

**10.64.4.19 SetValue()**

```
void gdcM::CSAElement::SetValue (
    Value const & vl ) [inline]
```

**10.64.4.20 SetVM()**

```
void gdcM::CSAElement::SetVM (
    const VM & vm ) [inline]
```



#### 10.64.4.21 SetVR()

```
void gdcmm::CSAElement::SetVR (
    VR const & vr ) [inline]
```

### 10.64.5 Friends And Related Function Documentation

#### 10.64.5.1 operator<<

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

### 10.64.6 Member Data Documentation

#### 10.64.6.1 DataField

```
DataPtr gdcmm::CSAElement::DataField [protected]
```

#### 10.64.6.2 KeyField

```
unsigned int gdcmm::CSAElement::KeyField [protected]
```

Referenced by operator==().

#### 10.64.6.3 NameField

```
std::string gdcmm::CSAElement::NameField [protected]
```

Referenced by operator==().

#### 10.64.6.4 NoOfItemsField

```
unsigned int gdcm::CSAElement::NoOfItemsField [protected]
```

#### 10.64.6.5 SyngoDTField

```
unsigned int gdcm::CSAElement::SyngoDTField [protected]
```

Referenced by operator==().

#### 10.64.6.6 ValueMultiplicityField

```
VM gdcm::CSAElement::ValueMultiplicityField [protected]
```

Referenced by operator==().

#### 10.64.6.7 VRField

```
VR gdcm::CSAElement::VRField [protected]
```

Referenced by operator==().

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

- [gdcmCSAElement.h](#)

## 10.65 gdcm::CSAHeader Class Reference

Class for [CSAHeader](#).

```
#include <gdcmCSAHeader.h>
```

### Public Types

- enum [CSAHeaderType](#) {  
    [UNKNOWN](#) = 0 ,  
    [SV10](#) ,  
    [NOMAGIC](#) ,  
    [DATASET\\_FORMAT](#) ,  
    [INTERFILE](#) ,  
    [ZEROED\\_OUT](#) }

*Divers 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)

### 10.65.1 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**

[DumpSiemensBase64.cxx](#), [MrProtocol.cxx](#), and [csa2img.cxx](#).

**10.65.2 Member Enumeration Documentation****10.65.2.1 CSAHeaderType**

enum `gdcm::CSAHeader::CSAHeaderType`

Divers format of [CSAHeader](#) as found 'in the wild'.

**Enumerator**

UNKNOWN	
SV10	
NOMAGIC	
DATASET_FORMAT	
INTERFILE	
ZEROED_OUT	

**10.65.3 Constructor & Destructor Documentation**

### 10.65.3.1 CSAHeader()

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

### 10.65.3.2 ~CSAHeader()

```
gdcm::CSAHeader::~~CSAHeader ( ) [default]
```

## 10.65.4 Member Function Documentation

### 10.65.4.1 FindCSAELEMENTByName()

```
bool gdcm::CSAHeader::FindCSAELEMENTByName (
    const char * name )
```

Return true if the CSA element matching 'name' is found or not

#### Warning

Case Sensitive

#### Examples

[DumpSiemensBase64.cxx](#), [MrProtocol.cxx](#), and [csa2img.cxx](#).

### 10.65.4.2 GetCSADATAInfo()

```
static const PrivateTag& gdcm::CSAHeader::GetCSADATAInfo ( ) [static]
```

Return the private tag used by SIEMENS to store the CSA Data Info This is: [PrivateTag](#)(0x0029,0x0010,"SIEMENS CSA NON-IMAGE");

### 10.65.4.3 GetCSAEEnd()

```
const CSAElement& gdcm::CSAHeader::GetCSAEEnd ( ) const [protected]
```

#### 10.65.4.4 GetCSAElementByName()

```
const CSAElement& gdcM::CSAHeader::GetCSAElementByName (
    const char * name )
```

Return the [CSAElement](#) corresponding to name 'name'

##### Warning

Case Sensitive

##### Examples

[DumpSiemensBase64.cxx](#), [MrProtocol.cxx](#), and [csa2img.cxx](#).

#### 10.65.4.5 GetCSAImageHeaderInfoTag()

```
static const PrivateTag& gdcM::CSAHeader::GetCSAImageHeaderInfoTag ( ) [static]
```

Return the private tag used by SIEMENS to store the CSA [Image](#) Header This is: [PrivateTag](#)(0x0029,0x0010,"SIEMENS CSA HEADER");

##### Examples

[DumpSiemensBase64.cxx](#), [PublicDict.cxx](#), and [csa2img.cxx](#).

#### 10.65.4.6 GetCSASeriesHeaderInfoTag()

```
static const PrivateTag& gdcM::CSAHeader::GetCSASeriesHeaderInfoTag ( ) [static]
```

Return the private tag used by SIEMENS to store the CSA [Series](#) Header This is: [PrivateTag](#)(0x0029,0x0020,"SIEMENS CSA HEADER");

##### Examples

[MrProtocol.cxx](#).

#### 10.65.4.7 GetDataSet()

```
const DataSet& gdcm::CSAHeader::GetDataSet ( ) const [inline]
```

Return the [DataSet](#) output (use only if Format == DATASET\_FORMAT )

#### 10.65.4.8 GetFormat()

```
CSAHeaderType gdcm::CSAHeader::GetFormat ( ) const
```

return the format of the [CSAHeader](#) SV10 and NOMAGIC are equivalent.

#### 10.65.4.9 GetInterfile()

```
const char* gdcm::CSAHeader::GetInterfile ( ) const [inline]
```

Return the string output (use only if Format == Interfile)

#### 10.65.4.10 GetMrProtocol()

```
bool gdcm::CSAHeader::GetMrProtocol (
    const DataSet & ds,
    MrProtocol & mrProtocol )
```

Retrieve the ASCII portion stored within the MrProtocol/MrPhoenixProtocol:

##### Examples

[MrProtocol.cxx](#).

#### 10.65.4.11 LoadFromDataElement()

```
bool gdcm::CSAHeader::LoadFromDataElement (
    DataElement const & de )
```

Decode the [CSAHeader](#) from element 'de'.

##### Examples

[DumpSiemensBase64.cxx](#), [MrProtocol.cxx](#), and [csa2img.cxx](#).

#### 10.65.4.12 Print()

```
void gdcM::CSAHeader::Print (
    std::ostream & os ) const
```

Print the [CSAHeader](#) (use only if Format == SV10 or NOMAGIC)

#### Examples

[csa2img.cxx](#).

### 10.65.5 Friends And Related Function Documentation

#### 10.65.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const CSAHeader & d ) [friend]
```

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

- [gdcMCSAHeader.h](#)

## 10.66 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)

### 10.66.1 Detailed Description

Class to represent a map of [CSAHeaderDictEntry](#).

#### Examples

[MrProtocol.cxx](#).

### 10.66.2 Member Typedef Documentation

#### 10.66.2.1 ConstIterator

```
typedef MapCSAHeaderDictEntry::const_iterator gdcm::CSAHeaderDict::ConstIterator
```

#### 10.66.2.2 Iterator

```
typedef MapCSAHeaderDictEntry::iterator gdcm::CSAHeaderDict::Iterator
```

#### 10.66.2.3 MapCSAHeaderDictEntry

```
typedef std::set<CSAHeaderDictEntry> gdcm::CSAHeaderDict::MapCSAHeaderDictEntry
```

### 10.66.3 Constructor & Destructor Documentation

### 10.66.3.1 CSAHeaderDict() [1/2]

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

### 10.66.3.2 CSAHeaderDict() [2/2]

```
gdcm::CSAHeaderDict::CSAHeaderDict (
    const CSAHeaderDict & _val ) [delete]
```

## 10.66.4 Member Function Documentation

### 10.66.4.1 AddCSAHeaderDictEntry()

```
void gdcm::CSAHeaderDict::AddCSAHeaderDictEntry (
    const CSAHeaderDictEntry & de ) [inline]
```

### 10.66.4.2 Begin()

```
ConstIterator gdcm::CSAHeaderDict::Begin ( ) const [inline]
```

### 10.66.4.3 End()

```
ConstIterator gdcm::CSAHeaderDict::End ( ) const [inline]
```

### 10.66.4.4 GetCSAHeaderDictEntry()

```
const CSAHeaderDictEntry& gdcm::CSAHeaderDict::GetCSAHeaderDictEntry (
    const char * name ) const [inline]
```

## Examples

[MrProtocol.cxx](#).

#### 10.66.4.5 IsEmpty()

```
bool gdcm::CSAHeaderDict::IsEmpty ( ) const [inline]
```

#### 10.66.4.6 LoadDefault()

```
void gdcm::CSAHeaderDict::LoadDefault ( ) [protected]
```

#### 10.66.4.7 operator=()

```
CSAHeaderDict& gdcm::CSAHeaderDict::operator= (
    const CSAHeaderDict & _val ) [delete]
```

### 10.66.5 Friends And Related Function Documentation

#### 10.66.5.1 Dicts

```
friend class Dicts [friend]
```

#### 10.66.5.2 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const CSAHeaderDict & _val ) [friend]
```

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

- [gdcmCSAHeaderDict.h](#)

## 10.67 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)

### 10.67.1 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](#).

### 10.67.2 Constructor & Destructor Documentation

### 10.67.2.1 CSAHeaderDictEntry()

```
gdcm::CSAHeaderDictEntry::CSAHeaderDictEntry (
    const char * name = "",
    VR const & vr = VR::INVALID,
    VM const & vm = VM::VMO,
    const char * desc = "" ) [inline]
```

## 10.67.3 Member Function Documentation

### 10.67.3.1 GetDescription()

```
const char* gdcm::CSAHeaderDictEntry::GetDescription ( ) const [inline]
```

Set/Get Description.

### 10.67.3.2 GetName()

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

Set/Get Name.

Referenced by operator<().

### 10.67.3.3 GetVM()

```
const VM& gdcm::CSAHeaderDictEntry::GetVM ( ) const [inline]
```

Set/Get VM.

### 10.67.3.4 GetVR()

```
const VR& gdcm::CSAHeaderDictEntry::GetVR ( ) const [inline]
```

Set/Get VR.

### 10.67.3.5 operator<()

```
bool gdcm::CSAHeaderDictEntry::operator< (
    const CSAHeaderDictEntry & entry ) const [inline]
```

References GetName().

### 10.67.3.6 SetDescription()

```
void gdcm::CSAHeaderDictEntry::SetDescription (
    const char * desc ) [inline]
```

### 10.67.3.7 SetName()

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

### 10.67.3.8 SetVM()

```
void gdcm::CSAHeaderDictEntry::SetVM (
    VM const & vm ) [inline]
```

### 10.67.3.9 SetVR()

```
void gdcm::CSAHeaderDictEntry::SetVR (
    const VR & vr ) [inline]
```

## 10.67.4 Friends And Related Function Documentation

#### 10.67.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const CSAHeaderDictEntry & _val ) [friend]
```

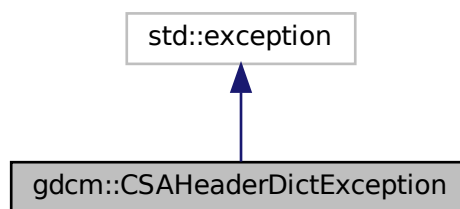
The documentation for this class was generated from the following file:

- [gdcmCSAHeaderDictEntry.h](#)

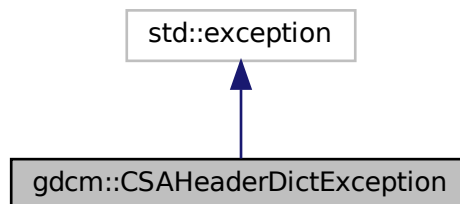
## 10.68 gdcm::CSAHeaderDictException Class Reference

```
#include <gdcmCSAHeaderDict.h>
```

Inheritance diagram for gdcm::CSAHeaderDictException:



Collaboration diagram for gdcm::CSAHeaderDictException:



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

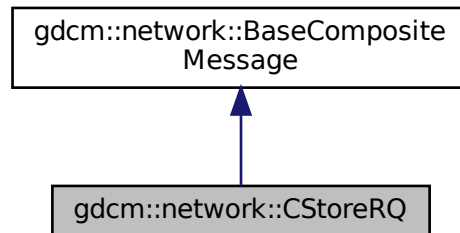
- [gdcmCSAHeaderDict.h](#)

## 10.69 gdcm::network::CStoreRQ Class Reference

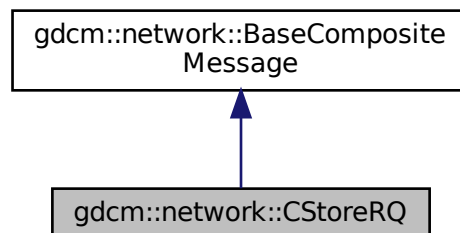
[CStoreRQ](#).

```
#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)`

### 10.69.1 Detailed Description

[CStoreRQ](#).

this file defines the messages for the cecho action



## 10.69.2 Member Function Documentation

### 10.69.2.1 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:

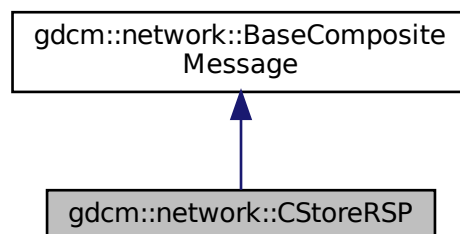
- [gdcmCStoreMessages.h](#)

## 10.70 gdcm::network::CStoreRSP Class Reference

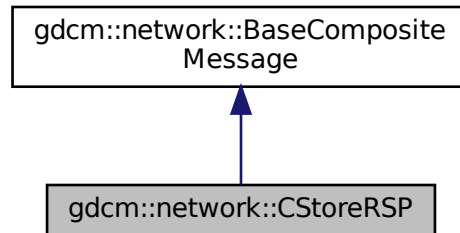
[CStoreRSP](#) this file defines the messages for the cecho action.

```
#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)

### 10.70.1 Detailed Description

[CStoreRSP](#) 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::CStoreRSP::ConstructPDV (
    const DataSet * inDataSet,
    const BasePDU * inPC )
```

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

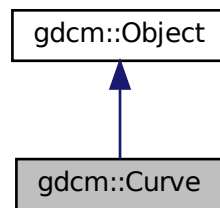
- [gdcmCStoreMessages.h](#)

## 10.71 gdcm::Curve Class Reference

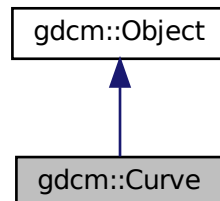
[Curve](#) class to handle element 50xx,3000 [Curve](#) Data.

```
#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)

## Static Public Member Functions

- static unsigned int [GetNumberOfCurves](#) ([DataSet](#) const &ds)

## Additional Inherited Members

### 10.71.1 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

### 10.71.2 Constructor & Destructor Documentation

### 10.71.2.1 Curve() [1/2]

```
gdcm::Curve::Curve ( )
```

### 10.71.2.2 ~Curve()

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

### 10.71.2.3 Curve() [2/2]

```
gdcm::Curve::Curve (
    Curve const & ov )
```

## 10.71.3 Member Function Documentation

### 10.71.3.1 Decode()

```
void gdcm::Curve::Decode (
    std::istream & is,
    std::ostream & os )
```

### 10.71.3.2 GetAsPoints()

```
void gdcm::Curve::GetAsPoints (
    float * array ) const
```

### 10.71.3.3 GetCurveDataDescriptor()

```
std::vector<unsigned short> const& gdcm::Curve::GetCurveDataDescriptor ( ) const
```

#### 10.71.3.4 GetDataValueRepresentation()

```
unsigned short gdcm::Curve::GetDataValueRepresentation ( ) const
```

#### 10.71.3.5 GetDimensions()

```
unsigned short gdcm::Curve::GetDimensions ( ) const
```

#### 10.71.3.6 GetGroup()

```
unsigned short gdcm::Curve::GetGroup ( ) const
```

#### 10.71.3.7 GetNumberOfCurves()

```
static unsigned int gdcm::Curve::GetNumberOfCurves (
    DataSet const & ds ) [static]
```

#### 10.71.3.8 GetNumberOfPoints()

```
unsigned short gdcm::Curve::GetNumberOfPoints ( ) const
```

#### 10.71.3.9 GetTypeInfoData()

```
const char* gdcm::Curve::GetTypeInfoData ( ) const
```

#### 10.71.3.10 GetTypeInfoDataDescription()

```
const char* gdcm::Curve::GetTypeInfoDataDescription ( ) const
```

### 10.71.3.11 IsEmpty()

```
bool gdcm::Curve::IsEmpty ( ) const
```

### 10.71.3.12 Print()

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

Reimplemented from [gdcm::Object](#).

### 10.71.3.13 SetCoordinateStartValue()

```
void gdcm::Curve::SetCoordinateStartValue (
    unsigned short v )
```

### 10.71.3.14 SetCoordinateStepValue()

```
void gdcm::Curve::SetCoordinateStepValue (
    unsigned short v )
```

### 10.71.3.15 SetCurve()

```
void gdcm::Curve::SetCurve (
    const char * array,
    unsigned int length )
```

### 10.71.3.16 SetCurveDataDescriptor()

```
void gdcm::Curve::SetCurveDataDescriptor (
    const uint16_t * values,
    size_t num )
```

**10.71.3.17 SetCurveDescription()**

```
void gdcM::Curve::SetCurveDescription (
    const char * curvedescription )
```

**10.71.3.18 SetDataValueRepresentation()**

```
void gdcM::Curve::SetDataValueRepresentation (
    unsigned short datavaluerepresentation )
```

**10.71.3.19 SetDimensions()**

```
void gdcM::Curve::SetDimensions (
    unsigned short dimensions )
```

**10.71.3.20 SetGroup()**

```
void gdcM::Curve::SetGroup (
    unsigned short group )
```

**10.71.3.21 SetNumberOfPoints()**

```
void gdcM::Curve::SetNumberOfPoints (
    unsigned short numberofpoints )
```

**10.71.3.22 SetTypeOfData()**

```
void gdcM::Curve::SetTypeOfData (
    const char * typeofdata )
```



### 10.71.3.23 Update()

```
void gdcm::Curve::Update (
    const DataElement & de )
```

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

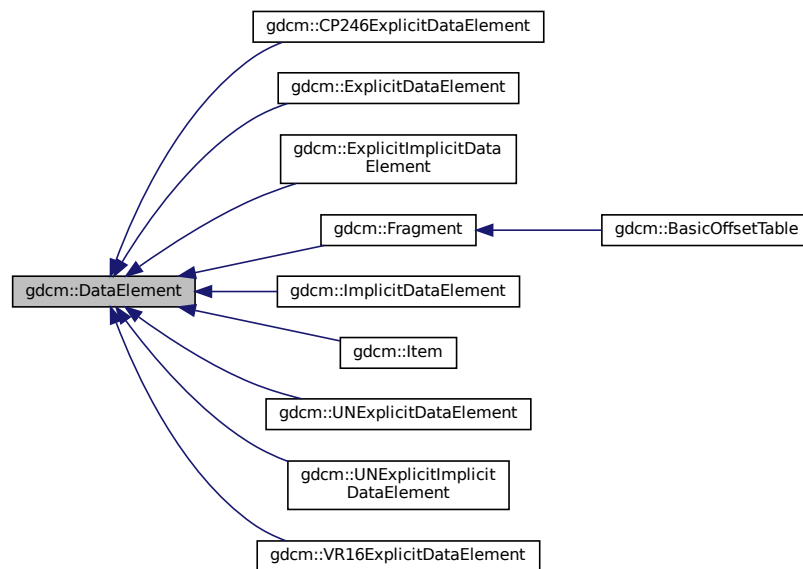
- [gdcmCurve.h](#)

## 10.72 gdcm::DataElement Class Reference

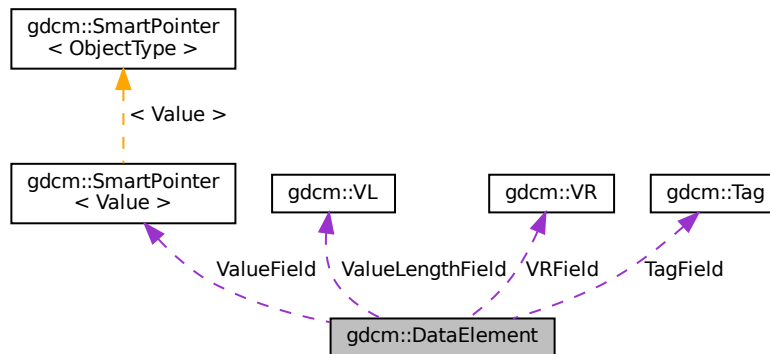
Class to represent a Data [Element](#) either Implicit or Explicit.

```
#include <gdcmDataElement.h>
```

Inheritance diagram for gdcm::DataElement:



Collaboration diagram for gdcm::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)

### 10.72.1 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

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [StreamImageReaderTest.cxx](#), [csa2img.cxx](#), [gdcmrtonplan.cxx](#), [gdcmrtpian.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

### 10.72.2 Member Typedef Documentation

#### 10.72.2.1 ValuePtr

```
typedef SmartPointer<Value> gdcmm::DataElement::ValuePtr [protected]
```

### 10.72.3 Constructor & Destructor Documentation

**10.72.3.1 DataElement() [1/2]**

```
gdcm::DataElement::DataElement (
    const Tag & t = Tag(0),
    const VL & vl = 0,
    const VR & vr = VR::INVALID ) [inline]
```

**10.72.3.2 DataElement() [2/2]**

```
gdcm::DataElement::DataElement (
    const DataElement & _val ) [inline]
```

**10.72.4 Member Function Documentation****10.72.4.1 Clear()**

```
void gdcm::DataElement::Clear ( ) [inline]
```

Clear Data [Element](#) (make [Value](#) empty and invalidate [Tag](#) & [VR](#))

**10.72.4.2 Empty()**

```
void gdcm::DataElement::Empty ( ) [inline]
```

Make Data [Element](#) empty (no [Value](#))

**10.72.4.3 GetByteValue()**

```
const ByteValue* gdcm::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](#), [DumpToshibaDTI.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAI Bug JPEGLS.cxx](#), [PatchFile.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

Referenced by `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()`.

#### 10.72.4.4 GetLength()

```
template<typename TDE >
VL gdcmm::DataElement::GetLength ( ) const [inline]
```

#### 10.72.4.5 GetSequenceOfFragments() [1/2]

```
SequenceOfFragments* gdcmm::DataElement::GetSequenceOfFragments ( )
```

#### 10.72.4.6 GetSequenceOfFragments() [2/2]

```
const SequenceOfFragments* gdcmm::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

[FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), and [GetJPEGSamplePrecision.cxx](#).

#### 10.72.4.7 GetTag() [1/2]

```
Tag& gdcmm::DataElement::GetTag ( ) [inline]
```

#### 10.72.4.8 GetTag() [2/2]

```
const Tag& gdcmm::DataElement::GetTag ( ) const [inline]
```

Get [Tag](#).

##### Examples

[DumpGEMSMovieGroup.cxx](#), [DuplicatePCDE.cxx](#), and [pmsct\\_rgb1.cxx](#).

Referenced by [gdcmm::DataSet::Insert\(\)](#), [gdcmm::FileMetaInformation::Insert\(\)](#), [gdcmm::CommandDataSet::Insert\(\)](#), [operator<\(\)](#), [gdcmm::SequenceOfItems::Read\(\)](#), [gdcmm::SequenceOfFragments::ReadValue\(\)](#), [gdcmm::FileMetaInformation::Replace\(\)](#), [gdcmm::CommandDataSet::Replace\(\)](#), [gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement\(\)](#), [gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement\(\)](#), and [gdcmm::Attribute< Group, Element, TVR, VM::VM1\\_n >::SetFromDataElement\(\)](#).

#### 10.72.4.9 GetValue() [1/2]

```
Value& gdcm::DataElement::GetValue ( ) [inline]
```

References [gdcmAssertAlwaysMacro](#).

#### 10.72.4.10 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\(\)](#).

#### 10.72.4.11 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](#), [DumpVisusChange.cxx](#), [ExtractEncryptedContent.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [gdcmrtonplan.cxx](#), and [gdcmrtpplan.cxx](#).

**10.72.4.12 GetVL()** [1/2]

```
VL& gdcM::DataElement::GetVL ( ) [inline]
```

**10.72.4.13 GetVL()** [2/2]

```
const VL& gdcM::DataElement::GetVL ( ) const [inline]
```

Get [VL](#).

Referenced by [gdcM::DataSet::InsertDataElement\(\)](#), [gdcM::SequenceOfItems::Read\(\)](#), and [gdcM::SequenceOfFragments::ReadValue\(\)](#).

**10.72.4.14 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\(\)](#).

**10.72.4.15 IsEmpty()**

```
bool gdcM::DataElement::IsEmpty ( ) const [inline]
```

Check if Data [Element](#) is empty.

**Examples**

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpToshibaDTI.cxx](#), [ELSCINT1WaveToText.cxx](#), [FixJAIBugJPEGs.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\(\)](#).



#### 10.72.4.16 IsUndefinedLength()

```
bool gdcm::DataElement::IsUndefinedLength ( ) const [inline]
```

return if [Value](#) Length if of undefined length

#### 10.72.4.17 operator<()

```
bool gdcm::DataElement::operator< (
    const DataElement & de ) const [inline]
```

References GetTag().

#### 10.72.4.18 operator=()

```
DataElement& gdcm::DataElement::operator= (
    const DataElement & ) [default]
```

#### 10.72.4.19 operator==( )

```
bool gdcm::DataElement::operator== (
    const DataElement & de ) const [inline]
```

References TagField, ValueField, ValueLengthField, and VRField.

#### 10.72.4.20 Read()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataElement::Read (
    std::istream & is ) [inline]
```

#### Examples

[DumpSiemensBase64.cxx](#).

#### 10.72.4.21 ReadOrSkip()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataElement::ReadOrSkip (
    std::istream & is,
    std::set< Tag > const & skiptags ) [inline]
```

#### 10.72.4.22 ReadPreValue()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataElement::ReadPreValue (
    std::istream & is,
    std::set< Tag > const & skiptags ) [inline]
```

#### 10.72.4.23 ReadValue()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataElement::ReadValue (
    std::istream & is,
    std::set< Tag > const & skiptags ) [inline]
```

#### 10.72.4.24 ReadValueWithLength()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataElement::ReadValueWithLength (
    std::istream & is,
    VL & length,
    std::set< Tag > const & skiptags ) [inline]
```

#### 10.72.4.25 ReadWithLength()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataElement::ReadWithLength (
    std::istream & is,
    VL & length ) [inline]
```

#### 10.72.4.26 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

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [FixBrokenJ2K.cxx](#), [FixJAI BugJPEGLS.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSubSequenceData.cxx](#), [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()`.

#### 10.72.4.27 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](#).

#### 10.72.4.28 SetValue()

```
void gdcM::DataElement::SetValue (
    Value const & vl ) [inline]
```

##### Warning

you need to set the ValueLengthField explicitly

##### Examples

[DuplicatePCDE.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), and [GenSeqs.cxx](#).

References [gdcM::Value::GetLength\(\)](#).

#### 10.72.4.29 SetValueFieldLength()

```
void gdcM::DataElement::SetValueFieldLength (
    VL vl,
    bool readvalues ) [protected]
```

#### 10.72.4.30 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](#)

#### 10.72.4.31 SetVLToUndefined()

```
void gdcM::DataElement::SetVLToUndefined ( )
```

##### Examples

[Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), and [NewSequence.cs](#).

### 10.72.4.32 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\(\)](#).

### 10.72.4.33 Write()

```
template<typename TDE , typename TSwap >
const std::ostream& gdcm::DataElement::Write (
    std::ostream & os ) const [inline]
```

## 10.72.5 Friends And Related Function Documentation

### 10.72.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const DataElement & _val ) [friend]
```

### 10.72.6 Member Data Documentation

#### 10.72.6.1 TagField

`Tag` `gdcm::DataElement::TagField` [protected]

Referenced by `operator==()`.

#### 10.72.6.2 ValueField

`ValuePtr` `gdcm::DataElement::ValueField` [protected]

Referenced by `operator==()`.

#### 10.72.6.3 ValueLengthField

`VL` `gdcm::DataElement::ValueLengthField` [protected]

Referenced by `operator==()`.

#### 10.72.6.4 VRField

`VR` `gdcm::DataElement::VRField` [protected]

Referenced by `operator==()`.

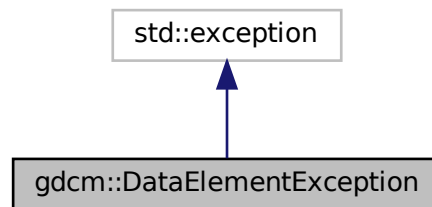
The documentation for this class was generated from the following file:

- [gdcmDataElement.h](#)

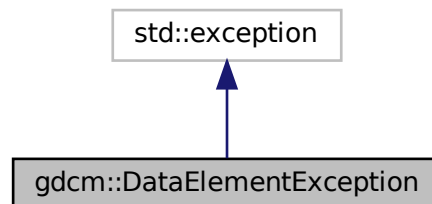
## 10.73 gdcm::DataElementException Class Reference

```
#include <gdcmDataSet.h>
```

Inheritance diagram for gdcm::DataElementException:



Collaboration diagram for gdcm::DataElementException:



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

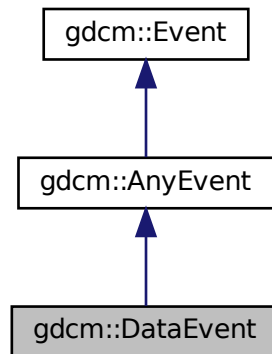
- [gdcmDataSet.h](#)

## 10.74 gdcm::DataEvent Class Reference

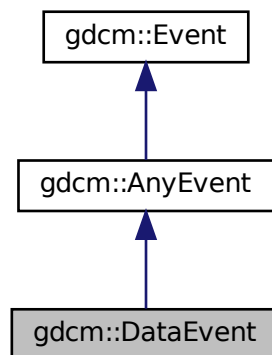
[DataEvent](#).

```
#include <gdcmDataEvent.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)

### 10.74.1 Detailed Description

[DataEvent](#).

### 10.74.2 Member Typedef Documentation

#### 10.74.2.1 Self

```
typedef DataEvent gdcm::DataEvent::Self
```

#### 10.74.2.2 Superclass

```
typedef AnyEvent gdcm::DataEvent::Superclass
```

### 10.74.3 Constructor & Destructor Documentation

#### 10.74.3.1 DataEvent() [1/2]

```
gdcm::DataEvent::DataEvent (  
    const char * bytes = nullptr,  
    size_t len = 0 ) [inline]
```

### 10.74.3.2 ~DataEvent()

```
gdcM::DataEvent::~~DataEvent ( ) [override], [default]
```

### 10.74.3.3 DataEvent() [2/2]

```
gdcM::DataEvent::DataEvent (
    const Self & s ) [inline]
```

## 10.74.4 Member Function Documentation

### 10.74.4.1 CheckEvent()

```
bool gdcM::DataEvent::CheckEvent (
    const ::gdcM::Event * e ) const [inline], [override]
```

### 10.74.4.2 GetData()

```
const char* gdcM::DataEvent::GetData ( ) const [inline]
```

### 10.74.4.3 GetDataLength()

```
size_t gdcM::DataEvent::GetDataLength ( ) const [inline]
```

### 10.74.4.4 GetEventName()

```
const char* gdcM::DataEvent::GetEventName ( ) const [inline], [override], [virtual]
```

Return the StringName associated with the event.

Implements [gdcM::Event](#).

#### 10.74.4.5 MakeObject()

```
::gdcm::Event* gdcm::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 [gdcm::Event](#).

#### 10.74.4.6 operator=()

```
void gdcm::DataEvent::operator= (
    const Self & ) [delete]
```

#### 10.74.4.7 SetData()

```
void gdcm::DataEvent::SetData (
    const char * bytes,
    size_t len ) [inline]
```

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

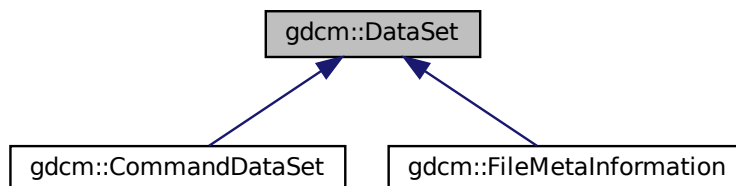
- [gdcmDataEvent.h](#)

## 10.75 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  
*Return the private creator of the private tag 't':*
- 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](#) &)

### 10.75.1 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](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Write.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](#), [PatchFile.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [SortImage.cxx](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), [VolumeSorter.cxx](#), [csa2img.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

**10.75.2 Member Typedef Documentation****10.75.2.1 ConstIterator**

```
typedef DataElementSet::const_iterator gdcm::DataSet::ConstIterator
```

**Examples**

[DumpVisusChange.cxx](#).

**10.75.2.2 DataElementSet**

```
typedef std::set<DataElement> gdcm::DataSet::DataElementSet
```

**10.75.2.3 Iterator**

```
typedef DataElementSet::iterator gdcm::DataSet::Iterator
```

**10.75.2.4 SizeType**

```
typedef DataElementSet::size_type gdcm::DataSet::SizeType
```

## 10.75.3 Member Function Documentation

### 10.75.3.1 Begin() [1/2]

```
Iterator gdcm::DataSet::Begin ( ) [inline]
```

### 10.75.3.2 Begin() [2/2]

```
ConstIterator gdcm::DataSet::Begin ( ) const [inline]
```

#### Examples

[DiffFile.cxx](#), and [DuplicatePCDE.cxx](#).

### 10.75.3.3 Clear()

```
void gdcm::DataSet::Clear ( ) [inline]
```

Referenced by `gdcm::Item::Read()`.

### 10.75.3.4 ComputeDataElement()

```
Tag gdcm::DataSet::ComputeDataElement (
    const PrivateTag & t ) const [protected]
```

### 10.75.3.5 ComputeGroupLength()

```
template<typename TDE >
unsigned int gdcm::DataSet::ComputeGroupLength (
    Tag const & tag ) const [inline]
```

References `gdcm::Tag::GetElement()`, and `gdcm::Tag::GetGroup()`.

**10.75.3.6 End()** [1/2]

```
Iterator gdcm::DataSet::End ( ) [inline]
```

**10.75.3.7 End()** [2/2]

```
ConstIterator gdcm::DataSet::End ( ) const [inline]
```

**Examples**

[DiffFile.cxx](#), and [DuplicatePCDE.cxx](#).

**10.75.3.8 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](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpVisusChange.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [MrProtocol.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.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()`.

**10.75.3.9 FindDataElement()** [2/2]

```
bool gdcm::DataSet::FindDataElement (
    const Tag & t ) const [inline]
```



**10.75.3.10 FindNextDataElement()**

```
const DataElement& gdcm::DataSet::FindNextDataElement (
    const Tag & t ) const [inline]
```

**Examples**

[DuplicatePCDE.cxx](#).

**10.75.3.11 GetDataElement() [1/2]**

```
const DataElement& gdcm::DataSet::GetDataElement (
    const PrivateTag & t ) const
```

Return the dataelement.

**10.75.3.12 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](#), [DeriveSeries.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpVisusChange.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadExplicitLengthSQIVR.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()`.

### 10.75.3.13 GetDEEnd()

```
const DataElement& gdcm::DataSet::GetDEEnd ( ) const [protected]
```

### 10.75.3.14 GetDES() [1/2]

```
DataElementSet& gdcm::DataSet::GetDES ( ) [inline]
```

### 10.75.3.15 GetDES() [2/2]

```
const DataElementSet& gdcm::DataSet::GetDES ( ) const [inline]
```

#### Examples

[ReadAndDumpDICOMDIR.cxx](#).

### 10.75.3.16 GetLength()

```
template<typename TDE >  
VL gdcm::DataSet::GetLength ( ) const [inline]
```

References [gdcm::VL::GetLength\(\)](#).

### 10.75.3.17 GetMediaStorage()

```
MediaStorage gdcm::DataSet::GetMediaStorage ( ) const
```

### 10.75.3.18 GetPrivateCreator()

```
std::string gdcm::DataSet::GetPrivateCreator (   
    const Tag & t ) const
```

Return the private creator of the private tag 't':

#### Examples

[DuplicatePCDE.cxx](#).

### 10.75.3.19 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](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [NewSequence.cs](#), [StreamImageReaderTest.cxx](#), and [TemplateEmptyImage.cxx](#).

References [gdcmErrorMacro](#), [gdcm::Tag::GetGroup\(\)](#), and [gdcm::DataElement::GetTag\(\)](#).

### 10.75.3.20 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\(\)](#).

### 10.75.3.21 IsEmpty()

```
bool gdcm::DataSet::IsEmpty ( ) const [inline]
```

Returns if the dataset is empty.

Referenced by [gdcm::Item::Read\(\)](#).

### 10.75.3.22 operator()()

```
const DataElement& gdcm::DataSet::operator() (
    uint16_t group,
    uint16_t element ) const [inline]
```

#### 10.75.3.23 operator=()

```
DataSet& gdcm::DataSet::operator= (
    DataSet const & ) [default]
```

#### 10.75.3.24 operator[]()

```
const DataElement& gdcm::DataSet::operator[] (
    const Tag & t ) const [inline]
```

#### 10.75.3.25 Print()

```
void gdcm::DataSet::Print (
    std::ostream & os,
    std::string const & indent = "" ) const [inline]
```

#### 10.75.3.26 Read()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataSet::Read (
    std::istream & is )
```

#### Examples

[DumpToshibaDTI.cxx](#).

#### 10.75.3.27 ReadNested()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataSet::ReadNested (
    std::istream & is )
```

### 10.75.3.28 ReadSelectedPrivateTags()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataSet::ReadSelectedPrivateTags (
    std::istream & is,
    const std::set< PrivateTag > & tags,
    bool readvalues = true )
```

### 10.75.3.29 ReadSelectedPrivateTagsWithLength()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataSet::ReadSelectedPrivateTagsWithLength (
    std::istream & is,
    const std::set< PrivateTag > & tags,
    VL & length,
    bool readvalues = true )
```

### 10.75.3.30 ReadSelectedTags()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataSet::ReadSelectedTags (
    std::istream & is,
    const std::set< Tag > & tags,
    bool readvalues = true )
```

### 10.75.3.31 ReadSelectedTagsWithLength()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataSet::ReadSelectedTagsWithLength (
    std::istream & is,
    const std::set< Tag > & tags,
    VL & length,
    bool readvalues = true )
```

### 10.75.3.32 ReadUpToTag()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataSet::ReadUpToTag (
    std::istream & is,
    const Tag & t,
    std::set< Tag > const & skiptags )
```

### 10.75.3.33 ReadUpToTagWithLength()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataSet::ReadUpToTagWithLength (
    std::istream & is,
    const Tag & t,
    std::set< Tag > const & skiptags,
    VL & length )
```

### 10.75.3.34 ReadWithLength()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataSet::ReadWithLength (
    std::istream & is,
    VL & length )
```

### 10.75.3.35 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](#).

### 10.75.3.36 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](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [PatchFile.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

References [gdcmAssertAlwaysMacro](#).

### 10.75.3.37 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`.

### 10.75.3.38 Size()

```
SizeType gdcm::DataSet::Size ( ) const [inline]
```

#### Examples

[DumpGEMSMovieGroup.cxx](#).

Referenced by `gdcm::SequenceOfItems::Read()`.

### 10.75.3.39 Write()

```
template<typename TDE , typename TSwap >
std::ostream const& gdcm::DataSet::Write (
    std::ostream & os ) const
```

## 10.75.4 Friends And Related Function Documentation

### 10.75.4.1 CSAHeader

```
friend class CSAHeader [friend]
```

### 10.75.4.2 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const DataSet & val ) [friend]
```

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

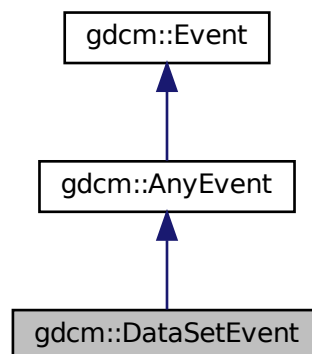
- [gdcmDataSet.h](#)

## 10.76 gdcm::DataSetEvent Class Reference

[DataSetEvent](#).

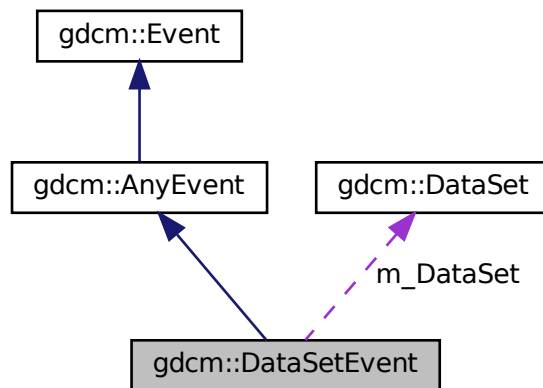
```
#include <gdcmDataSetEvent.h>
```

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 ::gdcm::Event \*e) const override
- [DataSet](#) const & [GetDataSet](#) () const
- const char \* [GetEventName](#) () const override
- ::gdcm::Event \* [MakeObject](#) () const override
- void [operator=](#) (const [Self](#) &)=delete

## Public Attributes

- const [DataSet](#) \* [m\\_DataSet](#)

### 10.76.1 Detailed Description

[DataSetEvent](#).

Special type of event triggered during the [DataSet](#) store/move process

See also

## 10.76.2 Member Typedef Documentation

### 10.76.2.1 Self

```
typedef DataSetEvent gdcM::DataSetEvent::Self
```

### 10.76.2.2 Superclass

```
typedef AnyEvent gdcM::DataSetEvent::Superclass
```

## 10.76.3 Constructor & Destructor Documentation

### 10.76.3.1 DataSetEvent() [1/2]

```
gdcM::DataSetEvent::DataSetEvent (
    DataSet const * ds = nullptr ) [inline]
```

### 10.76.3.2 ~DataSetEvent()

```
gdcM::DataSetEvent::~~DataSetEvent ( ) [override], [default]
```

### 10.76.3.3 DataSetEvent() [2/2]

```
gdcM::DataSetEvent::DataSetEvent (
    const Self & s ) [inline]
```

## 10.76.4 Member Function Documentation

#### 10.76.4.1 CheckEvent()

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

#### 10.76.4.2 GetDataSet()

```
DataSet const& gdcm::DataSetEvent::GetDataSet ( ) const [inline]
```

References `m_DataSet`.

#### 10.76.4.3 GetEventName()

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

Return the StringName associated with the event.

Implements [gdcm::Event](#).

#### 10.76.4.4 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](#).

#### 10.76.4.5 operator=()

```
void gdcm::DataSetEvent::operator= (
    const Self & ) [delete]
```

### 10.76.5 Member Data Documentation

### 10.76.5.1 m\_DataSet

```
const DataSet* gdcm::DataSetEvent::m_DataSet
```

Referenced by GetDataSet().

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

- [gdcmDataSetEvent.h](#)

## 10.77 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)

### 10.77.1 Detailed Description

[DataSetHelper](#) (internal class, not intended for user level)

### 10.77.2 Member Function Documentation

#### 10.77.2.1 ComputeVR()

```
static VR gdcm::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

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

- [gdcmDataSetHelper.h](#)



### 10.78.2.1 ~Decoder()

```
virtual gdcM::Decoder::~~Decoder ( ) [virtual], [default]
```

## 10.78.3 Member Function Documentation

### 10.78.3.1 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::RLECodec](#), [gdcM::RAWCodec](#), [gdcM::PVRGCodec](#), [gdcM::PNMCodec](#), [gdcM::PGXCodec](#), [gdcM::KAKADUCodec](#), [gdcM::JPEGLSCodec](#), [gdcM::JPEGCodec](#), [gdcM::JPEG2000Codec](#), [gdcM::PDFCodec](#), [gdcM::ImageCodec](#), and [gdcM::AudioCodec](#).

### 10.78.3.2 Decode()

```
virtual bool gdcM::Decoder::Decode (
    DataElement const & ,
    DataElement & ) [inline], [virtual]
```

Decode.

Reimplemented in [gdcM::ImageCodec](#), [gdcM::RLECodec](#), [gdcM::RAWCodec](#), [gdcM::PVRGCodec](#), [gdcM::PDFCodec](#), [gdcM::KAKADUCodec](#), [gdcM::JPEGLSCodec](#), [gdcM::JPEGCodec](#), [gdcM::JPEG2000Codec](#), [gdcM::AudioCodec](#), and [gdcM::DeltaEncodingCodec](#).

### 10.78.3.3 DecodeByStreams()

```
virtual bool gdcM::Decoder::DecodeByStreams (
    std::istream & ,
    std::ostream & ) [inline], [protected], [virtual]
```

Reimplemented in [gdcM::ImageCodec](#), [gdcM::RLECodec](#), [gdcM::RAWCodec](#), [gdcM::JPEGCodec](#), [gdcM::JPEG8Codec](#), [gdcM::JPEG2000Codec](#), [gdcM::JPEG16Codec](#), and [gdcM::JPEG12Codec](#).

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

- [gdcMDecoder.h](#)

## 10.79 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

#### 10.79.1 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.

#### 10.79.2 Constructor & Destructor Documentation

##### 10.79.2.1 DefinedTerms()

```
gdcm::DefinedTerms::DefinedTerms ( ) [default]
```

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

- [gdcmDefinedTerms.h](#)

## 10.80 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](#)

### 10.80.1 Detailed Description

FIXME I do not like the name '[Defs](#)'.

#### Note

bla

#### Examples

[GenerateStandardSOPClasses.cxx](#).

### 10.80.2 Constructor & Destructor Documentation



### 10.80.2.1 Defs() [1/2]

```
gdcm::Defs::Defs ( )
```

### 10.80.2.2 ~Defs()

```
gdcm::Defs::~~Defs ( )
```

### 10.80.2.3 Defs() [2/2]

```
gdcm::Defs::Defs (
    const Defs & val ) [delete]
```

## 10.80.3 Member Function Documentation

### 10.80.3.1 GetIODFromFile()

```
const IOD& gdcm::Defs::GetIODFromFile (
    const File & file ) const
```

### 10.80.3.2 GetIODNameFromMediaStorage()

```
static const char* gdcm::Defs::GetIODNameFromMediaStorage (
    MediaStorage const & ms ) [static]
```

#### Examples

[GenerateStandardSOPClasses.cxx](#).

### 10.80.3.3 GetIODs() [1/2]

```
IODs& gdcm::Defs::GetIODs ( ) [inline]
```

#### 10.80.3.4 GetIODs() [2/2]

```
const IODs& gdcM::Defs::GetIODs ( ) const [inline]
```

#### 10.80.3.5 GetMacros() [1/2]

```
Macros& gdcM::Defs::GetMacros ( ) [inline]
```

#### 10.80.3.6 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

#### 10.80.3.7 GetModules() [1/2]

```
Modules& gdcM::Defs::GetModules ( ) [inline]
```

#### 10.80.3.8 GetModules() [2/2]

```
const Modules& gdcM::Defs::GetModules ( ) const [inline]
```

#### 10.80.3.9 GetTypeFromTag()

```
Type gdcM::Defs::GetTypeFromTag (
    const File & file,
    const Tag & tag ) const
```

#### 10.80.3.10 IsEmpty()

```
bool gdcM::Defs::IsEmpty ( ) const [inline]
```

### 10.80.3.11 LoadDefaults()

```
void gdcmm::Defs::LoadDefaults ( ) [protected]
```

### 10.80.3.12 LoadFromFile()

```
void gdcmm::Defs::LoadFromFile (
    const char * filename ) [protected]
```

### 10.80.3.13 operator=()

```
Defs& gdcmm::Defs::operator= (
    const Defs & val ) [delete]
```

### 10.80.3.14 Verify() [1/2]

```
bool gdcmm::Defs::Verify (
    const DataSet & ds ) const
```

### 10.80.3.15 Verify() [2/2]

```
bool gdcmm::Defs::Verify (
    const File & file ) const
```

## 10.80.4 Friends And Related Function Documentation

### 10.80.4.1 Global

```
friend class Global [friend]
```

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

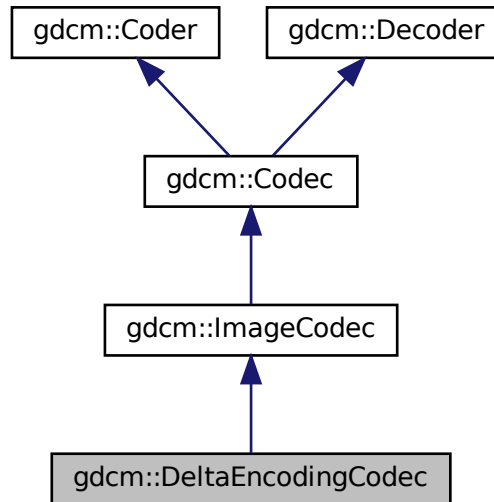
- [gdcmmDefs.h](#)

## 10.81 gdcm::DeltaEncodingCodec Class Reference

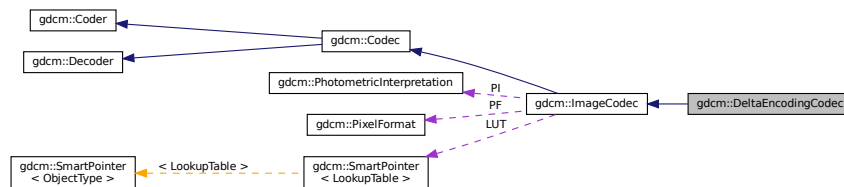
[DeltaEncodingCodec](#) compression used by some private vendor.

```
#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.*

## Protected Member Functions

- bool [Decode](#) (std::istream &is, std::ostream &os)

## Additional Inherited Members

### 10.81.1 Detailed Description

[DeltaEncodingCodec](#) compression used by some private vendor.

### 10.81.2 Constructor & Destructor Documentation

#### 10.81.2.1 DeltaEncodingCodec()

```
gdcm::DeltaEncodingCodec::DeltaEncodingCodec ( )
```

#### 10.81.2.2 ~DeltaEncodingCodec()

```
gdcm::DeltaEncodingCodec::~~DeltaEncodingCodec ( )
```

### 10.81.3 Member Function Documentation

#### 10.81.3.1 CanDecode()

```
bool gdcm::DeltaEncodingCodec::CanDecode (
    TransferSyntax const & ts )
```

#### 10.81.3.2 Decode() [1/2]

```
bool gdcm::DeltaEncodingCodec::Decode (
    DataElement const & ,
    DataElement & ) [virtual]
```

Decode.

Reimplemented from [gdcm::Decoder](#).

### 10.81.3.3 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](#)

## 10.82 gdcM::DICOMDIR Class Reference

[DICOMDIR](#) class.

```
#include <gdcMDICOMDIR.h>
```

### Public Member Functions

- [DICOMDIR](#) ()=default
- [DICOMDIR](#) ([FileSet](#) fs)

### 10.82.1 Detailed Description

[DICOMDIR](#) class.

Structured for handling [DICOMDIR](#)

### 10.82.2 Constructor & Destructor Documentation

#### 10.82.2.1 DICOMDIR() [1/2]

```
gdcM::DICOMDIR::DICOMDIR ( ) [default]
```

#### 10.82.2.2 DICOMDIR() [2/2]

```
gdcM::DICOMDIR::DICOMDIR (
    FileSet fs ) [inline]
```

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

- [gdcMDICOMDIR.h](#)

## 10.83 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](#) ()

### 10.83.1 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](#)

## 10.83.2 Member Typedef Documentation

### 10.83.2.1 FilenamesType

```
typedef Directory::FilenamesType gdcm::DICOMDIRGenerator::FilenamesType
```

### 10.83.2.2 FilenameType

```
typedef Directory::FilenameType gdcm::DICOMDIRGenerator::FilenameType
```

## 10.83.3 Constructor & Destructor Documentation

### 10.83.3.1 DICOMDIRGenerator()

```
gdcm::DICOMDIRGenerator::DICOMDIRGenerator ( )
```



### 10.83.3.2 ~DICOMDIRGenerator()

```
gdcm::DICOMDIRGenerator::~~DICOMDIRGenerator ( )
```

## 10.83.4 Member Function Documentation

### 10.83.4.1 AddImageDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddImageDirectoryRecord ( ) [protected]
```

### 10.83.4.2 AddPatientDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddPatientDirectoryRecord ( ) [protected]
```

### 10.83.4.3 AddSeriesDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddSeriesDirectoryRecord ( ) [protected]
```

### 10.83.4.4 AddStudyDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddStudyDirectoryRecord ( ) [protected]
```

### 10.83.4.5 Generate()

```
bool gdcm::DICOMDIRGenerator::Generate ( )
```

Main function to generate the [DICOMDIR](#).

#### 10.83.4.6 GetFile()

```
File& gdcm::DICOMDIRGenerator::GetFile ( )
```

#### 10.83.4.7 GetScanner()

```
Scanner& gdcm::DICOMDIRGenerator::GetScanner ( ) [protected]
```

#### 10.83.4.8 SetDescriptor()

```
void gdcm::DICOMDIRGenerator::SetDescriptor (
    const char * d )
```

Set the [File](#) Set ID.

##### Warning

this need to be a valid [VR::CS](#) value

#### 10.83.4.9 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.

#### 10.83.4.10 SetFileNames()

```
void gdcm::DICOMDIRGenerator::SetFileNames (
    FileNamesType const & fns )
```

Set the list of filenames from which the [DICOMDIR](#) should be generated from.

### 10.83.4.11 SetRootDirectory()

```
void gdcm::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:

- [gdcmDICOMDIRGenerator.h](#)

## 10.84 gdcm::Dict Class Reference

Class to represent a map of [DictEntry](#).

```
#include <gdcmDict.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`)

### 10.84.1 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](#).

### 10.84.2 Member Typedef Documentation

#### 10.84.2.1 ConstIterator

```
typedef MapDictEntry::const_iterator gdcm::Dict::ConstIterator
```

#### 10.84.2.2 Iterator

```
typedef MapDictEntry::iterator gdcm::Dict::Iterator
```

#### 10.84.2.3 MapDictEntry

```
typedef std::map<Tag, DictEntry> gdcm::Dict::MapDictEntry
```

### 10.84.3 Constructor & Destructor Documentation

### 10.84.3.1 Dict() [1/2]

```
gdcmm::Dict::Dict ( ) [inline]
```

### 10.84.3.2 Dict() [2/2]

```
gdcmm::Dict::Dict (
    const Dict & _val ) [delete]
```

## 10.84.4 Member Function Documentation

### 10.84.4.1 AddDictEntry()

```
void gdcmm::Dict::AddDictEntry (
    const Tag & tag,
    const DictEntry & de ) [inline]
```

### 10.84.4.2 Begin()

```
ConstIterator gdcmm::Dict::Begin ( ) const [inline]
```

#### Examples

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

### 10.84.4.3 End()

```
ConstIterator gdcmm::Dict::End ( ) const [inline]
```

#### Examples

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

#### 10.84.4.4 GetDictEntry()

```
const DictEntry& gdcm::Dict::GetDictEntry (
    const Tag & tag ) const [inline]
```

##### Examples

[GenFakelIdentifyFile.cxx](#), and [PublicDict.cxx](#).

#### 10.84.4.5 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](#).

#### 10.84.4.6 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](#).

#### 10.84.4.7 GetKeywordFromTag()

```
const char* gdcm::Dict::GetKeywordFromTag (
    Tag const & tag ) const [inline]
```

Function to return the Keyword from a [Tag](#).

#### 10.84.4.8 IsEmpty()

```
bool gdcm::Dict::IsEmpty ( ) const [inline]
```

#### 10.84.4.9 LoadDefault()

```
void gdcm::Dict::LoadDefault ( ) [protected]
```

#### 10.84.4.10 operator=()

```
Dict& gdcm::Dict::operator= (
    const Dict & _val ) [delete]
```

### 10.84.5 Friends And Related Function Documentation

#### 10.84.5.1 Dicts

```
friend class Dicts [friend]
```

#### 10.84.5.2 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const Dict & _val ) [friend]
```

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

- [gdcmDict.h](#)

## 10.85 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](#) ()

### 10.85.1 Detailed Description

Class to convert a .dic file into something else:

- CXX code : embeded dict into shared lib ([DICT\\_DEFAULT](#))
- Debug mode ([DICT\\_DEBUG](#))
- XML dict ([DICT\\_XML](#))

Note



## 10.85.2 Member Enumeration Documentation

### 10.85.2.1 OutputTypes

enum `gdcm::DictConverter::OutputTypes`

Enumerator

DICT_DEFAULT	
DICT_DEBUG	
DICT_XML	

## 10.85.3 Constructor & Destructor Documentation

### 10.85.3.1 DictConverter()

`gdcm::DictConverter::DictConverter ( )`

### 10.85.3.2 ~DictConverter()

`gdcm::DictConverter::~~DictConverter ( )`

## 10.85.4 Member Function Documentation

### 10.85.4.1 AddGroupLength()

`void gdcm::DictConverter::AddGroupLength ( )` [protected]

#### 10.85.4.2 Convert()

```
void gdcm::DictConverter::Convert ( )
```

#### 10.85.4.3 ConvertToCXX()

```
bool gdcm::DictConverter::ConvertToCXX (
    const char * raw,
    std::string & cxx ) [protected]
```

#### 10.85.4.4 ConvertToXML()

```
bool gdcm::DictConverter::ConvertToXML (
    const char * raw,
    std::string & cxx ) [protected]
```

#### 10.85.4.5 GetDictName()

```
const std::string& gdcm::DictConverter::GetDictName ( ) const
```

#### 10.85.4.6 GetInputFilename()

```
const std::string& gdcm::DictConverter::GetInputFilename ( ) const
```

#### 10.85.4.7 GetOutputFilename()

```
const std::string& gdcm::DictConverter::GetOutputFilename ( ) const
```

#### 10.85.4.8 GetOutputType()

```
int gdcm::DictConverter::GetOutputType ( ) const [inline]
```

#### 10.85.4.9 Readuint16()

```
static bool gdcm::DictConverter::Readuint16 (
    const char * raw,
    uint16_t & ov ) [static]
```

#### 10.85.4.10 ReadVM()

```
static bool gdcm::DictConverter::ReadVM (
    const char * raw,
    VM::VMType & type ) [static]
```

#### 10.85.4.11 ReadVR()

```
static bool gdcm::DictConverter::ReadVR (
    const char * raw,
    VR::VRType & type ) [static]
```

#### 10.85.4.12 SetDictName()

```
void gdcm::DictConverter::SetDictName (
    const char * name )
```

#### 10.85.4.13 SetInputFileName()

```
void gdcm::DictConverter::SetInputFileName (
    const char * filename )
```

#### 10.85.4.14 SetOutputFileName()

```
void gdcm::DictConverter::SetOutputFileName (
    const char * filename )
```

#### 10.85.4.15 SetOutputType()

```
void gdcM::DictConverter::SetOutputType (
    int type ) [inline]
```

#### 10.85.4.16 WriteFooter()

```
void gdcM::DictConverter::WriteFooter ( ) [protected]
```

#### 10.85.4.17 WriteHeader()

```
void gdcM::DictConverter::WriteHeader ( ) [protected]
```

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

- [gdcMDictConverter.h](#)

## 10.86 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)`

### 10.86.1 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 PublicDictEntry...indeed [DictEntry](#) has a notion of retired which does not exist in PrivateDictEntry...

#### See also

[gdcm::Dict](#)

#### Examples

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), and [PublicDict.cxx](#).

### 10.86.2 Constructor & Destructor Documentation

#### 10.86.2.1 DictEntry()

```
gdcm::DictEntry::DictEntry (  
    const char * name = "",  
    const char * keyword = "",  
    VR const & vr = VR::INVALID,  
    VM const & vm = VM::VM0,  
    bool ret = false ) [inline]
```

### 10.86.3 Member Function Documentation

### 10.86.3.1 GetKeyword()

```
const char* gdcm::DictEntry::GetKeyword ( ) const [inline]
```

same as GetName but without spaces...

### 10.86.3.2 GetName()

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

Set/Get Name.

Referenced by gdcm::PrivateDict::PrintXML().

### 10.86.3.3 GetRetired()

```
bool gdcm::DictEntry::GetRetired ( ) const [inline]
```

Set/Get Retired flag.

#### Examples

[GenAllVR.cxx](#).

### 10.86.3.4 GetVM()

```
const VM& gdcm::DictEntry::GetVM ( ) const [inline]
```

Set/Get [VM](#).

Referenced by gdcm::PrivateDict::AddDictEntry(), and gdcm::PrivateDict::PrintXML().

### 10.86.3.5 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()`.

### 10.86.3.6 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)

### 10.86.3.7 SetElementXX()

```
void gdcm::DictEntry::SetElementXX (
    bool v ) [inline]
```

Set whether element is shared in multiple elements (Source [Image](#) IDs typically)

### 10.86.3.8 SetGroupXX()

```
void gdcm::DictEntry::SetGroupXX (
    bool v ) [inline]
```

Set whether element is shared in multiple groups (Curve/Overlay typically)

### 10.86.3.9 SetKeyword()

```
void gdcm::DictEntry::SetKeyword (
    const char * keyword ) [inline]
```

#### 10.86.3.10 SetName()

```
void gdcM::DictEntry::SetName (
    const char * name ) [inline]
```

#### 10.86.3.11 SetRetired()

```
void gdcM::DictEntry::SetRetired (
    bool retired ) [inline]
```

#### 10.86.3.12 SetVM()

```
void gdcM::DictEntry::SetVM (
    VM const & vm ) [inline]
```

Referenced by gdcM::PrivateDict::AddDictEntry().

#### 10.86.3.13 SetVR()

```
void gdcM::DictEntry::SetVR (
    const VR & vr ) [inline]
```

Referenced by gdcM::PrivateDict::AddDictEntry().

### 10.86.4 Friends And Related Function Documentation

#### 10.86.4.1 Dict

```
friend class Dict [friend]
```



#### 10.86.4.2 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const DictEntry & _val ) [friend]
```

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

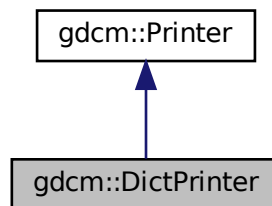
- [gdcmDictEntry.h](#)

## 10.87 gdcm::DictPrinter Class Reference

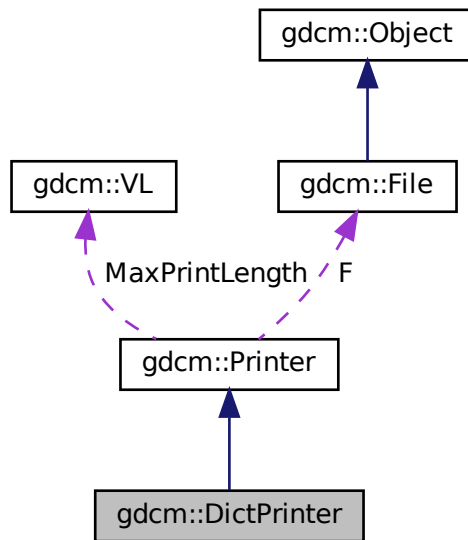
[DictPrinter](#) class.

```
#include <gdcmDictPrinter.h>
```

Inheritance diagram for gdcm::DictPrinter:



Collaboration diagram for `gdcm::DictPrinter`:



## Public Member Functions

- [DictPrinter](#) ()
- [~DictPrinter](#) ()
- void [Print](#) (std::ostream &os)

## Protected Member Functions

- void [PrintDataElement2](#) (std::ostream &os, const [DataSet](#) &ds, const [DataElement](#) &ide)
- void [PrintDataSet2](#) (std::ostream &os, const [DataSet](#) &ds)

## Additional Inherited Members

### 10.87.1 Detailed Description

[DictPrinter](#) class.

### 10.87.2 Constructor & Destructor Documentation

### 10.87.2.1 DictPrinter()

```
gdcm::DictPrinter::DictPrinter ( )
```

### 10.87.2.2 ~DictPrinter()

```
gdcm::DictPrinter::~~DictPrinter ( )
```

## 10.87.3 Member Function Documentation

### 10.87.3.1 Print()

```
void gdcm::DictPrinter::Print (
    std::ostream & os )
```

### 10.87.3.2 PrintDataElement2()

```
void gdcm::DictPrinter::PrintDataElement2 (
    std::ostream & os,
    const DataSet & ds,
    const DataElement & ide ) [protected]
```

### 10.87.3.3 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](#)

## 10.88 gdcmm::Dicts Class Reference

Class to manipulate the sum of knowledge (all the dict user load)

```
#include <gdcmmDicts.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
- NOT 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)

### 10.88.1 Detailed Description

Class to manipulate the sum of knowledge (all the dict user load)

Note

bla

Examples

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [PublicDict.cxx](#), and [ReadAndPrintAttributes.cxx](#).

### 10.88.2 Member Enumeration Documentation

#### 10.88.2.1 ConstructorType

```
enum gdcmm::Dicts::ConstructorType [protected]
```

Enumerator

PHILIPS	
GEMS	
SIEMENS	

### 10.88.3 Constructor & Destructor Documentation

#### 10.88.3.1 Dicts() [1/2]

```
gdcmm::Dicts::Dicts ( )
```

#### 10.88.3.2 ~Dicts()

```
gdcmm::Dicts::~~Dicts ( )
```

### 10.88.3.3 Dicts() [2/2]

```
gdcM::Dicts::Dicts (
    const Dicts & _val ) [delete]
```

## 10.88.4 Member Function Documentation

### 10.88.4.1 GetConstructorString()

```
static const char* gdcM::Dicts::GetConstructorString (
    ConstructorType type ) [static], [protected]
```

### 10.88.4.2 GetCSAHeaderDict()

```
const CSAHeaderDict& gdcM::Dicts::GetCSAHeaderDict ( ) const
```

#### Examples

[MrProtocol.cxx](#).

### 10.88.4.3 GetDictEntry() [1/2]

```
const DictEntry& gdcM::Dicts::GetDictEntry (
    const PrivateTag & tag ) const
```

### 10.88.4.4 GetDictEntry() [2/2]

```
const DictEntry& gdcM::Dicts::GetDictEntry (
    const Tag & tag,
    const char * owner = nullptr ) const
```

NOT 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](#).

#### 10.88.4.5 GetPrivateDict() [1/2]

```
PrivateDict& gdcmm::Dicts::GetPrivateDict ( )
```

#### 10.88.4.6 GetPrivateDict() [2/2]

```
const PrivateDict& gdcmm::Dicts::GetPrivateDict ( ) const
```

#### 10.88.4.7 GetPublicDict()

```
const Dict& gdcmm::Dicts::GetPublicDict ( ) const
```

#### Examples

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [PublicDict.cxx](#), and [ReadAndPrintAttributes.cxx](#).

#### 10.88.4.8 IsEmpty()

```
bool gdcmm::Dicts::IsEmpty ( ) const [inline]
```

#### 10.88.4.9 LoadDefaults()

```
void gdcmm::Dicts::LoadDefaults ( ) [protected]
```

#### 10.88.4.10 operator=()

```
Dicts& gdcmm::Dicts::operator= (
    const Dicts & _val ) [delete]
```

### 10.88.5 Friends And Related Function Documentation

### 10.88.5.1 Global

```
friend class Global [friend]
```

### 10.88.5.2 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const Dicts & d ) [friend]
```

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

- [gdcDicts.h](#)

## 10.89 gdc::network::DIMSE Class Reference

[DIMSE](#).

```
#include <gdcDIMSE.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 }



## 10.89.1 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)

## 10.89.2 Member Enumeration Documentation

### 10.89.2.1 CommandTypes

```
enum gdcmm::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:

- [gdcmmDIMSE.h](#)

## 10.90 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 void [Normalize](#) (double v[3])  
*Normalize in-place.*

### 10.90.1 Detailed Description

class to handle [DirectionCosines](#)

#### Examples

[DiscriminateVolume.cxx](#).

## 10.90.2 Constructor & Destructor Documentation

### 10.90.2.1 DirectionCosines() [1/2]

```
gdcmm::DirectionCosines::DirectionCosines ( )
```

### 10.90.2.2 DirectionCosines() [2/2]

```
gdcmm::DirectionCosines::DirectionCosines (
    const double dircos[6] )
```

### 10.90.2.3 ~DirectionCosines()

```
gdcmm::DirectionCosines::~~DirectionCosines ( )
```

## 10.90.3 Member Function Documentation

### 10.90.3.1 ComputeDistAlongNormal()

```
double gdcmm::DirectionCosines::ComputeDistAlongNormal (
    const double ipp[3] ) const
```

Compute the distance along the normal.

### 10.90.3.2 Cross()

```
void gdcmm::DirectionCosines::Cross (
    double z[3] ) const
```

Compute Cross product.

### 10.90.3.3 CrossDot()

```
double gdcM::DirectionCosines::CrossDot (
    DirectionCosines const & dc ) const
```

Compute the Dot product of the two cross vector of both [DirectionCosines](#) object.

#### Examples

[DiscriminateVolume.cxx](#).

### 10.90.3.4 Dot() [1/2]

```
double gdcM::DirectionCosines::Dot ( ) const
```

Compute Dot.

### 10.90.3.5 Dot() [2/2]

```
static double gdcM::DirectionCosines::Dot (
    const double x[3],
    const double y[3] ) [static]
```

Compute Dot.

### 10.90.3.6 IsValid()

```
bool gdcM::DirectionCosines::IsValid ( ) const
```

Return whether or not this is a valid direction cosines.

### 10.90.3.7 Normalize() [1/2]

```
void gdcM::DirectionCosines::Normalize ( )
```

Normalize in-place.

### 10.90.3.8 Normalize() [2/2]

```
static void gdcm::DirectionCosines::Normalize (
    double v[3] ) [static]
```

Normalize in-place.

### 10.90.3.9 operator const double \*()

```
gdcm::DirectionCosines::operator const double * ( ) const [inline]
```

Make the class behave like a const double \*.

### 10.90.3.10 Print()

```
void gdcm::DirectionCosines::Print (
    std::ostream & ) const
```

Print.

### 10.90.3.11 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](#)

## 10.91 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)

### 10.91.1 Detailed Description

Class for manipulation directories.

#### Note

This implementation provide a cross platform implementation for manipulating directorios: 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

[DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [DumpVisusChange.cxx](#), [GenerateRTSTRUCT.cxx](#), [ReadUTF8QtDir.cxx](#), [SortImage.cxx](#), [VolumeSorter.cxx](#), [gdcmorthoplanes.cxx](#), [reslicesphere.cxx](#), and [threadgdcm.cxx](#).

## 10.91.2 Member Typedef Documentation

### 10.91.2.1 FilenamesType

```
typedef std::vector<FilenameType> gdcmm::Directory::FilenamesType
```

#### Examples

[DiscriminateVolume.cxx](#), and [GenerateRTSTRUCT.cxx](#).

### 10.91.2.2 FilenameType

```
typedef std::string gdcmm::Directory::FilenameType
```

## 10.91.3 Constructor & Destructor Documentation

### 10.91.3.1 Directory()

```
gdcmm::Directory::Directory ( ) [default]
```

### 10.91.3.2 ~Directory()

```
gdcmm::Directory::~~Directory ( ) [default]
```

## 10.91.4 Member Function Documentation

#### 10.91.4.1 Explore()

```
unsigned int gdcm::Directory::Explore (
    FilenameType const & name,
    bool recursive ) [protected]
```

Return number of file found when 'recursive'ly exploring directory name

#### 10.91.4.2 GetDirectories()

```
FilenameType const& gdcm::Directory::GetDirectories ( ) const [inline]
```

Return the Directories traversed.

#### 10.91.4.3 GetFileNames()

```
FilenameType const& gdcm::Directory::GetFileNames ( ) const [inline]
```

Set/Get the file names within the directory.

#### Examples

[DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [DumpVisusChange.cxx](#), [GenerateRTSTRUCT.cxx](#), [ReadUTF8QtDir.cxx](#), [SortImage.cxx](#), [VolumeSorter.cxx](#), [gdcmorthoplanes.cxx](#), [reslicesphere.cxx](#), and [threadgdcm.cxx](#).

#### 10.91.4.4 GetToplevel()

```
FilenameType const& gdcm::Directory::GetToplevel ( ) const [inline]
```

Get the name of the toplevel directory.



#### 10.91.4.5 Load()

```
unsigned int gdcmm::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

[DecompressImageMultiframe.cs](#), [DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [DumpVisusChange.cxx](#), [GenerateRTSTRUCT.cxx](#), [ReadUTF8QtDir.cxx](#), [SortImage.cxx](#), [VolumeSorter.cxx](#), [gdcmmorthoplanes.cxx](#), [reslicesphere.cxx](#), and [threadgdcmm.cxx](#).

#### 10.91.4.6 Print()

```
void gdcmm::Directory::Print (
    std::ostream & os = std::cout ) const
```

Print.

##### Examples

[SortImage.cxx](#).

### 10.91.5 Friends And Related Function Documentation

#### 10.91.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const Directory & d ) [friend]
```

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

- [gdcmmDirectory.h](#)

## 10.92 gdcm::DirectoryHelper Class Reference

[DirectoryHelper](#).

```
#include <gdcmDirectoryHelper.h>
```

### Static Public Member Functions

- static [Directory::FilenamesType](#) [GetCTImageSeriesUIDs](#) (const std::string &inDirectory)
- static [Directory::FilenamesType](#) [GetFileNamesFromSeriesUIDs](#) (const std::string &inDirectory, const std::string &inSeriesUID)
- static std::string [GetFrameOfReference](#) (const std::vector< [DataSet](#) > &inDS)
- static [Directory::FilenamesType](#) [GetMRImageSeriesUIDs](#) (const std::string &inDirectory)
- static [Directory::FilenamesType](#) [GetRTStructSeriesUIDs](#) (const std::string &inDirectory)
- static [Directory::FilenamesType](#) [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)

### 10.92.1 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

### 10.92.2 Member Function Documentation

#### 10.92.2.1 GetCTImageSeriesUIDs()

```
static Directory::FilenamesType gdcm::DirectoryHelper::GetCTImageSeriesUIDs (
    const std::string & inDirectory ) [static]
```

### 10.92.2.2 GetFilenamesFromSeriesUIDs()

```
static Directory::FileNamesType gdcm::DirectoryHelper::GetFilenamesFromSeriesUIDs (
    const std::string & inDirectory,
    const std::string & inSeriesUID ) [static]
```

#### Examples

[GenerateRTSTRUCT.cxx](#).

### 10.92.2.3 GetFrameOfReference()

```
static std::string gdcm::DirectoryHelper::GetFrameOfReference (
    const std::vector< DataSet > & inDS ) [static]
```

### 10.92.2.4 GetMRImageSeriesUIDs()

```
static Directory::FileNamesType gdcm::DirectoryHelper::GetMRImageSeriesUIDs (
    const std::string & inDirectory ) [static]
```

### 10.92.2.5 GetRTStructSeriesUIDs()

```
static Directory::FileNamesType gdcm::DirectoryHelper::GetRTStructSeriesUIDs (
    const std::string & inDirectory ) [static]
```

#### Examples

[GenerateRTSTRUCT.cxx](#).

### 10.92.2.6 GetSeriesUIDsBySOPClassUID()

```
static Directory::FileNamesType gdcm::DirectoryHelper::GetSeriesUIDsBySOPClassUID (
    const std::string & inDirectory,
    const std::string & inSOPClassUID ) [static]
```

### 10.92.2.7 GetSOPClassUID()

```
static std::string gdc::DirectoryHelper::GetSOPClassUID (
    const std::vector< DataSet > & inDS ) [static]
```

### 10.92.2.8 GetStringValueFromTag()

```
static std::string gdc::DirectoryHelper::GetStringValueFromTag (
    const Tag & t,
    const DataSet & ds ) [static]
```

### 10.92.2.9 LoadImageFromFiles()

```
static std::vector<DataSet> gdc::DirectoryHelper::LoadImageFromFiles (
    const std::string & inDirectory,
    const std::string & inSeriesUID ) [static]
```

### 10.92.2.10 RetrieveSOPInstanceUIDFromIndex()

```
static std::string gdc::DirectoryHelper::RetrieveSOPInstanceUIDFromIndex (
    int inIndex,
    const std::vector< DataSet > & inDS ) [static]
```

### 10.92.2.11 RetrieveSOPInstanceUIDFromZPosition()

```
static std::string gdc::DirectoryHelper::RetrieveSOPInstanceUIDFromZPosition (
    double inZPos,
    const std::vector< DataSet > & inDS ) [static]
```

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

- [gdcDirectoryHelper.h](#)

## 10.93 gdc::DummyValueGenerator Class Reference

Class for generating dummy value.

```
#include <gdcDummyValueGenerator.h>
```

## Static Public Member Functions

- static const char \* [Generate](#) (const char \*input)

### 10.93.1 Detailed Description

Class for generating dummy value.

See also

[Anonymizer](#)

### 10.93.2 Member Function Documentation

#### 10.93.2.1 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:

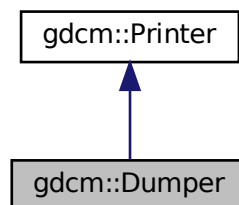
- [gdcmDummyValueGenerator.h](#)

## 10.94 gdcm::Dumper Class Reference

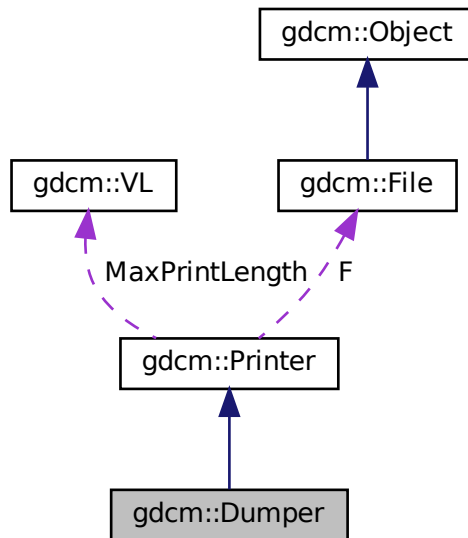
[Codec](#) class.

```
#include <gdcmDumper.h>
```

Inheritance diagram for gdcm::Dumper:



Collaboration diagram for `gdcm::Dumper`:



## Public Member Functions

- [Dumper](#) ()
- [~Dumper](#) ()=default

## Additional Inherited Members

### 10.94.1 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

### 10.94.2 Constructor & Destructor Documentation

### 10.94.2.1 Dumper()

```
gdcmm::Dumper::Dumper ( ) [inline]
```

### 10.94.2.2 ~Dumper()

```
gdcmm::Dumper::~~Dumper ( ) [default]
```

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

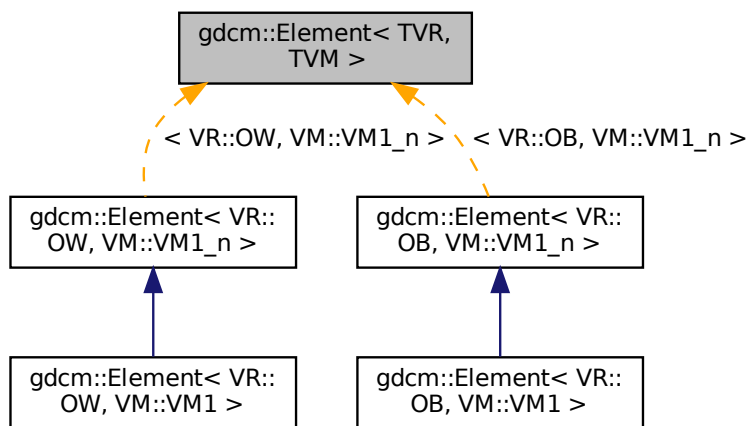
- [gdcmmDumper.h](#)

## 10.95 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 `VRTToType< TVR >::Type` `Type`

## Public Member Functions

- `DataElement GetAsDataElement ()` const
- unsigned long `GetLength ()` const
- `VRTToType< TVR >::Type & GetValue` (unsigned int idx=0)
- const `VRTToType< TVR >::Type & GetValue` (unsigned int idx=0) const
- const `VRTToType< TVR >::Type * GetValues ()` const
- `VRTToType< 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 `VRTToType< 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)

### 10.95.1 Detailed Description

```
template<long long TVR, int TVM>
class gdcm::Element< TVR, TVM >
```

[Element](#) class.

#### Note

TODO

#### Examples

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [iU22tomultisc.cxx](#), and [csa2img.cxx](#).

### 10.95.2 Member Typedef Documentation

#### 10.95.2.1 Type

```
template<long long TVR, int TVM>
typedef VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Type
```

### 10.95.3 Member Function Documentation

#### 10.95.3.1 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\(\)](#).

### 10.95.3.2 GetLength()

```
template<long long TVR, int TVM>
unsigned long gdcm::Element< TVR, TVM >::GetLength ( ) const [inline]
```

#### Examples

[DumpGEMSMovieGroup.cxx](#).

### 10.95.3.3 GetValue() [1/2]

```
template<long long TVR, int TVM>
VRToType<TVR>::Type& gdcm::Element< TVR, TVM >::GetValue (
    unsigned int idx = 0 ) [inline]
```

### 10.95.3.4 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](#), and [csa2img.cxx](#).

### 10.95.3.5 GetValues()

```
template<long long TVR, int TVM>
const VRToType<TVR>::Type* gdcm::Element< TVR, TVM >::GetValues ( ) const [inline]
```

### 10.95.3.6 GetVM()

```
template<long long TVR, int TVM>
static VM gdcm::Element< TVR, TVM >::GetVM ( ) [inline], [static]
```

### 10.95.3.7 GetVR()

```
template<long long TVR, int TVM>
static VR gdcm::Element< TVR, TVM >::GetVR ( ) [inline], [static]
```

### 10.95.3.8 operator[]()

```
template<long long TVR, int TVM>
VRToType<TVR>::Type gdcm::Element< TVR, TVM >::operator[] (
    unsigned int idx ) const [inline]
```

### 10.95.3.9 Print()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::Print (
    std::ostream & _os ) const [inline]
```

#### Examples

[DumpGEMSMovieGroup.cxx](#).

### 10.95.3.10 Read()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::Read (
    std::istream & _is ) [inline]
```

### 10.95.3.11 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()`.

### 10.95.3.12 SetFromDataElement()

```
template<long long TVR, int TVM>
void gdcmm::Element< TVR, TVM >::SetFromDataElement (
    DataElement< TVR, TVM > const & de ) [inline]
```

#### Examples

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), and [iU22tomultisc.cxx](#).

References [gdcmm::DataElement::GetByteValue\(\)](#), [gdcmm::DataElement::GetValue\(\)](#), and [gdcmm::DataElement::GetVR\(\)](#).

### 10.95.3.13 SetNoSwap()

```
template<long long TVR, int TVM>
void gdcmm::Element< TVR, TVM >::SetNoSwap (
    Value const & v ) [inline], [protected]
```

References [gdcmm::ByteValue::GetLength\(\)](#), and [gdcmm::ByteValue::GetPointer\(\)](#).

### 10.95.3.14 SetValue()

```
template<long long TVR, int TVM>
void gdcmm::Element< TVR, TVM >::SetValue (
    typename VRTToType< TVR >::Type v,
    unsigned int idx = 0 ) [inline]
```

#### Examples

[Extracting\\_All\\_Resolution.cxx](#), and [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#).

### 10.95.3.15 Write()

```
template<long long TVR, int TVM>
void gdcmm::Element< TVR, TVM >::Write (
    std::ostream & _os ) const [inline]
```

## 10.95.4 Member Data Documentation

#### 10.95.4.1 Internal

```
template<long long TVR, int TVM>
VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

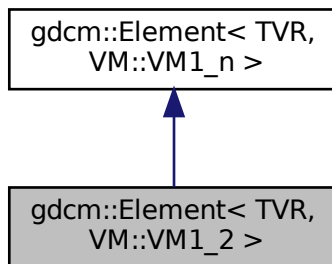
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

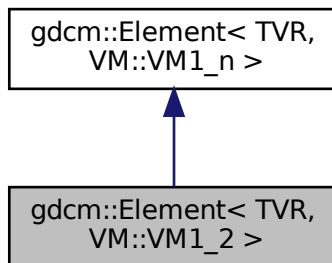
## 10.96 `gdcm::Element< TVR, VM::VM1_2 >` Class Template Reference

```
#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 Member Functions

- void [SetLength](#) (int len)

## Additional Inherited Members

### 10.96.1 Member Typedef Documentation

#### 10.96.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM1\_n> gdcm::Element< TVR, VM::VM1\_2 >::Parent
```

### 10.96.2 Member Function Documentation

#### 10.96.2.1 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1\_2 >::SetLength (
    int len ) [inline]
```

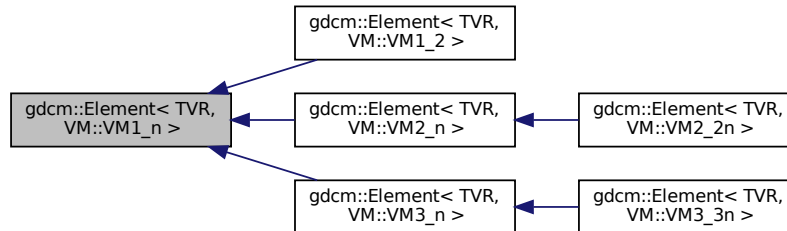
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

## 10.97 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`)

## 10.97.1 Member Typedef Documentation

### 10.97.1.1 Type

```
template<long long TVR>
typedef VRToType<TVR>::Type gdcM::Element< TVR, VM::VM1_n >::Type
```

## 10.97.2 Constructor & Destructor Documentation

### 10.97.2.1 `Element()` [1/2]

```
template<long long TVR>
gdcM::Element< TVR, VM::VM1_n >::Element ( ) [inline], [explicit]
```

### 10.97.2.2 `~Element()`

```
template<long long TVR>
gdcM::Element< TVR, VM::VM1_n >::~~Element ( ) [inline]
```

### 10.97.2.3 `Element()` [2/2]

```
template<long long TVR>
gdcM::Element< TVR, VM::VM1_n >::Element (
    const Element< TVR, VM::VM1_n > &_val ) [inline]
```

## 10.97.3 Member Function Documentation



### 10.97.3.1 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().

### 10.97.3.2 GetLength()

```
template<long long TVR>
unsigned long gdcm::Element< TVR, VM::VM1_n >::GetLength ( ) const [inline]
```

### 10.97.3.3 GetValue() [1/2]

```
template<long long TVR>
VRToType<TVR>::Type& gdcm::Element< TVR, VM::VM1_n >::GetValue (
    unsigned int idx = 0 ) [inline]
```

### 10.97.3.4 GetValue() [2/2]

```
template<long long TVR>
const VRToType<TVR>::Type& gdcm::Element< TVR, VM::VM1_n >::GetValue (
    unsigned int idx = 0 ) const [inline]
```

### 10.97.3.5 GetVM()

```
template<long long TVR>
static VM gdcm::Element< TVR, VM::VM1_n >::GetVM ( ) [inline], [static]
```

### 10.97.3.6 GetVR()

```
template<long long TVR>
static VR gdcm::Element< TVR, VM::VM1_n >::GetVR ( ) [inline], [static]
```

**10.97.3.7 operator=()**

```
template<long long TVR>
Element& gdcm::Element< TVR, VM::VM1_n >::operator= (
    const Element< TVR, VM::VM1_n > & _val ) [inline]
```

**10.97.3.8 operator[]()**

```
template<long long TVR>
VRToType<TVR>::Type gdcm::Element< TVR, VM::VM1_n >::operator[] (
    unsigned int idx ) const [inline]
```

**10.97.3.9 Print()**

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::Print (
    std::ostream & _os ) const [inline]
```

**10.97.3.10 Read()**

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::Read (
    std::istream & _is ) [inline]
```

**10.97.3.11 Set()**

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::Set (
    Value const & v ) [inline]
```

References `gdcm::ByteValue::GetLength()`, `gdcm::ByteValue::GetPointer()`, `gdcm::ByteValue::GetVoidPointer()`, and `gdcm::VRBINARY`.

**10.97.3.12 SetArray()**

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetArray (
    const Type * array,
    unsigned long len,
    bool save = false ) [inline]
```

**10.97.3.13 SetFromDataElement()**

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement (
    DataElement< TVR, VM::VM1_n > const & de ) [inline]
```

References `gdcm::DataElement::GetByteValue()`, `gdcm::DataElement::GetValue()`, and `gdcm::DataElement::GetVR()`.

**10.97.3.14 SetLength()**

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetLength (
    unsigned long len ) [inline]
```

**10.97.3.15 SetNoSwap()**

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetNoSwap (
    Value const & v ) [inline], [protected]
```

References `gdcm::ByteValue::GetLength()`, `gdcm::ByteValue::GetPointer()`, and `gdcm::VRBINARY`.

**10.97.3.16 SetValue()**

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetValue (
    typename VRToType< TVR >::Type v,
    unsigned int idx = 0 ) [inline]
```

### 10.97.3.17 Write()

```
template<long long TVR>
void gdcM::Element< TVR, VM::VM1_n >::Write (
    std::ostream & _os ) const [inline]
```

### 10.97.3.18 WriteASCII()

```
template<long long TVR>
void gdcM::Element< TVR, VM::VM1_n >::WriteASCII (
    std::ostream & os ) const [inline]
```

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

- [gdcMElement.h](#)

## 10.98 gdcM::Element< TVR, VM::VM2\_2n > Class Template Reference

```
#include <gdcMElement.h>
```

Inheritance diagram for gdcM::Element< TVR, VM::VM2\_2n >:



Collaboration diagram for gdcmm::Element< TVR, VM::VM2\_2n >:



## Public Types

- typedef `Element< TVR, VM::VM2_n >` `Parent`

## Public Member Functions

- void `SetLength` (int len)

## Additional Inherited Members

### 10.98.1 Member Typedef Documentation

#### 10.98.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM2_n> gdcmm::Element< TVR, VM::VM2_2n >::Parent
```

### 10.98.2 Member Function Documentation

### 10.98.2.1 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](#)

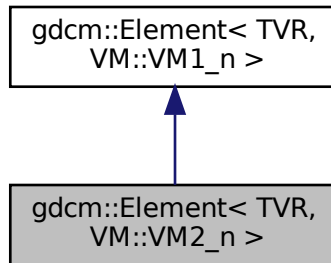
## 10.99 gdcM::Element< TVR, VM::VM2\_n > Class Template Reference

```
#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 Member Functions

- void `SetLength` (int len)

## Additional Inherited Members

### 10.99.1 Member Typedef Documentation

#### 10.99.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM2_n >::Parent
```

### 10.99.2 Member Function Documentation

### 10.99.2.1 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](#)

## 10.100 gdcM::Element< TVR, VM::VM3\_3n > Class Template Reference

```
#include <gdcMElement.h>
```

Inheritance diagram for gdcM::Element< TVR, VM::VM3\_3n >:





Collaboration diagram for gdcmm::Element< TVR, VM::VM3\_3n >:



## Public Types

- typedef [Element](#)< TVR, [VM::VM3\\_n](#) > [Parent](#)

## Public Member Functions

- void [SetLength](#) (int len)

## Additional Inherited Members

### 10.100.1 Member Typedef Documentation

#### 10.100.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM3\_n> gdcmm::Element< TVR, VM::VM3\_3n >::Parent
```

### 10.100.2 Member Function Documentation

### 10.100.2.1 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM3_3n >::SetLength (
    int len ) [inline]
```

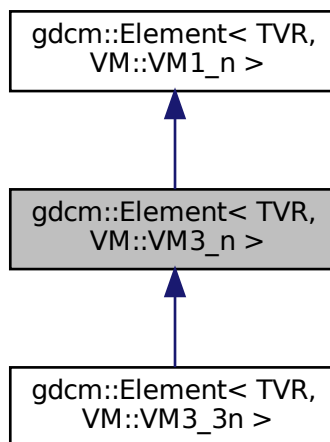
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

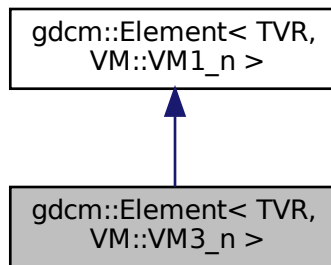
## 10.101 gdcm::Element< TVR, VM::VM3\_n > Class Template Reference

```
#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 Member Functions

- void `SetLength` (int len)

## Additional Inherited Members

### 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::VM3_n >::Parent
```

### 10.101.2 Member Function Documentation

### 10.101.2.1 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](#)

## 10.102 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([VRTToType](#)< VR::AS >::Type)]

## 10.102.1 Member Function Documentation

### 10.102.1.1 GetLength()

```
unsigned long gdcm::Element< VR::AS, VM::VM5 >::GetLength ( ) const [inline]
```

### 10.102.1.2 Print()

```
void gdcm::Element< VR::AS, VM::VM5 >::Print (
    std::ostream &_os ) const [inline]
```

## 10.102.2 Member Data Documentation

### 10.102.2.1 Internal

```
char gdcm::Element< VR::AS, VM::VM5 >::Internal[VMToLength< VM::VM5 >::Length *sizeof(VRToType< VR::AS >::Type)]
```

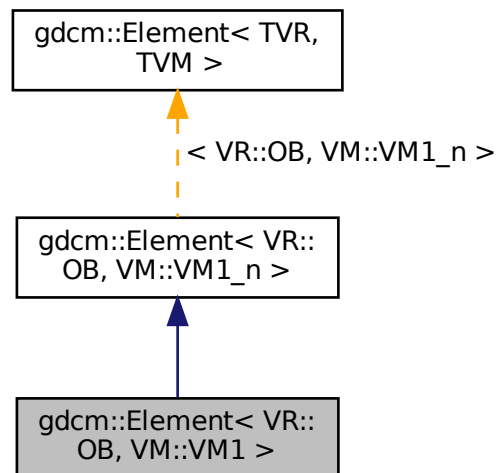
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

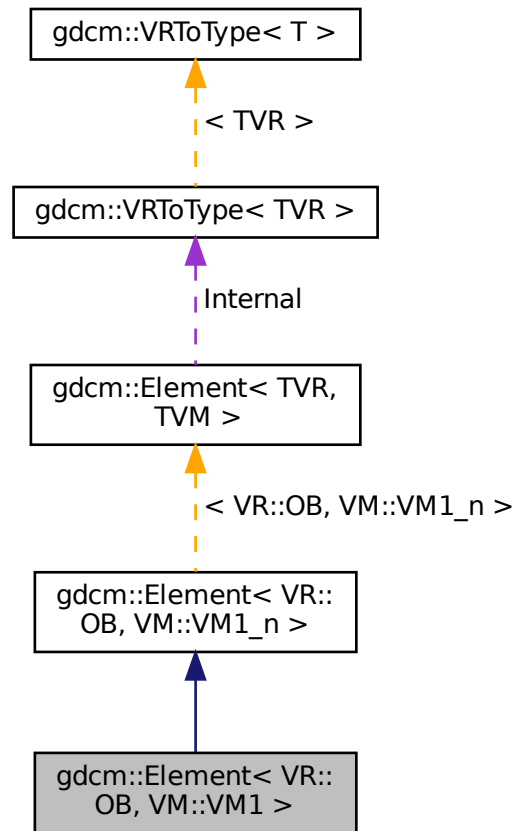
## 10.103 gdcm::Element< VR::OB, VM::VM1 > Class Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< VR::OB, VM::VM1 >:



Collaboration diagram for `gdcm::Element< VR::OB, VM::VM1 >`:



### Additional Inherited Members

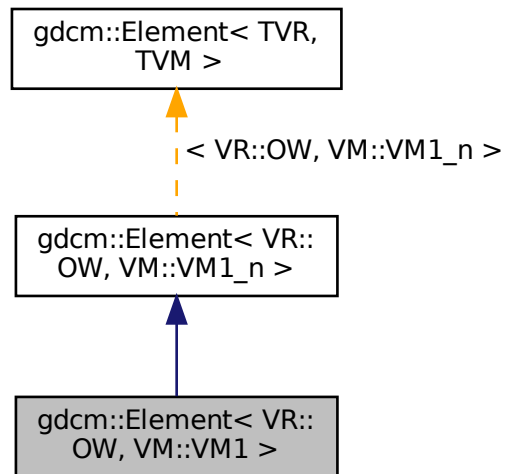
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

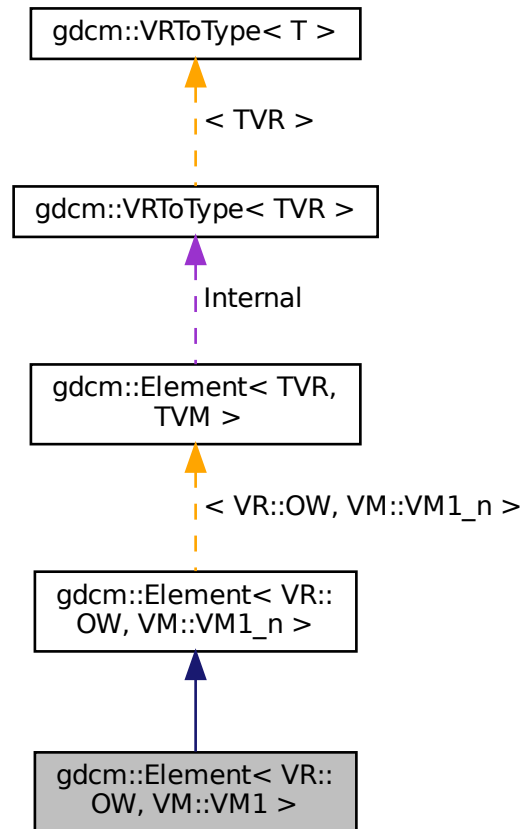
## 10.104 `gdcm::Element< VR::OW, VM::VM1 >` Class Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcM::Element< VR::OW, VM::VM1 >:



Collaboration diagram for `gdcm::Element< VR::OW, VM::VM1 >`:



### Additional Inherited Members

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

- [gdcmElement.h](#)

## 10.105 `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>
```



### 10.105.1 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](#)

## 10.106 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](#)

## 10.107 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](#)

## 10.108 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)

### 10.108.1 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](#).

## 10.108.2 Member Enumeration Documentation

### 10.108.2.1 SOPClassUIDMode

```
enum gdcm::EmptyMaskGenerator::SOPClassUIDMode
```

Enumerator

UseOriginalSOPClassUID	
UseGrayscaleSecondaryImageStorage	

## 10.108.3 Constructor & Destructor Documentation

### 10.108.3.1 EmptyMaskGenerator()

```
gdcm::EmptyMaskGenerator::EmptyMaskGenerator ( )
```

### 10.108.3.2 ~EmptyMaskGenerator()

```
gdcm::EmptyMaskGenerator::~~EmptyMaskGenerator ( )
```

## 10.108.4 Member Function Documentation

### 10.108.4.1 Execute()

```
bool gdcm::EmptyMaskGenerator::Execute ( )
```

Main loop.

Examples

[EmptyMask.cxx](#).

#### 10.108.4.2 SetInputDirectory()

```
void gdcM::EmptyMaskGenerator::SetInputDirectory (
    const char * dirname )
```

Specify input directory.

##### Examples

[EmptyMask.cxx](#).

#### 10.108.4.3 SetOutputDirectory()

```
void gdcM::EmptyMaskGenerator::SetOutputDirectory (
    const char * dirname )
```

Specify output directory.

##### Examples

[EmptyMask.cxx](#).

#### 10.108.4.4 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](#)

## 10.109 gdcM::EncapsulatedDocument Class Reference

[EncapsulatedDocument](#).

```
#include <gdcMEncapsulatedDocument.h>
```

## Public Member Functions

- [EncapsulatedDocument](#) ()=default

### 10.109.1 Detailed Description

[EncapsulatedDocument](#).

### 10.109.2 Constructor & Destructor Documentation

#### 10.109.2.1 EncapsulatedDocument()

```
gdcm::EncapsulatedDocument::EncapsulatedDocument ( ) [default]
```

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

- [gdcmEncapsulatedDocument.h](#)

## 10.110 gdcm::EncodingImplementation< T > Class Template Reference

[EncodingImplementation](#).

### 10.110.1 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](#)

## 10.111 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)

### 10.111.1 Member Function Documentation

#### 10.111.1.1 Read()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRASCII >::Read (
    T * data,
    unsigned long length,
    std::istream & _is ) [inline], [static]
```

#### 10.111.1.2 ReadComputeLength()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRASCII >::ReadComputeLength (
    T * data,
    unsigned int & length,
    std::istream & _is ) [inline], [static]
```

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

### 10.111.1.3 ReadNoSwap()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRASCII >::ReadNoSwap (
    T * data,
    unsigned long length,
    std::istream & _is ) [inline], [static]
```

### 10.111.1.4 Write() [1/2]

```
template<>
void gdcm::EncodingImplementation< VR::VRASCII >::Write (
    const double * data,
    unsigned long length,
    std::ostream & _os ) [inline]
```

References `gdcm::x16printf()`.

### 10.111.1.5 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](#)

## 10.112 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)

## 10.112.1 Member Function Documentation

### 10.112.1.1 Read()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRBINARY >::Read (
    T * data,
    unsigned long length,
    std::istream & _is ) [inline], [static]
```

### 10.112.1.2 ReadComputeLength()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadComputeLength (
    T * data,
    unsigned int & length,
    std::istream & _is ) [inline], [static]
```

### 10.112.1.3 ReadNoSwap()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadNoSwap (
    T * data,
    unsigned long length,
    std::istream & _is ) [inline], [static]
```

### 10.112.1.4 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:

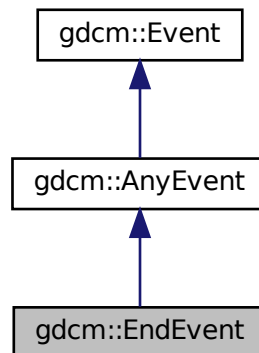
- [gdcmElement.h](#)



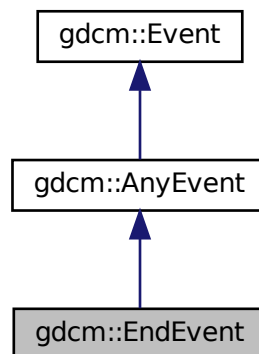
## 10.113 gdcm::EndEvent Class Reference

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

Inheritance diagram for gdcm::EndEvent:



Collaboration diagram for gdcm::EndEvent:



### Additional Inherited Members

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

- [gdcmEvent.h](#)

## 10.114 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

#### 10.114.1 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](#).

#### 10.114.2 Constructor & Destructor Documentation

##### 10.114.2.1 EnumeratedValues()

```
gdcm::EnumeratedValues::EnumeratedValues ( ) [default]
```

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

- [gdcmEnumeratedValues.h](#)

## 10.115 gdcm::EquipmentManufacturer Class Reference

```
#include <gdcmEquipmentManufacturer.h>
```

## Public Types

- enum [Type](#) {  
    [UNKNOWN](#) = 0 ,  
    [FUJI](#) ,  
    [GEMS](#) ,  
    [HITACHI](#) ,  
    [KODAK](#) ,  
    [MARCONI](#) ,  
    [PMS](#) ,  
    [SIEMENS](#) ,  
    [TOSHIBA](#) }

## Static Public Member Functions

- static [Type](#) [Compute](#) ([DataSet](#) const &ds)

### 10.115.1 Detailed Description

### 10.115.2 Member Enumeration Documentation

#### 10.115.2.1 Type

enum [gdcm::EquipmentManufacturer::Type](#)

##### Enumerator

UNKNOWN	
FUJI	
GEMS	
HITACHI	
KODAK	
MARCONI	
PMS	
SIEMENS	
TOSHIBA	

### 10.115.3 Member Function Documentation

#### 10.115.3.1 Compute()

```
static Type gdcM::EquipmentManufacturer::Compute (  
    DataSet const & ds ) [static]
```

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

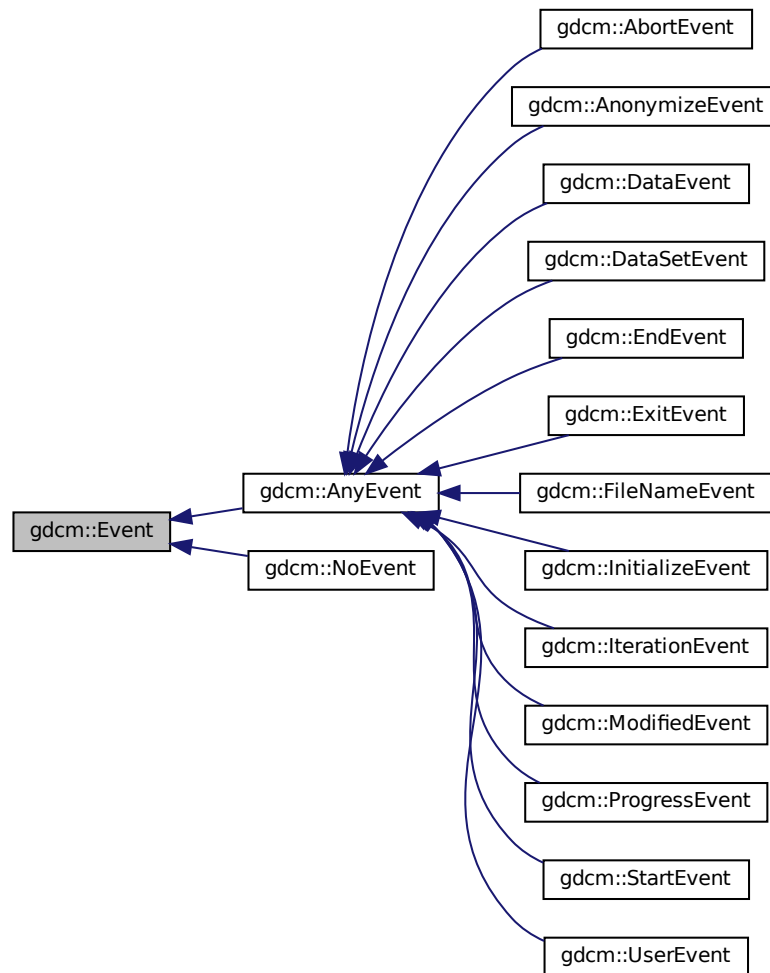
- [gdcMEquipmentManufacturer.h](#)

### 10.116 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

### 10.116.1 Detailed Description

superclass for callback/observer methods

See also

[Command Subject](#)

Examples

[SimpleScanner.cxx](#).

### 10.116.2 Constructor & Destructor Documentation

#### 10.116.2.1 Event() [1/2]

```
gdcm::Event::Event ( )
```

#### 10.116.2.2 ~Event()

```
virtual gdcm::Event::~~Event ( ) [virtual]
```

#### 10.116.2.3 Event() [2/2]

```
gdcm::Event::Event (
    const Event & )
```

### 10.116.3 Member Function Documentation

#### 10.116.3.1 CheckEvent()

```
virtual bool gdcm::Event::CheckEvent (
    const Event * ) const [pure virtual]
```

Check if given event matches or derives from this event.

### 10.116.3.2 GetEventName()

```
virtual const char* gdcm::Event::GetEventName ( ) const [pure virtual]
```

Return the StringName associated with the event.

Implemented in [gdcm::AnonymizeEvent](#), [gdcm::DataSetEvent](#), [gdcm::ProgressEvent](#), [gdcm::FileNameEvent](#), and [gdcm::DataEvent](#).

### 10.116.3.3 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::AnonymizeEvent](#), [gdcm::DataSetEvent](#), [gdcm::ProgressEvent](#), [gdcm::FileNameEvent](#), and [gdcm::DataEvent](#).

### 10.116.3.4 operator=()

```
void gdcm::Event::operator= (
    const Event & ) [delete]
```

### 10.116.3.5 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:

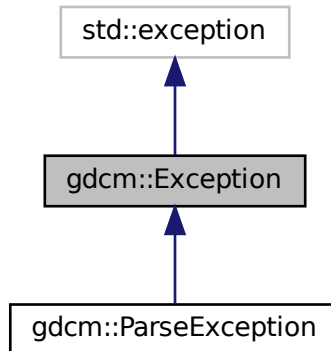
- [gdcmEvent.h](#)

## 10.117 gdcm::Exception Class Reference

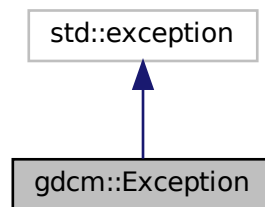
[Exception](#).

```
#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*



## 10.117.1 Detailed Description

[Exception.](#)

Standard exception handling object.

### Note

Its copy-constructor and assignment operator are generated by the compiler.

## 10.117.2 Constructor & Destructor Documentation

### 10.117.2.1 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.

### 10.117.2.2 ~Exception()

```
gdcm::Exception::~Exception ( ) throw ( ) [inline], [override]
```

## 10.117.3 Member Function Documentation

### 10.117.3.1 GetDescription()

```
const char* gdcm::Exception::GetDescription ( ) const [inline]
```

Return the Description.

Referenced by `gdcm::SequenceOfItems::Read()`.

### 10.117.3.2 what()

```
const char* gdcm::Exception::what ( ) const throw ( )    [inline], [override]
```

what implementation

Referenced by `gdcm::SequenceOfFragments::ReadValue()`.

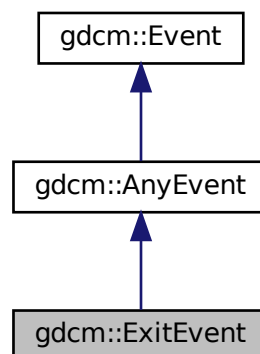
The documentation for this class was generated from the following file:

- [gdcmException.h](#)

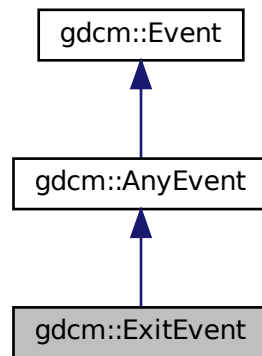
## 10.118 gdcm::ExitEvent Class Reference

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

Inheritance diagram for `gdcm::ExitEvent`:



Collaboration diagram for gdcm::ExitEvent:



### Additional Inherited Members

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

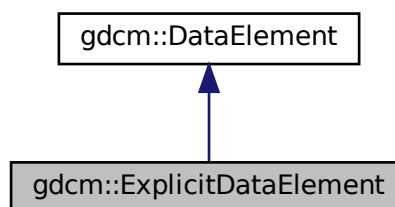
- [gdcmEvent.h](#)

## 10.119 gdcm::ExplicitDataElement Class Reference

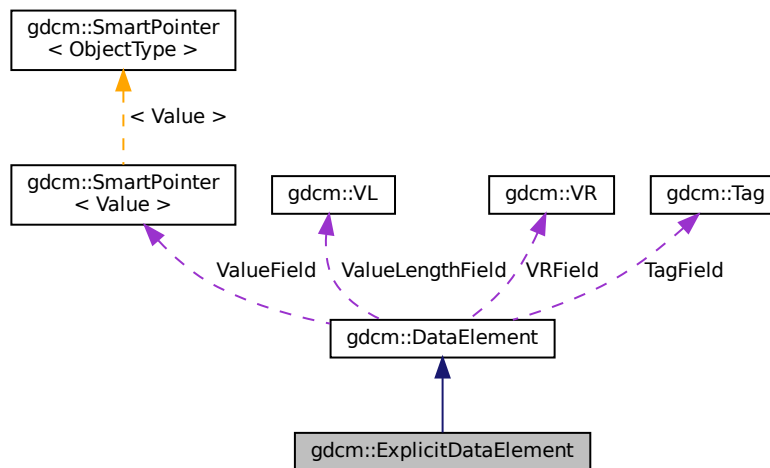
Class to read/write a [DataElement](#) as Explicit Data [Element](#).

```
#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

## Additional Inherited Members

### 10.119.1 Detailed Description

Class to read/write a [DataElement](#) as Explicit Data [Element](#).

#### Note

bla

#### Examples

[DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), and [ReadAndDumpDICOMDIR2.cxx](#).

## 10.119.2 Member Function Documentation

### 10.119.2.1 GetLength()

```
VL gdcm::ExplicitDataElement::GetLength ( ) const
```

### 10.119.2.2 Read()

```
template<typename TSwap >  
std::istream& gdcm::ExplicitDataElement::Read (   
    std::istream & is )
```

### 10.119.2.3 ReadPreValue()

```
template<typename TSwap >  
std::istream& gdcm::ExplicitDataElement::ReadPreValue (   
    std::istream & is )
```

### 10.119.2.4 ReadValue()

```
template<typename TSwap >  
std::istream& gdcm::ExplicitDataElement::ReadValue (   
    std::istream & is,   
    bool readvalues = true )
```

### 10.119.2.5 ReadWithLength()

```
template<typename TSwap >  
std::istream& gdcm::ExplicitDataElement::ReadWithLength (   
    std::istream & is,   
    VL & length )
```

### 10.119.2.6 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:

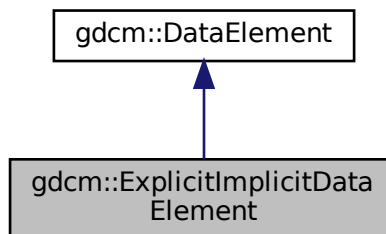
- [gdcmExplicitDataElement.h](#)

## 10.120 gdcm::ExplicitImplicitDataElement Class Reference

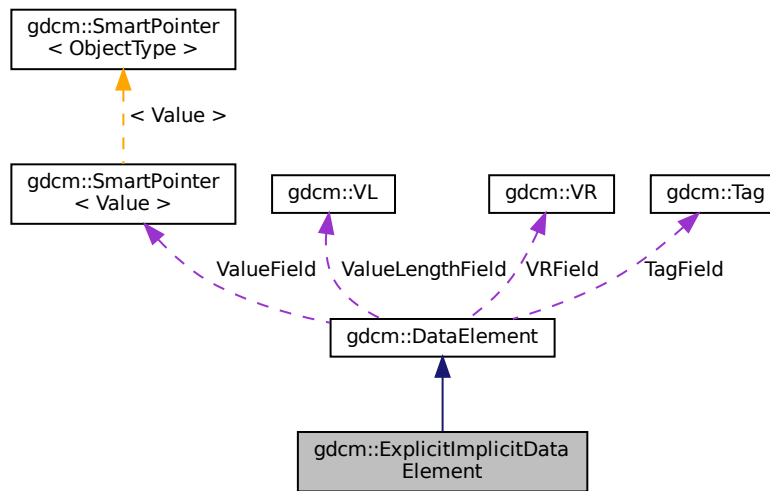
Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#).

```
#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)

## Additional Inherited Members

### 10.120.1 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.

## 10.120.2 Member Function Documentation

### 10.120.2.1 GetLength()

```
VL gdcm::ExplicitImplicitDataElement::GetLength ( ) const
```

### 10.120.2.2 Read()

```
template<typename TSwap >  
std::istream& gdcm::ExplicitImplicitDataElement::Read (   
    std::istream & is )
```

### 10.120.2.3 ReadPreValue()

```
template<typename TSwap >  
std::istream& gdcm::ExplicitImplicitDataElement::ReadPreValue (   
    std::istream & is )
```

### 10.120.2.4 ReadValue()

```
template<typename TSwap >  
std::istream& gdcm::ExplicitImplicitDataElement::ReadValue (   
    std::istream & is,   
    bool readvalues = true )
```

### 10.120.2.5 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](#)



## 10.121 gdcm::Fiducials Class Reference

[Fiducials.](#)

```
#include <gdcmFiducials.h>
```

### Public Member Functions

- [Fiducials](#) ()=default

#### 10.121.1 Detailed Description

[Fiducials.](#)

#### 10.121.2 Constructor & Destructor Documentation

##### 10.121.2.1 Fiducials()

```
gdcm::Fiducials::Fiducials ( ) [default]
```

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

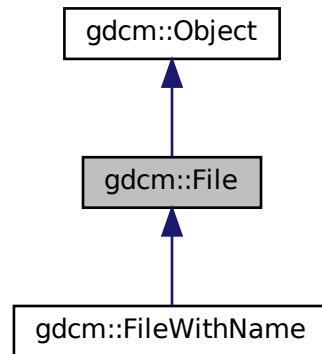
- [gdcmFiducials.h](#)

## 10.122 gdcm::File Class Reference

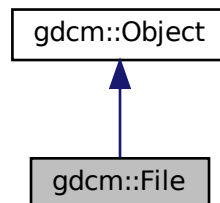
a DICOM [File](#)

```
#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.*

## Friends

- `std::ostream & operator<< (std::ostream &os, const File &val)`

## Additional Inherited Members

### 10.122.1 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](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [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](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), and [iU22tomultisc.cxx](#).

### 10.122.2 Constructor & Destructor Documentation

#### 10.122.2.1 File()

```
gdcm::File::File ( )
```

### 10.122.2.2 ~File()

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

## 10.122.3 Member Function Documentation

### 10.122.3.1 GetDataSet() [1/2]

```
DataSet& gdcm::File::GetDataSet ( ) [inline]
```

Get Data Set.

### 10.122.3.2 GetDataSet() [2/2]

```
const DataSet& gdcm::File::GetDataSet ( ) const [inline]
```

Get Data Set.

#### Examples

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CreateFakeRTDOSE.cxx](#), [CreateJIPDataSet.cxx](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Write.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenAIIVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), [csa2img.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

### 10.122.3.3 GetHeader() [1/2]

```
FileMetaInformation& gdcm::File::GetHeader ( ) [inline]
```

Get [File](#) Meta Information.

#### 10.122.3.4 GetHeader() [2/2]

```
const FileMetaInformation& gdcm::File::GetHeader ( ) const [inline]
```

Get [File](#) Meta Information.

##### Examples

[CreateJPIPDataSet.cxx](#), [DumpToshibaDTI.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](#), [StreamImageReaderTest.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

#### 10.122.3.5 Read()

```
std::istream& gdcm::File::Read (
    std::istream & is )
```

Read.

#### 10.122.3.6 SetDataSet()

```
void gdcm::File::SetDataSet (
    const DataSet & ds ) [inline]
```

Set Data Set.

#### 10.122.3.7 SetHeader()

```
void gdcm::File::SetHeader (
    const FileMetaInformation & fmi ) [inline]
```

Set [File](#) Meta Information.

#### 10.122.3.8 Write()

```
std::ostream const& gdcm::File::Write (
    std::ostream & os ) const
```

Write.

## 10.122.4 Friends And Related Function Documentation

### 10.122.4.1 operator<<

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

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

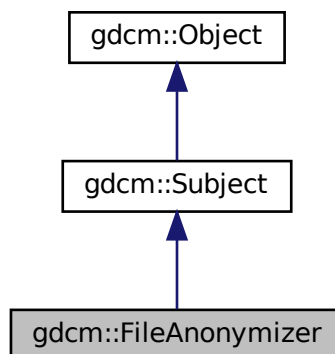
- [gdcmFile.h](#)

## 10.123 gdcm::FileAnonymizer Class Reference

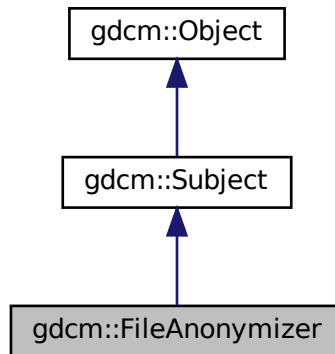
[FileAnonymizer](#).

```
#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.*

## Additional Inherited Members

### 10.123.1 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**

[MakeTemplate.cxx](#).

**10.123.2 Constructor & Destructor Documentation****10.123.2.1 FileAnonymizer()**

```
gdcm::FileAnonymizer::FileAnonymizer ( )
```

**10.123.2.2 ~FileAnonymizer()**

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

**10.123.3 Member Function Documentation****10.123.3.1 Empty()**

```
void gdcm::FileAnonymizer::Empty (
    Tag const & t )
```

Make [Tag](#) t empty Warning: does not handle SQ element

**Examples**

[MakeTemplate.cxx](#).



### 10.123.3.2 Remove()

```
void gdcm::FileAnonymizer::Remove (
    Tag const & t )
```

remove a tag (even a SQ can be removed)

### 10.123.3.3 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

### 10.123.3.4 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](#) !

### 10.123.3.5 SetInputFileName()

```
void gdcm::FileAnonymizer::SetInputFileName (
    const char * filename_native )
```

Set input filename.

#### Examples

[FileAnonymize.cs](#), and [MakeTemplate.cxx](#).

### 10.123.3.6 SetOutputFileName()

```
void gdcM::FileAnonymizer::SetOutputFileName (
    const char * filename_native )
```

Set output filename.

#### Examples

[MakeTemplate.cxx](#).

### 10.123.3.7 Write()

```
bool gdcM::FileAnonymizer::Write ( )
```

Write the output file.

#### Examples

[MakeTemplate.cxx](#).

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

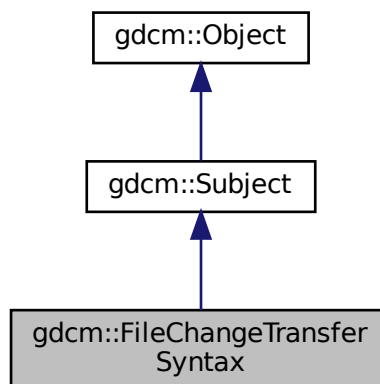
- [gdcMFileAnonymizer.h](#)

## 10.124 gdcM::FileChangeTransferSyntax Class Reference

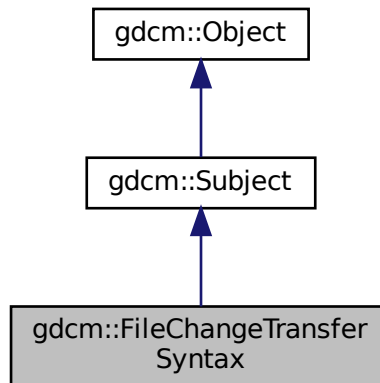
[FileChangeTransferSyntax](#).

```
#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.*

## Static Public Member Functions

- static [SmartPointer](#)< [FileChangeTransferSyntax](#) > [New](#) ()  
*for wrapped language: instantiate a reference counted object*

## Additional Inherited Members

### 10.124.1 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

## 10.124.2 Constructor & Destructor Documentation

### 10.124.2.1 FileChangeTransferSyntax()

```
gdcm::FileChangeTransferSyntax::FileChangeTransferSyntax ( )
```

### 10.124.2.2 ~FileChangeTransferSyntax()

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

## 10.124.3 Member Function Documentation

### 10.124.3.1 Change()

```
bool gdcm::FileChangeTransferSyntax::Change ( )
```

Change the transfer syntax.

### 10.124.3.2 GetCodec()

```
ImageCodec\* gdcm::FileChangeTransferSyntax::GetCodec ( )
```

Retrieve the actual codec (valid after calling SetTransferSyntax) Only advanced users should call this function.

### 10.124.3.3 New()

```
static SmartPointer<FileChangeTransferSyntax> gdcm::FileChangeTransferSyntax::New ( ) [inline],  
[static]
```

for wrapped language: instantiate a reference counted object

### 10.124.3.4 SetInputFileName()

```
void gdcm::FileChangeTransferSyntax::SetInputFileName (  
    const char * filename_native )
```

Set input filename (raw DICOM)

### 10.124.3.5 SetOutputFileName()

```
void gdcm::FileChangeTransferSyntax::SetOutputFileName (  
    const char * filename_native )
```

Set output filename (target compressed DICOM)

### 10.124.3.6 SetTransferSyntax()

```
void gdcm::FileChangeTransferSyntax::SetTransferSyntax (  
    TransferSyntax const & ts )
```

Specify the Target Transfer Syntax.

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

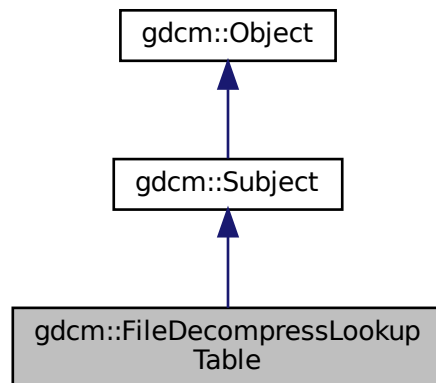
- [gdcmFileChangeTransferSyntax.h](#)

## 10.125 gdcm::FileDecompressLookupTable Class Reference

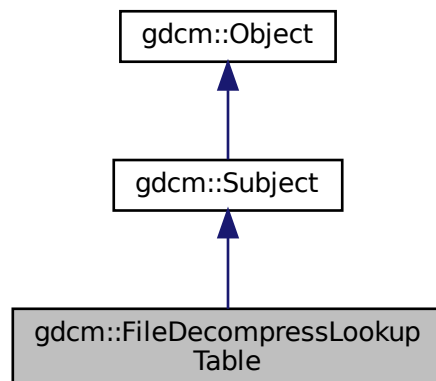
[FileDecompressLookupTable](#) class.

```
#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)

## Additional Inherited Members

### 10.125.1 Detailed Description

[FileDecompressLookupTable](#) class.

It decompress the segmented LUT into linearized one (only PALETTE\_COLOR images) Output will be a [PhotometricInterpretation](#)=RGB image

### 10.125.2 Constructor & Destructor Documentation

#### 10.125.2.1 FileDecompressLookupTable()

```
gdcm::FileDecompressLookupTable::FileDecompressLookupTable ( ) [default]
```

#### 10.125.2.2 ~FileDecompressLookupTable()

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

### 10.125.3 Member Function Documentation

### 10.125.3.1 Change()

```
bool gdcm::FileDecompressLookupTable::Change ( )
```

Decompress.

### 10.125.3.2 GetFile()

```
File& gdcm::FileDecompressLookupTable::GetFile ( ) [inline]
```

### 10.125.3.3 GetPixmap() [1/2]

```
Pixmap& gdcm::FileDecompressLookupTable::GetPixmap ( ) [inline]
```

### 10.125.3.4 GetPixmap() [2/2]

```
const Pixmap& gdcm::FileDecompressLookupTable::GetPixmap ( ) const [inline]
```

### 10.125.3.5 SetFile()

```
void gdcm::FileDecompressLookupTable::SetFile (
    const File & f ) [inline]
```

Set/Get [File](#).

### 10.125.3.6 SetPixmap()

```
void gdcm::FileDecompressLookupTable::SetPixmap (
    Pixmap const & img ) [inline]
```

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

- [gdcmFileDecompressLookupTable.h](#)



## 10.126 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](#) ()

### 10.126.1 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](#), and [GenFakelImage.cxx](#).

## 10.126.2 Constructor & Destructor Documentation

### 10.126.2.1 FileDerivation()

```
gdcm::FileDerivation::FileDerivation ( )
```

### 10.126.2.2 ~FileDerivation()

```
gdcm::FileDerivation::~~FileDerivation ( )
```

## 10.126.3 Member Function Documentation

### 10.126.3.1 AddDerivationDescription()

```
bool gdcm::FileDerivation::AddDerivationDescription ( ) [protected]
```

### 10.126.3.2 AddPurposeOfReferenceCodeSequence()

```
bool gdcm::FileDerivation::AddPurposeOfReferenceCodeSequence (
    DataSet & ds ) [protected]
```

### 10.126.3.3 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](#), and [GenFakelImage.cxx](#).

#### 10.126.3.4 AddSourceImageSequence()

```
bool gdcm::FileDerivation::AddSourceImageSequence ( ) [protected]
```

#### 10.126.3.5 Derive()

```
bool gdcm::FileDerivation::Derive ( )
```

Change.

##### Examples

[DeriveSeries.cxx](#), and [GenFakelImage.cxx](#).

#### 10.126.3.6 GetFile() [1/2]

```
File& gdcm::FileDerivation::GetFile ( ) [inline]
```

##### Examples

[GenFakelImage.cxx](#).

#### 10.126.3.7 GetFile() [2/2]

```
const File& gdcm::FileDerivation::GetFile ( ) const [inline]
```

#### 10.126.3.8 SetAppendDerivationHistory()

```
void gdcm::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.

### 10.126.3.9 SetDerivationCodeSequenceCodeValue()

```
void gdcM::FileDerivation::SetDerivationCodeSequenceCodeValue (
    unsigned int codevalue )
```

Specify the Derivation Code Sequence Code [Value](#). Eg 113040.

#### Examples

[DeriveSeries.cxx](#), and [GenFakelImage.cxx](#).

### 10.126.3.10 SetDerivationDescription()

```
void gdcM::FileDerivation::SetDerivationDescription (
    const char * dd )
```

Specify the Derivation Description. Eg "lossy conversion".

### 10.126.3.11 SetFile()

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

Set/Get [File](#).

#### Examples

[DeriveSeries.cxx](#), and [GenFakelImage.cxx](#).

### 10.126.3.12 SetPurposeOfReferenceCodeSequenceCodeValue()

```
void gdcM::FileDerivation::SetPurposeOfReferenceCodeSequenceCodeValue (
    unsigned int codevalue )
```

Specify the Purpose Of Reference Code [Value](#). Eg. 121320.

#### Examples

[DeriveSeries.cxx](#), and [GenFakelImage.cxx](#).

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

- [gdcMFileDerivation.h](#)

## 10.127 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)

### 10.127.1 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](#).

## 10.127.2 Constructor & Destructor Documentation

### 10.127.2.1 FileExplicitFilter()

```
gdcM::FileExplicitFilter::FileExplicitFilter ( ) [inline]
```

### 10.127.2.2 ~FileExplicitFilter()

```
gdcM::FileExplicitFilter::~~FileExplicitFilter ( ) [default]
```

## 10.127.3 Member Function Documentation

### 10.127.3.1 Change()

```
bool gdcM::FileExplicitFilter::Change ( )
```

Set FMI Transfer Syntax.

Change

#### Examples

[GenAIIVR.cxx](#), and [LargeVRDSEExplicit.cxx](#).

### 10.127.3.2 ChangeFMI()

```
bool gdcM::FileExplicitFilter::ChangeFMI ( ) [protected]
```

### 10.127.3.3 GetFile()

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

#### 10.127.3.4 ProcessDataSet()

```
bool gdcm::FileExplicitFilter::ProcessDataSet (
    DataSet & ds,
    Dicts const & dicts ) [protected]
```

#### 10.127.3.5 SetChangePrivateTags()

```
void gdcm::FileExplicitFilter::SetChangePrivateTags (
    bool b ) [inline]
```

Decide whether or not to [VR](#)ify private tags.

#### 10.127.3.6 SetFile()

```
void gdcm::FileExplicitFilter::SetFile (
    const File & f ) [inline]
```

Set/Get [File](#).

#### Examples

[GenAllVR.cxx](#), and [LargeVRDSExplicit.cxx](#).

#### 10.127.3.7 SetRecomputeItemLength()

```
void gdcm::FileExplicitFilter::SetRecomputeItemLength (
    bool b )
```

By default set Sequence & [Item](#) length to Undefined to avoid recomputing length:

#### 10.127.3.8 SetRecomputeSequenceLength()

```
void gdcm::FileExplicitFilter::SetRecomputeSequenceLength (
    bool b )
```

### 10.127.3.9 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:

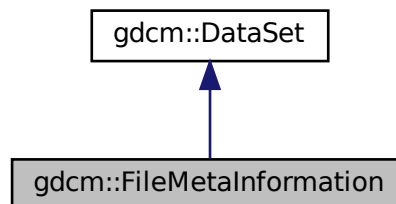
- [gdcmFileExplicitFilter.h](#)

## 10.128 gdcm::FileMetaInformation Class Reference

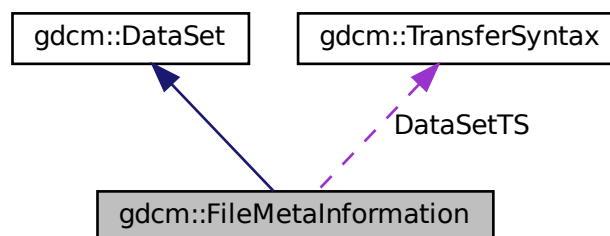
Class to represent a [File](#) Meta Information.

```
#include <gdcmFileMetaInformation.h>
```

Inheritance diagram for gdcm::FileMetaInformation:



Collaboration diagram for gdcm::FileMetaInformation:





## Public Member Functions

- [FileMetaInformation](#) ()
- [FileMetaInformation](#) ([FileMetaInformation](#) const &fmi)
- [~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
- 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.*

## 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)

## 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

### 10.128.1 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

[DumpToshibaDTI.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), and [ReadAndDumpDICOMDIR.cxx](#).

### 10.128.2 Constructor & Destructor Documentation

### 10.128.2.1 FileMetaInformation() [1/2]

```
gdcm::FileMetaInformation::FileMetaInformation ( )
```

### 10.128.2.2 ~FileMetaInformation()

```
gdcm::FileMetaInformation::~~FileMetaInformation ( )
```

### 10.128.2.3 FileMetaInformation() [2/2]

```
gdcm::FileMetaInformation::FileMetaInformation (
    FileMetaInformation const & fmi ) [inline]
```

References DataSetMS, DataSetTS, and MetaInformationTS.

## 10.128.3 Member Function Documentation

### 10.128.3.1 AppendImplementationClassUID()

```
static void gdcm::FileMetaInformation::AppendImplementationClassUID (
    const char * imp ) [static]
```

### 10.128.3.2 ComputeDataSetMediaStorageSOPClass()

```
void gdcm::FileMetaInformation::ComputeDataSetMediaStorageSOPClass ( ) [protected]
```

### 10.128.3.3 ComputeDataSetTransferSyntax()

```
void gdcm::FileMetaInformation::ComputeDataSetTransferSyntax ( ) [protected]
```

#### 10.128.3.4 Default()

```
void gdcm::FileMetaInformation::Default ( ) [protected]
```

#### 10.128.3.5 FillFromDataSet()

```
void gdcm::FileMetaInformation::FillFromDataSet (
    DataSet const & ds )
```

Construct a [FileMetaInformation](#) from an already existing [DataSet](#):

#### 10.128.3.6 GetDataSetTransferSyntax()

```
const TransferSyntax& gdcm::FileMetaInformation::GetDataSetTransferSyntax ( ) const [inline]
```

#### Examples

[GetJPEGSamplePrecision.cxx](#), and [MergeTwoFiles.cxx](#).

#### 10.128.3.7 GetFileMetaInformationVersion()

```
static const char* gdcm::FileMetaInformation::GetFileMetaInformationVersion ( ) [static], [protected]
```

#### 10.128.3.8 GetFullLength()

```
VL gdcm::FileMetaInformation::GetFullLength ( ) const [inline]
```

References [gdcm::VL::GetLength\(\)](#).

#### 10.128.3.9 GetGDCMImplementationClassUID()

```
static const char* gdcm::FileMetaInformation::GetGDCMImplementationClassUID ( ) [static], [protected]
```

#### 10.128.3.10 GetGDCMImplementationVersionName()

```
static const char* gdcm::FileMetaInformation::GetGDCMImplementationVersionName ( ) [static],  
[protected]
```

#### 10.128.3.11 GetGDCMSourceApplicationEntityTitle()

```
static const char* gdcm::FileMetaInformation::GetGDCMSourceApplicationEntityTitle ( ) [static],  
[protected]
```

#### 10.128.3.12 GetImplementationClassUID()

```
static const char* gdcm::FileMetaInformation::GetImplementationClassUID ( ) [static]
```

#### 10.128.3.13 GetImplementationVersionName()

```
static const char* gdcm::FileMetaInformation::GetImplementationVersionName ( ) [static]
```

#### 10.128.3.14 GetMediaStorage()

```
MediaStorage gdcm::FileMetaInformation::GetMediaStorage ( ) const
```

#### 10.128.3.15 GetMediaStorageAsString()

```
std::string gdcm::FileMetaInformation::GetMediaStorageAsString ( ) const
```

#### 10.128.3.16 GetMetaInformationTS()

```
TransferSyntax::NegociatedType gdcm::FileMetaInformation::GetMetaInformationTS ( ) const [inline]
```

**10.128.3.17 GetPreamble()** [1/2]

```
Preamble& gdcm::FileMetaInformation::GetPreamble ( ) [inline]
```

**10.128.3.18 GetPreamble()** [2/2]

```
const Preamble& gdcm::FileMetaInformation::GetPreamble ( ) const [inline]
```

Get [Preamble](#).

**10.128.3.19 GetSourceApplicationEntityTitle()**

```
static const char* gdcm::FileMetaInformation::GetSourceApplicationEntityTitle ( ) [static]
```

**10.128.3.20 Insert()**

```
void gdcm::FileMetaInformation::Insert (
    const DataElement & de ) [inline]
```

References [gdcmErrorMacro](#), [gdcm::Tag::GetGroup\(\)](#), and [gdcm::DataElement::GetTag\(\)](#).

**10.128.3.21 IsValid()**

```
bool gdcm::FileMetaInformation::IsValid ( ) const [inline]
```

**10.128.3.22 Read()**

```
std::istream& gdcm::FileMetaInformation::Read (
    std::istream & is )
```

Read.

### 10.128.3.23 ReadCompat()

```
std::istream& gdcm::FileMetaInformation::ReadCompat (
    std::istream & is )
```

### 10.128.3.24 ReadCompatInternal()

```
template<typename TSwap >
std::istream& gdcm::FileMetaInformation::ReadCompatInternal (
    std::istream & is ) [protected]
```

### 10.128.3.25 Replace()

```
void gdcm::FileMetaInformation::Replace (
    const DataElement & de ) [inline]
```

#### Examples

[LargeVRDSExplicit.cxx](#).

References [gdcm::DataElement::GetTag\(\)](#).

### 10.128.3.26 SetDataSetTransferSyntax()

```
void gdcm::FileMetaInformation::SetDataSetTransferSyntax (
    const TransferSyntax & ts )
```

#### Examples

[CreateJPIPDataSet.cxx](#), [DumpToshibaDTI.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](#).

### 10.128.3.27 SetImplementationClassUID()

```
static void gdcm::FileMetaInformation::SetImplementationClassUID (
    const char * imp ) [static]
```

Override the GDCM default values:

### 10.128.3.28 SetImplementationVersionName()

```
static void gdcm::FileMetaInformation::SetImplementationVersionName (
    const char * version ) [static]
```

### 10.128.3.29 SetPreamble()

```
void gdcm::FileMetaInformation::SetPreamble (
    const Preamble & p ) [inline]
```

### 10.128.3.30 SetSourceApplicationEntityTitle()

```
static void gdcm::FileMetaInformation::SetSourceApplicationEntityTitle (
    const char * title ) [static]
```

#### Examples

[ClinicalTrialIdentificationWorkflow.cs](#), [FixJAIBugJPEGLS.cxx](#), [GenerateDICOMDIR.cs](#), [ReformatFile.cs](#), and [StandardizeFiles.cs](#).

### 10.128.3.31 Write()

```
std::ostream& gdcm::FileMetaInformation::Write (
    std::ostream & os ) const
```

Write.

## 10.128.4 Friends And Related Function Documentation



#### 10.128.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const FileMetaInformation & _val ) [friend]
```

### 10.128.5 Member Data Documentation

#### 10.128.5.1 DataSetMS

[MediaStorage::MSType](#) gdcm::FileMetaInformation::DataSetMS [protected]

Referenced by FileMetaInformation().

#### 10.128.5.2 DataSetTS

[TransferSyntax](#) gdcm::FileMetaInformation::DataSetTS [protected]

Referenced by FileMetaInformation().

#### 10.128.5.3 MetaInformationTS

[TransferSyntax::NegociatedType](#) gdcm::FileMetaInformation::MetaInformationTS [protected]

Referenced by FileMetaInformation().

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

- [gdcmFileMetaInformation.h](#)

## 10.129 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)

### 10.129.1 Detailed Description

Class to manipulate file name's.

#### Note

OS independant representation of a filename (to query path, name and extension from a filename)

### 10.129.2 Constructor & Destructor Documentation

#### 10.129.2.1 [Filename\(\)](#)

```
gdcmm::Filename::Filename (
    const char * filename = "" ) [inline]
```

## 10.129.3 Member Function Documentation

### 10.129.3.1 EndWith()

```
bool gdcm::Filename::EndWith (
    const char ending[] ) const
```

Does the filename ends with a particular string ?

### 10.129.3.2 GetExtension()

```
const char* gdcm::Filename::GetExtension ( )
```

return only the extension part of a filename

### 10.129.3.3 GetFileName()

```
const char* gdcm::Filename::GetFileName ( ) const [inline]
```

Return the full filename.

### 10.129.3.4 GetName()

```
const char* gdcm::Filename::GetName ( )
```

return only the name part of a filename

### 10.129.3.5 GetPath()

```
const char* gdcm::Filename::GetPath ( )
```

Return only the path component of a filename.

#### Examples

[ClinicalTrialIdentificationWorkflow.cs](#).

### 10.129.3.6 IsEmpty()

```
bool gdcm::Filename::IsEmpty ( ) const [inline]
```

return whether the filename is empty

### 10.129.3.7 IsIdentical()

```
bool gdcm::Filename::IsIdentical (
    Filename const & fn ) const
```

### 10.129.3.8 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](#).

### 10.129.3.9 operator const char \*()

```
gdcm::Filename::operator const char * ( ) const [inline]
```

Simple operator to allow `Filename myfilename( "...")`; `const char * s = myfilename`;

### 10.129.3.10 ToUnixSlashes()

```
const char* gdcm::Filename::ToUnixSlashes ( )
```

Convert backslash (windows style) to UNIX style slash.

### 10.129.3.11 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:

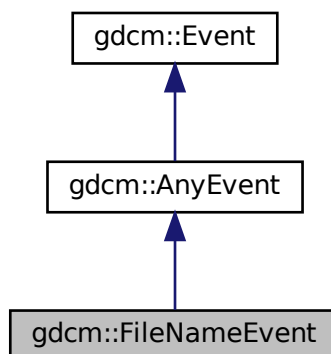
- [gdcmFilename.h](#)

## 10.130 gdcm::FileNameEvent Class Reference

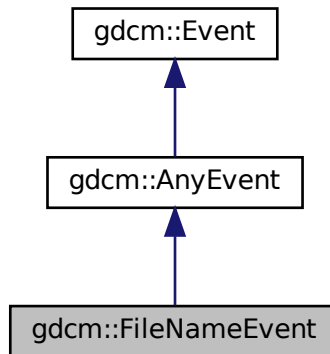
[FileNameEvent](#).

```
#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`)

### 10.130.1 Detailed Description

[FileNameEvent](#).

Special type of event triggered during processing of [FileSet](#)

See also

[AnyEvent](#)

Examples

[SimpleScanner.cxx](#).

## 10.130.2 Member Typedef Documentation

### 10.130.2.1 Self

```
typedef FileNameEvent gdcm::FileNameEvent::Self
```

### 10.130.2.2 Superclass

```
typedef AnyEvent gdcm::FileNameEvent::Superclass
```

## 10.130.3 Constructor & Destructor Documentation

### 10.130.3.1 FileNameEvent() [1/2]

```
gdcm::FileNameEvent::FileNameEvent (
    const char * s = "" ) [inline]
```

### 10.130.3.2 ~FileNameEvent()

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

### 10.130.3.3 FileNameEvent() [2/2]

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

## 10.130.4 Member Function Documentation

#### 10.130.4.1 CheckEvent()

```
bool gdcM::FileNameEvent::CheckEvent (
    const ::gdcM::Event * e ) const [inline], [override]
```

#### 10.130.4.2 GetEventName()

```
const char* gdcM::FileNameEvent::GetEventName ( ) const [inline], [override], [virtual]
```

Return the StringName associated with the event.

Implements [gdcM::Event](#).

#### 10.130.4.3 GetFileName()

```
const char* gdcM::FileNameEvent::GetFileName ( ) const [inline]
```

#### Examples

[SimpleScanner.cxx](#).

#### 10.130.4.4 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](#).

#### 10.130.4.5 operator=()

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



#### 10.130.4.6 SetFileName()

```
void gdcm::FileNameEvent::SetFileName (
    const char * f ) [inline]
```

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

- [gdcmFileNameEvent.h](#)

## 10.131 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.*

### 10.131.1 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](#).

### 10.131.2 Member Typedef Documentation

#### 10.131.2.1 FilenamesType

```
typedef std::vector<FilenameType> gdcm::FilenameGenerator::FilenamesType
```

#### 10.131.2.2 FilenameType

```
typedef std::string gdcm::FilenameGenerator::FilenameType
```

#### 10.131.2.3 SizeType

```
typedef FilenamesType::size\_type gdcm::FilenameGenerator::SizeType
```

### 10.131.3 Constructor & Destructor Documentation

### 10.131.3.1 FilenameGenerator()

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

### 10.131.3.2 ~FilenameGenerator()

```
gdcm::FilenameGenerator::~~FilenameGenerator ( ) [default]
```

## 10.131.4 Member Function Documentation

### 10.131.4.1 Generate()

```
bool gdcm::FilenameGenerator::Generate ( )
```

Generate (return success)

#### Examples

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

### 10.131.4.2 GetFilename()

```
const char* gdcm::FilenameGenerator::GetFilename (
    SizeType n ) const
```

Get a particular filename (call after Generate)

#### Examples

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

### 10.131.4.3 GetFileNames()

```
FilenameType const& gdcm::FilenameGenerator::GetFileNames ( ) const [inline]
```

Return all filenames.

#### 10.131.4.4 GetNumberOfFileNames()

```
SizeType gdcm::FilenameGenerator::GetNumberOfFileNames ( ) const
```

##### Examples

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

#### 10.131.4.5 GetPattern()

```
const char* gdcm::FilenameGenerator::GetPattern ( ) const [inline]
```

#### 10.131.4.6 GetPrefix()

```
const char* gdcm::FilenameGenerator::GetPrefix ( ) const [inline]
```

#### 10.131.4.7 SetNumberOfFileNames()

```
void gdcm::FilenameGenerator::SetNumberOfFileNames (
    SizeType nfiles )
```

Set/Get the number of filenames to generate.

##### Examples

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

#### 10.131.4.8 SetPattern()

```
void gdcm::FilenameGenerator::SetPattern (
    const char * pattern ) [inline]
```

Set/Get pattern.

##### Examples

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

#### 10.131.4.9 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](#)

## 10.132 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)

#### 10.132.1 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.

#### 10.132.2 Member Typedef Documentation

### 10.132.2.1 FileType

```
typedef std::vector<FileType> gdcm::FileSet::FileType
```

### 10.132.2.2 FileType

```
typedef std::string gdcm::FileSet::FileType
```

## 10.132.3 Constructor & Destructor Documentation

### 10.132.3.1 FileSet()

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

## 10.132.4 Member Function Documentation

### 10.132.4.1 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.

### 10.132.4.2 AddFile() [2/2]

```
void gdcm::FileSet::AddFile (
    File const & ) [inline]
```

**Deprecated** . Does nothing

### 10.132.4.3 GetFiles()

```
FileType const& gdcm::FileSet::GetFiles ( ) const [inline]
```

### 10.132.4.4 SetFiles()

```
void gdcm::FileSet::SetFiles (
    FileType const & files )
```

## 10.132.5 Friends And Related Function Documentation

### 10.132.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const FileSet & d ) [friend]
```

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

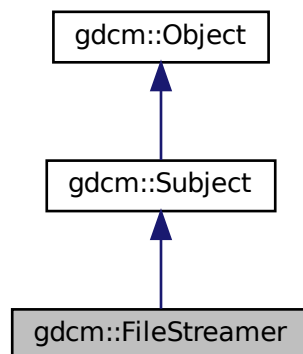
- [gdcmFileSet.h](#)

## 10.133 gdcm::FileStreamer Class Reference

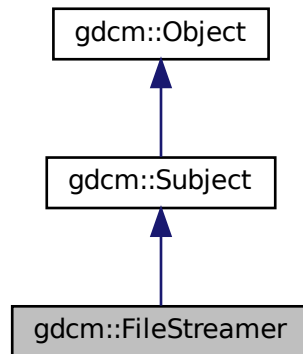
[FileStreamer.](#)

```
#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.*

## Static Public Member Functions

- static [SmartPointer](#)< [FileStreamer](#) > [New](#) ()  
*for wrapped language: instantiate a reference counted object*



## Additional Inherited Members

### 10.133.1 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.

### 10.133.2 Constructor & Destructor Documentation

#### 10.133.2.1 FileStreamer()

```
gdcm::FileStreamer::FileStreamer ( )
```

#### 10.133.2.2 ~FileStreamer()

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

### 10.133.3 Member Function Documentation

#### 10.133.3.1 AppendToDataElement()

```
bool gdcm::FileStreamer::AppendToDataElement (
    const Tag & t,
    const char * array,
    size_t len )
```

Append to previously started [Tag](#) t.

### 10.133.3.2 AppendToGroupDataElement()

```
bool gdcM::FileStreamer::AppendToGroupDataElement (
    const PrivateTag & pt,
    const char * array,
    size_t len )
```

Append to previously started private creator.

### 10.133.3.3 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

### 10.133.3.4 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.

### 10.133.3.5 New()

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

for wrapped language: instantiate a reference counted object

### 10.133.3.6 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

### 10.133.3.7 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

### 10.133.3.8 SetOutputFileName()

```
void gdcm::FileStreamer::SetOutputFileName (
    const char * filename_native )
```

Set output filename (target file)

### 10.133.3.9 SetTemplateFileName()

```
void gdcm::FileStreamer::SetTemplateFileName (
    const char * filename_native )
```

Set input DICOM template filename.

#### Examples

[FileStreaming.cs](#).

### 10.133.3.10 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.

### 10.133.3.11 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.
----------------	--

**10.133.3.12 StopDataElement()**

```
bool gdcM::FileStreamer::StopDataElement (
    const Tag & t )
```

Stop appending to tag t. This will compute the proper attribute length.

**10.133.3.13 StopGroupDataElement()**

```
bool gdcM::FileStreamer::StopGroupDataElement (
    const PrivateTag & pt )
```

Stop appending to private creator.

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

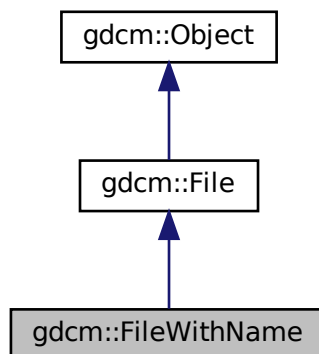
- [gdcMFileStreamer.h](#)

**10.134 gdcM::FileWithName Class Reference**

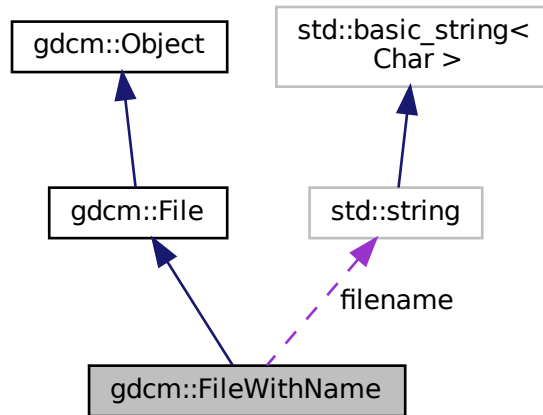
[FileWithName.](#)

```
#include <gdcMSerieHelper.h>
```

Inheritance diagram for gdcM::FileWithName:



Collaboration diagram for gdcm::FileWithName:



## Public Member Functions

- [FileWithName](#) ([File](#) &[f](#))

## Public Attributes

- `std::string` [filename](#)

## Additional Inherited Members

### 10.134.1 Detailed Description

[FileWithName](#).

Backward only class do not use in newer code

### 10.134.2 Constructor & Destructor Documentation

### 10.134.2.1 FileWithName()

```
gdcmm::FileWithName::FileWithName (
    File & f ) [inline]
```

## 10.134.3 Member Data Documentation

### 10.134.3.1 filename

```
std::string gdcmm::FileWithName::filename
```

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

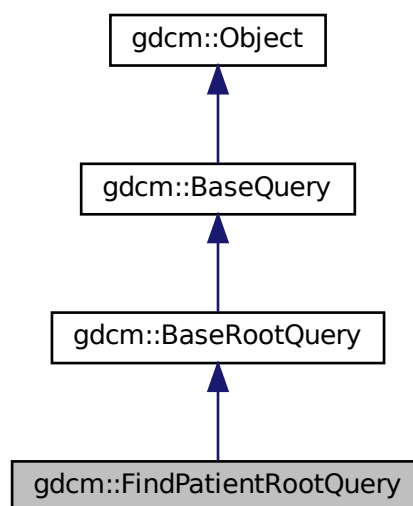
- [gdcmmSerieHelper.h](#)

## 10.135 gdcmm::FindPatientRootQuery Class Reference

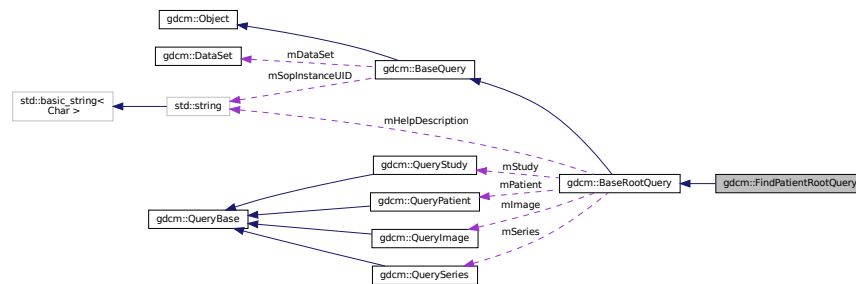
PatientRootQuery.

```
#include <gdcmmFindPatientRootQuery.h>
```

Inheritance diagram for gdcmm::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

## Friends

- class [QueryFactory](#)

## Additional Inherited Members

### 10.135.1 Detailed Description

PatientRootQuery.

contains: the class which will produce a dataset for c-find with patient root

### 10.135.2 Constructor & Destructor Documentation

#### 10.135.2.1 FindPatientRootQuery()

```
gdcm::FindPatientRootQuery::FindPatientRootQuery ( )
```

### 10.135.3 Member Function Documentation

#### 10.135.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcM::FindPatientRootQuery::GetAbstractSyntaxUID ( ) const [override], [virtual]
```

Implements [gdcM::BaseQuery](#).

#### 10.135.3.2 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](#).

#### 10.135.3.3 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](#).

#### 10.135.3.4 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](#).



## 10.135.4 Friends And Related Function Documentation

### 10.135.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

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

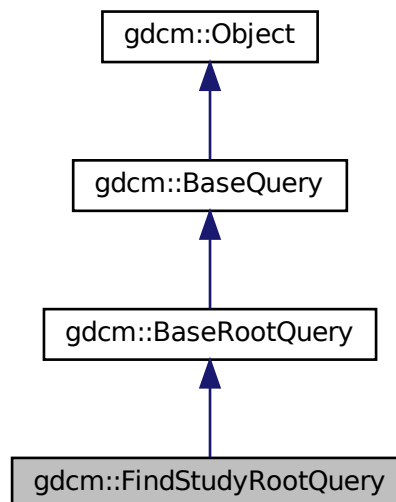
- [gdcmFindPatientRootQuery.h](#)

## 10.136 gdcm::FindStudyRootQuery Class Reference

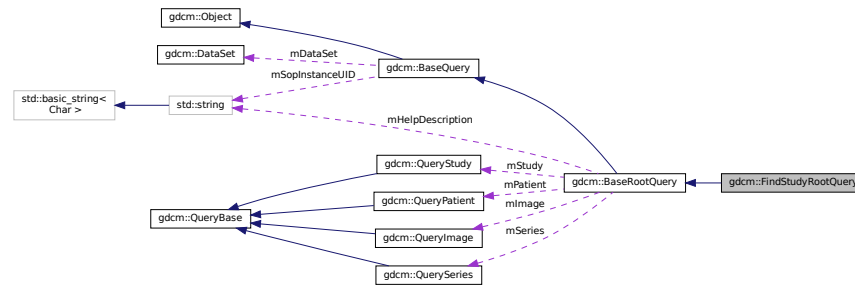
[FindStudyRootQuery](#).

```
#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

## Friends

- class [QueryFactory](#)

## Additional Inherited Members

### 10.136.1 Detailed Description

[FindStudyRootQuery](#).

contains: the class which will produce a dataset for C-FIND with study root

### 10.136.2 Constructor & Destructor Documentation

#### 10.136.2.1 FindStudyRootQuery()

```
gdcm::FindStudyRootQuery::FindStudyRootQuery ( )
```

## 10.136.3 Member Function Documentation

### 10.136.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::FindStudyRootQuery::GetAbstractSyntaxUID ( ) const [override], [virtual]
```

Implements [gdcm::BaseQuery](#).

### 10.136.3.2 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](#).

### 10.136.3.3 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](#).

### 10.136.3.4 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](#).

## 10.136.4 Friends And Related Function Documentation

### 10.136.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

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

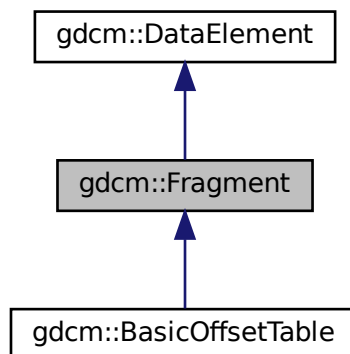
- [gdcmFindStudyRootQuery.h](#)

## 10.137 gdcm::Fragment Class Reference

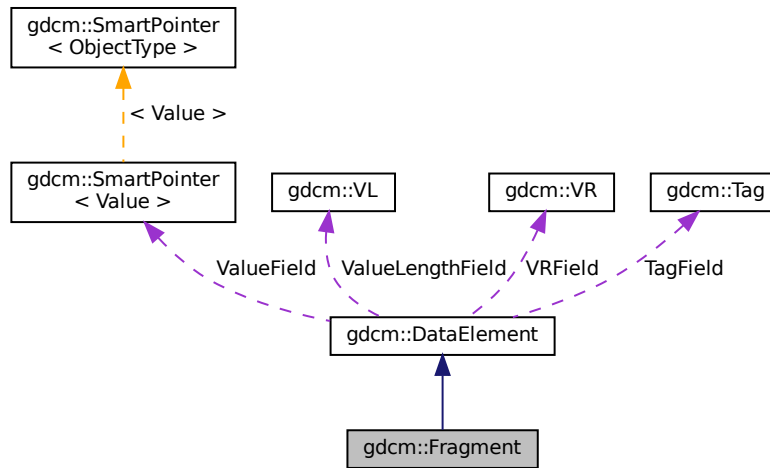
Class to represent a [Fragment](#).

```
#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

## Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [Fragment](#) &val)

## Additional Inherited Members

### 10.137.1 Detailed Description

Class to represent a [Fragment](#).

#### Examples

[FixBrokenJ2K.cxx](#), and [FixJAIBugJPEGLS.cxx](#).

## 10.137.2 Constructor & Destructor Documentation

### 10.137.2.1 Fragment()

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

## 10.137.3 Member Function Documentation

### 10.137.3.1 ComputeLength()

```
VL gdcm::Fragment::ComputeLength ( ) const
```

### 10.137.3.2 GetLength()

```
VL gdcm::Fragment::GetLength ( ) const
```

### 10.137.3.3 Read()

```
template<typename TSwap >  
std::istream& gdcm::Fragment::Read (   
    std::istream & is ) [inline]
```

Referenced by `gdcm::SequenceOfFragments::ReadValue()`.

### 10.137.3.4 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()`.

### 10.137.3.5 ReadPreValue()

```
template<typename TSwap >
std::istream& gdcm::Fragment::ReadPreValue (
    std::istream & is ) [inline]
```

### 10.137.3.6 ReadValue()

```
template<typename TSwap >
std::istream& gdcm::Fragment::ReadValue (
    std::istream & is ) [inline]
```

References `gdcmWarningMacro`, and `gdcm::ParseException::SetLastElement()`.

### 10.137.3.7 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()`.

## 10.137.4 Friends And Related Function Documentation

### 10.137.4.1 operator<<

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

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

- [gdcmFragment.h](#)

## 10.138 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)

### 10.138.1 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

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenerateStandardSOPClasses.cxx](#), [PublicDict.cxx](#), and [ReadAndPrintAttributes.cxx](#).

### 10.138.2 Constructor & Destructor Documentation



### 10.138.2.1 Global() [1/2]

```
gdcm::Global::Global ( )
```

### 10.138.2.2 ~Global()

```
gdcm::Global::~~Global ( )
```

### 10.138.2.3 Global() [2/2]

```
gdcm::Global::Global (
    const Global & _val ) [delete]
```

## 10.138.3 Member Function Documentation

### 10.138.3.1 Append()

```
bool gdcm::Global::Append (
    const char * path )
```

Append path at the end of the path list

#### Warning

not thread safe !

### 10.138.3.2 GetDefs()

```
Defs const& gdcm::Global::GetDefs ( ) const
```

retrieve the default/internal (Part 3) You need to explicitly call LoadResourcesFiles before

#### Examples

[GenerateStandardSOPClasses.cxx](#).

### 10.138.3.3 GetDicts() [1/2]

```
Dicts& gdcm::Global::GetDicts ( )
```

### 10.138.3.4 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](#), and [ReadAndPrintAttributes.cxx](#).

### 10.138.3.5 GetInstance()

```
static Global& gdcm::Global::GetInstance ( ) [static]
```

return the singleton instance

#### Examples

[BasicAnonymizer.cs](#), [ClinicalTrialIdentificationWorkflow.cs](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenerateStandardSOPClasses](#), [MrProtocol.cxx](#), [PublicDict.cxx](#), [ReadAndPrintAttributes.cxx](#), and [TraverseModules.cxx](#).

### 10.138.3.6 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

[GenerateStandardSOPClasses.cxx](#).

### 10.138.3.7 Locate()

```
const char* gdcm::Global::Locate (
    const char * resfile ) const [protected]
```

Locate a resource file.

### 10.138.3.8 operator=()

```
Global& gdcm::Global::operator= (
    const Global & _val ) [delete]
```

### 10.138.3.9 Prepend()

```
bool gdcm::Global::Prepend (
    const char * path )
```

Prepend path at the beginning of the path list

Warning

not thread safe !

## 10.138.4 Friends And Related Function Documentation

### 10.138.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const Global & g ) [friend]
```

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

- [gdcmGlobal.h](#)

## 10.139 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)

### 10.139.1 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 ?

### 10.139.2 Member Typedef Documentation

#### 10.139.2.1 GroupStringVector

```
typedef std::vector<std::string> gdc::GroupDict::GroupStringVector
```

### 10.139.3 Constructor & Destructor Documentation

### 10.139.3.1 GroupDict()

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

### 10.139.3.2 ~GroupDict()

```
gdcm::GroupDict::~~GroupDict ( ) [default]
```

## 10.139.4 Member Function Documentation

### 10.139.4.1 Add()

```
void gdcm::GroupDict::Add (
    std::string const & abbreviation,
    std::string const & name ) [protected]
```

### 10.139.4.2 GetAbbreviation()

```
std::string const& gdcm::GroupDict::GetAbbreviation (
    uint16_t num ) const
```

### 10.139.4.3 GetName()

```
std::string const& gdcm::GroupDict::GetName (
    uint16_t num ) const
```

### 10.139.4.4 Insert()

```
void gdcm::GroupDict::Insert (
    uint16_t num,
    std::string const & abbreviation,
    std::string const & name ) [protected]
```

#### 10.139.4.5 Size()

```
size_t gdcmm::GroupDict::Size ( ) const [inline]
```

### 10.139.5 Friends And Related Function Documentation

#### 10.139.5.1 operator<<

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

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

- [gdcmmGroupDict.h](#)

## 10.140 gdcmm::IconImageFilter Class Reference

[IconImageFilter](#).

```
#include <gdcmmIconImageFilter.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](#) ()

### 10.140.1 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](#).

### 10.140.2 Constructor & Destructor Documentation

#### 10.140.2.1 IconImageFilter()

```
gdcm::IconImageFilter::IconImageFilter ( )
```

#### 10.140.2.2 ~IconImageFilter()

```
gdcm::IconImageFilter::~~IconImageFilter ( )
```

### 10.140.3 Member Function Documentation

#### 10.140.3.1 Extract()

```
bool gdcM::IconImageFilter::Extract ( )
```

Extract all Icon found in [File](#).

##### Examples

[ExtractIconFromFile.cxx](#).

#### 10.140.3.2 ExtractIconImages()

```
void gdcM::IconImageFilter::ExtractIconImages ( ) [protected]
```

#### 10.140.3.3 ExtractVeprolIconImages()

```
void gdcM::IconImageFilter::ExtractVeproIconImages ( ) [protected]
```

#### 10.140.3.4 GetFile() [1/2]

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

#### 10.140.3.5 GetFile() [2/2]

```
const File& gdcM::IconImageFilter::GetFile ( ) const [inline]
```



### 10.140.3.6 GetIconImage()

```
IconImage& gdcm::IconImageFilter::GetIconImage (
    unsigned int i ) const
```

#### Examples

[ExtractIconFromFile.cxx](#).

### 10.140.3.7 GetNumberOfIconImages()

```
unsigned int gdcm::IconImageFilter::GetNumberOfIconImages ( ) const
```

Retrieve extract IconImage (need to call Extract first)

#### Examples

[ExtractIconFromFile.cxx](#).

### 10.140.3.8 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](#)

## 10.141 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.*

### 10.141.1 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](#).

### 10.141.2 Constructor & Destructor Documentation

### 10.141.2.1 IconImageGenerator()

```
gdcm::IconImageGenerator::IconImageGenerator ( )
```

### 10.141.2.2 ~IconImageGenerator()

```
gdcm::IconImageGenerator::~~IconImageGenerator ( )
```

## 10.141.3 Member Function Documentation

### 10.141.3.1 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](#).

### 10.141.3.2 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

### 10.141.3.3 Generate()

```
bool gdcm::IconImageGenerator::Generate ( )
```

Generate Icon.

#### Examples

[ExtractIconFromFile.cxx](#).

#### 10.141.3.4 GetIconImage()

```
const IconImage& gdcm::IconImageGenerator::GetIconImage ( ) const [inline]
```

Retrieve generated Icon.

##### Examples

[ExtractIconFromFile.cxx](#).

#### 10.141.3.5 GetPixmap() [1/2]

```
Pixmap& gdcm::IconImageGenerator::GetPixmap ( ) [inline]
```

#### 10.141.3.6 GetPixmap() [2/2]

```
const Pixmap& gdcm::IconImageGenerator::GetPixmap ( ) const [inline]
```

#### 10.141.3.7 SetOutputDimensions()

```
void gdcm::IconImageGenerator::SetOutputDimensions (
    const unsigned int dims[2] )
```

Set Target dimension of output Icon.

##### Examples

[ExtractIconFromFile.cxx](#).

#### 10.141.3.8 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)`

### 10.141.3.9 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.

### 10.141.3.10 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](#)

## 10.142 gdcm::ignore\_char Struct Reference

```
#include <gdcmElement.h>
```

### Public Member Functions

- [ignore\\_char](#) (char *c*)

### Public Attributes

- char [m\\_char](#)

### 10.142.1 Constructor & Destructor Documentation

### 10.142.1.1 ignore\_char()

```
gdcM::ignore_char::ignore_char (  
    char c )    [inline]
```

## 10.142.2 Member Data Documentation

### 10.142.2.1 m\_char

```
char gdcM::ignore_char::m_char
```

Referenced by `gdcM::operator>>()`.

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

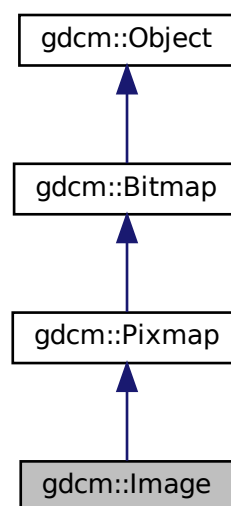
- [gdcMElement.h](#)

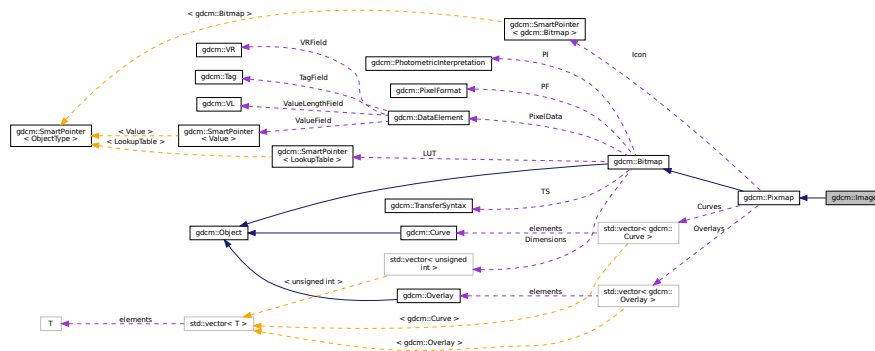
## 10.143 gdcM::Image Class Reference

[Image](#).

```
#include <gdcMImage.h>
```

Inheritance diagram for `gdcM::Image`:





- **Image** ()
- **~Image** () override=default
- const double \* **GetDirectionCosines** () const
- double **GetDirectionCosines** (unsigned int idx) const
- double **GetIntercept** () const
- const double \* **GetOrigin** () const
- double **GetOrigin** (unsigned int idx) const
- double **GetSlope** () const
- const double \* **GetSpacing** () const
- double **GetSpacing** (unsigned int idx) const
- void **Print** (std::ostream &os) const override

*print*

- void **SetDirectionCosines** (const double dircos[6])
- void **SetDirectionCosines** (const float dircos[6])
- void **SetDirectionCosines** (unsigned int idx, double dircos)
- void **SetIntercept** (double intercept)

*intercept*

- void **SetOrigin** (const double origin[3])
- void **SetOrigin** (const float origin[3])
- void **SetOrigin** (unsigned int idx, double ori)
- void **SetSlope** (double slope)

*slope*

- void **SetSpacing** (const double spacing[3])
- void **SetSpacing** (unsigned int idx, double spacing)

## Additional Inherited Members

### 10.143.1 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

[CompressImage.cxx](#), [ConvertToQImage.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [ExtractIconFromFile.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenFakeImage.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [PatchFile.cxx](#), [PrintLUT.cxx](#), [ReadMultiTimesException.cxx](#), [TemplateEmptyImage.cxx](#), [csa2img.cxx](#), [iU22tomultisc.cxx](#), and [threadgdc.cxx](#).

### 10.143.2 Constructor & Destructor Documentation

#### 10.143.2.1 Image()

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



### 10.143.2.2 ~Image()

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

## 10.143.3 Member Function Documentation

### 10.143.3.1 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.

### 10.143.3.2 GetDirectionCosines() [2/2]

```
double gdcm::Image::GetDirectionCosines (
    unsigned int idx ) const
```

### 10.143.3.3 GetIntercept()

```
double gdcm::Image::GetIntercept ( ) const [inline]
```

### 10.143.3.4 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](#).

#### 10.143.3.5 GetOrigin() [2/2]

```
double gdcM::Image::GetOrigin (
    unsigned int idx ) const
```

#### 10.143.3.6 GetSlope()

```
double gdcM::Image::GetSlope ( ) const [inline]
```

#### 10.143.3.7 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 specified, a default value of 1 will be returned

#### 10.143.3.8 GetSpacing() [2/2]

```
double gdcM::Image::GetSpacing (
    unsigned int idx ) const
```

#### 10.143.3.9 Print()

```
void gdcM::Image::Print (
    std::ostream & os ) const [override], [virtual]
```

print

Reimplemented from [gdcM::Bitmap](#).

#### Examples

[CompressImage.cxx](#), and [PatchFile.cxx](#).

**10.143.3.10 SetDirectionCosines()** [1/3]

```
void gdcm::Image::SetDirectionCosines (
    const double dircos[6] )
```

**10.143.3.11 SetDirectionCosines()** [2/3]

```
void gdcm::Image::SetDirectionCosines (
    const float dircos[6] )
```

**10.143.3.12 SetDirectionCosines()** [3/3]

```
void gdcm::Image::SetDirectionCosines (
    unsigned int idx,
    double dircos )
```

**10.143.3.13 SetIntercept()**

```
void gdcm::Image::SetIntercept (
    double intercept ) [inline]
```

*intercept*

**Examples**

[TemplateEmptyImage.cxx](#).

**10.143.3.14 SetOrigin()** [1/3]

```
void gdcm::Image::SetOrigin (
    const double origin[3] )
```

**10.143.3.15 SetOrigin() [2/3]**

```
void gdcM::Image::SetOrigin (
    const float origin[3] )
```

**10.143.3.16 SetOrigin() [3/3]**

```
void gdcM::Image::SetOrigin (
    unsigned int idx,
    double ori )
```

**10.143.3.17 SetSlope()**

```
void gdcM::Image::SetSlope (
    double slope ) [inline]
```

*slope*

**Examples**

[TemplateEmptyImage.cxx](#).

**10.143.3.18 SetSpacing() [1/2]**

```
void gdcM::Image::SetSpacing (
    const double spacing[3] )
```

**Examples**

[csa2img.cxx](#), and [iU22tomultisc.cxx](#).

**10.143.3.19 SetSpacing() [2/2]**

```
void gdcM::Image::SetSpacing (
    unsigned int idx,
    double spacing )
```

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

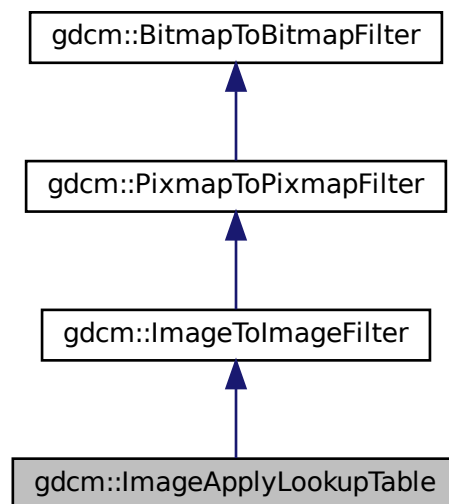
- [gdcMImage.h](#)

## 10.144 gdcm::ImageApplyLookupTable Class Reference

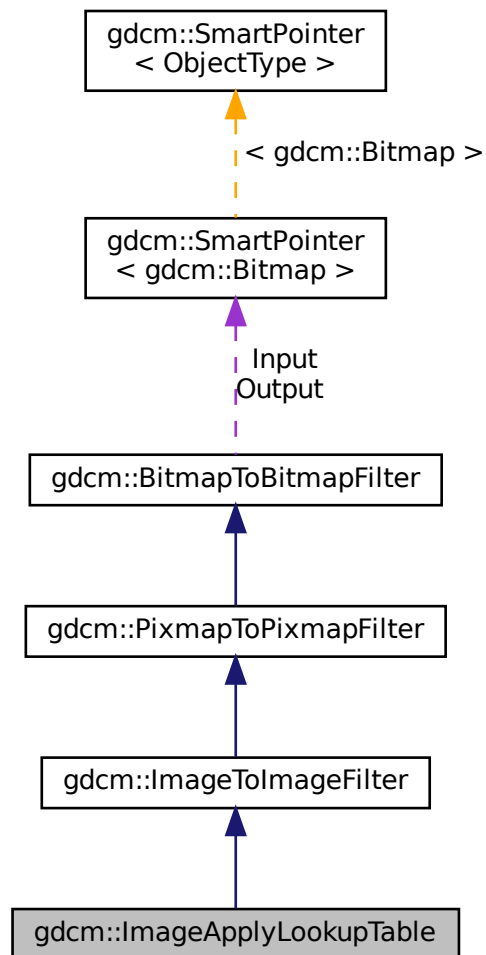
[ImageApplyLookupTable](#) class.

```
#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 ?*

## Additional Inherited Members

### 10.144.1 Detailed Description

[ImageApplyLookupTable](#) class.

It applies the LUT the PixelData (only PALETTE\_COLOR images) Output will be a [PhotometricInterpretation=RGB](#) image

### 10.144.2 Constructor & Destructor Documentation

#### 10.144.2.1 ImageApplyLookupTable()

```
gdcm::ImageApplyLookupTable::ImageApplyLookupTable ( )
```

#### 10.144.2.2 ~ImageApplyLookupTable()

```
gdcm::ImageApplyLookupTable::~~ImageApplyLookupTable ( )
```

### 10.144.3 Member Function Documentation

#### 10.144.3.1 Apply()

```
bool gdcm::ImageApplyLookupTable::Apply ( )
```

Apply.

#### 10.144.3.2 SetRGB8()

```
void gdcm::ImageApplyLookupTable::SetRGB8 (
    bool b )
```

RGB8 ?

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

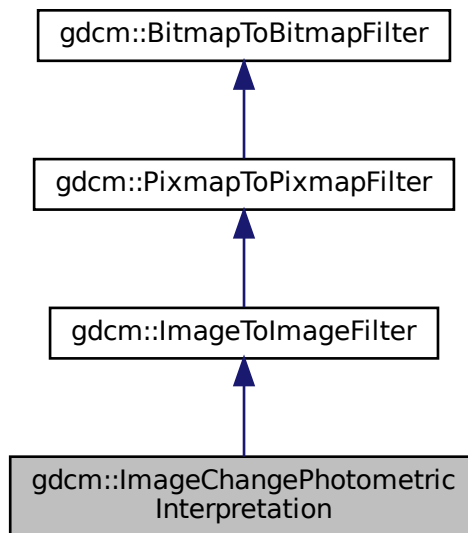
- [gdcmImageApplyLookupTable.h](#)

## 10.145 gdcm::ImageChangePhotometricInterpretation Class Reference

[ImageChangePhotometricInterpretation](#) class.

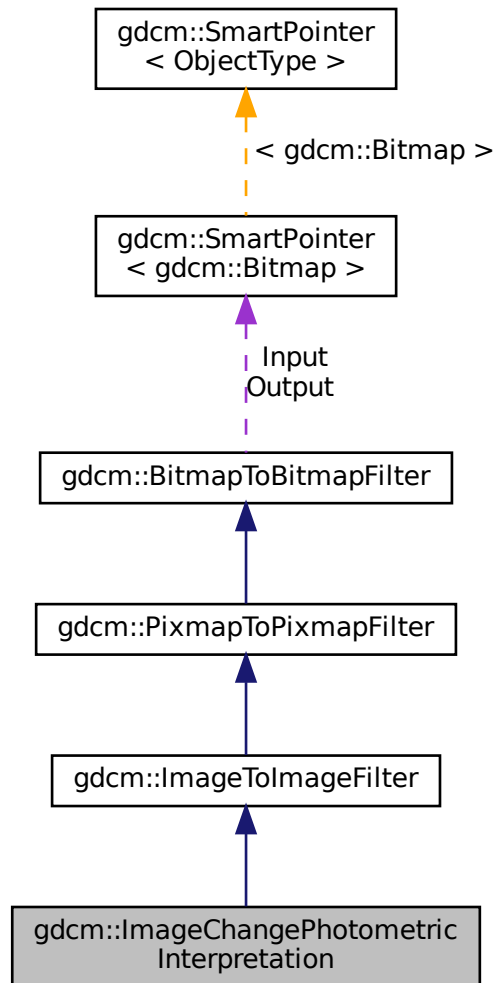
```
#include <gdcmImageChangePhotometricInterpretation.h>
```

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](#).*

## 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

### 10.145.1 Detailed Description

[ImageChangePhotometricInterpretation](#) class.

Class to change the Photometric Interpretation of an input DICOM

### 10.145.2 Constructor & Destructor Documentation

#### 10.145.2.1 [ImageChangePhotometricInterpretation\(\)](#)

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

#### 10.145.2.2 [~ImageChangePhotometricInterpretation\(\)](#)

```
gdcm::ImageChangePhotometricInterpretation::~~ImageChangePhotometricInterpretation ( ) [default]
```

### 10.145.3 Member Function Documentation

### 10.145.3.1 Change()

```
bool gdcm::ImageChangePhotometricInterpretation::Change ( )
```

Change.

### 10.145.3.2 ChangeMonochrome()

```
bool gdcm::ImageChangePhotometricInterpretation::ChangeMonochrome ( ) [protected]
```

### 10.145.3.3 ChangeRGB2YBR()

```
bool gdcm::ImageChangePhotometricInterpretation::ChangeRGB2YBR ( ) [protected]
```

### 10.145.3.4 ChangeYBR2RGB()

```
bool gdcm::ImageChangePhotometricInterpretation::ChangeYBR2RGB ( ) [protected]
```

### 10.145.3.5 GetPhotometricInterpretation()

```
const PhotometricInterpretation& gdcm::ImageChangePhotometricInterpretation::GetPhotometricInterpretation ( ) const [inline]
```

### 10.145.3.6 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().

### 10.145.3.7 SetPhotometricInterpretation()

```
void gdcm::ImageChangePhotometricInterpretation::SetPhotometricInterpretation (
    PhotometricInterpretation const & pi ) [inline]
```

Set/Get requested [PhotometricInterpretation](#).

### 10.145.3.8 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:

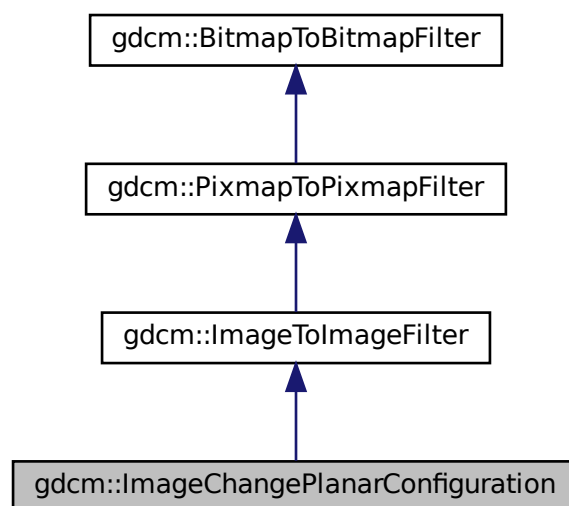
- [gdcmImageChangePhotometricInterpretation.h](#)

## 10.146 gdcm::ImageChangePlanarConfiguration Class Reference

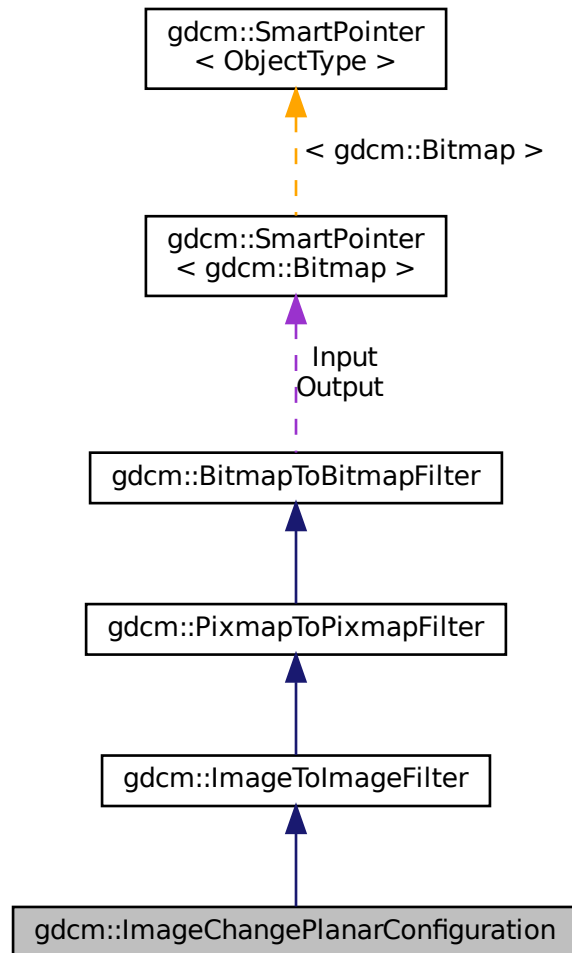
[ImageChangePlanarConfiguration](#) class.

```
#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.*

## 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

### 10.146.1 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

### 10.146.2 Constructor & Destructor Documentation

#### 10.146.2.1 ImageChangePlanarConfiguration()

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

#### 10.146.2.2 ~ImageChangePlanarConfiguration()

```
gdcm::ImageChangePlanarConfiguration::~~ImageChangePlanarConfiguration ( ) [default]
```

### 10.146.3 Member Function Documentation

#### 10.146.3.1 Change()

```
bool gdcm::ImageChangePlanarConfiguration::Change ( )
```

Change.

### 10.146.3.2 GetPlanarConfiguration()

```
unsigned int gdcm::ImageChangePlanarConfiguration::GetPlanarConfiguration ( ) const [inline]
```

### 10.146.3.3 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

### 10.146.3.4 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

### 10.146.3.5 SetPlanarConfiguration()

```
void gdcm::ImageChangePlanarConfiguration::SetPlanarConfiguration (
    unsigned int pc ) [inline]
```

Set/Get requested PlanarConfiguration.

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

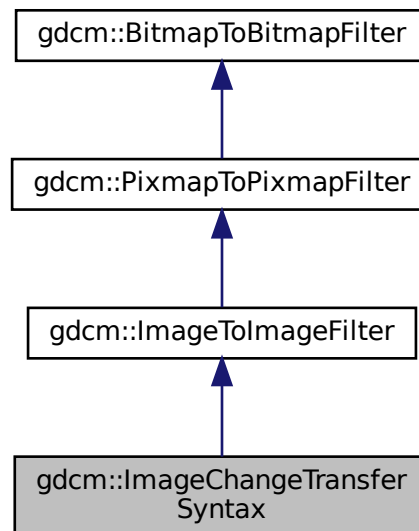
- [gdcmImageChangePlanarConfiguration.h](#)

## 10.147 gdcm::ImageChangeTransferSyntax Class Reference

[ImageChangeTransferSyntax](#) class.

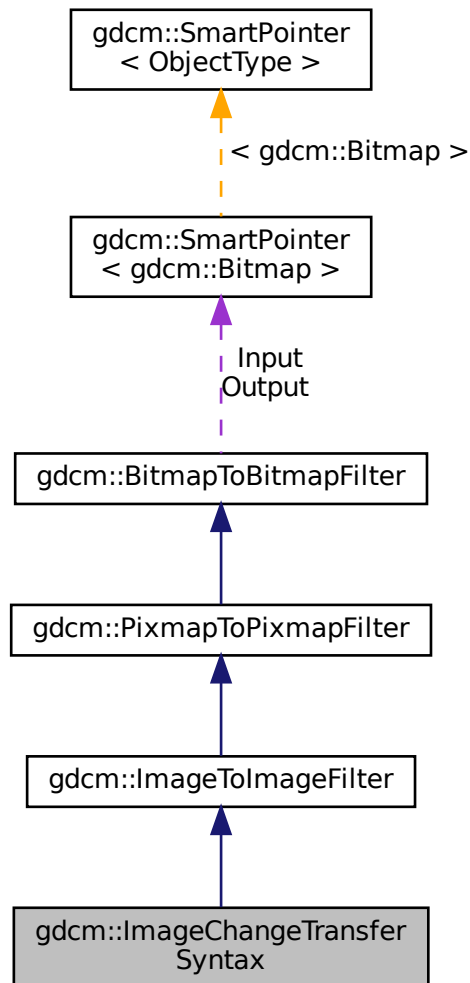
```
#include <gdcmImageChangeTransferSyntax.h>
```

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)

## Protected Member Functions

- bool [TryJPEG2000Codec](#) (const [DataElement](#) &pixelde, [Bitmap](#) const &input, [Bitmap](#) &output)
- bool [TryJPEGCodec](#) (const [DataElement](#) &pixelde, [Bitmap](#) const &input, [Bitmap](#) &output)
- bool [TryJPEGLSCodec](#) (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

### 10.147.1 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

[CompressImage.cxx](#).

### 10.147.2 Constructor & Destructor Documentation

#### 10.147.2.1 ImageChangeTransferSyntax()

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

#### 10.147.2.2 ~ImageChangeTransferSyntax()

```
gdcm::ImageChangeTransferSyntax::~~ImageChangeTransferSyntax ( ) [default]
```

### 10.147.3 Member Function Documentation

#### 10.147.3.1 Change()

```
bool gdcm::ImageChangeTransferSyntax::Change ( )
```

Change.

##### Examples

[CompressImage.cxx](#).

#### 10.147.3.2 GetTransferSyntax()

```
const TransferSyntax& gdcm::ImageChangeTransferSyntax::GetTransferSyntax ( ) const [inline]
```

Get Transfer Syntax.

#### 10.147.3.3 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

#### 10.147.3.4 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

### 10.147.3.5 SetTransferSyntax()

```
void gdcm::ImageChangeTransferSyntax::SetTransferSyntax (
    const TransferSyntax & ts ) [inline]
```

Set target Transfer Syntax.

#### Examples

[CompressImage.cxx](#).

### 10.147.3.6 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 )`

### 10.147.3.7 TryJPEG2000Codec()

```
bool gdcm::ImageChangeTransferSyntax::TryJPEG2000Codec (
    const DataElement & pixelde,
    Bitmap const & input,
    Bitmap & output ) [protected]
```

### 10.147.3.8 TryJPEGCodec()

```
bool gdcm::ImageChangeTransferSyntax::TryJPEGCodec (
    const DataElement & pixelde,
    Bitmap const & input,
    Bitmap & output ) [protected]
```

### 10.147.3.9 TryJPEGLSCodec()

```
bool gdcm::ImageChangeTransferSyntax::TryJPEGLSCodec (
    const DataElement & pixelde,
    Bitmap const & input,
    Bitmap & output ) [protected]
```

### 10.147.3.10 TryRAWCodec()

```
bool gdcm::ImageChangeTransferSyntax::TryRAWCodec (
    const DataElement & pixelde,
    Bitmap const & input,
    Bitmap & output ) [protected]
```

### 10.147.3.11 TryRLECodec()

```
bool gdcm::ImageChangeTransferSyntax::TryRLECodec (
    const DataElement & pixelde,
    Bitmap const & input,
    Bitmap & output ) [protected]
```

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

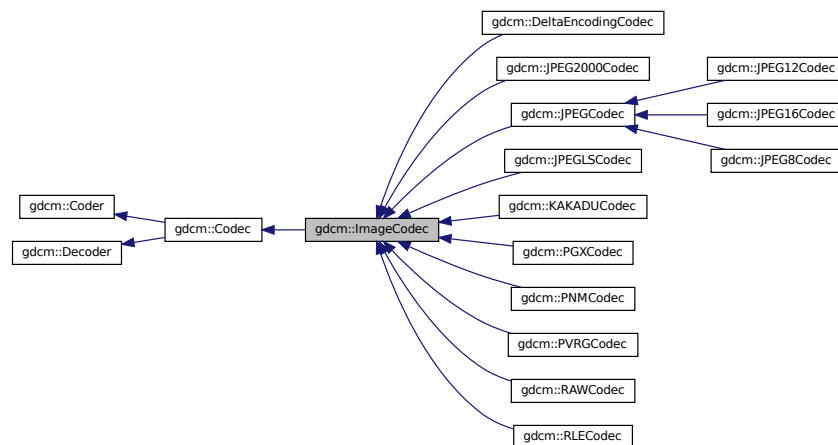
- [gdcmImageChangeTransferSyntax.h](#)

## 10.148 gdcm::ImageCodec Class Reference

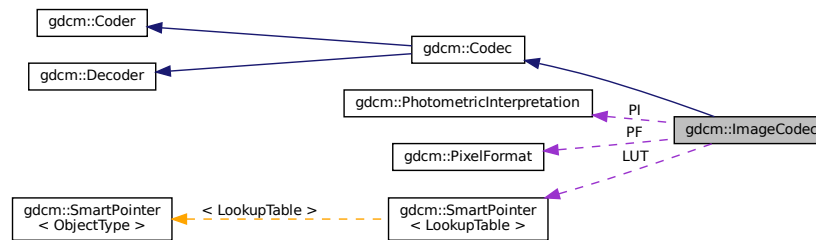
[ImageCodec](#).

```
#include <gdcmImageCodec.h>
```

Inheritance diagram for gdcm::ImageCodec:



Collaboration diagram for `gdcm::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)

## 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 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](#)

### 10.148.1 Detailed Description

[ImageCodec](#).

#### Note

Main codec, this is a central place for all implementation

## 10.148.2 Member Typedef Documentation

### 10.148.2.1 LUTPtr

```
typedef SmartPointer<LookupTable> gdcM::ImageCodec::LUTPtr [protected]
```

## 10.148.3 Constructor & Destructor Documentation

### 10.148.3.1 ImageCodec()

```
gdcM::ImageCodec::ImageCodec ( )
```

### 10.148.3.2 ~ImageCodec()

```
gdcM::ImageCodec::~~ImageCodec ( ) [override]
```

## 10.148.4 Member Function Documentation

### 10.148.4.1 AppendFrameEncode()

```
virtual bool gdcM::ImageCodec::AppendFrameEncode (  
    std::ostream & out,  
    const char * data,  
    size_t datalen ) [protected], [virtual]
```

Reimplemented in [gdcM::RLECodec](#), [gdcM::JPEGLSCodec](#), [gdcM::JPEGCodec](#), and [gdcM::JPEG2000Codec](#).



#### 10.148.4.2 AppendRowEncode()

```
virtual bool gdcm::ImageCodec::AppendRowEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [protected], [virtual]
```

Reimplemented in [gdcm::RLECodec](#), [gdcm::JPEGLSCodec](#), [gdcm::JPEGCodec](#), and [gdcm::JPEG2000Codec](#).

#### 10.148.4.3 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::RLECodec](#), [gdcm::RAWCodec](#), [gdcm::PVRGCodec](#), [gdcm::PNMCodec](#), [gdcm::PGXCodec](#), [gdcm::KAKADUCodec](#), [gdcm::JPEGLSCodec](#), [gdcm::JPEGCodec](#), and [gdcm::JPEG2000Codec](#).

#### 10.148.4.4 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::RLECodec](#), [gdcm::RAWCodec](#), [gdcm::PVRGCodec](#), [gdcm::PNMCodec](#), [gdcm::PGXCodec](#), [gdcm::KAKADUCodec](#), [gdcm::JPEGLSCodec](#), [gdcm::JPEGCodec](#), and [gdcm::JPEG2000Codec](#).

#### 10.148.4.5 CleanupUnusedBits()

```
bool gdcm::ImageCodec::CleanupUnusedBits (
    char * data,
    size_t datalen )
```

#### 10.148.4.6 Clone()

```
virtual ImageCodec\* gdcM::ImageCodec::Clone ( ) const [pure virtual]
```

Implemented in [gdcM::RLECodec](#), [gdcM::RAWCodec](#), [gdcM::PVRGCodec](#), [gdcM::PNMCodec](#), [gdcM::PGXCodec](#), [gdcM::KAKADUCodec](#), [gdcM::JPEGLSCodec](#), [gdcM::JPEGCodec](#), and [gdcM::JPEG2000Codec](#).

#### 10.148.4.7 Decode()

```
bool gdcM::ImageCodec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcM::Decoder](#).

Reimplemented in [gdcM::RLECodec](#), [gdcM::RAWCodec](#), [gdcM::PVRGCodec](#), [gdcM::KAKADUCodec](#), [gdcM::JPEGLSCodec](#), [gdcM::JPEGCodec](#), and [gdcM::JPEG2000Codec](#).

#### 10.148.4.8 DecodeByStreams()

```
bool gdcM::ImageCodec::DecodeByStreams (
    std::istream & is_,
    std::ostream & os ) [override], [protected], [virtual]
```

Reimplemented from [gdcM::Decoder](#).

Reimplemented in [gdcM::RLECodec](#), [gdcM::RAWCodec](#), [gdcM::JPEGCodec](#), [gdcM::JPEG8Codec](#), [gdcM::JPEG2000Codec](#), [gdcM::JPEG16Codec](#), and [gdcM::JPEG12Codec](#).

#### 10.148.4.9 DoByteSwap()

```
bool gdcM::ImageCodec::DoByteSwap (
    std::istream & is_,
    std::ostream & os ) [protected]
```

#### 10.148.4.10 DoInvertMonochrome()

```
bool gdcm::ImageCodec::DoInvertMonochrome (
    std::istream & is_,
    std::ostream & os ) [protected]
```

#### 10.148.4.11 DoOverlayCleanup()

```
bool gdcm::ImageCodec::DoOverlayCleanup (
    std::istream & is_,
    std::ostream & os ) [protected]
```

#### 10.148.4.12 DoPaddedCompositePixelCode()

```
bool gdcm::ImageCodec::DoPaddedCompositePixelCode (
    std::istream & is_,
    std::ostream & os ) [protected]
```

#### 10.148.4.13 DoPlanarConfiguration()

```
bool gdcm::ImageCodec::DoPlanarConfiguration (
    std::istream & is_,
    std::ostream & os ) [protected]
```

#### 10.148.4.14 DoSimpleCopy()

```
bool gdcm::ImageCodec::DoSimpleCopy (
    std::istream & is_,
    std::ostream & os ) [protected]
```

#### 10.148.4.15 DoYBR()

```
bool gdcm::ImageCodec::DoYBR (
    std::istream & is_,
    std::ostream & os ) [protected]
```

#### 10.148.4.16 DoYBRFull422()

```
bool gdcM::ImageCodec::DoYBRFull422 (
    std::istream & is_,
    std::ostream & os ) [protected]
```

#### 10.148.4.17 GetDimensions()

```
const unsigned int* gdcM::ImageCodec::GetDimensions ( ) const [inline]
```

#### 10.148.4.18 GetHeaderInfo()

```
virtual bool gdcM::ImageCodec::GetHeaderInfo (
    std::istream & is_,
    TransferSyntax & ts ) [virtual]
```

Reimplemented in [gdcM::RLECodec](#), [gdcM::RAWCodec](#), [gdcM::PNMCodec](#), [gdcM::PGXCodec](#), [gdcM::JPEGLSCodec](#), [gdcM::JPEGCodec](#), [gdcM::JPEG8Codec](#), [gdcM::JPEG2000Codec](#), [gdcM::JPEG16Codec](#), and [gdcM::JPEG12Codec](#).

#### 10.148.4.19 GetLossyFlag()

```
bool gdcM::ImageCodec::GetLossyFlag ( ) const
```

#### 10.148.4.20 GetLUT()

```
const LookupTable& gdcM::ImageCodec::GetLUT ( ) const [inline]
```

#### 10.148.4.21 GetNeedByteSwap()

```
bool gdcM::ImageCodec::GetNeedByteSwap ( ) const [inline]
```

#### 10.148.4.22 GetNumberOfDimensions()

```
unsigned int gdcm::ImageCodec::GetNumberOfDimensions ( ) const
```

#### 10.148.4.23 GetPhotometricInterpretation()

```
const PhotometricInterpretation& gdcm::ImageCodec::GetPhotometricInterpretation ( ) const
```

#### 10.148.4.24 GetPixelFormat() [1/2]

```
PixelFormat& gdcm::ImageCodec::GetPixelFormat ( ) [inline]
```

##### Examples

[GetJPEGSamplePrecision.cxx](#).

#### 10.148.4.25 GetPixelFormat() [2/2]

```
const PixelFormat& gdcm::ImageCodec::GetPixelFormat ( ) const [inline]
```

#### 10.148.4.26 GetPlanarConfiguration()

```
unsigned int gdcm::ImageCodec::GetPlanarConfiguration ( ) const [inline]
```

#### 10.148.4.27 IsFrameEncoder()

```
virtual bool gdcm::ImageCodec::IsFrameEncoder ( ) [protected], [virtual]
```

Reimplemented in [gdcm::RLECodec](#), [gdcm::JPEGLSCodec](#), [gdcm::JPEGCodec](#), and [gdcm::JPEG2000Codec](#).

#### 10.148.4.28 IsLossy()

```
bool gdcm::ImageCodec::IsLossy ( ) const
```

#### 10.148.4.29 IsRowEncoder()

```
virtual bool gdcm::ImageCodec::IsRowEncoder ( ) [protected], [virtual]
```

Reimplemented in [gdcm::RLECodec](#), [gdcm::JPEGLSCodec](#), [gdcm::JPEGCodec](#), and [gdcm::JPEG2000Codec](#).

#### 10.148.4.30 IsValid()

```
virtual bool gdcm::ImageCodec::IsValid (
    PhotometricInterpretation const & pi ) [protected], [virtual]
```

Reimplemented in [gdcm::JPEGCodec](#).

#### 10.148.4.31 SetDimensions() [1/2]

```
void gdcm::ImageCodec::SetDimensions (
    const std::vector< unsigned int > & d )
```

#### 10.148.4.32 SetDimensions() [2/2]

```
void gdcm::ImageCodec::SetDimensions (
    const unsigned int d[3] )
```

#### Examples

[ExtractIconFromFile.cxx](#).

#### 10.148.4.33 SetLossyFlag()

```
void gdcm::ImageCodec::SetLossyFlag (
    bool l )
```

#### 10.148.4.34 SetLUT()

```
void gdcm::ImageCodec::SetLUT (
    LookupTable const & lut ) [inline]
```

#### Examples

[ExtractIconFromFile.cxx](#).

#### 10.148.4.35 SetNeedByteSwap()

```
void gdcm::ImageCodec::SetNeedByteSwap (
    bool b ) [inline]
```

#### 10.148.4.36 SetNeedOverlayCleanup()

```
void gdcm::ImageCodec::SetNeedOverlayCleanup (
    bool b ) [inline]
```

#### 10.148.4.37 SetNumberOfDimensions()

```
void gdcm::ImageCodec::SetNumberOfDimensions (
    unsigned int dim )
```

#### 10.148.4.38 SetPhotometricInterpretation()

```
void gdcM::ImageCodec::SetPhotometricInterpretation (
    PhotometricInterpretation const & pi )
```

##### Examples

[ExtractIconFromFile.cxx](#).

#### 10.148.4.39 SetPixelFormat()

```
virtual void gdcM::ImageCodec::SetPixelFormat (
    PixelFormat const & pf ) [inline], [virtual]
```

Reimplemented in [gdcM::JPEGCodec](#).

##### Examples

[ExtractIconFromFile.cxx](#).

#### 10.148.4.40 SetPlanarConfiguration()

```
void gdcM::ImageCodec::SetPlanarConfiguration (
    unsigned int pc ) [inline]
```

#### 10.148.4.41 StartEncode()

```
virtual bool gdcM::ImageCodec::StartEncode (
    std::ostream & os ) [protected], [virtual]
```

Reimplemented in [gdcM::RLECodec](#), [gdcM::JPEGLSCodec](#), [gdcM::JPEGCodec](#), and [gdcM::JPEG2000Codec](#).

#### 10.148.4.42 StopEncode()

```
virtual bool gdcM::ImageCodec::StopEncode (
    std::ostream & os ) [protected], [virtual]
```

Reimplemented in [gdcM::RLECodec](#), [gdcM::JPEGLSCodec](#), [gdcM::JPEGCodec](#), and [gdcM::JPEG2000Codec](#).



## 10.148.5 Friends And Related Function Documentation

### 10.148.5.1 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

### 10.148.5.2 ImageChangePhotometricInterpretation

```
friend class ImageChangePhotometricInterpretation [friend]
```

## 10.148.6 Member Data Documentation

### 10.148.6.1 Dimensions

```
unsigned int gdcm::ImageCodec::Dimensions[3] [protected]
```

### 10.148.6.2 LossyFlag

```
bool gdcm::ImageCodec::LossyFlag [protected]
```

### 10.148.6.3 LUT

```
LUTPtr gdcm::ImageCodec::LUT [protected]
```

#### 10.148.6.4 NeedByteSwap

```
bool gdcm::ImageCodec::NeedByteSwap [protected]
```

#### 10.148.6.5 NeedOverlayCleanup

```
bool gdcm::ImageCodec::NeedOverlayCleanup [protected]
```

#### 10.148.6.6 NumberOfDimensions

```
unsigned int gdcm::ImageCodec::NumberOfDimensions [protected]
```

#### 10.148.6.7 PF

```
PixelFormat gdcm::ImageCodec::PF [protected]
```

#### 10.148.6.8 PI

```
PhotometricInterpretation gdcm::ImageCodec::PI [protected]
```

#### 10.148.6.9 PlanarConfiguration

```
unsigned int gdcm::ImageCodec::PlanarConfiguration [protected]
```

#### 10.148.6.10 RequestPaddedCompositePixelCode

```
bool gdcm::ImageCodec::RequestPaddedCompositePixelCode [protected]
```

### 10.148.6.11 RequestPlanarConfiguration

```
bool gdcm::ImageCodec::RequestPlanarConfiguration [protected]
```

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

- [gdcmImageCodec.h](#)

## 10.149 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)

### 10.149.1 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

### 10.149.2 Constructor & Destructor Documentation

#### 10.149.2.1 ImageConverter()

```
gdcm::ImageConverter::ImageConverter ( )
```

#### 10.149.2.2 ~ImageConverter()

```
gdcm::ImageConverter::~ImageConverter ( )
```

### 10.149.3 Member Function Documentation

#### 10.149.3.1 Convert()

```
void gdcm::ImageConverter::Convert ( )
```

#### 10.149.3.2 GetOutput()

```
const Image& gdcm::ImageConverter::GetOutput ( ) const
```

#### 10.149.3.3 SetInput()

```
void gdcm::ImageConverter::SetInput (
    Image const & input )
```

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

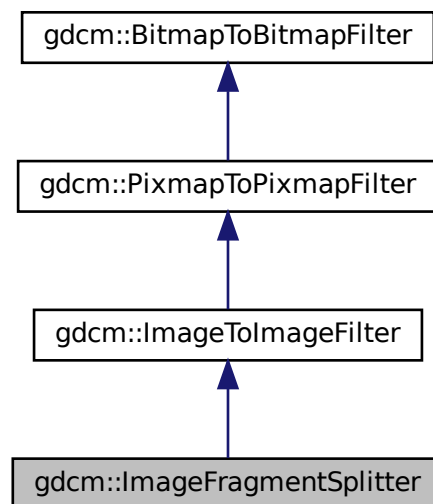
- [gdcmImageConverter.h](#)

## 10.150 gdcm::ImageFragmentSplitter Class Reference

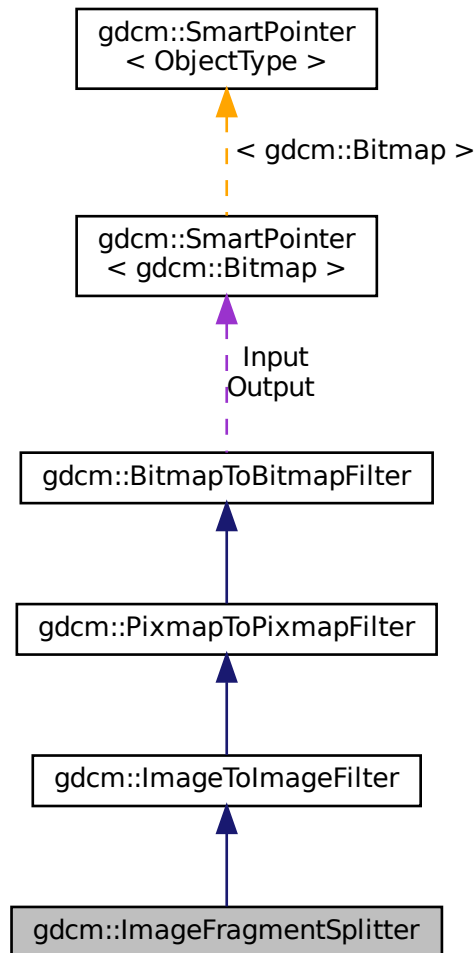
[ImageFragmentSplitter](#) class.

```
#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.*

## Additional Inherited Members

### 10.150.1 Detailed Description

[ImageFragmentSplitter](#) class.

For single frame image, DICOM standard allow splitting the frame into multiple fragments

### 10.150.2 Constructor & Destructor Documentation

#### 10.150.2.1 ImageFragmentSplitter()

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

#### 10.150.2.2 ~ImageFragmentSplitter()

```
gdcm::ImageFragmentSplitter::~~ImageFragmentSplitter ( ) [default]
```

### 10.150.3 Member Function Documentation

#### 10.150.3.1 GetFragmentSizeMax()

```
unsigned int gdcm::ImageFragmentSplitter::GetFragmentSizeMax ( ) const [inline]
```

#### 10.150.3.2 SetForce()

```
void gdcm::ImageFragmentSplitter::SetForce (
    bool f ) [inline]
```

When file already has all it's segment < FragmentSizeMax there is not need to run the filter. Unless the user explicitly say 'force' recomputation !

### 10.150.3.3 SetFragmentSizeMax()

```
void gdcmm::ImageFragmentSplitter::SetFragmentSizeMax (
    unsigned int fragsize )
```

FragmentSizeMax needs to be an even number.

### 10.150.3.4 Split()

```
bool gdcmm::ImageFragmentSplitter::Split ( )
```

Split.

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

- [gdcmmImageFragmentSplitter.h](#)

## 10.151 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 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 [SetSpacingValue](#) ([DataSet](#) &ds, const std::vector< double > &spacing)

## Static Protected Member Functions

- static [Tag](#) [GetSpacingTagFromMediaStorage](#) ([MediaStorage](#) const &ms)
- static [Tag](#) [GetZSpacingTagFromMediaStorage](#) ([MediaStorage](#) const &ms)

### 10.151.1 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.

### 10.151.2 Member Function Documentation

#### 10.151.2.1 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](#).

### 10.151.2.2 ComputeSpacingFromImagePositionPatient()

```
static bool gdcm::ImageHelper::ComputeSpacingFromImagePositionPatient (
    const std::vector< double > & imageposition,
    std::vector< double > & spacing ) [static]
```

DO NOT USE.

### 10.151.2.3 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

[Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

### 10.151.2.4 GetDirectionCosinesFromDataSet()

```
static bool gdcm::ImageHelper::GetDirectionCosinesFromDataSet (
    DataSet const & ds,
    std::vector< double > & dircos ) [static]
```

### 10.151.2.5 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

### 10.151.2.6 GetForcePixelSpacing()

```
static bool gdcm::ImageHelper::GetForcePixelSpacing ( ) [static]
```

### 10.151.2.7 GetForceRescaleInterceptSlope()

```
static bool gdcm::ImageHelper::GetForceRescaleInterceptSlope ( ) [static]
```

### 10.151.2.8 GetLUT()

```
static SmartPointer<LookupTable> gdcm::ImageHelper::GetLUT (
    File const & f ) [static]
```

returns the lookup table of an image file

### 10.151.2.9 GetOriginValue()

```
static std::vector<double> gdcm::ImageHelper::GetOriginValue (
    File const & f ) [static]
```

Set/Get Origin (IPP) from/to a file.

### 10.151.2.10 GetPhotometricInterpretationValue()

```
static PhotometricInterpretation gdcm::ImageHelper::GetPhotometricInterpretationValue (
    File const & f ) [static]
```

### 10.151.2.11 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)

### 10.151.2.12 GetPlanarConfigurationValue()

```
static unsigned int gdcm::ImageHelper::GetPlanarConfigurationValue (
    const File & f ) [static]
```

#### 10.151.2.13 GetPMSRescaleInterceptSlope()

```
static bool gdcm::ImageHelper::GetPMSRescaleInterceptSlope ( ) [static]
```

#### 10.151.2.14 GetPointerFromElement()

```
static const ByteValue* gdcm::ImageHelper::GetPointerFromElement (
    Tag const & tag,
    File const & f ) [static]
```

#### 10.151.2.15 GetRealWorldValueMappingContent()

```
static bool gdcm::ImageHelper::GetRealWorldValueMappingContent (
    File const & f,
    RealWorldValueMappingContent & rwvmc ) [static]
```

#### 10.151.2.16 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

#### 10.151.2.17 GetSpacingTagFromMediaStorage()

```
static Tag gdcm::ImageHelper::GetSpacingTagFromMediaStorage (
    MediaStorage const & ms ) [static], [protected]
```

#### 10.151.2.18 GetSpacingValue()

```
static std::vector<double> gdcm::ImageHelper::GetSpacingValue (
    File const & f ) [static]
```

Set/Get [Spacing](#) from/to a [File](#).

#### 10.151.2.19 GetZSpacingTagFromMediaStorage()

```
static Tag gdcm::ImageHelper::GetZSpacingTagFromMediaStorage (
    MediaStorage const & ms ) [static], [protected]
```

#### 10.151.2.20 SetDimensionsValue()

```
static void gdcm::ImageHelper::SetDimensionsValue (
    File & f,
    const Pixmap & img ) [static]
```

#### 10.151.2.21 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.

#### 10.151.2.22 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](#)

#### 10.151.2.23 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.

#### 10.151.2.24 SetOriginValue()

```
static void gdcm::ImageHelper::SetOriginValue (
    DataSet & ds,
    const Image & img ) [static]
```

#### 10.151.2.25 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.

#### 10.151.2.26 SetRescaleInterceptSlopeValue()

```
static void gdcm::ImageHelper::SetRescaleInterceptSlopeValue (
    File & f,
    const Image & img ) [static]
```

#### 10.151.2.27 SetSpacingValue()

```
static void gdcm::ImageHelper::SetSpacingValue (
    DataSet & ds,
    const std::vector< double > & spacing ) [static]
```

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

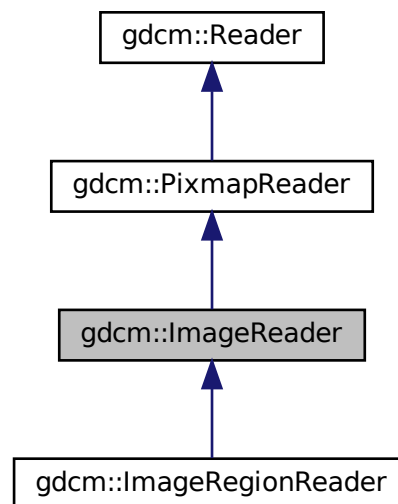
- [gdcmImageHelper.h](#)

## 10.152 gdcm::ImageReader Class Reference

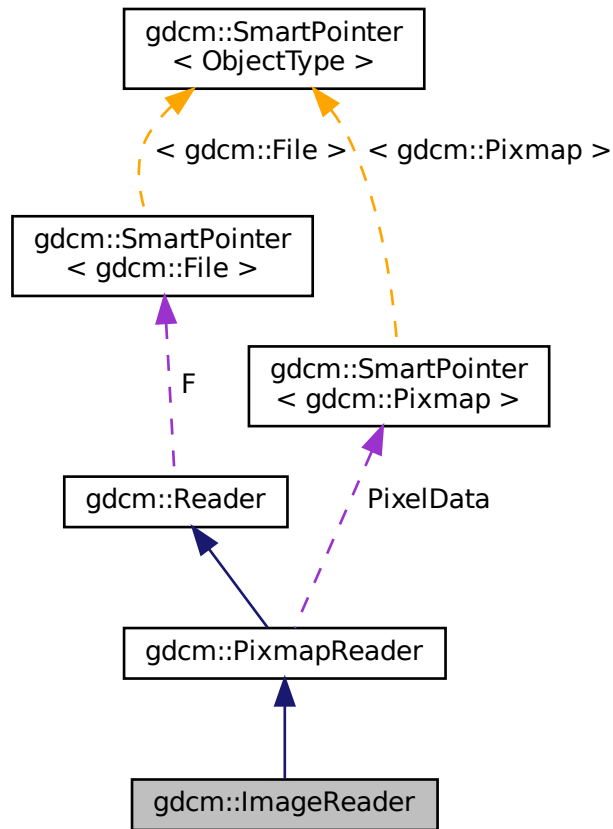
[ImageReader](#).

```
#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

## Protected Member Functions

- `bool ReadACRNEMAIImage ()` override
- `bool ReadImage (MediaStorage const &ms)` override



## Additional Inherited Members

### 10.152.1 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

[CheckBigEndianBug.cxx](#), [CompressImage.cxx](#), [ConvertToQImage.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetJPEGSamplePrecision.cxx](#), [HelloVizWorld.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [PrintLUT.cxx](#), [ReadMultiTimesException.cxx](#), and [threadgdcm.cxx](#).

### 10.152.2 Constructor & Destructor Documentation

#### 10.152.2.1 ImageReader()

```
gdcm::ImageReader::ImageReader ( )
```

#### 10.152.2.2 ~ImageReader()

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

### 10.152.3 Member Function Documentation

#### 10.152.3.1 GetImage() [1/2]

```
Image& gdcm::ImageReader::GetImage ( )
```

### 10.152.3.2 GetImage() [2/2]

```
const Image& gdcm::ImageReader::GetImage ( ) const
```

Return the read image.

#### Examples

[CompressImage.cxx](#), [ConvertToQImage.cxx](#), [ExtractIconFromFile.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetJPEGSamplePrecision.cxx](#), [HelloVizWorld.cxx](#), [MergeTwoFiles.cxx](#), [PatchFile.cxx](#), [PrintLUT.cxx](#), [ReadMultiTimesException.cxx](#), [TemplateEmptyImage.cxx](#), and [threadgdcm.cxx](#).

### 10.152.3.3 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

[CheckBigEndianBug.cxx](#), [CompressImage.cxx](#), [ConvertToQImage.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetJPEGSamplePrecision.cxx](#), [HelloVizWorld.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [PrintLUT.cxx](#), [ReadMultiTimesException.cxx](#), and [threadgdcm.cxx](#).

### 10.152.3.4 ReadACRNEMAIImage()

```
bool gdcm::ImageReader::ReadACRNEMAIImage ( ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::PixmapReader](#).

### 10.152.3.5 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:

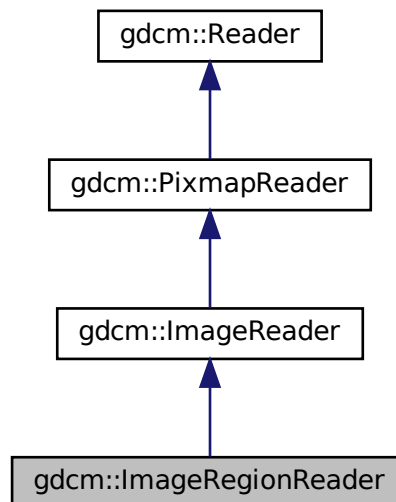
- [gdcmImageReader.h](#)

## 10.153 gdcm::ImageRegionReader Class Reference

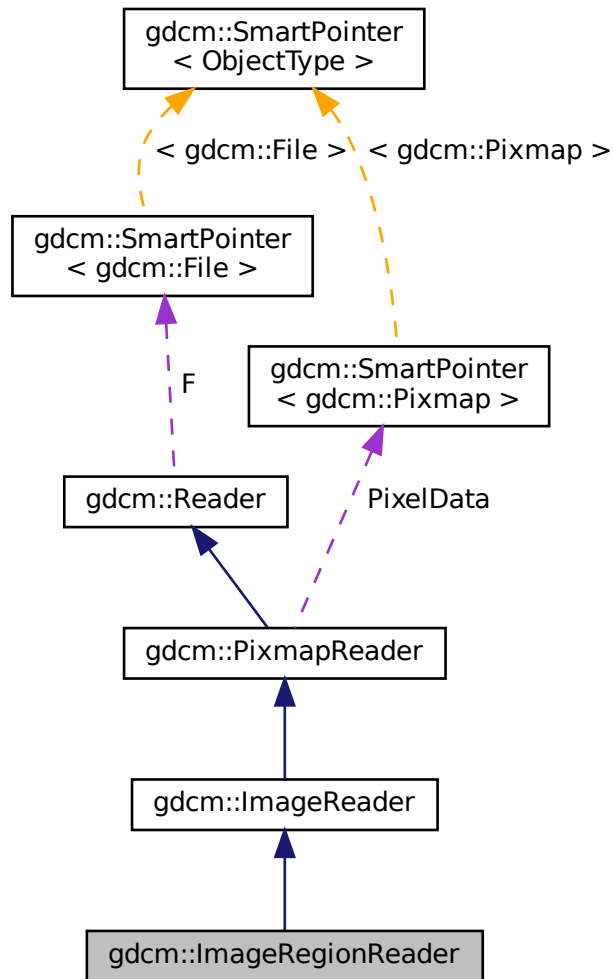
[ImageRegionReader](#).

```
#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.*

## Protected Member Functions

- bool [Read](#) () override

*To prevent user from calling super class [Read\(\)](#) function.*

## Additional Inherited Members

### 10.153.1 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

[TemplateEmptyImage.cxx](#).

### 10.153.2 Constructor & Destructor Documentation

#### 10.153.2.1 ImageRegionReader()

```
gdcm::ImageRegionReader::ImageRegionReader ( )
```

#### 10.153.2.2 ~ImageRegionReader()

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

### 10.153.3 Member Function Documentation

#### 10.153.3.1 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

#### 10.153.3.2 GetRegion()

```
Region const& gdcm::ImageRegionReader::GetRegion ( ) const
```

#### 10.153.3.3 Read()

```
bool gdcm::ImageRegionReader::Read ( ) [override], [protected], [virtual]
```

To prevent user from calling super class [Read\(\)](#) function.

Reimplemented from [gdcm::ImageReader](#).

#### 10.153.3.4 ReadInformation()

```
bool gdcm::ImageRegionReader::ReadInformation ( )
```

Read meta information (not Pixel Data) from the DICOM file.

##### Returns

false upon error

##### Examples

[TemplateEmptyImage.cxx](#).

### 10.153.3.5 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

### 10.153.3.6 SetRegion()

```
void gdcm::ImageRegionReader::SetRegion (
    Region const & region )
```

Set/Get [Region](#) to be read.

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

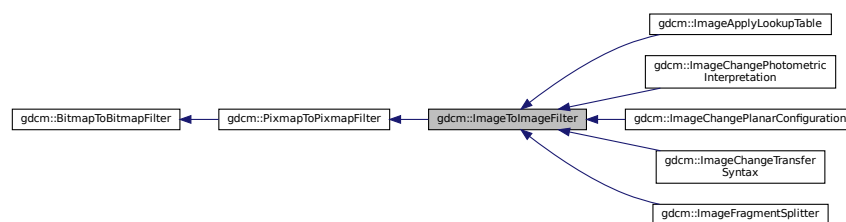
- [gdcmImageRegionReader.h](#)

## 10.154 gdcm::ImageToImageFilter Class Reference

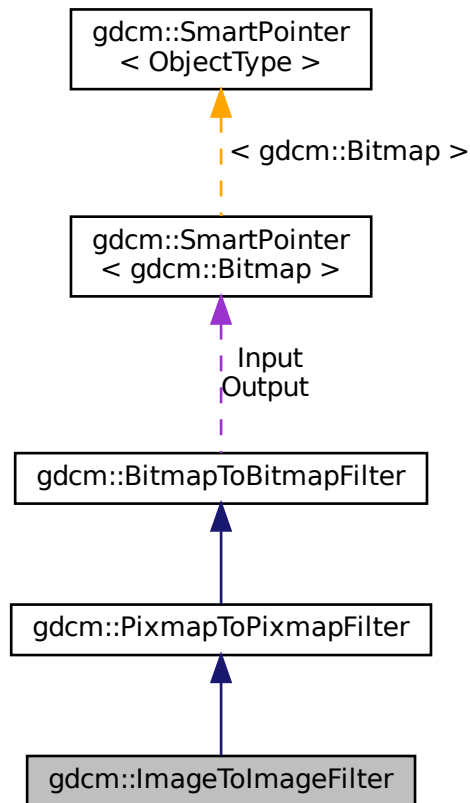
[ImageToImageFilter](#) class.

```
#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.*

## Additional Inherited Members

### 10.154.1 Detailed Description

[ImageToImageFilter](#) class.

Super class for all filter taking an image and producing an output image



## 10.154.2 Constructor & Destructor Documentation

### 10.154.2.1 ImageToImageFilter()

```
gdcm::ImageToImageFilter::ImageToImageFilter ( )
```

### 10.154.2.2 ~ImageToImageFilter()

```
gdcm::ImageToImageFilter::~~ImageToImageFilter ( ) [default]
```

## 10.154.3 Member Function Documentation

### 10.154.3.1 GetInput()

```
Image& gdcm::ImageToImageFilter::GetInput ( )
```

### 10.154.3.2 GetOutput()

```
const Image& gdcm::ImageToImageFilter::GetOutput ( ) const
```

Get Output image.

#### Examples

[CompressImage.cxx](#).

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

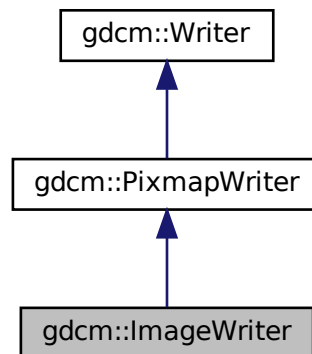
- [gdcmImageToImageFilter.h](#)

## 10.155 gdcm::ImageWriter Class Reference

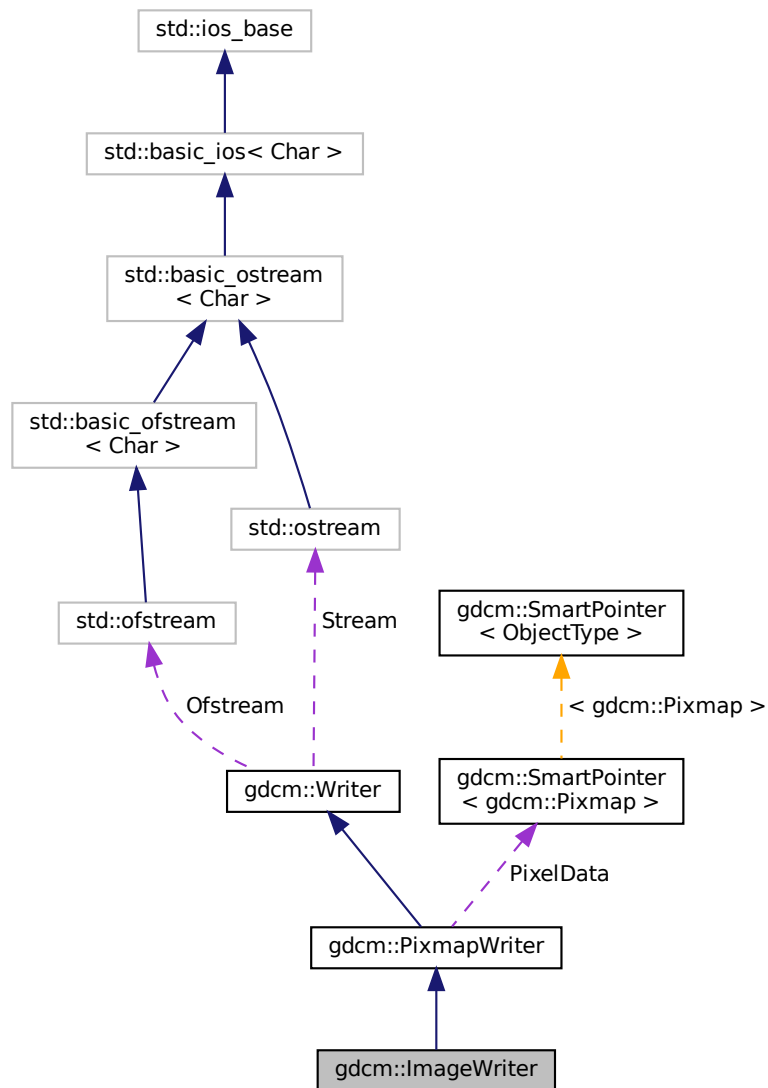
[ImageWriter](#).

```
#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.*

## Additional Inherited Members

### 10.155.1 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

[CompressImage.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [MergeTwoFiles.cxx](#), [TemplateEmptyImage.cxx](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

### 10.155.2 Constructor & Destructor Documentation

#### 10.155.2.1 ImageWriter()

```
gdcm::ImageWriter::ImageWriter ( )
```

#### 10.155.2.2 ~ImageWriter()

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

### 10.155.3 Member Function Documentation

### 10.155.3.1 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`.

### 10.155.3.2 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`.

### 10.155.3.3 GetImage() [2/2]

`Image&` gdcm::ImageWriter::GetImage ( ) [inline], [override], [virtual]

Reimplemented from `gdcm::PixmapWriter`.

### 10.155.3.4 Write()

`bool` gdcm::ImageWriter::Write ( ) [override], [virtual]

Write.

Reimplemented from `gdcm::Writer`.

#### Examples

`CompressImage.cxx`, `CreateARGBImage.cxx`, `CreateCMYKImage.cxx`, `GenFakelImage.cxx`, `GetSubSequenceData.cxx`, `HelloVizWorld.cxx`, `MergeTwoFiles.cxx`, `TemplateEmptyImage.cxx`, `csa2img.cxx`, and `iU22tomultisc.cxx`.

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

- `gdcmImageWriter.h`

## 10.156 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

### 10.156.1 Detailed Description

[ImplementationClassUIDSub](#).

PS 3.7 [Table](#) D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

### 10.156.2 Constructor & Destructor Documentation

#### 10.156.2.1 ImplementationClassUIDSub()

```
gdcm::network::ImplementationClassUIDSub::ImplementationClassUIDSub ( )
```

### 10.156.3 Member Function Documentation

#### 10.156.3.1 Print()

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

**10.156.3.2 Read()**

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

**10.156.3.3 Size()**

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

**10.156.3.4 Write()**

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

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

- [gdcmImplementationClassUIDSub.h](#)

**10.157 gdcm::network::ImplementationUIDSub Class Reference**

[ImplementationUIDSub.](#)

```
#include <gdcmImplementationUIDSub.h>
```

**Public Member Functions**

- [ImplementationUIDSub](#) ()
- const std::ostream & [Write](#) (std::ostream &os) const

**10.157.1 Detailed Description**

[ImplementationUIDSub.](#)

[Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS \(A-ASSOCIATE-AC\)](#)

**10.157.2 Constructor & Destructor Documentation**

### 10.157.2.1 ImplementationUIDSub()

```
gdcm::network::ImplementationUIDSub::ImplementationUIDSub ( )
```

## 10.157.3 Member Function Documentation

### 10.157.3.1 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](#)

## 10.158 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

### 10.158.1 Detailed Description

[ImplementationVersionNameSub.](#)

[Table](#) D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

### 10.158.2 Constructor & Destructor Documentation



### 10.158.2.1 ImplementationVersionNameSub()

```
gdcm::network::ImplementationVersionNameSub::ImplementationVersionNameSub ( )
```

## 10.158.3 Member Function Documentation

### 10.158.3.1 Print()

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

### 10.158.3.2 Read()

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

### 10.158.3.3 Size()

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

### 10.158.3.4 Write()

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

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

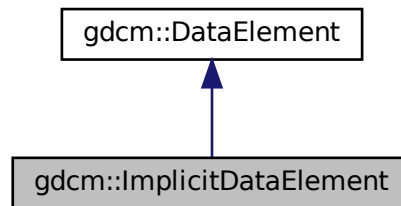
- [gdcmImplementationVersionNameSub.h](#)

## 10.159 gdcm::ImplicitDataElement Class Reference

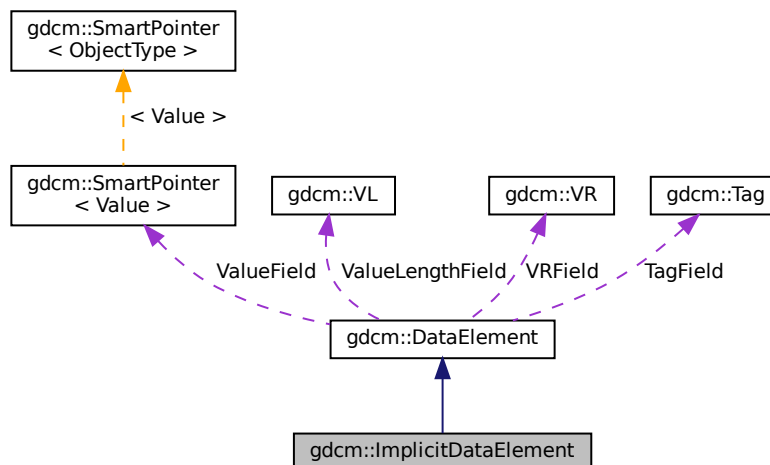
Class to represent an *Implicit VR Data Element*.

```
#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

## Additional Inherited Members

### 10.159.1 Detailed Description

Class to represent an *Implicit VR Data Element*.

Note

bla

### 10.159.2 Member Function Documentation

#### 10.159.2.1 GetLength()

```
VL gdcm::ImplicitDataElement::GetLength ( ) const
```

#### 10.159.2.2 Read()

```
template<typename TSwap >
std::istream& gdcm::ImplicitDataElement::Read (
    std::istream & is )
```

#### 10.159.2.3 ReadPreValue()

```
template<typename TSwap >
std::istream& gdcm::ImplicitDataElement::ReadPreValue (
    std::istream & is )
```

#### 10.159.2.4 ReadValue()

```
template<typename TSwap >
std::istream& gdcM::ImplicitDataElement::ReadValue (
    std::istream & is,
    bool readvalues = true )
```

#### 10.159.2.5 ReadValueWithLength()

```
template<typename TSwap >
std::istream& gdcM::ImplicitDataElement::ReadValueWithLength (
    std::istream & is,
    VL & length,
    bool readvalues = true )
```

#### 10.159.2.6 ReadWithLength()

```
template<typename TSwap >
std::istream& gdcM::ImplicitDataElement::ReadWithLength (
    std::istream & is,
    VL & length,
    bool readvalues = true )
```

#### 10.159.2.7 Write()

```
template<typename TSwap >
const std::ostream& gdcM::ImplicitDataElement::Write (
    std::ostream & os ) const
```

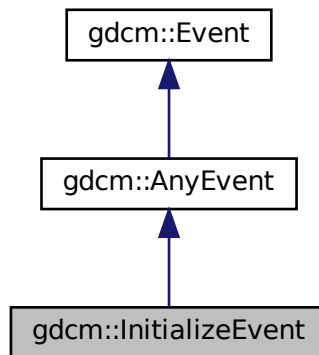
The documentation for this class was generated from the following file:

- [gdcMImplicitDataElement.h](#)

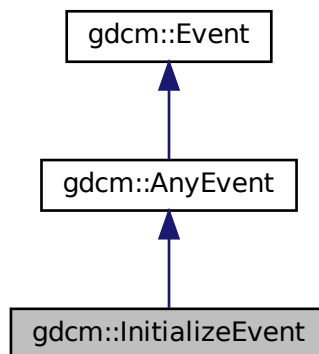
## 10.160 gdcm::InitializeEvent Class Reference

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

Inheritance diagram for gdcm::InitializeEvent:



Collaboration diagram for gdcm::InitializeEvent:



### Additional Inherited Members

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

- [gdcmEvent.h](#)

## 10.161 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)

#### 10.161.1 Detailed Description

Class for representing a [IOD](#).

Note

bla

See also

[Dict](#)

#### 10.161.2 Member Typedef Documentation

### 10.161.2.1 MapIODEntry

```
typedef std::vector<IODEntry> gdcm::IOD::MapIODEntry
```

### 10.161.2.2 SizeType

```
typedef MapIODEntry::size_type gdcm::IOD::SizeType
```

## 10.161.3 Constructor & Destructor Documentation

### 10.161.3.1 IOD()

```
gdcm::IOD::IOD ( ) [default]
```

## 10.161.4 Member Function Documentation

### 10.161.4.1 AddIODEntry()

```
void gdcm::IOD::AddIODEntry (
    const IODEntry & iode ) [inline]
```

### 10.161.4.2 Clear()

```
void gdcm::IOD::Clear ( ) [inline]
```

### 10.161.4.3 GetIODEntry()

```
const IODEntry& gdcm::IOD::GetIODEntry (
    SizeType idx ) const [inline]
```

#### 10.161.4.4 GetNumberOfIODs()

```
SizeType gdcml::IOD::GetNumberOfIODs ( ) const [inline]
```

#### 10.161.4.5 GetTypeFromTag()

```
Type gdcml::IOD::GetTypeFromTag (
    const Defs & defs,
    const Tag & tag ) const
```

### 10.161.5 Friends And Related Function Documentation

#### 10.161.5.1 operator<<

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

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

- [gdcmlIOD.h](#)

## 10.162 gdcml::IODEntry Class Reference

Class for representing a [IODEntry](#).

```
#include <gdcmlIODEntry.h>
```

### Public Member Functions

- [IODEntry](#) (const char \*name="", const char \*ref="", const char \*usag="")
- 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 \*usag)



## Friends

- `std::ostream & operator<< (std::ostream &_os, const IODEntry &_val)`

### 10.162.1 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](#)

### 10.162.2 Constructor & Destructor Documentation

#### 10.162.2.1 IODEntry()

```
gdcm::IODEntry::IODEntry (
    const char * name = "",
    const char * ref = "",
    const char * usag = "" ) [inline]
```

### 10.162.3 Member Function Documentation

**10.162.3.1 GetIE()**

```
const char* gdcm::IODEntry::GetIE ( ) const [inline]
```

**10.162.3.2 GetName()**

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

**10.162.3.3 GetRef()**

```
const char* gdcm::IODEntry::GetRef ( ) const [inline]
```

**10.162.3.4 GetUsage()**

```
const char* gdcm::IODEntry::GetUsage ( ) const [inline]
```

**10.162.3.5 GetUsageType()**

```
Usage::UsageType gdcm::IODEntry::GetUsageType ( ) const
```

**10.162.3.6 SetIE()**

```
void gdcm::IODEntry::SetIE (
    const char * ie ) [inline]
```

**10.162.3.7 SetName()**

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

### 10.162.3.8 SetRef()

```
void gdcm::IODEntry::SetRef (
    const char * ref ) [inline]
```

### 10.162.3.9 SetUsage()

```
void gdcm::IODEntry::SetUsage (
    const char * usag ) [inline]
```

## 10.162.4 Friends And Related Function Documentation

### 10.162.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const IODEntry & _val ) [friend]
```

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

- [gdcmIODEntry.h](#)

## 10.163 gdcm::IODs Class Reference

Class for representing a [IODs](#).

```
#include <gdcmIODs.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)

### 10.163.1 Detailed Description

Class for representing a [IODs](#).

#### Note

bla

#### See also

[IOD](#)

### 10.163.2 Member Typedef Documentation

#### 10.163.2.1 IODMapType

```
typedef std::map<IODName, IOD> gdcm::IODs::IODMapType
```

#### 10.163.2.2 IODMapTypeConstIterator

```
typedef IODMapType::const\_iterator gdcm::IODs::IODMapTypeConstIterator
```

#### Examples

[TraverseModules.cxx](#).

### 10.163.2.3 IODName

```
typedef std::string gdcm::IODs::IODName
```

#### Examples

[TraverseModules.cxx](#).

## 10.163.3 Constructor & Destructor Documentation

### 10.163.3.1 IODs()

```
gdcm::IODs::IODs ( ) [default]
```

## 10.163.4 Member Function Documentation

### 10.163.4.1 AddIOD()

```
void gdcm::IODs::AddIOD (
    const char * name,
    const IOD & module ) [inline]
```

### 10.163.4.2 Begin()

```
IODMapTypeConstIterator gdcm::IODs::Begin ( ) const [inline]
```

### 10.163.4.3 Clear()

```
void gdcm::IODs::Clear ( ) [inline]
```

#### 10.163.4.4 End()

```
IODMapTypeConstIterator gdcM::IODs::End ( ) const [inline]
```

#### 10.163.4.5 GetIOD()

```
const IOD& gdcM::IODs::GetIOD (
    const char * name ) const [inline]
```

### 10.163.5 Friends And Related Function Documentation

#### 10.163.5.1 operator<<

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

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

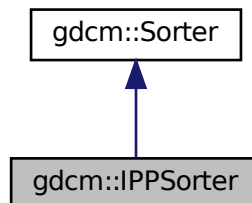
- [gdcMIODs.h](#)

## 10.164 gdcM::IPPSorter Class Reference

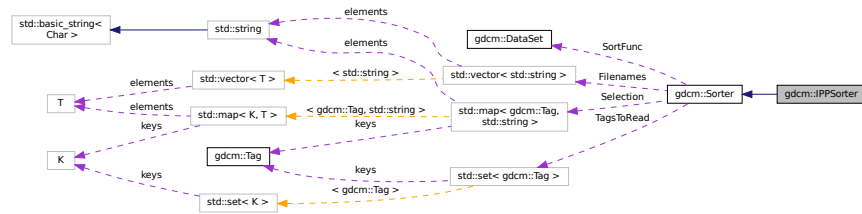
[IPPSorter.](#)

```
#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

## Protected Attributes

- bool [ComputeZSpacing](#)
- double [DirCosTolerance](#)
- bool [DropDuplicatePositions](#)
- double [ZSpacing](#)
- double [ZTolerance](#)

## Additional Inherited Members

### 10.164.1 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](#).

## 10.164.2 Constructor & Destructor Documentation

### 10.164.2.1 IPPSorter()

```
gdcm::IPPSorter::IPPSorter ( )
```

## 10.164.3 Member Function Documentation

### 10.164.3.1 GetDirectionCosinesTolerance()

```
double gdcm::IPPSorter::GetDirectionCosinesTolerance ( ) const [inline]
```



### 10.164.3.2 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](#).

### 10.164.3.3 GetZSpacingTolerance()

```
double gdcm::IPPSorter::GetZSpacingTolerance ( ) const [inline]
```

### 10.164.3.4 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](#).

### 10.164.3.5 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 define 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 imply a [DirectionCosines](#) tolerance of exactly 0.0 (default).

### 10.164.3.6 SetDropDuplicatePositions()

```
void gdcmm::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.

### 10.164.3.7 SetZSpacingTolerance()

```
void gdcmm::IPPSorter::SetZSpacingTolerance (
    double tol ) [inline]
```

1. Another reason for failure is that that Z-Spacing is only slightly changing (eg 1e-3) along the serie, a human can determine that this is ok and change the tolerance from its default value: 1e-6

#### Examples

[Compute3DSpacing.cxx](#), [gdcmmorthoplanes.cxx](#), and [reslicesphere.cxx](#).

### 10.164.3.8 Sort()

```
bool gdcmm::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 [gdcmm::Sorter](#).

#### Examples

[Compute3DSpacing.cxx](#), [VolumeSorter.cxx](#), [gdcmmorthoplanes.cxx](#), and [reslicesphere.cxx](#).

## 10.164.4 Member Data Documentation

### 10.164.4.1 ComputeZSpacing

```
bool gdcmm::IPPSorter::ComputeZSpacing [protected]
```

#### 10.164.4.2 DirCosTolerance

```
double gdcm::IPPSorter::DirCosTolerance [protected]
```

#### 10.164.4.3 DropDuplicatePositions

```
bool gdcm::IPPSorter::DropDuplicatePositions [protected]
```

#### 10.164.4.4 ZSpacing

```
double gdcm::IPPSorter::ZSpacing [protected]
```

#### 10.164.4.5 ZTolerance

```
double gdcm::IPPSorter::ZTolerance [protected]
```

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

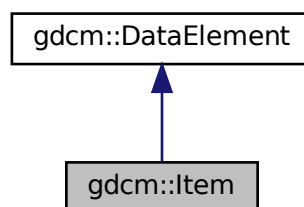
- [gdcmIPPSorter.h](#)

### 10.165 gdcm::Item Class Reference

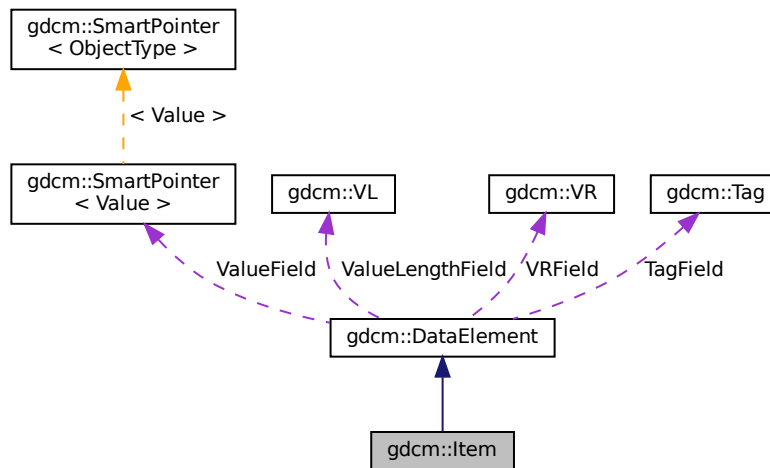
Class to represent an [Item](#).

```
#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

## Friends

- std::ostream & `operator<<` (std::ostream &os, const `Item` &val)

## Additional Inherited Members

### 10.165.1 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 Standart 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](#), [DumpToshibaDTI.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [LargeVRDSExplicit.cxx](#), [gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

### 10.165.2 Constructor & Destructor Documentation

#### 10.165.2.1 Item() [1/2]

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

#### 10.165.2.2 Item() [2/2]

```
gdcm::Item::Item (
    Item const & val ) [inline]
```

### 10.165.3 Member Function Documentation

### 10.165.3.1 Clear()

```
void gdcm::Item::Clear ( ) [inline]
```

Referenced by `gdcm::SequenceOfItems::Read()`.

### 10.165.3.2 FindDataElement()

```
bool gdcm::Item::FindDataElement (
    const Tag & t ) const [inline]
```

### 10.165.3.3 GetDataElement()

```
const DataElement& gdcm::Item::GetDataElement (
    const Tag & t ) const [inline]
```

### 10.165.3.4 GetLength()

```
template<typename TDE >
VL gdcm::Item::GetLength ( ) const
```

### 10.165.3.5 GetNestedDataSet() [1/2]

```
DataSet& gdcm::Item::GetNestedDataSet ( ) [inline]
```

### 10.165.3.6 GetNestedDataSet() [2/2]

```
const DataSet& gdcm::Item::GetNestedDataSet ( ) const [inline]
```

#### Examples

[ChangeSequenceUltrasound.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpToshibaDTI.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [LargeVRDSExplicit.cxx](#), [gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

Referenced by `gdcm::SequenceOfItems::Read()`.

### 10.165.3.7 InsertDataElement()

```
void gdcm::Item::InsertDataElement (
    const DataElement & de ) [inline]
```

### 10.165.3.8 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\(\)](#).

### 10.165.3.9 SetNestedDataSet()

```
void gdcm::Item::SetNestedDataSet (
    const DataSet & nested ) [inline]
```

### 10.165.3.10 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\(\)](#).

## 10.165.4 Friends And Related Function Documentation

### 10.165.4.1 operator<<

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

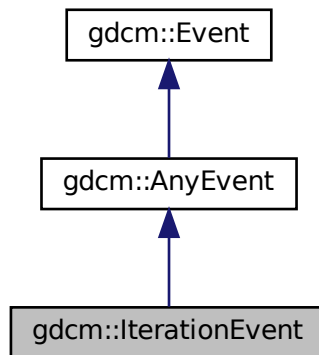
The documentation for this class was generated from the following file:

- [gdcmItem.h](#)

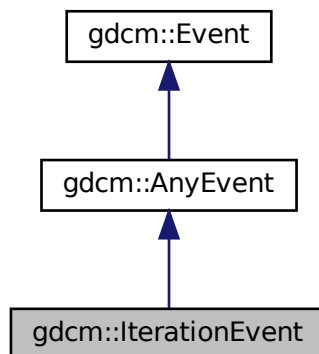
## 10.166 gdcm::IterationEvent Class Reference

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

Inheritance diagram for gdcm::IterationEvent:



Collaboration diagram for gdcm::IterationEvent:



### Additional Inherited Members

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

- [gdcmEvent.h](#)

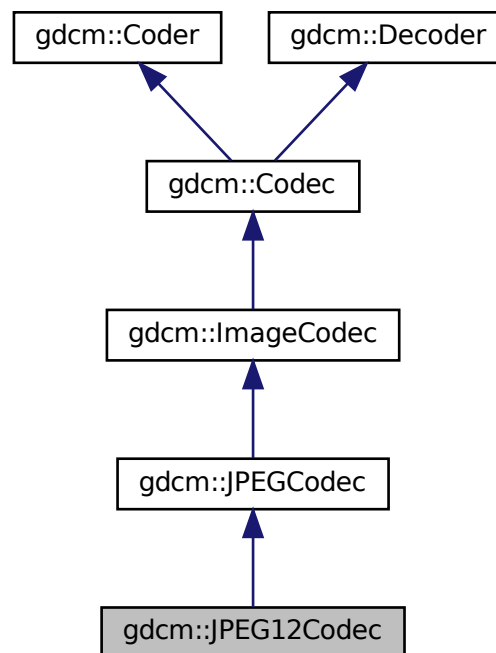


## 10.167 gdcm::JPEG12Codec Class Reference

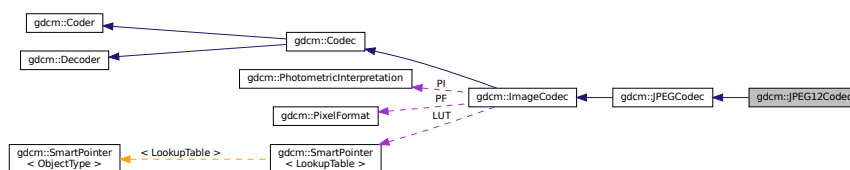
Class to do JPEG 12bits (lossy & lossless)

```
#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

## Protected Member Functions

- bool [EncodeBuffer](#) (std::ostream &os, const char \*data, size\_t datalen) override
- bool [IsStateSuspension](#) () const override

## Additional Inherited Members

### 10.167.1 Detailed Description

Class to do JPEG 12bits (lossy & lossless)

#### Note

internal class

### 10.167.2 Constructor & Destructor Documentation

#### 10.167.2.1 JPEG12Codec()

```
gdcm::JPEG12Codec::JPEG12Codec ( )
```

#### 10.167.2.2 ~JPEG12Codec()

```
gdcm::JPEG12Codec::~~JPEG12Codec ( ) [override]
```

### 10.167.3 Member Function Documentation

### 10.167.3.1 DecodeByStreams()

```
bool gdcm::JPEG12Codec::DecodeByStreams (
    std::istream & is,
    std::ostream & os ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.167.3.2 EncodeBuffer()

```
bool gdcm::JPEG12Codec::EncodeBuffer (
    std::ostream & os,
    const char * data,
    size_t datalen ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

### 10.167.3.3 GetHeaderInfo()

```
bool gdcm::JPEG12Codec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.167.3.4 InternalCode()

```
bool gdcm::JPEG12Codec::InternalCode (
    const char * input,
    unsigned long len,
    std::ostream & os ) [override], [virtual]
```

Reimplemented from [gdcm::Coder](#).

### 10.167.3.5 IsStateSuspension()

```
bool gdcm::JPEG12Codec::IsStateSuspension ( ) const [override], [protected], [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

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

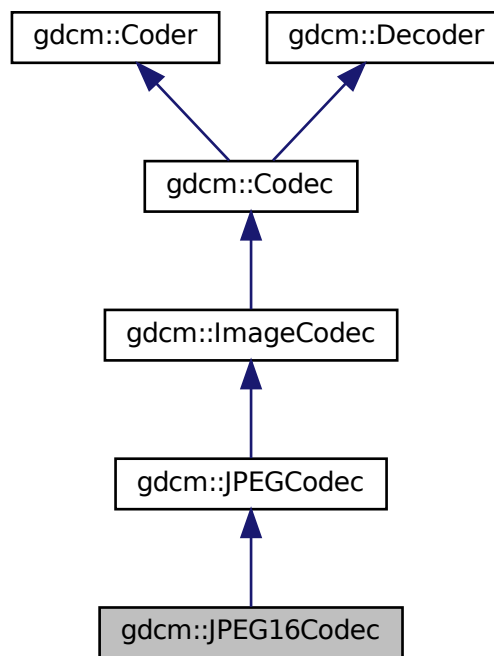
- [gdcmJPEG12Codec.h](#)

## 10.168 gdcm::JPEG16Codec Class Reference

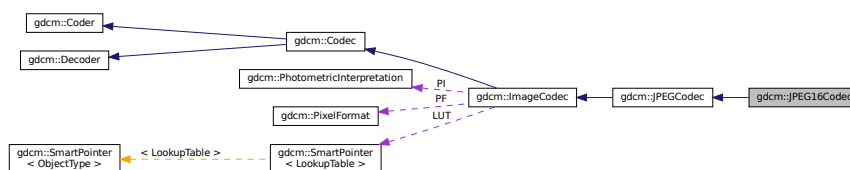
Class to do JPEG 16bits (lossless)

```
#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

## Protected Member Functions

- bool [EncodeBuffer](#) (std::ostream &os, const char \*data, size\_t datalen) override
- bool [IsStateSuspension](#) () const override

## Additional Inherited Members

### 10.168.1 Detailed Description

Class to do JPEG 16bits (lossless)

#### Note

internal class

### 10.168.2 Constructor & Destructor Documentation

#### 10.168.2.1 JPEG16Codec()

```
gdcm::JPEG16Codec::JPEG16Codec ( )
```

#### 10.168.2.2 ~JPEG16Codec()

```
gdcm::JPEG16Codec::~~JPEG16Codec ( ) [override]
```

### 10.168.3 Member Function Documentation

### 10.168.3.1 DecodeByStreams()

```
bool gdcm::JPEG16Codec::DecodeByStreams (
    std::istream & is,
    std::ostream & os ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.168.3.2 EncodeBuffer()

```
bool gdcm::JPEG16Codec::EncodeBuffer (
    std::ostream & os,
    const char * data,
    size_t datalen ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

### 10.168.3.3 GetHeaderInfo()

```
bool gdcm::JPEG16Codec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.168.3.4 InternalCode()

```
bool gdcm::JPEG16Codec::InternalCode (
    const char * input,
    unsigned long len,
    std::ostream & os ) [override], [virtual]
```

Reimplemented from [gdcm::Coder](#).

### 10.168.3.5 IsStateSuspension()

```
bool gdcm::JPEG16Codec::IsStateSuspension ( ) const [override], [protected], [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

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

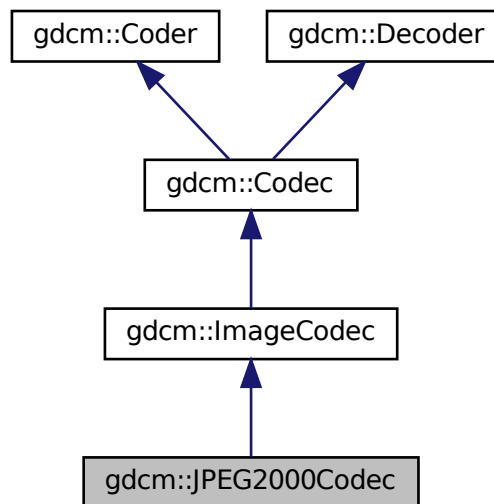
- [gdcmJPEG16Codec.h](#)

## 10.169 gdcm::JPEG2000Codec Class Reference

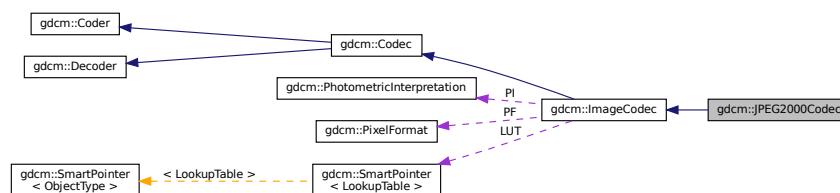
Class to do JPEG 2000.

```
#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 [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)

## 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

## Friends

- class [Bitmap](#)
- class [ImageRegionReader](#)

## Additional Inherited Members

### 10.169.1 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](#)



## 10.169.2 Constructor & Destructor Documentation

### 10.169.2.1 JPEG2000Codec()

```
gdcm::JPEG2000Codec::JPEG2000Codec ( )
```

### 10.169.2.2 ~JPEG2000Codec()

```
gdcm::JPEG2000Codec::~~JPEG2000Codec ( ) [override]
```

## 10.169.3 Member Function Documentation

### 10.169.3.1 AppendFrameEncode()

```
bool gdcm::JPEG2000Codec::AppendFrameEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.169.3.2 AppendRowEncode()

```
bool gdcm::JPEG2000Codec::AppendRowEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.169.3.3 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](#).

### 10.169.3.4 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](#).

### 10.169.3.5 Clone()

```
ImageCodec* gdcm::JPEG2000Codec::Clone ( ) const [override], [virtual]
```

Implements [gdcm::ImageCodec](#).

### 10.169.3.6 Code()

```
bool gdcm::JPEG2000Codec::Code (
    DataElement const & in_,
    DataElement & out_ ) [override], [virtual]
```

Code.

Reimplemented from [gdcm::Coder](#).

### 10.169.3.7 Decode()

```
bool gdcm::JPEG2000Codec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcm::ImageCodec](#).

### 10.169.3.8 DecodeByStreams()

```
bool gdcm::JPEG2000Codec::DecodeByStreams (
    std::istream & is,
    std::ostream & os ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.169.3.9 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]
```

### 10.169.3.10 GetHeaderInfo()

```
bool gdcm::JPEG2000Codec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

**10.169.3.11 GetQuality()**

```
double gdcM::JPEG2000Codec::GetQuality (
    unsigned int idx = 0 ) const
```

**10.169.3.12 GetRate()**

```
double gdcM::JPEG2000Codec::GetRate (
    unsigned int idx = 0 ) const
```

**10.169.3.13 IsFrameEncoder()**

```
bool gdcM::JPEG2000Codec::IsFrameEncoder ( ) [override], [protected], [virtual]
```

Reimplemented from [gdcM::ImageCodec](#).

**10.169.3.14 IsRowEncoder()**

```
bool gdcM::JPEG2000Codec::IsRowEncoder ( ) [override], [protected], [virtual]
```

Reimplemented from [gdcM::ImageCodec](#).

**10.169.3.15 SetNumberOfResolutions()**

```
void gdcM::JPEG2000Codec::SetNumberOfResolutions (
    unsigned int nres )
```

**10.169.3.16 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
-----------------	---

**10.169.3.17 SetQuality()**

```
void gdcm::JPEG2000Codec::SetQuality (
    unsigned int idx,
    double q )
```

**10.169.3.18 SetRate()**

```
void gdcm::JPEG2000Codec::SetRate (
    unsigned int idx,
    double rate )
```

**10.169.3.19 SetReversible()**

```
void gdcm::JPEG2000Codec::SetReversible (
    bool res )
```

**10.169.3.20 SetTileSize()**

```
void gdcm::JPEG2000Codec::SetTileSize (
    unsigned int tx,
    unsigned int ty )
```

**10.169.3.21 StartEncode()**

```
bool gdcm::JPEG2000Codec::StartEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.169.3.22 StopEncode()

```
bool gdcmm::JPEG2000Codec::StopEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcmm::ImageCodec](#).

## 10.169.4 Friends And Related Function Documentation

### 10.169.4.1 Bitmap

```
friend class Bitmap [friend]
```

### 10.169.4.2 ImageRegionReader

```
friend class ImageRegionReader [friend]
```

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

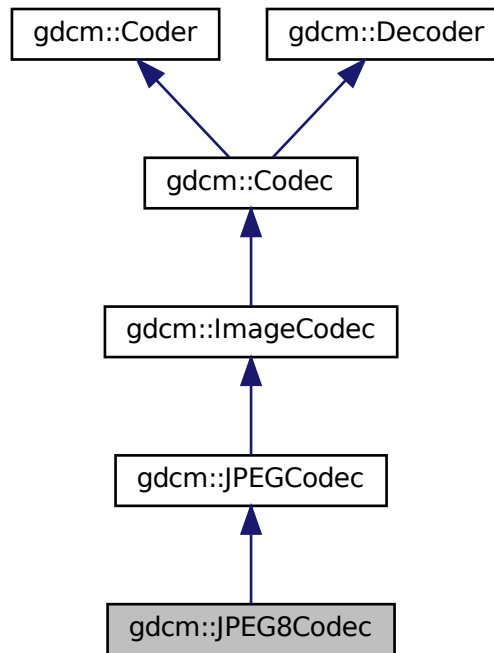
- [gdcmmJPEG2000Codec.h](#)

## 10.170 gdcmm::JPEG8Codec Class Reference

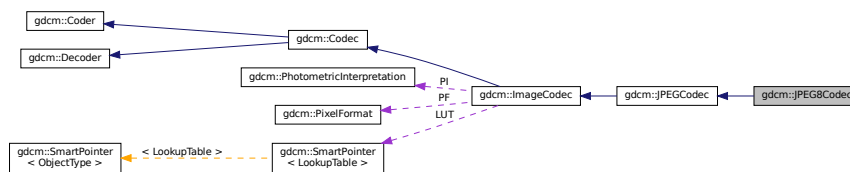
Class to do JPEG 8bits (lossy & lossless)

```
#include <gdcmmJPEG8Codec.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

## Protected Member Functions

- bool [EncodeBuffer](#) (std::ostream &os, const char \*data, size\_t datalen) override
- bool [IsStateSuspension](#) () const override

## Additional Inherited Members

### 10.170.1 Detailed Description

Class to do JPEG 8bits (lossy & lossless)

#### Note

internal class

### 10.170.2 Constructor & Destructor Documentation

#### 10.170.2.1 JPEG8Codec()

```
gdcm::JPEG8Codec::JPEG8Codec ( )
```

#### 10.170.2.2 ~JPEG8Codec()

```
gdcm::JPEG8Codec::~~JPEG8Codec ( ) [override]
```

### 10.170.3 Member Function Documentation

#### 10.170.3.1 DecodeByStreams()

```
bool gdcm::JPEG8Codec::DecodeByStreams (
    std::istream & is,
    std::ostream & os ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).



### 10.170.3.2 EncodeBuffer()

```
bool gdcm::JPEG8Codec::EncodeBuffer (
    std::ostream & os,
    const char * data,
    size_t datalen ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

### 10.170.3.3 GetHeaderInfo()

```
bool gdcm::JPEG8Codec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.170.3.4 InternalCode()

```
bool gdcm::JPEG8Codec::InternalCode (
    const char * input,
    unsigned long len,
    std::ostream & os ) [override], [virtual]
```

Reimplemented from [gdcm::Coder](#).

### 10.170.3.5 IsStateSuspension()

```
bool gdcm::JPEG8Codec::IsStateSuspension ( ) const [override], [protected], [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

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

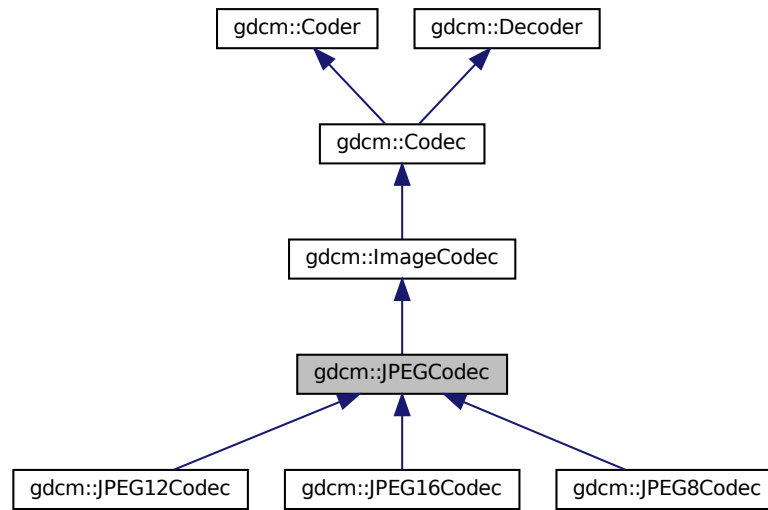
- [gdcmJPEG8Codec.h](#)

## 10.171 gdcm::JPEGCodec Class Reference

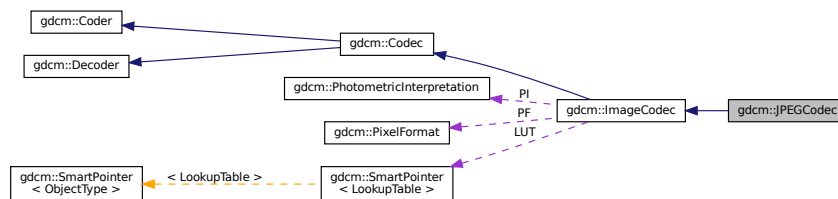
JPEG codec.

```
#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)

## 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 Attributes

- int [BitSample](#)
- int [Quality](#)

## Friends

- class [ImageRegionReader](#)

## Additional Inherited Members

### 10.171.1 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/2d525ef6a2f08](http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/2d525ef6a2f08)
- [http://groups.google.com/group/comp.protocols.dicom/browse\\_thread/thread/6b93af410f8c8](http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/6b93af410f8c8)

#### Examples

[GetJPEGSamplePrecision.cxx](#).

### 10.171.2 Constructor & Destructor Documentation

#### 10.171.2.1 JPEGCodec()

```
gdcm::JPEGCodec::JPEGCodec ( )
```

#### 10.171.2.2 ~JPEGCodec()

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

### 10.171.3 Member Function Documentation

### 10.171.3.1 AppendFrameEncode()

```
bool gdcm::JPEGCodec::AppendFrameEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.171.3.2 AppendRowEncode()

```
bool gdcm::JPEGCodec::AppendRowEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.171.3.3 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](#).

### 10.171.3.4 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](#).

### 10.171.3.5 Clone()

```
ImageCodec* gdcm::JPEGCodec::Clone ( ) const [override], [virtual]
```

Implements [gdcm::ImageCodec](#).

### 10.171.3.6 Code()

```
bool gdcm::JPEGCodec::Code (
    DataElement const & in,
    DataElement & out ) [override], [virtual]
```

Compress into JPEG.

Reimplemented from [gdcm::Coder](#).

### 10.171.3.7 ComputeOffsetTable()

```
void gdcm::JPEGCodec::ComputeOffsetTable (
    bool b )
```

Compute the offset table:

### 10.171.3.8 Decode()

```
bool gdcm::JPEGCodec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcm::ImageCodec](#).

### 10.171.3.9 DecodeByStreams()

```
bool gdcm::JPEGCodec::DecodeByStreams (
    std::istream & is,
    std::ostream & os ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.171.3.10 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]
```

### 10.171.3.11 EncodeBuffer()

```
virtual bool gdcm::JPEGCodec::EncodeBuffer (
    std::ostream & out,
    const char * inbuffer,
    size_t inlen ) [virtual]
```

Reimplemented in [gdcm::JPEG8Codec](#), [gdcm::JPEG16Codec](#), and [gdcm::JPEG12Codec](#).

### 10.171.3.12 GetHeaderInfo()

```
bool gdcm::JPEGCodec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### Examples

[GetJPEGSamplePrecision.cxx](#).

### 10.171.3.13 GetLossless()

```
bool gdcm::JPEGCodec::GetLossless ( ) const
```

#### 10.171.3.14 GetQuality()

```
double gdcm::JPEGCodec::GetQuality ( ) const
```

#### 10.171.3.15 IsFrameEncoder()

```
bool gdcm::JPEGCodec::IsFrameEncoder ( ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### 10.171.3.16 IsRowEncoder()

```
bool gdcm::JPEGCodec::IsRowEncoder ( ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### 10.171.3.17 IsStateSuspension()

```
virtual bool gdcm::JPEGCodec::IsStateSuspension ( ) const [protected], [virtual]
```

Reimplemented in [gdcm::JPEG8Codec](#), [gdcm::JPEG16Codec](#), and [gdcm::JPEG12Codec](#).

#### 10.171.3.18 IsValid()

```
bool gdcm::JPEGCodec::IsValid (
    PhotometricInterpretation const & pi ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### 10.171.3.19 SetBitSample()

```
void gdcm::JPEGCodec::SetBitSample (
    int bit ) [protected]
```



### 10.171.3.20 SetLossless()

```
void gdcm::JPEGCodec::SetLossless (
    bool l )
```

### 10.171.3.21 SetPixelFormat()

```
void gdcm::JPEGCodec::SetPixelFormat (
    PixelFormat const & pf ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### Examples

[GetJPEGSamplePrecision.cxx](#).

### 10.171.3.22 SetQuality()

```
void gdcm::JPEGCodec::SetQuality (
    double q )
```

### 10.171.3.23 StartEncode()

```
bool gdcm::JPEGCodec::StartEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.171.3.24 StopEncode()

```
bool gdcm::JPEGCodec::StopEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

## 10.171.4 Friends And Related Function Documentation

### 10.171.4.1 ImageRegionReader

```
friend class ImageRegionReader [friend]
```

## 10.171.5 Member Data Documentation

### 10.171.5.1 BitSample

```
int gdcM::JPEGCodec::BitSample [protected]
```

### 10.171.5.2 Quality

```
int gdcM::JPEGCodec::Quality [protected]
```

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

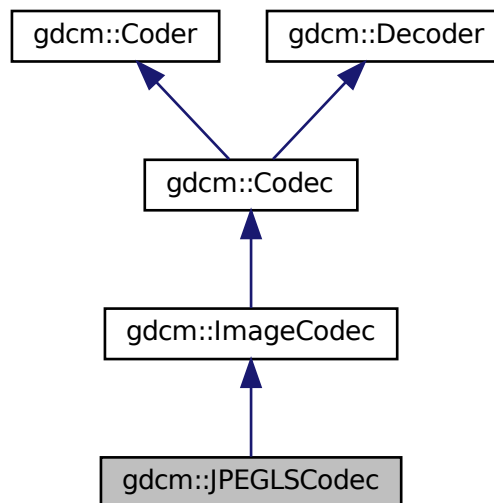
- [gdcMJPEGCodec.h](#)

## 10.172 gdcM::JPEGLSCodec Class Reference

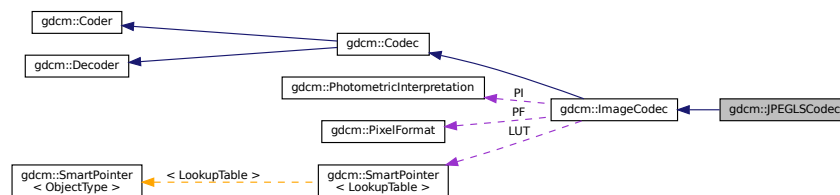
JPEG-LS.

```
#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)
- 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*

## 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

## Friends

- class [ImageRegionReader](#)

## Additional Inherited Members

### 10.172.1 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>

### 10.172.2 Constructor & Destructor Documentation

### 10.172.2.1 JPEGLSCodec()

```
gdcm::JPEGLSCodec::JPEGLSCodec ( )
```

### 10.172.2.2 ~JPEGLSCodec()

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

## 10.172.3 Member Function Documentation

### 10.172.3.1 AppendFrameEncode()

```
bool gdcm::JPEGLSCodec::AppendFrameEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.172.3.2 AppendRowEncode()

```
bool gdcm::JPEGLSCodec::AppendRowEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.172.3.3 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](#).

#### 10.172.3.4 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](#).

#### 10.172.3.5 Clone()

```
ImageCodec* gdcm::JPEGLSCodec::Clone ( ) const [override], [virtual]
```

Implements [gdcm::ImageCodec](#).

#### 10.172.3.6 Code()

```
bool gdcm::JPEGLSCodec::Code (
    DataElement const & in_,
    DataElement & out_ ) [override], [virtual]
```

Code.

Reimplemented from [gdcm::Coder](#).

#### 10.172.3.7 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 )
```

### 10.172.3.8 Decode() [2/2]

```
bool gdcm::JPEGLSCodec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcm::ImageCodec](#).

### 10.172.3.9 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]
```

### 10.172.3.10 GetBufferLength()

```
unsigned long gdcm::JPEGLSCodec::GetBufferLength ( ) const [inline]
```

### 10.172.3.11 GetHeaderInfo()

```
bool gdcm::JPEGLSCodec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.172.3.12 GetLossless()

```
bool gdcm::JPEGLSCodec::GetLossless ( ) const
```

**10.172.3.13 IsFrameEncoder()**

```
bool gdcM::JPEGLSCodec::IsFrameEncoder ( ) [override], [protected], [virtual]
```

Reimplemented from [gdcM::ImageCodec](#).

**10.172.3.14 IsRowEncoder()**

```
bool gdcM::JPEGLSCodec::IsRowEncoder ( ) [override], [protected], [virtual]
```

Reimplemented from [gdcM::ImageCodec](#).

**10.172.3.15 SetBufferLength()**

```
void gdcM::JPEGLSCodec::SetBufferLength (
    unsigned long l ) [inline]
```

**10.172.3.16 SetLossless()**

```
void gdcM::JPEGLSCodec::SetLossless (
    bool l )
```

**10.172.3.17 SetLossyError()**

```
void gdcM::JPEGLSCodec::SetLossyError (
    int error )
```

[0-3] generally

**10.172.3.18 StartEncode()**

```
bool gdcM::JPEGLSCodec::StartEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcM::ImageCodec](#).



### 10.172.3.19 StopEncode()

```
bool gdcm::JPEGLSCodec::StopEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

## 10.172.4 Friends And Related Function Documentation

### 10.172.4.1 ImageRegionReader

```
friend class ImageRegionReader [friend]
```

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

- [gdcmJPEGLSCodec.h](#)

## 10.173 gdcm::JSON Class Reference

```
#include <gdcmJSON.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)

### 10.173.1 Detailed Description

Examples

[QIDO-RS.cxx](#).

### 10.173.2 Constructor & Destructor Documentation

### 10.173.2.1 JSON()

```
gdcm::JSON::JSON ( )
```

### 10.173.2.2 ~JSON()

```
gdcm::JSON::~~JSON ( )
```

## 10.173.3 Member Function Documentation

### 10.173.3.1 Code()

```
bool gdcm::JSON::Code (
    DataSet const & in,
    std::ostream & os )
```

#### Examples

[QIDO-RS.cxx](#).

### 10.173.3.2 Decode()

```
bool gdcm::JSON::Decode (
    std::istream & is,
    DataSet & out )
```

#### Examples

[QIDO-RS.cxx](#).

### 10.173.3.3 GetPrettyPrint()

```
bool gdcm::JSON::GetPrettyPrint ( ) const
```

#### 10.173.3.4 PrettyPrintOff()

```
void gdcm::JSON::PrettyPrintOff ( )
```

#### 10.173.3.5 PrettyPrintOn()

```
void gdcm::JSON::PrettyPrintOn ( )
```

#### Examples

[QIDO-RS.cxx](#).

#### 10.173.3.6 SetPrettyPrint()

```
void gdcm::JSON::SetPrettyPrint (
    bool onoff )
```

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

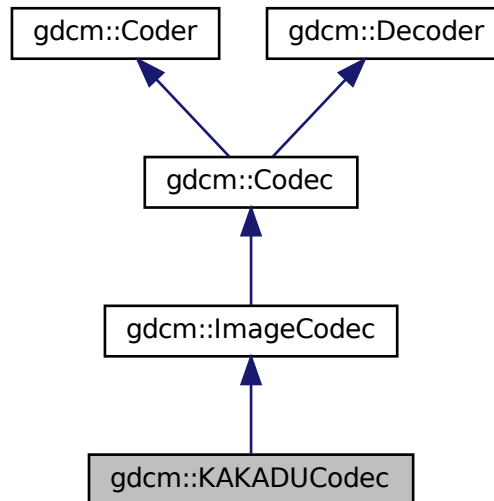
- [gdcmJSON.h](#)

## 10.174 gdcm::KAKADUCodec Class Reference

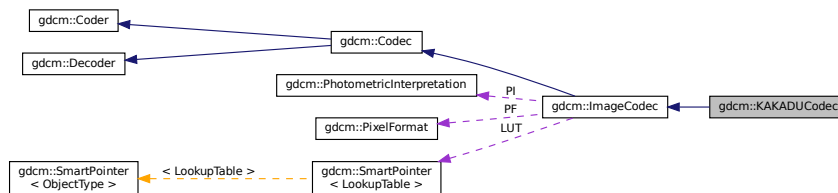
[KAKADUCodec](#).

```
#include <gdcmKAKADUCodec.h>
```

Inheritance diagram for `gdcm::KAKADUCodec`:



Collaboration diagram for `gdcm::KAKADUCodec`:



## 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.*

## Additional Inherited Members

### 10.174.1 Detailed Description

[KAKADUCodec](#).

### 10.174.2 Constructor & Destructor Documentation

#### 10.174.2.1 KAKADUCodec()

```
gdcm::KAKADUCodec::KAKADUCodec ( )
```

#### 10.174.2.2 ~KAKADUCodec()

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

### 10.174.3 Member Function Documentation

#### 10.174.3.1 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](#).

#### 10.174.3.2 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](#).

### 10.174.3.3 Clone()

```
ImageCodec* gdcM::KAKADUCodec::Clone ( ) const [override], [virtual]
```

Implements [gdcM::ImageCodec](#).

### 10.174.3.4 Code()

```
bool gdcM::KAKADUCodec::Code (
    DataElement const & in_,
    DataElement & out_ ) [override], [virtual]
```

Code.

Reimplemented from [gdcM::Coder](#).

### 10.174.3.5 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:

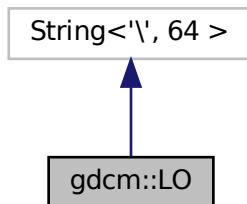
- [gdcMKAKADUCodec.h](#)

## 10.175 gdcm::LO Class Reference

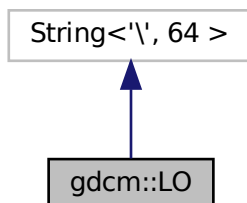
[LO](#).

```
#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=npow)
- [LO](#) (const [value\\_type](#) \*s)
- [LO](#) (const [value\\_type](#) \*s, [size\\_type](#) n)
- bool [IsValid](#) () const

### 10.175.1 Detailed Description

[LO](#).

Note

TODO

### 10.175.2 Member Typedef Documentation

#### 10.175.2.1 `const_iterator`

```
typedef Superclass::const\_iterator gdcm::LO::const\_iterator
```

#### 10.175.2.2 `const_reference`

```
typedef Superclass::const\_reference gdcm::LO::const\_reference
```

#### 10.175.2.3 `const_reverse_iterator`

```
typedef Superclass::const\_reverse\_iterator gdcm::LO::const\_reverse\_iterator
```

#### 10.175.2.4 `difference_type`

```
typedef Superclass::difference\_type gdcm::LO::difference\_type
```



#### 10.175.2.5 iterator

```
typedef Superclass::iterator gdcm::LO::iterator
```

#### 10.175.2.6 pointer

```
typedef Superclass::pointer gdcm::LO::pointer
```

#### 10.175.2.7 reference

```
typedef Superclass::reference gdcm::LO::reference
```

#### 10.175.2.8 reverse\_iterator

```
typedef Superclass::reverse_iterator gdcm::LO::reverse_iterator
```

#### 10.175.2.9 size\_type

```
typedef Superclass::size_type gdcm::LO::size_type
```

#### 10.175.2.10 Superclass

```
typedef String<'\\', 64> gdcm::LO::Superclass
```

#### 10.175.2.11 value\_type

```
typedef Superclass::value_type gdcm::LO::value_type
```

### 10.175.3 Constructor & Destructor Documentation

#### 10.175.3.1 LO() [1/4]

```
gdcmm::LO::LO ( ) [inline]
```

#### 10.175.3.2 LO() [2/4]

```
gdcmm::LO::LO (
    const value\_type * s ) [inline]
```

#### 10.175.3.3 LO() [3/4]

```
gdcmm::LO::LO (
    const value\_type * s,
    size\_type n ) [inline]
```

#### 10.175.3.4 LO() [4/4]

```
gdcmm::LO::LO (
    const Superclass & s,
    size\_type pos = 0,
    size\_type n = npos ) [inline]
```

### 10.175.4 Member Function Documentation

#### 10.175.4.1 IsValid()

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

References `gdcmm::String< TDelimiter, TMaxLength, TPadChar >::IsValid()`.

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

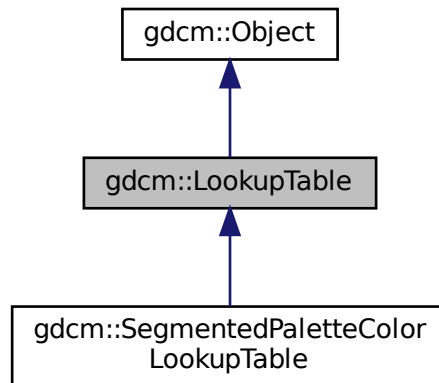
- [gdcmmLO.h](#)

## 10.176 gdcm::LookupTable Class Reference

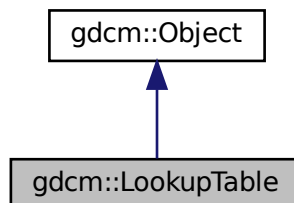
[LookupTable](#) class.

```
#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.*

## Protected Attributes

- unsigned short [BitSample](#)
- bool [IncompleteLUT](#):1
- [LookupTableInternal](#) \* [Internal](#)

## Additional Inherited Members

### 10.176.1 Detailed Description

[LookupTable](#) class.

#### Examples

[PrintLUT.cxx](#).

### 10.176.2 Member Enumeration Documentation

#### 10.176.2.1 LookupTableType

```
enum gdcm::LookupTable::LookupTableType
```

##### Enumerator

RED	
GREEN	
BLUE	
GRAY	
UNKNOWN	

### 10.176.3 Constructor & Destructor Documentation

#### 10.176.3.1 LookupTable() [1/2]

```
gdcm::LookupTable::LookupTable ( )
```

#### 10.176.3.2 ~LookupTable()

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

### 10.176.3.3 LookupTable() [2/2]

```
gdcmm::LookupTable::LookupTable (
    LookupTable const & lut ) [inline]
```

## 10.176.4 Member Function Documentation

### 10.176.4.1 Allocate()

```
void gdcmm::LookupTable::Allocate (
    unsigned short bitsample = 8 )
```

Allocate the LUT.

### 10.176.4.2 Clear()

```
void gdcmm::LookupTable::Clear ( )
```

Clear the LUT.

### 10.176.4.3 Decode() [1/2]

```
bool gdcmm::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

### 10.176.4.4 Decode() [2/2]

```
void gdcmm::LookupTable::Decode (
    std::istream & is,
    std::ostream & os ) const
```

Decode the LUT.

#### 10.176.4.5 Decode8()

```
bool gdcm::LookupTable::Decode8 (
    char * outputbuffer,
    size_t outlen,
    const char * inputbuffer,
    size_t inlen ) const
```

Decode into RGB 8 bits space.

#### 10.176.4.6 GetBitSample()

```
unsigned short gdcm::LookupTable::GetBitSample ( ) const [inline]
```

return the bit sample

#### 10.176.4.7 GetBufferAsRGBA()

```
bool gdcm::LookupTable::GetBufferAsRGBA (
    unsigned char * rgba ) const
```

return the LUT as RGBA buffer

#### 10.176.4.8 GetLUT()

```
void gdcm::LookupTable::GetLUT (
    LookupTableType type,
    unsigned char * array,
    unsigned int & length ) const
```

#### Examples

[ExtractImageRegionWithLUT.cs](#).

**10.176.4.9 GetLUTDescriptor()**

```
void gdcm::LookupTable::GetLUTDescriptor (
    LookupTableType type,
    unsigned short & length,
    unsigned short & subscript,
    unsigned short & bitsize ) const
```

**10.176.4.10 GetLUTLength()**

```
unsigned int gdcm::LookupTable::GetLUTLength (
    LookupTableType type ) const
```

**10.176.4.11 GetPointer()**

```
const unsigned char* gdcm::LookupTable::GetPointer ( ) const
```

return a raw pointer to the LUT

**10.176.4.12 InitializeBlueLUT()**

```
void gdcm::LookupTable::InitializeBlueLUT (
    unsigned short length,
    unsigned short subscript,
    unsigned short bitsize )
```

**10.176.4.13 Initialized()**

```
bool gdcm::LookupTable::Initialized ( ) const
```

return whether the LUT has been initialized



#### 10.176.4.14 InitializeGreenLUT()

```
void gdcm::LookupTable::InitializeGreenLUT (
    unsigned short length,
    unsigned short subscript,
    unsigned short bitsize )
```

#### 10.176.4.15 InitializeLUT()

```
void gdcm::LookupTable::InitializeLUT (
    LookupTableType type,
    unsigned short length,
    unsigned short subscript,
    unsigned short bitsize )
```

Generic interface:

#### 10.176.4.16 InitializeRedLUT()

```
void gdcm::LookupTable::InitializeRedLUT (
    unsigned short length,
    unsigned short subscript,
    unsigned short bitsize )
```

RED / GREEN / BLUE specific:

#### 10.176.4.17 IsRGB8()

```
bool gdcm::LookupTable::IsRGB8 ( ) const
```

Return whether 16 bits LUT is in RGB 8 bits space.

#### 10.176.4.18 Print()

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

Reimplemented from [gdcm::Object](#).

Reimplemented in [gdcm::SegmentedPaletteColorLookupTable](#).

Examples

[PrintLUT.cxx](#).

#### 10.176.4.19 SetBlueLUT()

```
void gdcm::LookupTable::SetBlueLUT (
    const unsigned char * blue,
    unsigned int length )
```

#### 10.176.4.20 SetGreenLUT()

```
void gdcm::LookupTable::SetGreenLUT (
    const unsigned char * green,
    unsigned int length )
```

#### 10.176.4.21 SetLUT()

```
virtual void gdcm::LookupTable::SetLUT (
    LookupTableType type,
    const unsigned char * array,
    unsigned int length ) [virtual]
```

Reimplemented in [gdcm::SegmentedPaletteColorLookupTable](#).

#### 10.176.4.22 SetRedLUT()

```
void gdcm::LookupTable::SetRedLUT (
    const unsigned char * red,
    unsigned int length )
```

#### 10.176.4.23 WriteBufferAsRGBA()

```
bool gdcm::LookupTable::WriteBufferAsRGBA (
    const unsigned char * rgba )
```

Write the LUT as RGBA.

### 10.176.5 Member Data Documentation

### 10.176.5.1 BitSample

```
unsigned short gdcm::LookupTable::BitSample [protected]
```

### 10.176.5.2 IncompleteLUT

```
bool gdcm::LookupTable::IncompleteLUT [protected]
```

### 10.176.5.3 Internal

```
LookupTableInternal* gdcm::LookupTable::Internal [protected]
```

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

- [gdcmLookupTable.h](#)

## 10.177 gdcm::Scanner::ltstr Struct Reference

```
#include <gdcmScanner.h>
```

### Public Member Functions

- bool [operator\(\)](#) (const char \*s1, const char \*s2) const

### 10.177.1 Member Function Documentation

#### 10.177.1.1 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](#)

## 10.178 gdcm::StrictScanner::ltstr Struct Reference

```
#include <gdcmStrictScanner.h>
```

### Public Member Functions

- bool [operator\(\)](#) (const char \*s1, const char \*s2) const

### 10.178.1 Member Function Documentation

#### 10.178.1.1 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](#)

## 10.179 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)`

### 10.179.1 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](#)

### 10.179.2 Member Typedef Documentation

#### 10.179.2.1 ArrayIncludeMacrosType

```
typedef std::vector<std::string> gdcmmacro::Macro::ArrayIncludeMacrosType
```

#### 10.179.2.2 MapModuleEntry

```
typedef std::map<Tag, MacroEntry> gdcmmacro::Macro::MapModuleEntry
```

### 10.179.3 Constructor & Destructor Documentation

#### 10.179.3.1 Macro()

```
gdcmmacro::Macro::Macro ( ) [default]
```

### 10.179.4 Member Function Documentation

#### 10.179.4.1 AddMacroEntry()

```
void gdcmm::Macro::AddMacroEntry (
    const Tag & tag,
    const MacroEntry & module ) [inline]
```

Will add a [ModuleEntry](#) directly at root-level. See [Macro](#) for nested-included level.

#### 10.179.4.2 Clear()

```
void gdcmm::Macro::Clear ( ) [inline]
```

#### 10.179.4.3 FindMacroEntry()

```
bool gdcmm::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.

#### 10.179.4.4 GetMacroEntry()

```
const MacroEntry& gdcmm::Macro::GetMacroEntry (
    const Tag & tag ) const
```

#### 10.179.4.5 GetName()

```
const char* gdcmm::Macro::GetName ( ) const [inline]
```

#### 10.179.4.6 SetName()

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

#### 10.179.4.7 Verify()

```
bool gdcmmacros::Macro::Verify (
    const DataSet & ds,
    Usage const & usage ) const
```

### 10.179.5 Friends And Related Function Documentation

#### 10.179.5.1 operator<<

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

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

- [gdcmmacros.h](#)

## 10.180 gdcmmacros 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)

### 10.180.1 Detailed Description

Class for representing a [Modules](#).

Note

bla

See also

[Module](#)

### 10.180.2 Member Typedef Documentation

#### 10.180.2.1 ModuleMapType

```
typedef std::map<std::string, Macro> gdcmm::Macros::ModuleMapType
```

### 10.180.3 Constructor & Destructor Documentation

#### 10.180.3.1 Macros()

```
gdcmm::Macros::Macros ( ) [default]
```

### 10.180.4 Member Function Documentation

#### 10.180.4.1 AddMacro()

```
void gdcmm::Macros::AddMacro (  
    const char * ref,  
    const Macro & module ) [inline]
```



#### 10.180.4.2 Clear()

```
void gdcm::Macros::Clear ( ) [inline]
```

#### 10.180.4.3 GetMacro()

```
const Macro& gdcm::Macros::GetMacro (
    const char * name ) const [inline]
```

#### 10.180.4.4 IsEmpty()

```
bool gdcm::Macros::IsEmpty ( ) const [inline]
```

### 10.180.5 Friends And Related Function Documentation

#### 10.180.5.1 operator<<

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

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

- [gdcmMacros.h](#)

## 10.181 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

### 10.181.1 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)

### 10.181.2 Constructor & Destructor Documentation

#### 10.181.2.1 MaximumLengthSub()

```
gdcn::network::MaximumLengthSub::MaximumLengthSub ( )
```

### 10.181.3 Member Function Documentation

#### 10.181.3.1 GetMaximumLength()

```
uint32_t gdcn::network::MaximumLengthSub::GetMaximumLength ( ) const [inline]
```

#### 10.181.3.2 Print()

```
void gdcn::network::MaximumLengthSub::Print (
    std::ostream & os ) const
```

### 10.181.3.3 Read()

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

### 10.181.3.4 SetMaximumLength()

```
void gdcm::network::MaximumLengthSub::SetMaximumLength (
    uint32_t maximumlength )
```

### 10.181.3.5 Size()

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

### 10.181.3.6 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](#)

## 10.182 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*

### 10.182.1 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

### 10.182.2 Member Function Documentation

#### 10.182.2.1 Compute()

```
static bool gdcM::MD5::Compute (
    const char * buffer,
    size_t buf_len,
    char digest_str[33] ) [static]
```

#### 10.182.2.2 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](#)

## 10.183 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 ,  
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 ,  
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 ,
FujiPrivateCRIImageStorage ,
OphthalmicPhotography8BitImageStorage ,
OphthalmicTomographyImageStorage ,
VLMicroscopicImageStorage ,
EnhancedPETImageStorage ,
VideoPhotographicImageStorage ,
XRay3DCraniofacialImageStorage ,
IVOCTForPresentation ,
IVOCTForProcessing ,
LegacyConvertedEnhancedCTImageStorage ,
LegacyConvertedEnhancedMRIImageStorage ,
LegacyConvertedEnhancedPETImageStorage ,
BreastProjectionXRayImageStorageForPresentation ,
BreastProjectionXRayImageStorageForProcessing ,
HardcopyColorImageStorage ,
EnhancedMRColorImageStorage ,
FujiPrivateMammoCRIImageStorage ,
OphthalmicPhotography16BitImageStorage ,
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 MSType](#) () 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](#) (MSType ts)  
*Return the Media [String](#) associated. Will return NULL for MS\_END.*
- static MSType [GetMSType](#) (const char \*str)
- static unsigned int [GetNumberOfModality](#) ()
- static unsigned int [GetNumberOfMSString](#) ()
- static unsigned int [GetNumberOfMSType](#) ()
- static bool [IsImage](#) (MSType ts)

## Protected Member Functions

- void [SetFromSourceImageSequence](#) ([DataSet](#) const &ds)

## Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [MediaStorage](#) &ms)

### 10.183.1 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](#), [GenAIIVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenerateStandardSOPClasses.cxx](#), [GetSubSequenceData.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), [TestReader.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), and [iU22tomultisc.cxx](#).

### 10.183.2 Member Enumeration Documentation

#### 10.183.2.1 MSType

```
enum gdcm::MediaStorage::MSType
```

#### Enumerator

<a href="#">MediaStorageDirectoryStorage</a>	
<a href="#">ComputedRadiographyImageStorage</a>	
<a href="#">DigitalXRayImageStorageForPresentation</a>	
<a href="#">DigitalXRayImageStorageForProcessing</a>	
<a href="#">DigitalMammographyImageStorageForPresentation</a>	
<a href="#">DigitalMammographyImageStorageForProcessing</a>	
<a href="#">DigitalIntraoralXrayImageStorageForPresentation</a>	
<a href="#">DigitalIntraoralXRayImageStorageForProcessing</a>	
<a href="#">CTImageStorage</a>	
<a href="#">EnhancedCTImageStorage</a>	
<a href="#">UltrasoundImageStorageRetired</a>	
<a href="#">UltrasoundImageStorage</a>	
<a href="#">UltrasoundMultiFrameImageStorageRetired</a>	
<a href="#">UltrasoundMultiFrameImageStorage</a>	
<a href="#">MRImageStorage</a>	



## Enumerator

EnhancedMRIImageStorage	
MRSpectroscopyStorage	
NuclearMedicineImageStorageRetired	
SecondaryCaptureImageStorage	
MultiframeSingleBitSecondaryCaptureImageStorage	
MultiframeGrayscaleByteSecondaryCaptureImageStorage	
MultiframeGrayscaleWordSecondaryCaptureImageStorage	
MultiframeTrueColorSecondaryCaptureImageStorage	
StandaloneOverlayStorage	
StandaloneCurveStorage	
LeadECGWaveformStorage	
GeneralECGWaveformStorage	
AmbulatoryECGWaveformStorage	
HemodynamicWaveformStorage	
CardiacElectrophysiologyWaveformStorage	
BasicVoiceAudioWaveformStorage	
StandaloneModalityLUTStorage	
StandaloneVOILUTStorage	
GrayscaleSoftcopyPresentationStateStorageSOPClass	
XR-RayAngiographicImageStorage	
XR-RayRadiofluoroscopicImageStorage	
XR-RayAngiographicBiPlaneImageStorageRetired	
NuclearMedicineImageStorage	
RawDataStorage	
SpacialRegistrationStorage	
SpacialFiducialsStorage	
PETImageStorage	
RTImageStorage	
RTDoseStorage	
RTStructureSetStorage	
RTPlanStorage	
CSANonImageStorage	
Philips3D	
EnhancedSR	
BasicTextSR	
HardcopyGrayscaleImageStorage	
ComprehensiveSR	
DetachedStudyManagementSOPClass	
EncapsulatedPDFStorage	
EncapsulatedCDASStorage	
StudyComponentManagementSOPClass	
DetachedVisitManagementSOPClass	
DetachedPatientManagementSOPClass	

## Enumerator

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	
XRay3DCraniofacialImageStorage	
IVOCTForPresentation	
IVOCTForProcessing	
LegacyConvertedEnhancedCTImageStorage	
LegacyConvertedEnhancedMRIImageStorage	
LegacyConvertedEnhancedPETImageStorage	
BreastProjectionXRayImageStorageForPresentation	
BreastProjectionXRayImageStorageForProcessing	
HardcopyColorImageStorage	
EnhancedMRColorImageStorage	
FujiPrivateMammoCRImageStorage	
OphthalmicPhotography16BitImageStorage	
MS_END	

## Examples

[GenerateStandardSOPClasses.cxx](#).

### 10.183.2.2 ObjectType

```
enum gdcm::MediaStorage::ObjectType
```

#### Enumerator

NoObject	
Video	
Waveform	
Audio	
PDF	
URI	
Segmentation	
ObjectEnd	

## 10.183.3 Constructor & Destructor Documentation

### 10.183.3.1 MediaStorage()

```
gdcm::MediaStorage::MediaStorage (
    MSType type = MS_END ) [inline]
```

## 10.183.4 Member Function Documentation

### 10.183.4.1 GetModality()

```
const char* gdcm::MediaStorage::GetModality ( ) const
```

#### 10.183.4.2 GetModalityDimension()

```
unsigned int gdcm::MediaStorage::GetModalityDimension ( ) const
```

#### 10.183.4.3 GetMSString()

```
static const char* gdcm::MediaStorage::GetMSString (
    MSType ts ) [static]
```

Return the Media [String](#) associated. Will return NULL for MS\_END.

##### Examples

[GenerateStandardSOPClasses.cxx](#).

#### 10.183.4.4 GetMSType()

```
static MSType gdcm::MediaStorage::GetMSType (
    const char * str ) [static]
```

##### Examples

[MetaImageMD5Activiz.cs](#), and [TestReader.cxx](#).

#### 10.183.4.5 GetNumberOfModality()

```
static unsigned int gdcm::MediaStorage::GetNumberOfModality ( ) [static]
```

#### 10.183.4.6 GetNumberOfMSString()

```
static unsigned int gdcm::MediaStorage::GetNumberOfMSString ( ) [static]
```

#### 10.183.4.7 GetNumberOfMSType()

```
static unsigned int gdcm::MediaStorage::GetNumberOfMSType ( ) [static]
```

#### 10.183.4.8 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](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), and [iU22tomultisc.cxx](#).

#### 10.183.4.9 GuessFromModality()

```
void gdcm::MediaStorage::GuessFromModality (
    const char * modality,
    unsigned int dimension = 2 )
```

#### 10.183.4.10 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](#).

#### 10.183.4.11 IsUndefined()

```
bool gdcM::MediaStorage::IsUndefined ( ) const [inline]
```

##### Examples

[TestReader.cxx](#).

#### 10.183.4.12 operator MStype()

```
gdcM::MediaStorage::operator MStype ( ) const [inline]
```

#### 10.183.4.13 SetFromDataSet()

```
bool gdcM::MediaStorage::SetFromDataSet (
    DataSet const & ds )
```

Advanced user only (functions should be protected level...) Those function are lower level than SetFromFile

#### 10.183.4.14 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](#).

#### 10.183.4.15 SetFromHeader()

```
bool gdcM::MediaStorage::SetFromHeader (
    FileMetaInformation const & fmi )
```

#### 10.183.4.16 SetFromModality()

```
bool gdcm::MediaStorage::SetFromModality (
    DataSet const & ds )
```

#### 10.183.4.17 SetFromSourceImageSequence()

```
void gdcm::MediaStorage::SetFromSourceImageSequence (
    DataSet const & ds ) [protected]
```

### 10.183.5 Friends And Related Function Documentation

#### 10.183.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const MediaStorage & ms ) [friend]
```

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

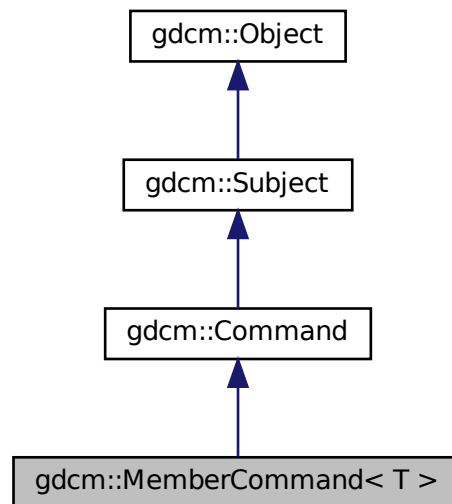
- [gdcmMediaStorage.h](#)

## 10.184 gdcm::MemberCommand< T > Class Template Reference

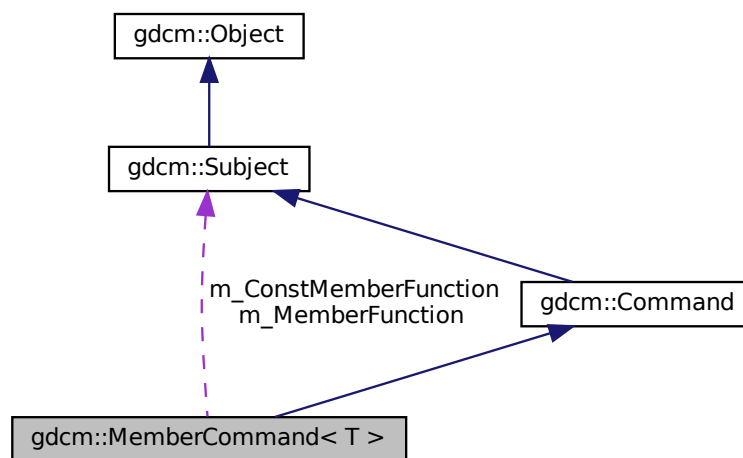
[Command](#) subclass that calls a pointer to a member function.

```
#include <gdcmCommand.h>
```

Inheritance diagram for `gdc::MemberCommand< T >`:



Collaboration diagram for `gdc::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)

## Static Public Member Functions

- static [SmartPointer](#)< [MemberCommand](#) > [New](#) ()

## Protected Member Functions

- [MemberCommand](#) ()
- [~MemberCommand](#) () override=default

## Protected Attributes

- [TConstMemberFunctionPointer](#) m\_ConstMemberFunction
- [TMemberFunctionPointer](#) m\_MemberFunction
- T \* m\_This

### 10.184.1 Detailed Description

```
template<class T>
class gdcmm::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](#).

### 10.184.2 Member Typedef Documentation

### 10.184.2.1 Self

```
template<class T >
typedef MemberCommand gdcM::MemberCommand< T >::Self
```

Standard class typedefs.

### 10.184.2.2 TConstMemberFunctionPointer

```
template<class T >
typedef void(T::* gdcM::MemberCommand< T >::TConstMemberFunctionPointer) (const Subject *, const Event &)
```

### 10.184.2.3 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

## 10.184.3 Constructor & Destructor Documentation

### 10.184.3.1 MemberCommand() [1/2]

```
template<class T >
gdcM::MemberCommand< T >::MemberCommand (
    const Self & ) [delete]
```

### 10.184.3.2 MemberCommand() [2/2]

```
template<class T >
gdcM::MemberCommand< T >::MemberCommand ( ) [inline], [protected]
```

Referenced by gdcM::MemberCommand< T >::New().

### 10.184.3.3 ~MemberCommand()

```
template<class T >
gdcmmembercommand< T >::~~MemberCommand ( ) [override], [protected], [default]
```

## 10.184.4 Member Function Documentation

### 10.184.4.1 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::Command](#).

References [gdcmmembercommand< T >::m\\_ConstMemberFunction](#).

### 10.184.4.2 Execute() [2/2]

```
template<class T >
void gdcmmembercommand< T >::Execute (
    Subject * caller,
    const Event & event ) [inline], [override], [virtual]
```

Invoke the member function.

Implements [gdcmmembercommand::Command](#).

References [gdcmmembercommand< T >::m\\_MemberFunction](#).

### 10.184.4.3 New()

```
template<class T >
static SmartPointer<MemberCommand> gdcmmembercommand< T >::New ( ) [inline], [static]
```

Method for creation through the object factory.

References [gdcmmembercommand< T >::MemberCommand\(\)](#).

#### 10.184.4.4 operator=()

```
template<class T >
void gdcm::MemberCommand< T >::operator= (
    const Self & ) [delete]
```

#### 10.184.4.5 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.

#### 10.184.4.6 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.

### 10.184.5 Member Data Documentation

#### 10.184.5.1 m\_ConstMemberFunction

```
template<class T >
TConstMemberFunctionPointer gdcm::MemberCommand< T >::m_ConstMemberFunction [protected]
```

Referenced by gdcm::MemberCommand< T >::Execute(), and gdcm::MemberCommand< T >::SetCallbackFunction().

### 10.184.5.2 m\_MemberFunction

```
template<class T >
TMemberFunctionPointer gdcm::MemberCommand< T >::m_MemberFunction [protected]
```

Referenced by `gdcm::MemberCommand< T >::Execute()`, and `gdcm::MemberCommand< T >::SetCallbackFunction()`.

### 10.184.5.3 m\_This

```
template<class T >
T* gdcm::MemberCommand< T >::m_This [protected]
```

Referenced by `gdcm::MemberCommand< T >::SetCallbackFunction()`.

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

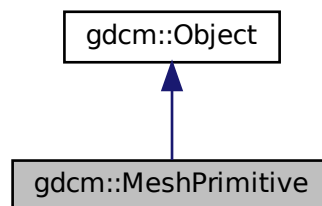
- [gdcmCommand.h](#)

## 10.185 gdcm::MeshPrimitive Class Reference

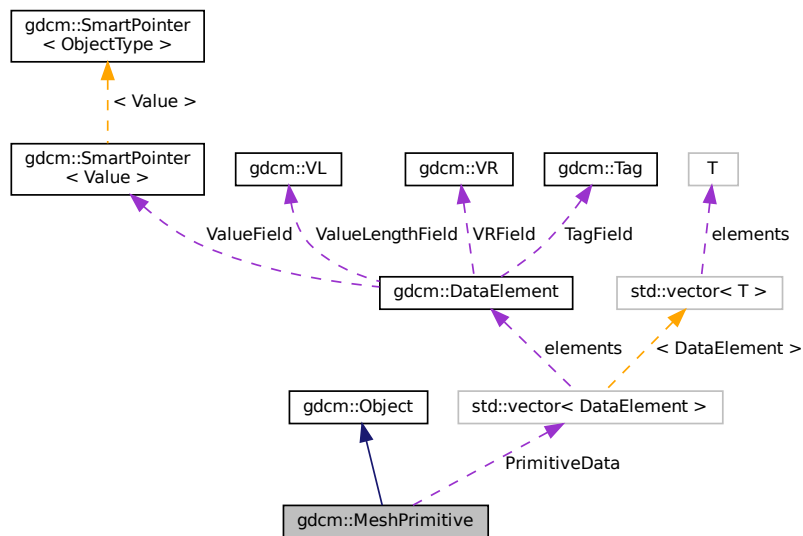
This class defines surface mesh primitives.

```
#include <gdcmMeshPrimitive.h>
```

Inheritance diagram for `gdcm::MeshPrimitive`:



Collaboration diagram for `gdcM::MeshPrimitive`:



## Public Types

- enum `MPType` {  
`VERTEX` = 0 ,  
`EDGE` ,  
`TRIANGLE` ,  
`TRIANGLE_STRIP` ,  
`TRIANGLE_FAN` ,  
`LINE` ,  
`FACET` ,  
`MPType_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

- [MPType](#) [GetPrimitiveType](#) () const
- void [SetPrimitiveData](#) (const unsigned int idx, [DataElement](#) const &de)
- void [SetPrimitiveData](#) ([DataElement](#) const &de)
- void [SetPrimitivesData](#) ([PrimitivesData](#) const &DEs)
- void [SetPrimitiveType](#) (const [MPType](#) type)

### Static Public Member Functions

- static [MPType](#) [GetMPType](#) (const char \*type)
- static const char \* [GetMPTypeString](#) (const [MPType](#) type)

### Protected Attributes

- [PrimitivesData](#) [PrimitiveData](#)
- [MPType](#) [PrimitiveType](#)

### Additional Inherited Members

#### 10.185.1 Detailed Description

This class defines surface mesh primitives.

It is designed from surface mesh primitives macro.

See also

PS 3.3 C.27.4

#### 10.185.2 Member Typedef Documentation

##### 10.185.2.1 PrimitivesData

```
typedef std::vector< DataElement > gdcm::MeshPrimitive::PrimitivesData
```

#### 10.185.3 Member Enumeration Documentation

##### 10.185.3.1 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	
MPTYPE_END	

## 10.185.4 Constructor & Destructor Documentation

### 10.185.4.1 MeshPrimitive()

```
gdcM::MeshPrimitive::MeshPrimitive ( )
```

### 10.185.4.2 ~MeshPrimitive()

```
gdcM::MeshPrimitive::~~MeshPrimitive ( ) [override]
```

## 10.185.5 Member Function Documentation

### 10.185.5.1 AddPrimitiveData()

```
void gdcM::MeshPrimitive::AddPrimitiveData (
    DataElement const & de )
```

### 10.185.5.2 GetMPTYPE()

```
static MPTYPE gdcM::MeshPrimitive::GetMPTYPE (
    const char * type ) [static]
```



### 10.185.5.3 GetMPTypeString()

```
static const char* gdcm::MeshPrimitive::GetMPTypeString (
    const MPType type ) [static]
```

### 10.185.5.4 GetNumberOfPrimitivesData()

```
unsigned int gdcm::MeshPrimitive::GetNumberOfPrimitivesData ( ) const
```

### 10.185.5.5 GetPrimitiveData() [1/4]

```
DataElement& gdcm::MeshPrimitive::GetPrimitiveData ( )
```

### 10.185.5.6 GetPrimitiveData() [2/4]

```
const DataElement& gdcm::MeshPrimitive::GetPrimitiveData ( ) const
```

### 10.185.5.7 GetPrimitiveData() [3/4]

```
DataElement& gdcm::MeshPrimitive::GetPrimitiveData (
    const unsigned int idx )
```

### 10.185.5.8 GetPrimitiveData() [4/4]

```
const DataElement& gdcm::MeshPrimitive::GetPrimitiveData (
    const unsigned int idx ) const
```

### 10.185.5.9 GetPrimitivesData() [1/2]

```
PrimitivesData& gdcm::MeshPrimitive::GetPrimitivesData ( )
```

**10.185.5.10 GetPrimitivesData() [2/2]**

```
const PrimitivesData& gdcM::MeshPrimitive::GetPrimitivesData ( ) const
```

**10.185.5.11 GetPrimitiveType()**

```
MPType gdcM::MeshPrimitive::GetPrimitiveType ( ) const
```

**10.185.5.12 SetPrimitiveData() [1/2]**

```
void gdcM::MeshPrimitive::SetPrimitiveData (
    const unsigned int idx,
    DataElement const & de )
```

**10.185.5.13 SetPrimitiveData() [2/2]**

```
void gdcM::MeshPrimitive::SetPrimitiveData (
    DataElement const & de )
```

**10.185.5.14 SetPrimitivesData()**

```
void gdcM::MeshPrimitive::SetPrimitivesData (
    PrimitivesData const & DEs )
```

**10.185.5.15 SetPrimitiveType()**

```
void gdcM::MeshPrimitive::SetPrimitiveType (
    const MPType type )
```

**10.185.6 Member Data Documentation**

### 10.185.6.1 PrimitiveData

`PrimitivesData` gdcm::MeshPrimitive::PrimitiveData [protected]

### 10.185.6.2 PrimitiveType

`MPTType` gdcm::MeshPrimitive::PrimitiveType [protected]

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

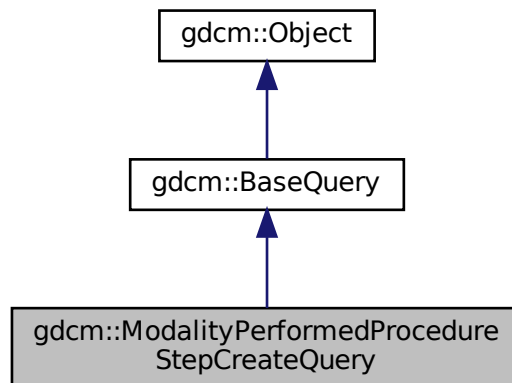
- [gdcmMeshPrimitive.h](#)

## 10.186 gdcm::ModalityPerformedProcedureStepCreateQuery Class Reference

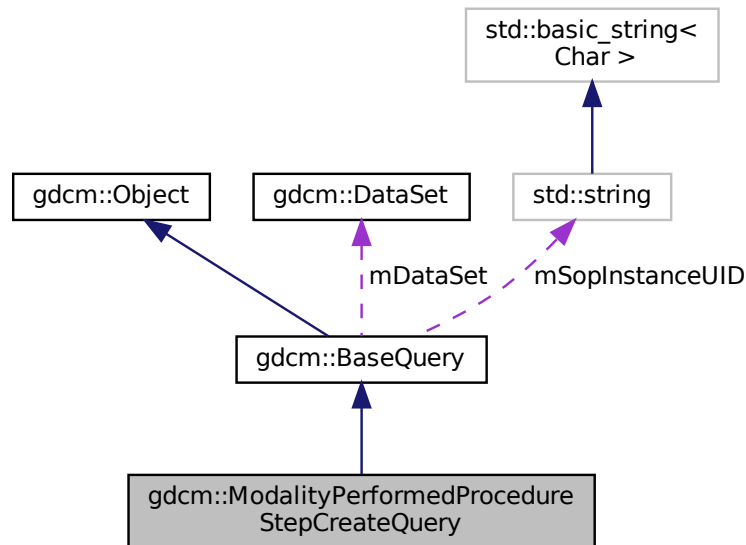
[ModalityPerformedProcedureStepCreateQuery.](#)

```
#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

## Friends

- class [QueryFactory](#)

## Additional Inherited Members

### 10.186.1 Detailed Description

[ModalityPerformedProcedureStepCreateQuery](#).

contains: the class which will produce a dataset for n-create for Modality Performed Procedure Step sop class

## 10.186.2 Constructor & Destructor Documentation

### 10.186.2.1 ModalityPerformedProcedureStepCreateQuery()

```
gdcm::ModalityPerformedProcedureStepCreateQuery::ModalityPerformedProcedureStepCreateQuery (
    const std::string & iSopInstanceUID )
```

## 10.186.3 Member Function Documentation

### 10.186.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::ModalityPerformedProcedureStepCreateQuery::GetAbstractSyntaxUID ( ) const [override],
[virtual]
```

Implements [gdcm::BaseQuery](#).

### 10.186.3.2 GetRequiredDataSet()

```
gdcm::DataSet gdcm::ModalityPerformedProcedureStepCreateQuery::GetRequiredDataSet ( ) const
```

### 10.186.3.3 ValidateQuery()

```
bool gdcm::ModalityPerformedProcedureStepCreateQuery::ValidateQuery (
    bool inStrict = true ) const [override], [virtual]
```

Implements [gdcm::BaseQuery](#).

## 10.186.4 Friends And Related Function Documentation

#### 10.186.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

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

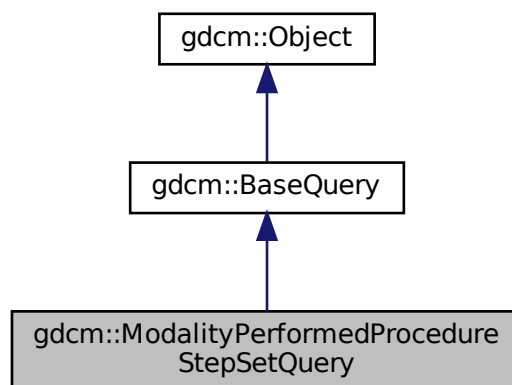
- [gdcmModalityPerformedProcedureStepCreateQuery.h](#)

### 10.187 gdcm::ModalityPerformedProcedureStepSetQuery Class Reference

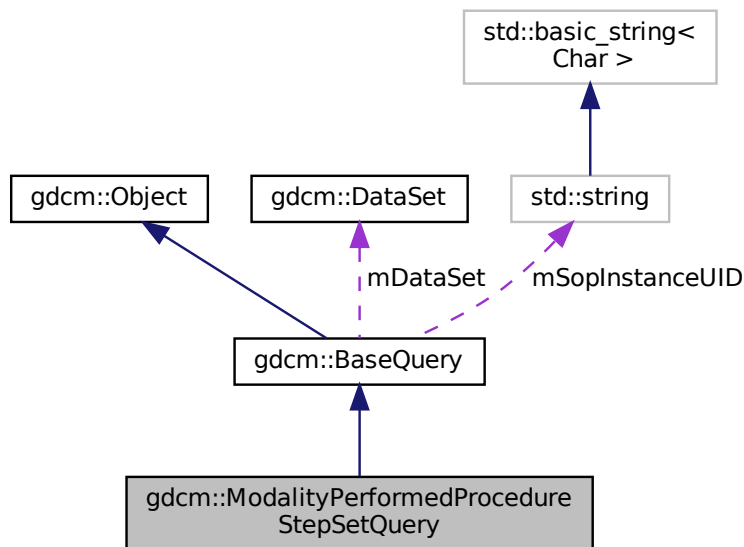
[ModalityPerformedProcedureStepSetQuery](#).

```
#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

## Friends

- class [QueryFactory](#)

## Additional Inherited Members

### 10.187.1 Detailed Description

[ModalityPerformedProcedureStepSetQuery](#).

contains: the class which will produce a dataset for n-set for Modality Performed Procedure Step sop class

## 10.187.2 Constructor & Destructor Documentation

### 10.187.2.1 ModalityPerformedProcedureStepSetQuery()

```
gdcm::ModalityPerformedProcedureStepSetQuery::ModalityPerformedProcedureStepSetQuery (
    const std::string & iSopInstanceUID )
```

## 10.187.3 Member Function Documentation

### 10.187.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::ModalityPerformedProcedureStepSetQuery::GetAbstractSyntaxUID ( ) const [override],
[virtual]
```

Implements [gdcm::BaseQuery](#).

### 10.187.3.2 GetRequiredDataSet()

```
gdcm::DataSet gdcm::ModalityPerformedProcedureStepSetQuery::GetRequiredDataSet ( ) const
```

### 10.187.3.3 ValidateQuery()

```
bool gdcm::ModalityPerformedProcedureStepSetQuery::ValidateQuery (
    bool inStrict = true ) const [override], [virtual]
```

Implements [gdcm::BaseQuery](#).

## 10.187.4 Friends And Related Function Documentation



### 10.187.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

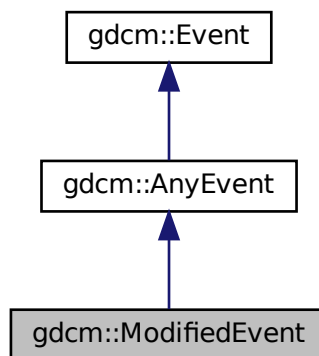
The documentation for this class was generated from the following file:

- [gdcmModalityPerformedProcedureStepSetQuery.h](#)

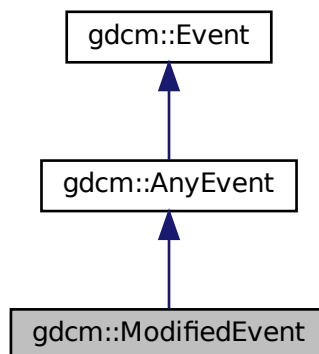
## 10.188 gdcm::ModifiedEvent Class Reference

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

Inheritance diagram for gdcm::ModifiedEvent:



Collaboration diagram for gdcm::ModifiedEvent:



## Additional Inherited Members

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

- [gdcmEvent.h](#)

## 10.189 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)

### 10.189.1 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](#)

## 10.189.2 Member Typedef Documentation

### 10.189.2.1 ArrayIncludeMacroType

```
typedef std::vector<std::string> gdcm::Module::ArrayIncludeMacroType
```

### 10.189.2.2 MapModuleEntry

```
typedef std::map<Tag, ModuleEntry> gdcm::Module::MapModuleEntry
```

## 10.189.3 Constructor & Destructor Documentation

### 10.189.3.1 Module()

```
gdcm::Module::Module ( ) [default]
```

## 10.189.4 Member Function Documentation

### 10.189.4.1 AddMacro()

```
void gdcm::Module::AddMacro (
    const char * include ) [inline]
```

### 10.189.4.2 AddModuleEntry()

```
void gdcm::Module::AddModuleEntry (
    const Tag & tag,
    const ModuleEntry & module ) [inline]
```

Will add a [ModuleEntry](#) directly at root-level. See [Macro](#) for nested-included level.

#### 10.189.4.3 Clear()

```
void gdcmm::Module::Clear ( ) [inline]
```

#### 10.189.4.4 FindModuleEntryInMacros()

```
bool gdcmm::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.

#### 10.189.4.5 GetModuleEntryInMacros()

```
const ModuleEntry& gdcmm::Module::GetModuleEntryInMacros (
    Macros const & macros,
    const Tag & tag ) const
```

#### 10.189.4.6 GetName()

```
const char* gdcmm::Module::GetName ( ) const [inline]
```

#### 10.189.4.7 SetName()

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

#### 10.189.4.8 Verify()

```
bool gdcmm::Module::Verify (
    const DataSet & ds,
    Usage const & usage ) const
```

### 10.189.5 Friends And Related Function Documentation

### 10.189.5.1 operator<<

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

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

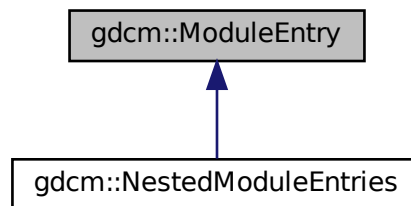
- [gdcmModule.h](#)

## 10.190 gdcm::ModuleEntry Class Reference

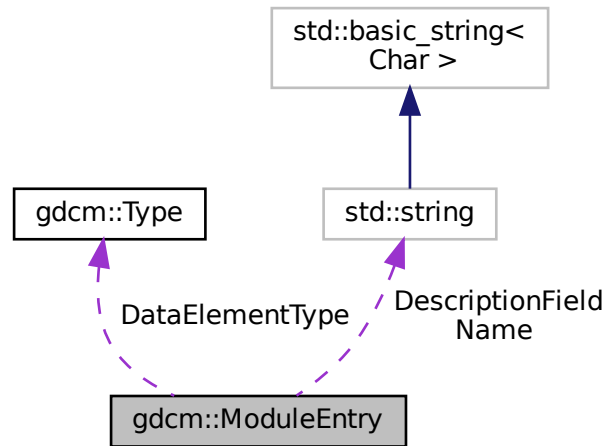
Class for representing a [ModuleEntry](#).

```
#include <gdcmModuleEntry.h>
```

Inheritance diagram for gdcm::ModuleEntry:



Collaboration diagram for `gdcmmoduleentry`:



## 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)

### 10.190.1 Detailed Description

Class for representing a [ModuleEntry](#).

#### Note

bla

#### See also

[DictEntry](#)

### 10.190.2 Member Typedef Documentation

#### 10.190.2.1 Description

```
typedef std::string gdcmm::ModuleEntry::Description
```

### 10.190.3 Constructor & Destructor Documentation

#### 10.190.3.1 ModuleEntry()

```
gdcmm::ModuleEntry::ModuleEntry (
    const char * name = "",
    const char * type = "3",
    const char * description = "" ) [inline]
```

References [gdcmm::Type::GetTypeType\(\)](#).

#### 10.190.3.2 ~ModuleEntry()

```
virtual gdcmm::ModuleEntry::~~ModuleEntry ( ) [virtual], [default]
```

### 10.190.4 Member Function Documentation

#### 10.190.4.1 GetDescription()

```
const Description& gdcmm::ModuleEntry::GetDescription ( ) const [inline]
```

#### 10.190.4.2 GetName()

```
const char* gdcmm::ModuleEntry::GetName ( ) const [inline]
```

#### 10.190.4.3 GetType()

```
const Type& gdcmm::ModuleEntry::GetType ( ) const [inline]
```

#### 10.190.4.4 SetDescription()

```
void gdcmm::ModuleEntry::SetDescription (  
    const char * d ) [inline]
```

#### 10.190.4.5 SetName()

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

#### 10.190.4.6 SetType()

```
void gdcmm::ModuleEntry::SetType (  
    const Type & type ) [inline]
```

### 10.190.5 Friends And Related Function Documentation



### 10.190.5.1 operator<<

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

## 10.190.6 Member Data Documentation

### 10.190.6.1 DataElementType

```
Type gdcm::ModuleEntry::DataElementType [protected]
```

### 10.190.6.2 DescriptionField

```
Description gdcm::ModuleEntry::DescriptionField [protected]
```

### 10.190.6.3 Name

```
std::string gdcm::ModuleEntry::Name [protected]
```

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

- [gdcmModuleEntry.h](#)

## 10.191 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)

### 10.191.1 Detailed Description

Class for representing a [Modules](#).

Note

bla

See also

[Module](#)

### 10.191.2 Member Typedef Documentation

#### 10.191.2.1 ModuleMapType

```
typedef std::map<std::string, Module> gdcm::Modules::ModuleMapType
```

### 10.191.3 Constructor & Destructor Documentation

#### 10.191.3.1 Modules()

```
gdcm::Modules::Modules ( ) [default]
```

## 10.191.4 Member Function Documentation

### 10.191.4.1 AddModule()

```
void gdcm::Modules::AddModule (
    const char * ref,
    const Module & module ) [inline]
```

### 10.191.4.2 Clear()

```
void gdcm::Modules::Clear ( ) [inline]
```

### 10.191.4.3 GetModule()

```
const Module& gdcm::Modules::GetModule (
    const char * name ) const [inline]
```

### 10.191.4.4 IsEmpty()

```
bool gdcm::Modules::IsEmpty ( ) const [inline]
```

## 10.191.5 Friends And Related Function Documentation

### 10.191.5.1 operator<<

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

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

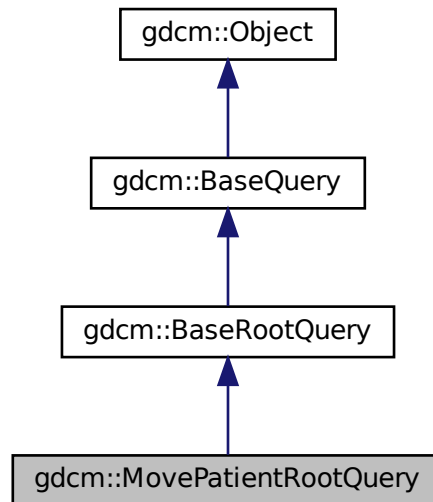
- [gdcmModules.h](#)

## 10.192 gdcm::MovePatientRootQuery Class Reference

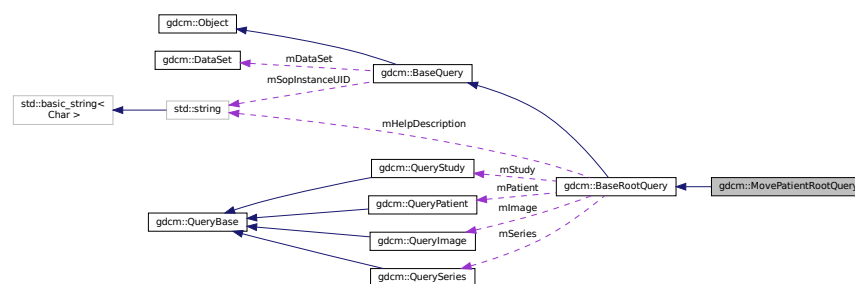
[MovePatientRootQuery](#).

```
#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

## Friends

- class [QueryFactory](#)

## Additional Inherited Members

### 10.192.1 Detailed Description

[MovePatientRootQuery](#).

contains: the class which will produce a dataset for c-move with patient root

### 10.192.2 Constructor & Destructor Documentation

#### 10.192.2.1 MovePatientRootQuery()

```
gdcm::MovePatientRootQuery::MovePatientRootQuery ( )
```

### 10.192.3 Member Function Documentation

#### 10.192.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::MovePatientRootQuery::GetAbstractSyntaxUID ( ) const [override], [virtual]
```

Implements [gdcm::BaseQuery](#).

#### 10.192.3.2 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](#).

### 10.192.3.3 InitializeDataSet()

```
void gdcmm::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 [gdcmm::BaseRootQuery](#).

### 10.192.3.4 ValidateQuery()

```
bool gdcmm::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 [gdcmm::BaseRootQuery](#).

## 10.192.4 Friends And Related Function Documentation

### 10.192.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

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

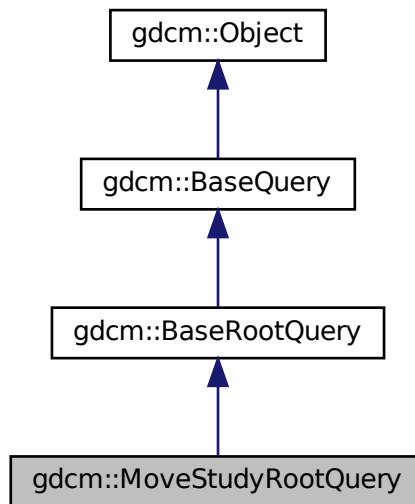
- [gdcmmMovePatientRootQuery.h](#)

## 10.193 gdcm::MoveStudyRootQuery Class Reference

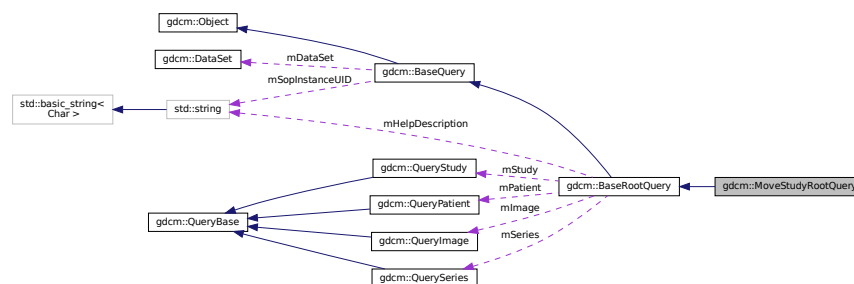
[MoveStudyRootQuery](#).

```
#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

## Friends

- class [QueryFactory](#)

## Additional Inherited Members

### 10.193.1 Detailed Description

[MoveStudyRootQuery](#).

contains: the class which will produce a dataset for C-MOVE with study root

### 10.193.2 Constructor & Destructor Documentation

#### 10.193.2.1 MoveStudyRootQuery()

```
gdcm::MoveStudyRootQuery::MoveStudyRootQuery ( )
```

### 10.193.3 Member Function Documentation

#### 10.193.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::MoveStudyRootQuery::GetAbstractSyntaxUID ( ) const [override], [virtual]
```

Implements [gdcm::BaseQuery](#).

#### 10.193.3.2 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](#).



### 10.193.3.3 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](#).

### 10.193.3.4 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](#).

## 10.193.4 Friends And Related Function Documentation

### 10.193.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

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

- [gdcmMoveStudyRootQuery.h](#)

## 10.194 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)

### 10.194.1 Detailed Description

Class for [MrProtocol](#).

#### Examples

[MrProtocol.cxx](#).

### 10.194.2 Constructor & Destructor Documentation

#### 10.194.2.1 MrProtocol()

```
gdcmm::MrProtocol::MrProtocol ( )
```

#### 10.194.2.2 ~MrProtocol()

```
gdcmm::MrProtocol::~~MrProtocol ( )
```

### 10.194.3 Member Function Documentation

#### 10.194.3.1 FindMrProtocolByName()

```
bool gdcm::MrProtocol::FindMrProtocolByName (
    const char * name ) const
```

#### 10.194.3.2 GetMrProtocolByName()

```
const char* gdcm::MrProtocol::GetMrProtocolByName (
    const char * name ) const
```

#### 10.194.3.3 GetSliceArray()

```
bool gdcm::MrProtocol::GetSliceArray (
    MrProtocol::SliceArray & sa ) const
```

#### 10.194.3.4 GetVersion()

```
int gdcm::MrProtocol::GetVersion ( ) const
```

#### 10.194.3.5 Load()

```
bool gdcm::MrProtocol::Load (
    const ByteValue * bv,
    const char * str,
    int version )
```

#### 10.194.3.6 Print()

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

## 10.194.4 Friends And Related Function Documentation

### 10.194.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const MrProtocol & d ) [friend]
```

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

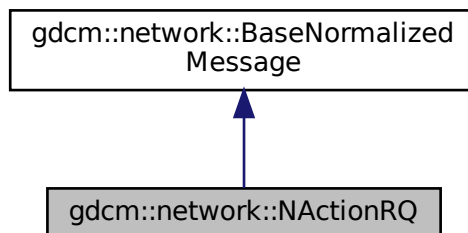
- [gdcmMrProtocol.h](#)

## 10.195 gdcm::network::NActionRQ Class Reference

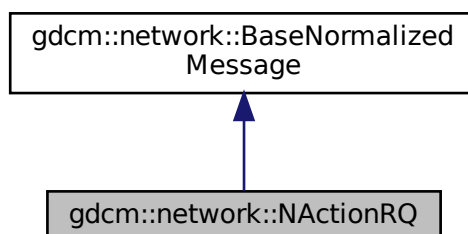
[NActionRQ](#).

```
#include <gdcmNActionMessages.h>
```

Inheritance diagram for gdcm::network::NActionRQ:



Collaboration diagram for gdcm::network::NActionRQ:



## Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseQuery](#) \*inQuery) override

### 10.195.1 Detailed Description

[NActionRQ](#).

this file defines the messages for the NAction action

### 10.195.2 Member Function Documentation

#### 10.195.2.1 ConstructPDV()

```
std::vector<PresentationDataValue> gdcm::network::NActionRQ::ConstructPDV (  
    const ULConnection & inConnection,  
    const BaseQuery * inQuery ) [override], [virtual]
```

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

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

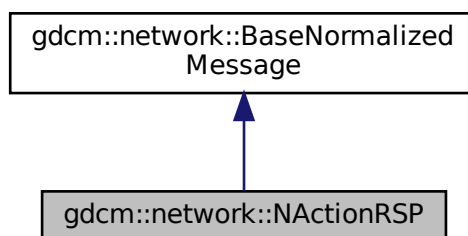
- [gdcmNActionMessages.h](#)

## 10.196 gdcm::network::NActionRSP Class Reference

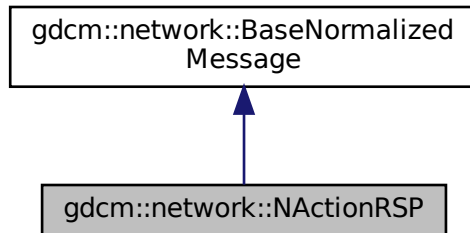
[NActionRSP](#) this file defines the messages for the NAction action.

```
#include <gdcmNActionMessages.h>
```

Inheritance diagram for `gdcm::network::NActionRSP`:



Collaboration diagram for `gdcm::network::NActionRSP`:



## Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet` (const `DataSet` \*inDataSet)

### 10.196.1 Detailed Description

[NActionRSP](#) this file defines the messages for the NAction action.

### 10.196.2 Member Function Documentation

#### 10.196.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcm::network::NActionRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

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

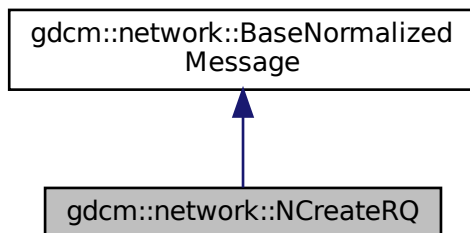
- [gdcmNActionMessages.h](#)

## 10.197 gdcm::network::NCreateRQ Class Reference

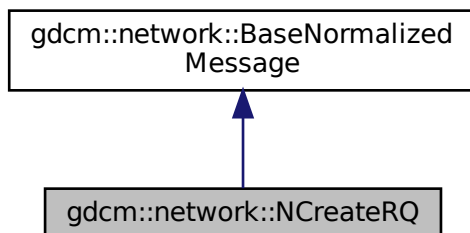
[NCreateRQ](#).

```
#include <gdcmNCreateMessages.h>
```

Inheritance diagram for gdcm::network::NCreateRQ:



Collaboration diagram for gdcm::network::NCreateRQ:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseQuery](#) \*inQuery) override

### 10.197.1 Detailed Description

[NCreateRQ](#).

this file defines the messages for the ncreate action

## 10.197.2 Member Function Documentation

### 10.197.2.1 ConstructPDV()

```
std::vector<PresentationDataValue> gdcn::network::NCreateRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [override], [virtual]
```

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

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

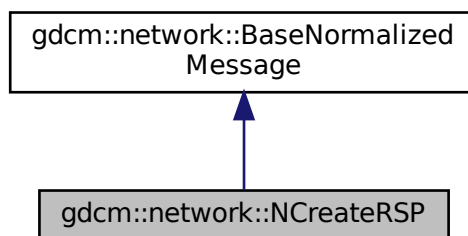
- [gdcnNCreateMessages.h](#)

## 10.198 gdcn::network::NCreateRSP Class Reference

[NCreateRSP](#) this file defines the messages for the ncreate action.

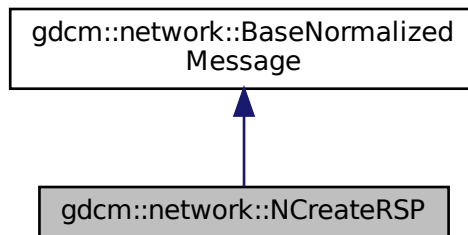
```
#include <gdcnNCreateMessages.h>
```

Inheritance diagram for gdcn::network::NCreateRSP:





Collaboration diagram for gdcm::network::NCreateRSP:



## Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)`

### 10.198.1 Detailed Description

[NCreateRSP](#) this file defines the messages for the ncreate action.

### 10.198.2 Member Function Documentation

#### 10.198.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcm::network::NCreateRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

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

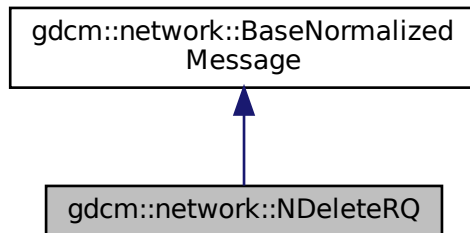
- [gdcmNCreateMessages.h](#)

## 10.199 gdcmm::network::NDeleteRQ Class Reference

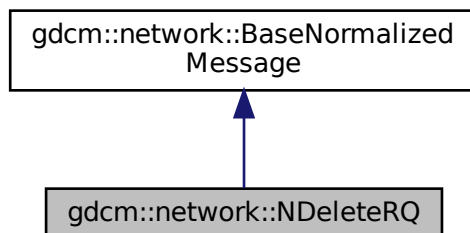
[NDeleteRQ](#).

```
#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

### 10.199.1 Detailed Description

[NDeleteRQ](#).

this file defines the messages for the ndelete action

## 10.199.2 Member Function Documentation

### 10.199.2.1 ConstructPDV()

```
std::vector<PresentationDataValue> gdcm::network::NDeleteRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [override], [virtual]
```

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

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

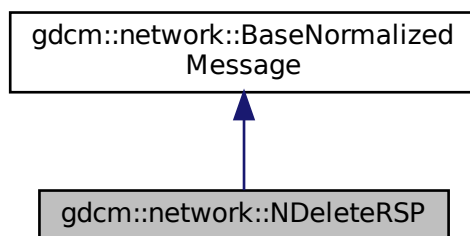
- [gdcmNDeleteMessages.h](#)

## 10.200 gdcm::network::NDeleteRSP Class Reference

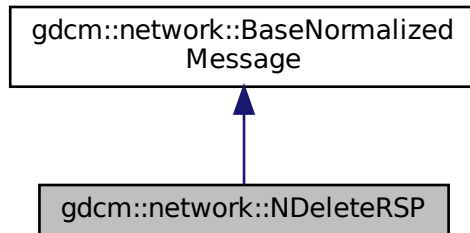
[NDeleteRSP](#) this file defines the messages for the ndelete action.

```
#include <gdcmNDeleteMessages.h>
```

Inheritance diagram for `gdcm::network::NDeleteRSP`:



Collaboration diagram for `gdcm::network::NDeleteRSP`:



## Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet` (`const DataSet *inDataSet`)

### 10.200.1 Detailed Description

[NDeleteRSP](#) this file defines the messages for the ndelete action.

### 10.200.2 Member Function Documentation

#### 10.200.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcm::network::NDeleteRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

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

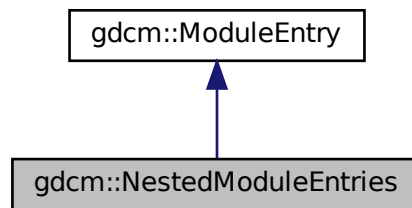
- [gdcmNDeleteMessages.h](#)

## 10.201 gdcm::NestedModuleEntries Class Reference

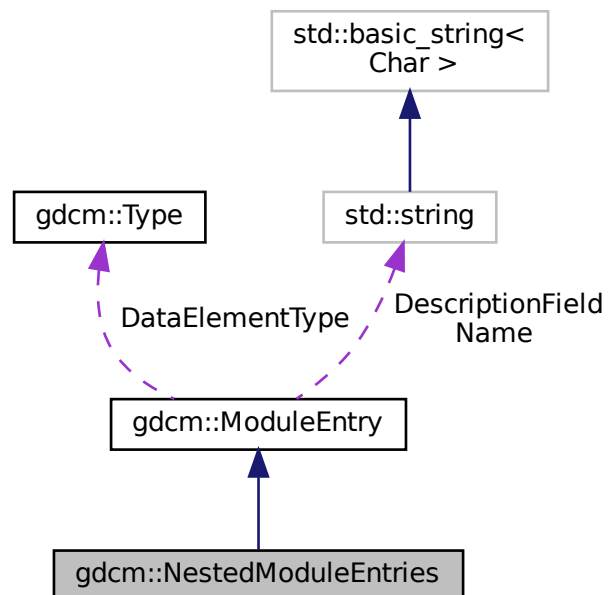
Class for representing a [NestedModuleEntries](#).

```
#include <gdcmNestedModuleEntries.h>
```

Inheritance diagram for gdcm::NestedModuleEntries:



Collaboration diagram for gdcm::NestedModuleEntries:



## Public Types

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

## Public 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](#) ()

## Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [NestedModuleEntries](#) &\_val)

## Additional Inherited Members

### 10.201.1 Detailed Description

Class for representing a [NestedModuleEntries](#).

#### Note

bla

#### See also

[ModuleEntry](#)

### 10.201.2 Member Typedef Documentation

#### 10.201.2.1 SizeType

```
typedef std::vector<ModuleEntry>::size_type gdcn::NestedModuleEntries::SizeType
```

### 10.201.3 Constructor & Destructor Documentation

### 10.201.3.1 NestedModuleEntries()

```
gdcmm::NestedModuleEntries::NestedModuleEntries (
    const char * name = "",
    const char * type = "3",
    const char * description = "" ) [inline]
```

## 10.201.4 Member Function Documentation

### 10.201.4.1 AddModuleEntry()

```
void gdcmm::NestedModuleEntries::AddModuleEntry (
    const ModuleEntry & me ) [inline]
```

### 10.201.4.2 GetModuleEntry() [1/2]

```
ModuleEntry& gdcmm::NestedModuleEntries::GetModuleEntry (
    SizeType idx ) [inline]
```

### 10.201.4.3 GetModuleEntry() [2/2]

```
const ModuleEntry& gdcmm::NestedModuleEntries::GetModuleEntry (
    SizeType idx ) const [inline]
```

### 10.201.4.4 GetNumberOfModuleEntries()

```
SizeType gdcmm::NestedModuleEntries::GetNumberOfModuleEntries ( ) [inline]
```

## 10.201.5 Friends And Related Function Documentation

### 10.201.5.1 operator<<

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

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

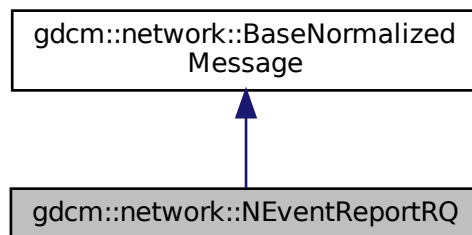
- [gdcmNestedModuleEntries.h](#)

## 10.202 gdcm::network::NEventReportRQ Class Reference

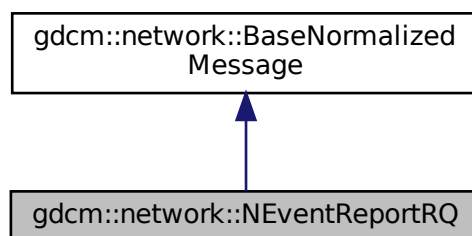
[NEventReportRQ](#).

```
#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

### 10.202.1 Detailed Description

[NEventReportRQ](#).

this file defines the messages for the neventreport action

### 10.202.2 Member Function Documentation

#### 10.202.2.1 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:

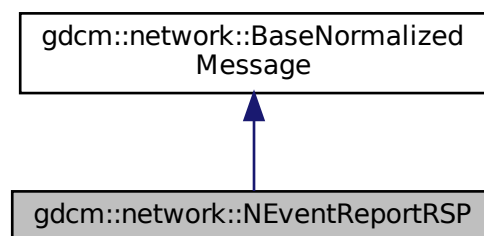
- [gdcmNEventReportMessages.h](#)

## 10.203 gdcm::network::NEventReportRSP Class Reference

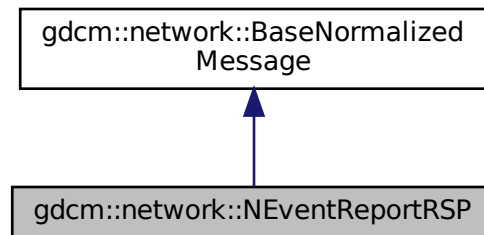
[NEventReportRSP](#) this file defines the messages for the neventreport action.

```
#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`)

### 10.203.1 Detailed Description

[NEventReportRSP](#) this file defines the messages for the neventreport action.

### 10.203.2 Member Function Documentation

#### 10.203.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcm::network::NEventReportRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

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

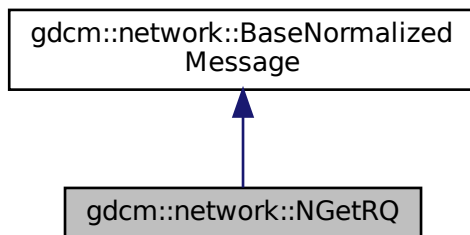
- [gdcmNEventReportMessages.h](#)

## 10.204 gdcm::network::NGetRQ Class Reference

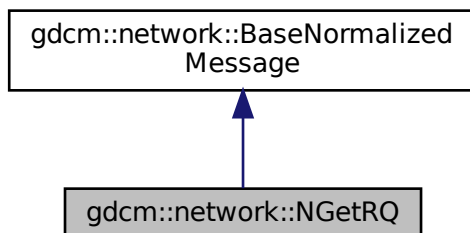
[NGetRQ](#).

```
#include <gdcmNGetMessages.h>
```

Inheritance diagram for gdcm::network::NGetRQ:



Collaboration diagram for gdcm::network::NGetRQ:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseQuery](#) \*inQuery) override

### 10.204.1 Detailed Description

[NGetRQ](#).

this file defines the messages for the nget action

## 10.204.2 Member Function Documentation

### 10.204.2.1 ConstructPDV()

```
std::vector<PresentationDataValue> gdcM::network::NGetRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [override], [virtual]
```

Implements [gdcM::network::BaseNormalizedMessage](#).

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

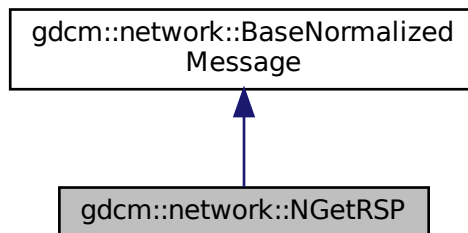
- [gdcMNGetMessages.h](#)

## 10.205 gdcM::network::NGetRSP Class Reference

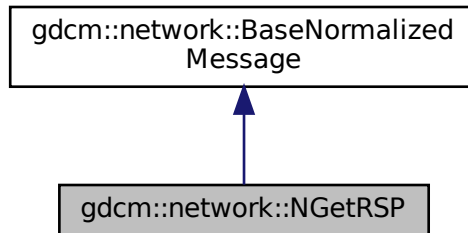
[NGetRSP](#) this file defines the messages for the nget action.

```
#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)`

### 10.205.1 Detailed Description

[NGetRSP](#) this file defines the messages for the nget action.

### 10.205.2 Member Function Documentation

#### 10.205.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcm::network::NGetRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

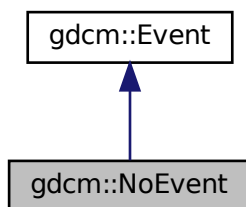
The documentation for this class was generated from the following file:

- [gdcmNGetMessages.h](#)

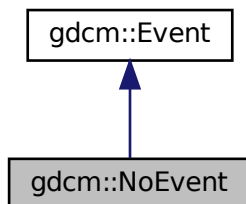
## 10.206 gdcm::NoEvent Class Reference

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

Inheritance diagram for gdcm::NoEvent:



Collaboration diagram for gdcm::NoEvent:



### Additional Inherited Members

#### 10.206.1 Detailed Description

Define some common GDCM events

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

- [gdcmEvent.h](#)

## 10.207 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)

### 10.207.1 Member Function Documentation

#### 10.207.1.1 ConstructNAction()

```
static std::vector<PresentationDataValue> gdcm::network::NormalizedMessageFactory::Construct←
NAction (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

#### 10.207.1.2 ConstructNCreate()

```
static std::vector<PresentationDataValue> gdcm::network::NormalizedMessageFactory::Construct←
NCreate (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

### 10.207.1.3 ConstructNDelete()

```
static std::vector<PresentationDataValue> gdcM::network::NormalizedMessageFactory::ConstructNDelete (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

### 10.207.1.4 ConstructNEventReport()

```
static std::vector<PresentationDataValue> gdcM::network::NormalizedMessageFactory::ConstructNEventReport (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

### 10.207.1.5 ConstructNGet()

```
static std::vector<PresentationDataValue> gdcM::network::NormalizedMessageFactory::ConstructNGet (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

### 10.207.1.6 ConstructNSet()

```
static std::vector<PresentationDataValue> gdcM::network::NormalizedMessageFactory::ConstructNSet (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

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

- [gdcMNormalizedMessageFactory.h](#)

## 10.208 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)

### 10.208.1 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

### 10.208.2 Member Function Documentation

#### 10.208.2.1 ConstructQuery()

```
static BaseQuery* gdcm::NormalizedNetworkFunctions::ConstructQuery (
    const std::string & sopInstanceUID,
    const DataSet & queryds,
    ENQueryType queryType = eCreateMMPS ) [static]
```

### 10.208.2.2 NAction()

```
static bool gdcm::NormalizedNetworkFunctions::NAction (
    const char * remote,
    uint16_t portno,
    const BaseQuery * query,
    std::vector< DataSet > & retDataSets,
    const char * aetitle,
    const char * call ) [static]
```

### 10.208.2.3 NCreate()

```
static bool gdcm::NormalizedNetworkFunctions::NCreate (
    const char * remote,
    uint16_t portno,
    BaseQuery * query,
    std::vector< DataSet > & retDataSets,
    const char * aetitle,
    const char * call ) [static]
```

### 10.208.2.4 NDelete()

```
static bool gdcm::NormalizedNetworkFunctions::NDelete (
    const char * remote,
    uint16_t portno,
    const BaseQuery * query,
    std::vector< DataSet > & retDataSets,
    const char * aetitle,
    const char * call ) [static]
```

### 10.208.2.5 NEventReport()

```
static bool gdcm::NormalizedNetworkFunctions::NEventReport (
    const char * remote,
    uint16_t portno,
    const BaseQuery * query,
    std::vector< DataSet > & retDataSets,
    const char * aetitle,
    const char * call ) [static]
```

### 10.208.2.6 NGet()

```
static bool gdcm::NormalizedNetworkFunctions::NGet (
    const char * remote,
    uint16_t portno,
    const BaseQuery * query,
    std::vector< DataSet > & retDataSets,
    const char * aetitle,
    const char * call ) [static]
```

### 10.208.2.7 NSet()

```
static bool gdcm::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:

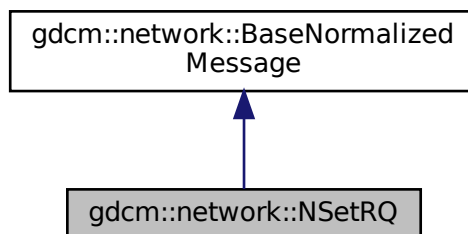
- [gdcmNormalizedNetworkFunctions.h](#)

## 10.209 gdcm::network::NSetRQ Class Reference

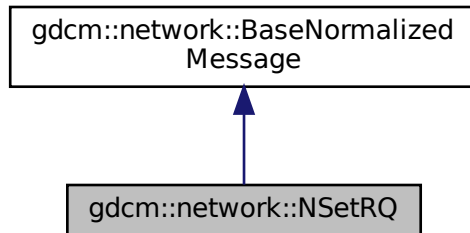
[NSetRQ](#).

```
#include <gdcmNSetMessages.h>
```

Inheritance diagram for gdcm::network::NSetRQ:



Collaboration diagram for `gdcm::network::NSetRQ`:



## Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &`inConnection`, const [BaseQuery](#) \*`inQuery`) override

### 10.209.1 Detailed Description

[NSetRQ](#).

this file defines the messages for the nset action

### 10.209.2 Member Function Documentation

#### 10.209.2.1 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:

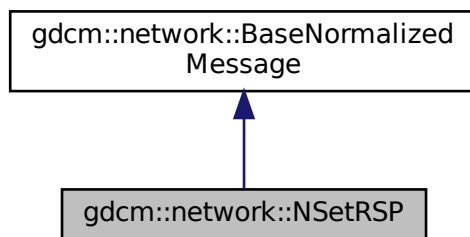
- [gdcmNSetMessages.h](#)

## 10.210 gdcm::network::NSetRSP Class Reference

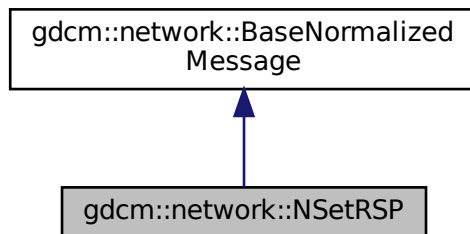
[NSetRSP](#) this file defines the messages for the nset action.

```
#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)`

#### 10.210.1 Detailed Description

[NSetRSP](#) this file defines the messages for the nset action.

## 10.210.2 Member Function Documentation

### 10.210.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcmm::network::NSetRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

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

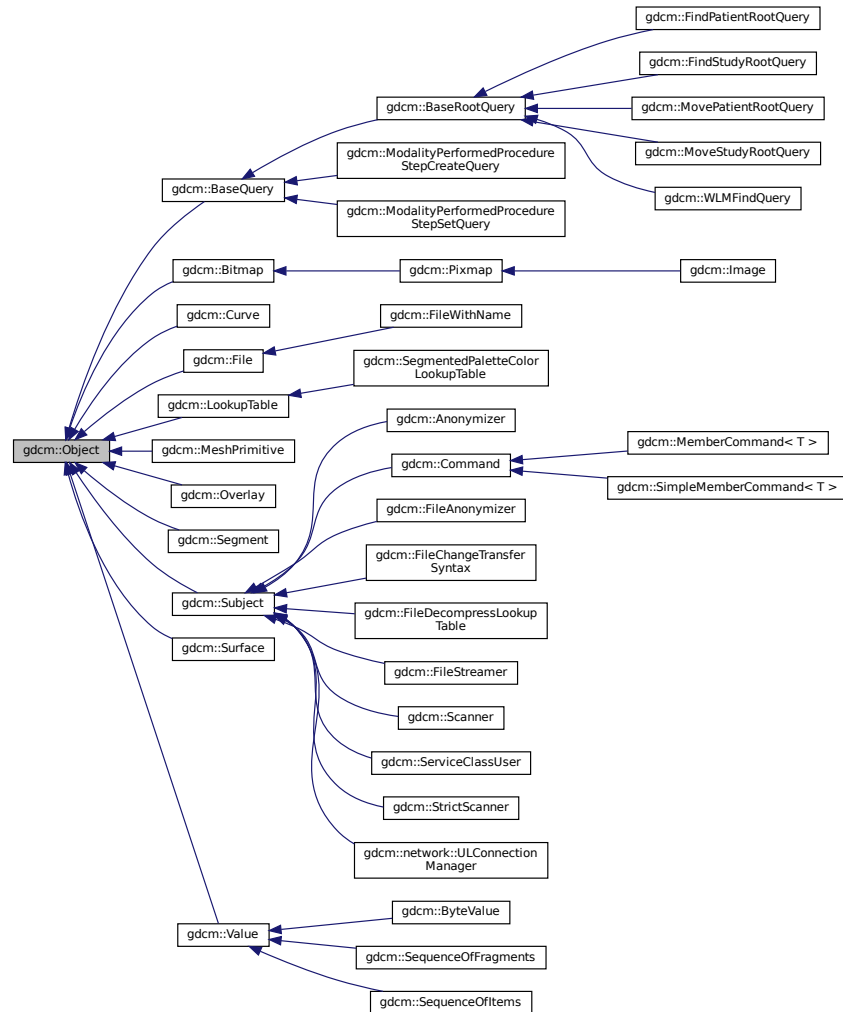
- [gdcmmNSetMessages.h](#)

## 10.211 gdcmm::Object Class Reference

[Object.](#)

```
#include <gdcmmObject.h>
```

Inheritance diagram for gdcmm::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`

### 10.211.1 Detailed Description

[Object](#).

#### Note

main superclass for object that want to use [SmartPointer](#) invasive ref counting system

#### See also

[SmartPointer](#)

### 10.211.2 Constructor & Destructor Documentation

#### 10.211.2.1 [Object\(\)](#) [1/2]

```
gdcmm::Object::Object ( ) [inline]
```

#### 10.211.2.2 [~Object\(\)](#)

```
virtual gdcmm::Object::~~Object ( ) [inline], [virtual]
```

#### 10.211.2.3 [Object\(\)](#) [2/2]

```
gdcmm::Object::Object (
    const Object & ) [inline]
```

Special requirement for copy/cstor, assignment operator.



### 10.211.3 Member Function Documentation

#### 10.211.3.1 operator=()

```
void gdcm::Object::operator= (
    const Object & ) [inline]
```

#### 10.211.3.2 Print()

```
virtual void gdcm::Object::Print (
    std::ostream & ) const [inline], [virtual]
```

Reimplemented in [gdcm::BaseQuery](#), [gdcm::StrictScanner](#), [gdcm::Scanner](#), [gdcm::Image](#), [gdcm::SequenceOfItems](#), [gdcm::SequenceOfFragments](#), [gdcm::ByteValue](#), [gdcm::SegmentedPaletteColorLookupTable](#), [gdcm::Pixmap](#), [gdcm::Overlay](#), [gdcm::LookupTable](#), [gdcm::Curve](#), and [gdcm::Bitmap](#).

#### Examples

[ReadAndDumpDICOMDIR.cxx](#).

#### 10.211.3.3 Register()

```
void gdcm::Object::Register ( ) [inline], [protected]
```

#### 10.211.3.4 UnRegister()

```
void gdcm::Object::UnRegister ( ) [inline], [protected]
```

### 10.211.4 Friends And Related Function Documentation

#### 10.211.4.1 operator<<

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

#### 10.211.4.2 SmartPointer

```
template<class ObjectType >
friend class SmartPointer [friend]
```

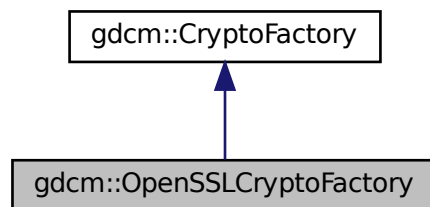
The documentation for this class was generated from the following file:

- [gdcmmObject.h](#)

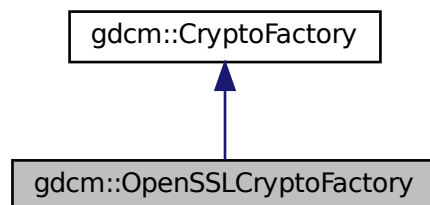
### 10.212 gdcmm::OpenSSLCryptoFactory Class Reference

```
#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](#) ()

## Additional Inherited Members

### 10.212.1 Constructor & Destructor Documentation

#### 10.212.1.1 OpenSSLCryptoFactory()

```
gdcm::OpenSSLCryptoFactory::OpenSSLCryptoFactory (
    CryptoLib id ) [inline]
```

References [gdcmDebugMacro](#).

### 10.212.2 Member Function Documentation

#### 10.212.2.1 CreateCMSProvider()

```
CryptographicMessageSyntax* gdcm::OpenSSLCryptoFactory::CreateCMSProvider ( ) [inline], [virtual]
```

Implements [gdcm::CryptoFactory](#).

#### 10.212.2.2 InitOpenSSL()

```
void gdcm::OpenSSLCryptoFactory::InitOpenSSL ( ) [protected]
```

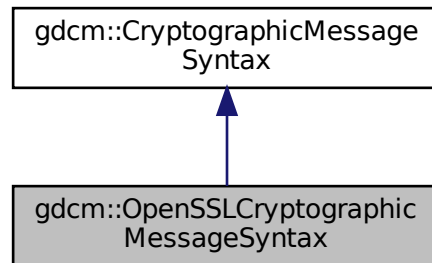
The documentation for this class was generated from the following file:

- [gdcmOpenSSLCryptoFactory.h](#)

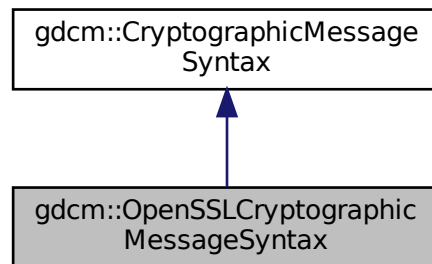
## 10.213 gdcm::OpenSSLCryptographicMessageSyntax Class Reference

```
#include <gdcmOpenSSLCryptographicMessageSyntax.h>
```

Inheritance diagram for gdcm::OpenSSLCryptographicMessageSyntax:



Collaboration diagram for gdcm::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)

## Additional Inherited Members

### 10.213.1 Constructor & Destructor Documentation

#### 10.213.1.1 OpenSSLCryptographicMessageSyntax()

```
gdcmm::OpenSSLCryptographicMessageSyntax::OpenSSLCryptographicMessageSyntax ( )
```

#### 10.213.1.2 ~OpenSSLCryptographicMessageSyntax()

```
gdcmm::OpenSSLCryptographicMessageSyntax::~~OpenSSLCryptographicMessageSyntax ( )
```

### 10.213.2 Member Function Documentation

#### 10.213.2.1 Decrypt()

```
bool gdcmm::OpenSSLCryptographicMessageSyntax::Decrypt (
    char * output,
    size_t & outlen,
    const char * array,
    size_t len ) const [virtual]
```

decrypt content from a PKCS#7 envelopedData structure

Implements [gdcmm::CryptographicMessageSyntax](#).

### 10.213.2.2 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](#).

### 10.213.2.3 GetCipherType()

```
CipherTypes gdcM::OpenSSLCryptographicMessageSyntax::GetCipherType ( ) const [virtual]
```

Implements [gdcM::CryptographicMessageSyntax](#).

### 10.213.2.4 ParseCertificateFile()

```
bool gdcM::OpenSSLCryptographicMessageSyntax::ParseCertificateFile (
    const char * filename ) [virtual]
```

Implements [gdcM::CryptographicMessageSyntax](#).

### 10.213.2.5 ParseKeyFile()

```
bool gdcM::OpenSSLCryptographicMessageSyntax::ParseKeyFile (
    const char * filename ) [virtual]
```

Implements [gdcM::CryptographicMessageSyntax](#).

### 10.213.2.6 SetCipherType()

```
void gdcM::OpenSSLCryptographicMessageSyntax::SetCipherType (
    CipherTypes type ) [virtual]
```

Set Cipher [Type](#). Default is: AES256\_CIPHER

Implements [gdcM::CryptographicMessageSyntax](#).

### 10.213.2.7 SetPassword()

```
bool gdcm::OpenSSLCryptographicMessageSyntax::SetPassword (
    const char * pass,
    size_t passLen ) [virtual]
```

Implements [gdcm::CryptographicMessageSyntax](#).

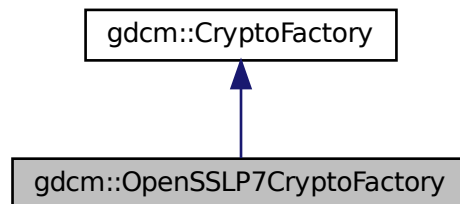
The documentation for this class was generated from the following file:

- [gdcmOpenSSLCryptographicMessageSyntax.h](#)

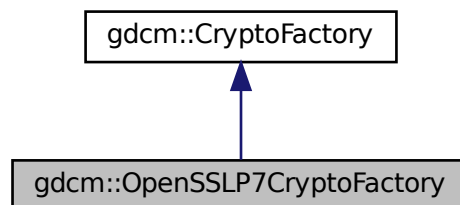
## 10.214 gdcm::OpenSSLP7CryptoFactory Class Reference

```
#include <gdcmOpenSSLP7CryptoFactory.h>
```

Inheritance diagram for gdcm::OpenSSLP7CryptoFactory:



Collaboration diagram for gdcm::OpenSSLP7CryptoFactory:



## Public Member Functions

- [OpenSSLP7CryptoFactory](#) ([CryptoLib](#) id)
- [CryptographicMessageSyntax](#) \* [CreateCMSProvider](#) ()

## Additional Inherited Members

### 10.214.1 Constructor & Destructor Documentation

#### 10.214.1.1 OpenSSLP7CryptoFactory()

```
gdcmm::OpenSSLP7CryptoFactory::OpenSSLP7CryptoFactory (  
    CryptoLib id ) [inline]
```

References [gdcmmDebugMacro](#).

### 10.214.2 Member Function Documentation

#### 10.214.2.1 CreateCMSProvider()

```
CryptographicMessageSyntax* gdcmm::OpenSSLP7CryptoFactory::CreateCMSProvider ( ) [inline], [virtual]
```

Implements [gdcmm::CryptoFactory](#).

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

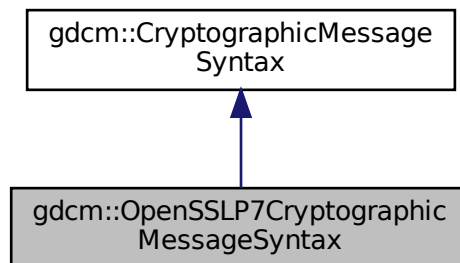
- [gdcmmOpenSSLP7CryptoFactory.h](#)



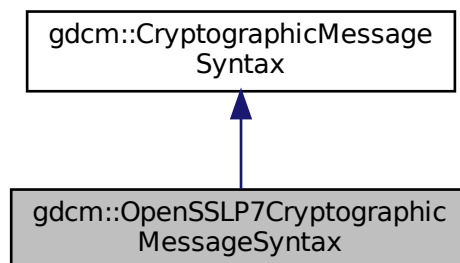
## 10.215 gdcM::OpenSSL7CryptographicMessageSyntax Class Reference

```
#include <gdcMOpenSSL7CryptographicMessageSyntax.h>
```

Inheritance diagram for gdcM::OpenSSL7CryptographicMessageSyntax:



Collaboration diagram for gdcM::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)

## Additional Inherited Members

### 10.215.1 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)

### 10.215.2 Constructor & Destructor Documentation

#### 10.215.2.1 OpenSSLP7CryptographicMessageSyntax()

```
gdcM::OpenSSLP7CryptographicMessageSyntax::OpenSSLP7CryptographicMessageSyntax ( )
```

#### 10.215.2.2 ~OpenSSLP7CryptographicMessageSyntax()

```
gdcM::OpenSSLP7CryptographicMessageSyntax::~~OpenSSLP7CryptographicMessageSyntax ( )
```

### 10.215.3 Member Function Documentation

#### 10.215.3.1 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](#).

### 10.215.3.2 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](#).

### 10.215.3.3 GetCipherType()

```
CipherTypes gdcmm::OpenSSLP7CryptographicMessageSyntax::GetCipherType ( ) const [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

### 10.215.3.4 ParseCertificateFile()

```
bool gdcmm::OpenSSLP7CryptographicMessageSyntax::ParseCertificateFile (
    const char * filename ) [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

### 10.215.3.5 ParseKeyFile()

```
bool gdcmm::OpenSSLP7CryptographicMessageSyntax::ParseKeyFile (
    const char * filename ) [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

### 10.215.3.6 SetCipherType()

```
void gdcmm::OpenSSLP7CryptographicMessageSyntax::SetCipherType (
    CipherTypes type ) [virtual]
```

Set Cipher [Type](#). Default is: AES256\_CIPHER

Implements [gdcmm::CryptographicMessageSyntax](#).

### 10.215.3.7 SetPassword()

```
bool gdcmm::OpenSSL7CryptographicMessageSyntax::SetPassword (
    const char * ,
    size_t ) [inline], [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

References [gdcmmWarningMacro](#).

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

- [gdcmmOpenSSL7CryptographicMessageSyntax.h](#)

## 10.216 gdcmm::Orientation Class Reference

class to handle [Orientation](#)

```
#include <gdcmmOrientation.h>
```

### Public Types

- enum [OrientationType](#) {  
[UNKNOWN](#) ,  
[AXIAL](#) ,  
[CORONAL](#) ,  
[SAGITTAL](#) ,  
[OBLIQUE](#) }

### Public Member Functions

- [Orientation](#) ()
- [~Orientation](#) ()
- 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)

### 10.216.1 Detailed Description

class to handle [Orientation](#)

### 10.216.2 Member Enumeration Documentation

#### 10.216.2.1 OrientationType

enum [gdcm::Orientation::OrientationType](#)

Enumerator

UNKNOWN	
AXIAL	
CORONAL	
SAGITTAL	
OBLIQUE	

### 10.216.3 Constructor & Destructor Documentation

#### 10.216.3.1 Orientation()

[gdcm::Orientation::Orientation](#) ( )

#### 10.216.3.2 ~Orientation()

[gdcm::Orientation::~~Orientation](#) ( )

## 10.216.4 Member Function Documentation

### 10.216.4.1 GetLabel()

```
static const char* gdcm::Orientation::GetLabel (
    OrientationType type ) [static]
```

Return the label of an [Orientation](#).

#### Examples

[FixOrientation.cxx](#).

### 10.216.4.2 GetMajorAxisFromPatientRelativeDirectionCosine()

```
static char gdcm::Orientation::GetMajorAxisFromPatientRelativeDirectionCosine (
    double x,
    double y,
    double z ) [static], [protected]
```

### 10.216.4.3 GetObliquityThresholdCosineValue()

```
static double gdcm::Orientation::GetObliquityThresholdCosineValue ( ) [static]
```

### 10.216.4.4 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](#).

#### 10.216.4.5 Print()

```
void gdcm::Orientation::Print (
    std::ostream & ) const
```

Print.

#### 10.216.4.6 SetObliquityThresholdCosineValue()

```
static void gdcm::Orientation::SetObliquityThresholdCosineValue (
    double val ) [static]
```

ObliquityThresholdCosineValue stuff.

### 10.216.5 Friends And Related Function Documentation

#### 10.216.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const Orientation & o ) [friend]
```

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

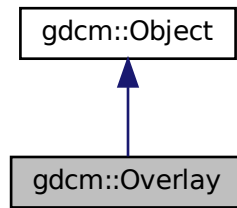
- [gdcmOrientation.h](#)

## 10.217 gdcm::Overlay Class Reference

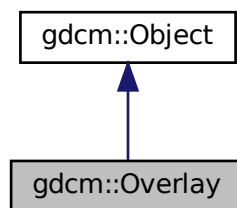
[Overlay](#) class.

```
#include <gdcmOverlay.h>
```

Inheritance diagram for `gdcm::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:*

## Static Public Member Functions

- static const char \* [GetOverlayTypeAsString](#) ([OverlayType](#) ot)
- static [OverlayType](#) [GetOverlayTypeFromString](#) (const char \*)

## Additional Inherited Members

### 10.217.1 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:

### 10.217.2 Member Enumeration Documentation

#### 10.217.2.1 OverlayType

```
enum gdcm::Overlay::OverlayType
```

#### Enumerator

Invalid	
Graphics	
ROI	

## 10.217.3 Constructor & Destructor Documentation

### 10.217.3.1 Overlay() [1/2]

```
gdcm::Overlay::Overlay ( )
```

### 10.217.3.2 ~Overlay()

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

### 10.217.3.3 Overlay() [2/2]

```
gdcm::Overlay::Overlay (
    Overlay const & ov )
```

## 10.217.4 Member Function Documentation

### 10.217.4.1 Decompress()

```
void gdcm::Overlay::Decompress (
    std::ostream & os ) const
```

Decode the internal OverlayData (packed bits) into unpacked representation.

### 10.217.4.2 GetBitPosition()

```
unsigned short gdcm::Overlay::GetBitPosition ( ) const
```

return bit position

#### 10.217.4.3 GetBitsAllocated()

```
unsigned short gdcm::Overlay::GetBitsAllocated ( ) const
```

return bits allocated

#### 10.217.4.4 GetColumns()

```
unsigned short gdcm::Overlay::GetColumns ( ) const
```

get columns

#### 10.217.4.5 GetDescription()

```
const char* gdcm::Overlay::GetDescription ( ) const
```

get description

#### 10.217.4.6 GetGroup()

```
unsigned short gdcm::Overlay::GetGroup ( ) const
```

Get Group number.

#### 10.217.4.7 GetOrigin()

```
const signed short* gdcm::Overlay::GetOrigin ( ) const
```

get origin

#### 10.217.4.8 GetOverlayData()

```
const ByteValue& gdcm::Overlay::GetOverlayData ( ) const
```

Return the [Overlay](#) Data as [ByteValue](#): Not thread safe

#### 10.217.4.9 GetOverlayTypeAsString()

```
static const char* gdcm::Overlay::GetOverlayTypeAsString (
    OverlayType ot ) [static]
```

#### 10.217.4.10 GetOverlayTypeFromString()

```
static OverlayType gdcm::Overlay::GetOverlayTypeFromString (
    const char * ) [static]
```

#### 10.217.4.11 GetRows()

```
unsigned short gdcm::Overlay::GetRows ( ) const
```

get rows

#### 10.217.4.12 GetType()

```
const char* gdcm::Overlay::GetType ( ) const
```

get type

#### 10.217.4.13 GetTypeAsEnum()

```
OverlayType gdcm::Overlay::GetTypeAsEnum ( ) const
```

#### 10.217.4.14 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\(\)](#)

**10.217.4.15 GetUnpackBufferLength()**

```
size_t gdcm::Overlay::GetUnpackBufferLength ( ) const
```

Retrieve the size of the buffer needed to hold the [Overlay](#) as specified by Col & Row parameters

**10.217.4.16 GrabOverlayFromPixelData()**

```
bool gdcm::Overlay::GrabOverlayFromPixelData (
    DataSet const & ds )
```

**10.217.4.17 IsEmpty()**

```
bool gdcm::Overlay::IsEmpty ( ) const
```

Return whether or not the [Overlay](#) is empty:

**10.217.4.18 IsInPixelData() [1/2]**

```
bool gdcm::Overlay::IsInPixelData ( ) const
```

return if the [Overlay](#) is stored in the pixel data or not

**10.217.4.19 IsInPixelData() [2/2]**

```
void gdcm::Overlay::IsInPixelData (
    bool b )
```

Set whether or no the OverlayData is in the Pixel Data:

**10.217.4.20 IsZero()**

```
bool gdcm::Overlay::IsZero ( ) const
```

return true if all bits are set to 0

#### 10.217.4.21 operator=()

```
Overlay& gdcm::Overlay::operator= (
    Overlay const & ov )
```

#### 10.217.4.22 Print()

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

Print.

Reimplemented from [gdcm::Object](#).

#### 10.217.4.23 SetBitPosition()

```
void gdcm::Overlay::SetBitPosition (
    unsigned short bitposition )
```

set bit position

#### 10.217.4.24 SetBitsAllocated()

```
void gdcm::Overlay::SetBitsAllocated (
    unsigned short bitsallocated )
```

set bits allocated

#### 10.217.4.25 SetColumns()

```
void gdcm::Overlay::SetColumns (
    unsigned short columns )
```

set columns

**10.217.4.26 SetDescription()**

```
void gdcm::Overlay::SetDescription (
    const char * description )
```

set description

**10.217.4.27 SetFrameOrigin()**

```
void gdcm::Overlay::SetFrameOrigin (
    unsigned short frameorigin )
```

set frame origin

**10.217.4.28 SetGroup()**

```
void gdcm::Overlay::SetGroup (
    unsigned short group )
```

Set Group number.

**10.217.4.29 SetNumberOfFrames()**

```
void gdcm::Overlay::SetNumberOfFrames (
    unsigned int numberofframes )
```

set number of frames

**10.217.4.30 SetOrigin()**

```
void gdcm::Overlay::SetOrigin (
    const signed short origin[2] )
```

set origin



#### 10.217.4.31 SetOverlay()

```
void gdcm::Overlay::SetOverlay (
    const char * array,
    size_t length )
```

set overlay from byte array + length

#### 10.217.4.32 SetRows()

```
void gdcm::Overlay::SetRows (
    unsigned short rows )
```

set rows

#### 10.217.4.33 SetType()

```
void gdcm::Overlay::SetType (
    const char * type )
```

set type

#### 10.217.4.34 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:

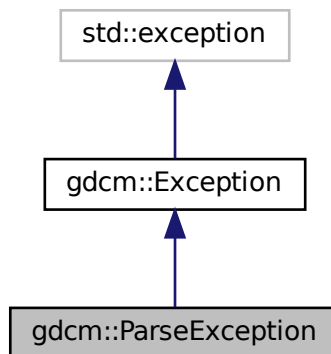
- [gdcmOverlay.h](#)

## 10.218 gdcm::ParseException Class Reference

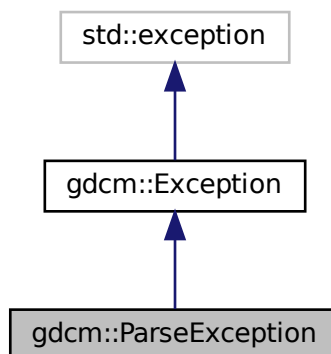
[ParseException](#) Standard exception handling object.

```
#include <gdcmParseException.h>
```

Inheritance diagram for `gdcm::ParseException`:



Collaboration diagram for `gdcm::ParseException`:



## Public Member Functions

- [ParseException](#) ()=default
- [~ParseException](#) () override throw ()
- const [DataElement](#) & [GetLastElement](#) () const
- [ParseException](#) & [operator=](#) (const [ParseException](#) &orig)
- void [SetLastElement](#) ([DataElement](#) &de)

### 10.218.1 Detailed Description

[ParseException](#) Standard exception handling object.

### 10.218.2 Constructor & Destructor Documentation

#### 10.218.2.1 [ParseException](#)()

```
gdcm::ParseException::ParseException ( ) [default]
```

#### 10.218.2.2 [~ParseException](#)()

```
gdcm::ParseException::~~ParseException ( ) throw ( ) [inline], [override]
```

### 10.218.3 Member Function Documentation

#### 10.218.3.1 [GetLastElement](#)()

```
const DataElement& gdcm::ParseException::GetLastElement ( ) const [inline]
```

#### 10.218.3.2 [operator=](#)()

```
ParseException& gdcm::ParseException::operator= (
    const ParseException & orig ) [inline]
```

Assignment operator.

### 10.218.3.3 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](#)

## 10.219 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](#) ()

### 10.219.1 Detailed Description

[Parser](#) ala XML\_Parser from expat (SAX)

Detailed description here

#### Note

Simple API for DICOM

### 10.219.2 Member Typedef Documentation

#### 10.219.2.1 EndElementHandler

```
typedef void(* gdcm::Parser::EndElementHandler) (void *userData, const Tag &name)
```

#### 10.219.2.2 StartElementHandler

```
typedef void(* gdcm::Parser::StartElementHandler) (void *userData, const Tag &tag, const char  
*atts[])
```

### 10.219.3 Member Enumeration Documentation

#### 10.219.3.1 ErrorType

```
enum gdcm::Parser::ErrorType
```

**Enumerator**

NoError	
NoMemoryError	
SyntaxError	
NoElementsError	
TagMismatchError	
DuplicateAttributeError	
JunkAfterDocElementError	
UndefinedEntityError	
UnexpectedStateError	

**10.219.4 Constructor & Destructor Documentation****10.219.4.1 Parser()**

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

**10.219.4.2 ~Parser()**

```
gdcm::Parser::~~Parser ( ) [inline]
```

**10.219.5 Member Function Documentation****10.219.5.1 GetBuffer()**

```
char* gdcm::Parser::GetBuffer (
    int len ) [protected]
```

**10.219.5.2 GetCurrentByteIndex()**

```
unsigned long gdcm::Parser::GetCurrentByteIndex ( ) const
```

### 10.219.5.3 GetErrorCode()

```
ErrorType gdcm::Parser::GetErrorCode ( ) const
```

### 10.219.5.4 GetErrorString()

```
static const char* gdcm::Parser::GetErrorString (
    ErrorType const & err ) [static]
```

### 10.219.5.5 GetUserData()

```
void* gdcm::Parser::GetUserData ( ) const
```

### 10.219.5.6 Parse()

```
bool gdcm::Parser::Parse (
    const char * s,
    int len,
    bool isFinal )
```

### 10.219.5.7 ParseBuffer()

```
bool gdcm::Parser::ParseBuffer (
    int len,
    bool isFinal ) [protected]
```

### 10.219.5.8 Process()

```
ErrorType gdcm::Parser::Process ( ) [protected]
```

#### 10.219.5.9 SetElementHandler()

```
void gdcM::Parser::SetElementHandler (
    StartElementHandler start,
    EndElementHandler end )
```

#### 10.219.5.10 SetUserData()

```
void gdcM::Parser::SetUserData (
    void * userData )
```

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

- [gdcMParser.h](#)

### 10.220 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

#### 10.220.1 Detailed Description

See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

#### 10.220.2 Constructor & Destructor Documentation

##### 10.220.2.1 Patient()

```
gdcM::Patient::Patient ( ) [default]
```

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

- [gdcMPatient.h](#)

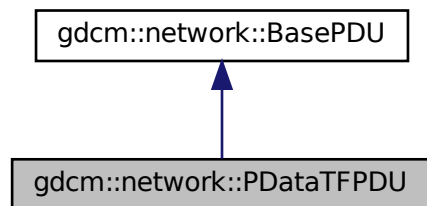


## 10.221 gdcm::network::PDataTFPDU Class Reference

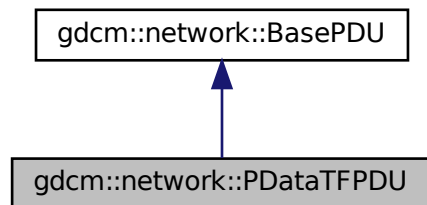
[PDataTFPDU](#).

```
#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

## Protected Member Functions

- `std::istream & ReadInfo (std::istream &is, std::ostream &os)`

### 10.221.1 Detailed Description

[PDataTFPDU](#).

[Table](#) 9-22 P-DATA-TF PDU FIELDS

### 10.221.2 Member Typedef Documentation

#### 10.221.2.1 SizeType

```
typedef std::vector<PresentationDataValue>::size_type gdcm::network::PDataTFPDU::SizeType
```

### 10.221.3 Constructor & Destructor Documentation

#### 10.221.3.1 PDataTFPDU()

```
gdcm::network::PDataTFPDU::PDataTFPDU ( )
```

### 10.221.4 Member Function Documentation

#### 10.221.4.1 AddPresentationDataValue()

```
void gdcm::network::PDataTFPDU::AddPresentationDataValue (  
    PresentationDataValue const & pdv ) [inline]
```

#### 10.221.4.2 GetNumberOfPresentationDataValues()

```
SizeType gdcm::network::PDataTFPDU::GetNumberOfPresentationDataValues ( ) const [inline]
```

#### 10.221.4.3 GetPresentationDataValue()

```
PresentationDataValue const& gdcm::network::PDataTFPDU::GetPresentationDataValue (
    SizeType i ) const [inline]
```

#### 10.221.4.4 IsLastFragment()

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

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

#### 10.221.4.5 Print()

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

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

#### 10.221.4.6 Read()

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

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

#### 10.221.4.7 ReadInto()

```
std::istream& gdcm::network::PDataTFPDU::ReadInto (
    std::istream & is,
    std::ostream & os ) [protected]
```

#### 10.221.4.8 Size()

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

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

#### 10.221.4.9 Write()

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

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

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

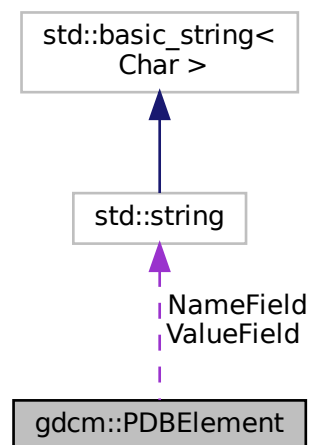
- [gdcnPDataTFPDU.h](#)

### 10.222 gdcn::PDBelement Class Reference

Class to represent a PDB [Element](#).

```
#include <gdcnPDBelement.h>
```

Collaboration diagram for gdcn::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)

### 10.222.1 Detailed Description

Class to represent a PDB [Element](#).

See also

[PDBHeader](#)

### 10.222.2 Constructor & Destructor Documentation

#### 10.222.2.1 PDBelement()

```
gdcm::PDBelement::PDBelement ( ) [default]
```

### 10.222.3 Member Function Documentation

#### 10.222.3.1 GetName()

```
const char* gdcM::PDBelement::GetName ( ) const [inline]
```

Set/Get Name.

#### 10.222.3.2 GetValue()

```
const char* gdcM::PDBelement::GetValue ( ) const [inline]
```

Set/Get [Value](#).

#### 10.222.3.3 operator==()

```
bool gdcM::PDBelement::operator== (
    const PDBelement & de ) const [inline]
```

References NameField, and ValueField.

#### 10.222.3.4 SetName()

```
void gdcM::PDBelement::SetName (
    const char * name ) [inline]
```

#### 10.222.3.5 SetValue()

```
void gdcM::PDBelement::SetValue (
    const char * value ) [inline]
```

### 10.222.4 Friends And Related Function Documentation

#### 10.222.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const PDBElement & val ) [friend]
```

### 10.222.5 Member Data Documentation

#### 10.222.5.1 NameField

```
std::string gdcm::PDBElement::NameField [protected]
```

Referenced by operator==().

#### 10.222.5.2 ValueField

```
std::string gdcm::PDBElement::ValueField [protected]
```

Referenced by operator==().

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

- [gdcmPDBElement.h](#)

## 10.223 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)

### 10.223.1 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.

#### See also

[CSAHeader](#)

### 10.223.2 Constructor & Destructor Documentation

#### 10.223.2.1 PDBHeader()

```
gdcm::PDBHeader::PDBHeader ( ) [default]
```

#### 10.223.2.2 ~PDBHeader()

```
gdcm::PDBHeader::~~PDBHeader ( ) [default]
```



### 10.223.3 Member Function Documentation

#### 10.223.3.1 FindPDBelementByName()

```
bool gdcm::PDBHeader::FindPDBelementByName (
    const char * name )
```

Return true if the PDB element matching name is found or not.

#### 10.223.3.2 GetPDBeEnd()

```
const PDBelement& gdcm::PDBHeader::GetPDBeEnd ( ) const [protected]
```

#### 10.223.3.3 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

#### 10.223.3.4 GetPDBInfoTag()

```
static const PrivateTag& gdcm::PDBHeader::GetPDBInfoTag ( ) [static]
```

Return the Private [Tag](#) where the PDB header is stored within a DICOM [DataSet](#).

#### 10.223.3.5 LoadFromDataElement()

```
bool gdcM::PDBHeader::LoadFromDataElement (
    DataElement const & de )
```

Load the PDB Header from a [DataElement](#) of a [DataSet](#).

#### 10.223.3.6 Print()

```
void gdcM::PDBHeader::Print (
    std::ostream & os ) const
```

Print.

### 10.223.4 Friends And Related Function Documentation

#### 10.223.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const PDBHeader & d ) [friend]
```

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

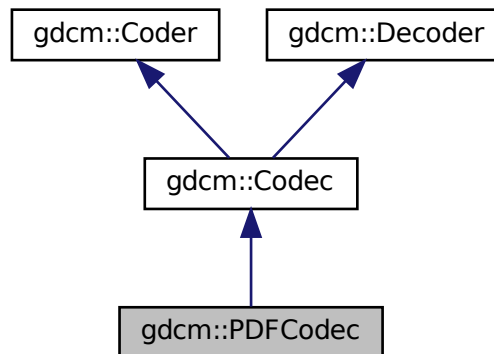
- [gdcM\\_PDBHeader.h](#)

## 10.224 gdcM::PDFCodec Class Reference

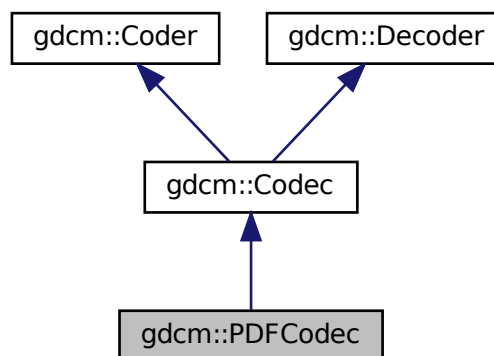
[PDFCodec](#) class.

```
#include <gdcM_PDFCodec.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.*

## Additional Inherited Members

### 10.224.1 Detailed Description

[PDFCodec](#) class.

### 10.224.2 Constructor & Destructor Documentation

#### 10.224.2.1 PDFCodec()

```
gdcm::PDFCodec::PDFCodec ( )
```

#### 10.224.2.2 ~PDFCodec()

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

### 10.224.3 Member Function Documentation

#### 10.224.3.1 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](#).

#### 10.224.3.2 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](#).

### 10.224.3.3 Decode()

```
bool gdcm::PDFCodec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcm::Decoder](#).

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

- [gdcmPDFCodec.h](#)

## 10.225 gdcm::network::PDUFactory Class Reference

[PDUFactory](#) basically, given an initial byte, construct the.

```
#include <gdcmPDUFactory.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)

### 10.225.1 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.

### 10.225.2 Member Function Documentation

#### 10.225.2.1 ConstructAbortPDU()

```
static BasePDU* gdcn::network::PDUFactory::ConstructAbortPDU ( ) [static]
```

#### 10.225.2.2 ConstructPDU()

```
static BasePDU* gdcn::network::PDUFactory::ConstructPDU (
    uint8_t itemtype ) [static]
```

#### 10.225.2.3 ConstructReleasePDU()

```
static BasePDU* gdcn::network::PDUFactory::ConstructReleasePDU ( ) [static]
```

#### 10.225.2.4 CreateCEchoPDU()

```
static std::vector<BasePDU> gdcn::network::PDUFactory::CreateCEchoPDU (
    const ULConnection & inConnection ) [static]
```

#### 10.225.2.5 CreateCFindPDU()

```
static std::vector<BasePDU> gdcn::network::PDUFactory::CreateCFindPDU (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery ) [static]
```

### 10.225.2.6 CreateCMovePDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateCMovePDU (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery ) [static]
```

### 10.225.2.7 CreateCStoreRQPDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateCStoreRQPDU (
    const ULConnection & inConnection,
    const File & file,
    bool writeDataSet = true ) [static]
```

### 10.225.2.8 CreateCStoreRSPPDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateCStoreRSPPDU (
    const DataSet * inDataSet,
    const BasePDU * inPC ) [static]
```

### 10.225.2.9 CreateNActionPDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateNActionPDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

### 10.225.2.10 CreateNCreatePDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateNCreatePDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

### 10.225.2.11 CreateNDeletePDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateNDeletePDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

#### 10.225.2.12 CreateNEventReportPDU()

```
static std::vector<BasePDU*> gdc::network::PDUFactory::CreateNEventReportPDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

#### 10.225.2.13 CreateNGetPDU()

```
static std::vector<BasePDU*> gdc::network::PDUFactory::CreateNGetPDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

#### 10.225.2.14 CreateNSetPDU()

```
static std::vector<BasePDU*> gdc::network::PDUFactory::CreateNSetPDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

#### 10.225.2.15 DetermineEventByPDU()

```
static EEventID gdc::network::PDUFactory::DetermineEventByPDU (
    const BasePDU * inPDU ) [static]
```

#### 10.225.2.16 GetPDVs()

```
static std::vector<PresentationDataValue> gdc::network::PDUFactory::GetPDVs (
    const std::vector< BasePDU * > & inDataPDUs ) [static]
```

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

- [gdcPDUFactory.h](#)

## 10.226 gdc::PersonName Class Reference

[PersonName](#) class.

```
#include <gdcPersonName.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](#) = '^'

### 10.226.1 Detailed Description

[PersonName](#) class.

### 10.226.2 Member Function Documentation

#### 10.226.2.1 GetMaxLength()

```
unsigned int gdcmm::PersonName::GetMaxLength ( ) const [inline]
```

#### 10.226.2.2 GetNumberOfComponents()

```
unsigned int gdcmm::PersonName::GetNumberOfComponents ( ) const [inline]
```

### 10.226.2.3 Print()

```
void gdcm::PersonName::Print (
    std::ostream & os ) const [inline]
```

### 10.226.2.4 SetBlob()

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

### 10.226.2.5 SetComponents() [1/2]

```
void gdcm::PersonName::SetComponents (
    const char * comp1 = "",
    const char * comp2 = "",
    const char * comp3 = "",
    const char * comp4 = "",
    const char * comp5 = "" ) [inline]
```

### 10.226.2.6 SetComponents() [2/2]

```
void gdcm::PersonName::SetComponents (
    const char * components[] ) [inline]
```

## 10.226.3 Member Data Documentation

### 10.226.3.1 Component

```
char gdcm::PersonName::Component [MaxNumberOfComponents] [MaxLength+1]
```

### 10.226.3.2 MaxLength

```
const unsigned int gdcm::PersonName::MaxLength = 64 [static]
```

### 10.226.3.3 MaxNumberOfComponents

```
const unsigned int gdcm::PersonName::MaxNumberOfComponents = 5 [static]
```

### 10.226.3.4 Padding

```
const char gdcm::PersonName::Padding = ' ' [static]
```

### 10.226.3.5 Separator

```
const char gdcm::PersonName::Separator = '^' [static]
```

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

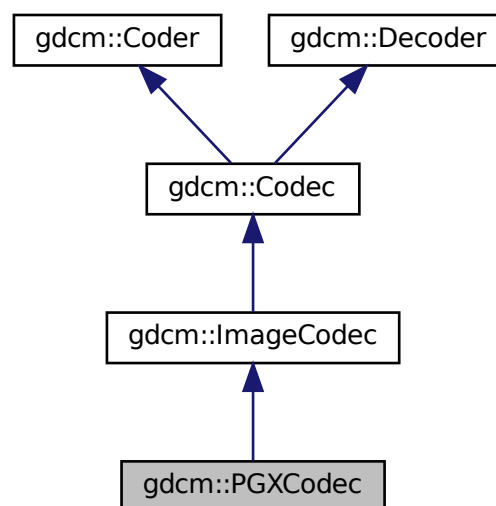
- [gdcmPersonName.h](#)

## 10.227 gdcm::PGXCodec Class Reference

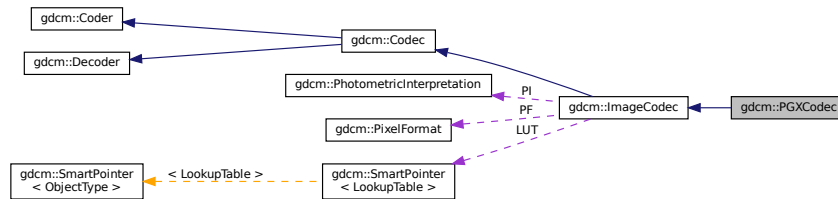
Class to do PGX.

```
#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`

## Additional Inherited Members

### 10.227.1 Detailed Description

Class to do PGX.

See PGX as used in JPEG 2000 implementation and reference images

### 10.227.2 Constructor & Destructor Documentation

#### 10.227.2.1 PGXCodec()

```
gdcm::PGXCodec::PGXCodec ( )
```

### 10.227.2.2 ~PGXCodec()

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

## 10.227.3 Member Function Documentation

### 10.227.3.1 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](#).

### 10.227.3.2 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](#).

### 10.227.3.3 Clone()

```
ImageCodec* gdcm::PGXCodec::Clone ( ) const [override], [virtual]
```

Implements [gdcm::ImageCodec](#).

### 10.227.3.4 GetHeaderInfo()

```
bool gdcm::PGXCodec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.227.3.5 Read()

```
bool gdcM::PGXCodec::Read (
    const char * filename,
    DataElement & out ) const
```

### 10.227.3.6 Write()

```
bool gdcM::PGXCodec::Write (
    const char * filename,
    const DataElement & out ) const
```

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

- [gdcMPGXCodec.h](#)

## 10.228 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)

### 10.228.1 Detailed Description

Class to represent an [PhotometricInterpretation](#).

#### Examples

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [HelloVizWorld.cxx](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

### 10.228.2 Member Enumeration Documentation

## Enumerator

---

### 10.228.2.1 PType

```
enum gdcM::PhotometricInterpretation::PType
```

#### Enumerator

UNKNOWN	
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	

## 10.228.3 Constructor & Destructor Documentation

### 10.228.3.1 PhotometricInterpretation()

```
gdcM::PhotometricInterpretation::PhotometricInterpretation (
    PType pi = UNKNOWN ) [inline]
```

## 10.228.4 Member Function Documentation

### 10.228.4.1 GetPIString()

```
static const char* gdcM::PhotometricInterpretation::GetPIString (
    PType pi ) [static]
```



#### 10.228.4.2 GetPIType()

```
static PType gdcm::PhotometricInterpretation::GetPIType (
    const char * pi ) [static]
```

#### 10.228.4.3 GetSamplesPerPixel()

```
unsigned short gdcm::PhotometricInterpretation::GetSamplesPerPixel ( ) const
```

return the value for Sample Per Pixel associated with a particular Photometric Interpretation

#### 10.228.4.4 GetString()

```
const char* gdcm::PhotometricInterpretation::GetString ( ) const
```

#### 10.228.4.5 GetType()

```
PType gdcm::PhotometricInterpretation::GetType ( ) const [inline]
```

#### 10.228.4.6 IsLossless()

```
bool gdcm::PhotometricInterpretation::IsLossless ( ) const
```

#### 10.228.4.7 IsLossy()

```
bool gdcm::PhotometricInterpretation::IsLossy ( ) const
```

#### 10.228.4.8 IsRetired()

```
static bool gdcm::PhotometricInterpretation::IsRetired (
    PType pi ) [static]
```

#### 10.228.4.9 IsSameColorSpace()

```
bool gdcM::PhotometricInterpretation::IsSameColorSpace (
    PhotometricInterpretation const & pi ) const
```

#### 10.228.4.10 operator PType()

```
gdcM::PhotometricInterpretation::operator PType ( ) const [inline]
```

### 10.228.5 Friends And Related Function Documentation

#### 10.228.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const PhotometricInterpretation & pi ) [friend]
```

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

- [gdcMPhotometricInterpretation.h](#)

## 10.229 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)

## 10.229.1 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](#), [FixJAIBugJPEGLS.cxx](#), [GetJPEGSamplePrecision.cxx](#), [TemplateEmptyImage.cxx](#), [csa2img.cxx](#), [iU22tomultisc.cxx](#), and [threadgdc.cxx](#).

## 10.229.2 Member Enumeration Documentation

### 10.229.2.1 ScalarType

```
enum gdcm::PixelFormat::ScalarType
```

#### Enumerator

UINT8	
INT8	
UINT12	
INT12	
UINT16	
INT16	
UINT32	
INT32	
UINT64	
INT64	
FLOAT16	
FLOAT32	
FLOAT64	
SINGLEBIT	
UNKNOWN	

## 10.229.3 Constructor & Destructor Documentation

### 10.229.3.1 PixelFormat() [1/3]

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

### 10.229.3.2 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]
```

### 10.229.3.3 PixelFormat() [3/3]

```
gdcm::PixelFormat::PixelFormat (
    ScalarType st )
```

## 10.229.4 Member Function Documentation

### 10.229.4.1 GetBitsAllocated()

```
unsigned short gdcm::PixelFormat::GetBitsAllocated ( ) const [inline]
```

BitsAllocated see [Tag](#) (0028,0100) US Bits Allocated.

#### Examples

[GetJPEGSamplePrecision.cxx](#).

#### 10.229.4.2 GetBitsStored()

```
unsigned short gdcm::PixelFormat::GetBitsStored ( ) const [inline]
```

BitsStored see [Tag](#) (0028,0101) US Bits Stored.

##### Examples

[GetJPEGSamplePrecision.cxx](#).

#### 10.229.4.3 GetHighBit()

```
unsigned short gdcm::PixelFormat::GetHighBit ( ) const [inline]
```

HighBit see [Tag](#) (0028,0102) US High Bit.

#### 10.229.4.4 GetMax()

```
int64_t gdcm::PixelFormat::GetMax ( ) const
```

return the max possible of the pixel

#### 10.229.4.5 GetMin()

```
int64_t gdcm::PixelFormat::GetMin ( ) const
```

return the min possible of the pixel

#### 10.229.4.6 GetPixelRepresentation()

```
unsigned short gdcm::PixelFormat::GetPixelRepresentation ( ) const [inline]
```

PixelRepresentation: 0 or 1, see [Tag](#) (0028,0103) US Pixel Representation.

#### 10.229.4.7 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

[threadgdcm.cxx](#).

#### 10.229.4.8 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](#).

#### 10.229.4.9 GetScalarType()

```
ScalarType gdcm::PixelFormat::GetScalarType ( ) const
```

ScalarType does not take into account the sample per pixel.

#### 10.229.4.10 GetScalarTypeAsString()

```
const char* gdcm::PixelFormat::GetScalarTypeAsString ( ) const
```

#### 10.229.4.11 IsCompatible()

```
bool gdcm::PixelFormat::IsCompatible (
    const TransferSyntax & ts ) const
```

#### 10.229.4.12 IsValid()

```
bool gdcm::PixelFormat::IsValid ( ) const
```

```
return IsValid
```

#### 10.229.4.13 operator ScalarType()

```
gdcm::PixelFormat::operator ScalarType ( ) const [inline]
```

#### 10.229.4.14 operator!=( ) [1/2]

```
bool gdcm::PixelFormat::operator!= (
    const PixelFormat & pf ) const [inline]
```

#### 10.229.4.15 operator!=( ) [2/2]

```
bool gdcm::PixelFormat::operator!= (
    ScalarType st ) const [inline]
```

#### 10.229.4.16 operator==( ) [1/2]

```
bool gdcm::PixelFormat::operator== (
    const PixelFormat & pf ) const [inline]
```



**10.229.4.17 operator==( ) [2/2]**

```
bool gdcm::PixelFormat::operator== (
    ScalarType st ) const [inline]
```

**10.229.4.18 Print()**

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

Print.

**10.229.4.19 SetBitsAllocated()**

```
void gdcm::PixelFormat::SetBitsAllocated (
    unsigned short ba ) [inline]
```

**10.229.4.20 SetBitsStored()**

```
void gdcm::PixelFormat::SetBitsStored (
    unsigned short bs ) [inline]
```

**10.229.4.21 SetHighBit()**

```
void gdcm::PixelFormat::SetHighBit (
    unsigned short hb ) [inline]
```

**10.229.4.22 SetPixelRepresentation()**

```
void gdcm::PixelFormat::SetPixelRepresentation (
    unsigned short pr ) [inline]
```

**Examples**

[TemplateEmptyImage.cxx](#).

#### 10.229.4.23 SetSamplesPerPixel()

```
void gdcM::PixelFormat::SetSamplesPerPixel (
    unsigned short spp ) [inline]
```

##### Examples

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), and [GenFakeImage.cxx](#).

References `gdcMAssertMacro`.

#### 10.229.4.24 SetScalarType()

```
void gdcM::PixelFormat::SetScalarType (
    ScalarType st )
```

Set [PixelFormat](#) based only on the [ScalarType](#)

##### Warning

: You need to call `SetScalarType` *before* `SetSamplesPerPixel`

#### 10.229.4.25 Validate()

```
bool gdcM::PixelFormat::Validate ( ) [protected]
```

When image with 24/24/23 was read, need to validate.

Referenced by `gdcM::Bitmap::SetPixelFormat()`.

### 10.229.5 Friends And Related Function Documentation

#### 10.229.5.1 Bitmap

```
friend class Bitmap [friend]
```

### 10.229.5.2 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const PixelFormat & pf ) [friend]
```

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

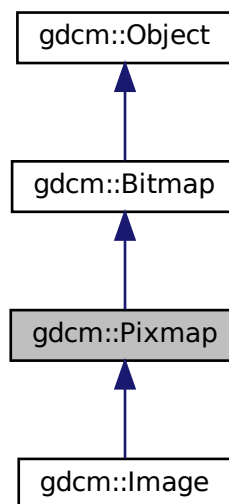
- [gdcmPixelFormat.h](#)

## 10.230 gdcm::Pixmap Class Reference

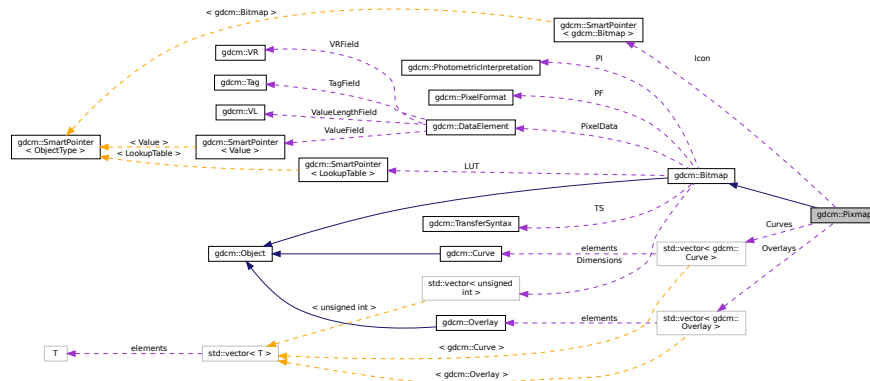
[Pixmap](#) class.

```
#include <gdcmPixmap.h>
```

Inheritance diagram for gdcm::Pixmap:



Collaboration diagram for `gdcm::Pixmap`:



## 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*

## Protected Attributes

- `std::vector< Curve > Curves`
- `SmartPointer< IconImage > Icon`
- `std::vector< Overlay > Overlays`

## Additional Inherited Members

### 10.230.1 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](#)

### 10.230.2 Constructor & Destructor Documentation

#### 10.230.2.1 Pixmap()

```
gdcm::Pixmap::Pixmap ( )
```

#### 10.230.2.2 ~Pixmap()

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

### 10.230.3 Member Function Documentation

#### 10.230.3.1 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](#).

### 10.230.3.2 GetCurve() [1/2]

```
Curve& gdcm::Pixmap::GetCurve (
    size_t i = 0 ) [inline]
```

Curve: group 50xx.

### 10.230.3.3 GetCurve() [2/2]

```
const Curve& gdcm::Pixmap::GetCurve (
    size_t i = 0 ) const [inline]
```

### 10.230.3.4 GetIconImage() [1/2]

```
IconImage& gdcm::Pixmap::GetIconImage ( ) [inline]
```

### 10.230.3.5 GetIconImage() [2/2]

```
const IconImage& gdcm::Pixmap::GetIconImage ( ) const [inline]
```

Set/Get Icon Image.

### 10.230.3.6 GetNumberOfCurves()

```
size_t gdcm::Pixmap::GetNumberOfCurves ( ) const [inline]
```

### 10.230.3.7 GetNumberOfOverlays()

```
size_t gdcm::Pixmap::GetNumberOfOverlays ( ) const [inline]
```

### 10.230.3.8 GetOverlay() [1/2]

```
Overlay& gdcm::Pixmap::GetOverlay (
    size_t i = 0 ) [inline]
```

[Overlay](#): group 60xx.

### 10.230.3.9 GetOverlay() [2/2]

```
const Overlay& gdcm::Pixmap::GetOverlay (
    size_t i = 0 ) const [inline]
```

### 10.230.3.10 Print()

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

Reimplemented from [gdcm::Bitmap](#).

### 10.230.3.11 RemoveOverlay()

```
void gdcm::Pixmap::RemoveOverlay (
    size_t i ) [inline]
```

### 10.230.3.12 SetIconImage()

```
void gdcm::Pixmap::SetIconImage (
    IconImage const & ii ) [inline]
```

### 10.230.3.13 SetNumberOfCurves()

```
void gdcm::Pixmap::SetNumberOfCurves (
    size_t n ) [inline]
```

#### 10.230.3.14 SetNumberOfOverlays()

```
void gdcM::Pixmap::SetNumberOfOverlays (
    size_t n ) [inline]
```

#### 10.230.3.15 UnusedBitsPresentInPixelData()

```
bool gdcM::Pixmap::UnusedBitsPresentInPixelData ( ) const [override], [virtual]
```

returns if there are unused bits in the pixel data

Reimplemented from [gdcM::Bitmap](#).

### 10.230.4 Member Data Documentation

#### 10.230.4.1 Curves

```
std::vector<Curve> gdcM::Pixmap::Curves [protected]
```

#### 10.230.4.2 Icon

```
SmartPointer<IconImage> gdcM::Pixmap::Icon [protected]
```

#### 10.230.4.3 Overlays

```
std::vector<Overlay> gdcM::Pixmap::Overlays [protected]
```

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

- [gdcMPixmap.h](#)

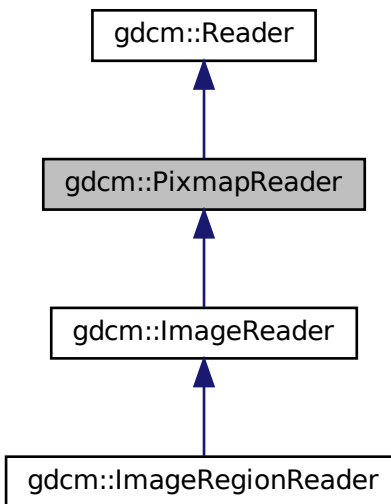


## 10.231 gdcm::PixmapReader Class Reference

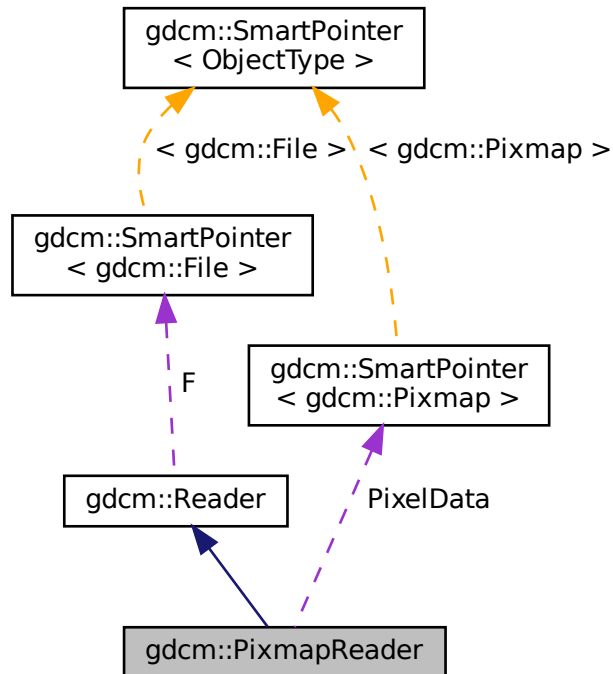
[PixmapReader](#).

```
#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

## Protected Member Functions

- virtual bool [ReadACRNEMAIImage](#) ()
- virtual bool [ReadImage](#) ([MediaStorage](#) const &ms)
- bool [ReadImageInternal](#) ([MediaStorage](#) const &ms, bool handlepixeldata=true)

## Protected Attributes

- [SmartPointer](#)< [Pixmap](#) > [PixelData](#)

## 10.231.1 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 of 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](#)

## 10.231.2 Constructor & Destructor Documentation

### 10.231.2.1 PixmapReader()

```
gdcm::PixmapReader::PixmapReader ( )
```

### 10.231.2.2 ~PixmapReader()

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

## 10.231.3 Member Function Documentation

### 10.231.3.1 GetPixmap() [1/2]

```
Pixmap& gdcm::PixmapReader::GetPixmap ( )
```

### 10.231.3.2 GetPixmap() [2/2]

```
const Pixmap& gdcm::PixmapReader::GetPixmap ( ) const
```

Return the read image (need to call [Read\(\)](#) first)

### 10.231.3.3 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](#).

### 10.231.3.4 ReadACRNEMAIImage()

```
virtual bool gdcm::PixmapReader::ReadACRNEMAIImage ( ) [protected], [virtual]
```

Reimplemented in [gdcm::ImageReader](#).

### 10.231.3.5 ReadImage()

```
virtual bool gdcm::PixmapReader::ReadImage (
    MediaStorage const & ms ) [protected], [virtual]
```

Reimplemented in [gdcm::ImageReader](#).

### 10.231.3.6 ReadImageInternal()

```
bool gdcm::PixmapReader::ReadImageInternal (
    MediaStorage const & ms,
    bool handlepixeldata = true ) [protected]
```

## 10.231.4 Member Data Documentation

### 10.231.4.1 PixelData

```
SmartPointer<Pixmap> gdcm::PixmapReader::PixelData [protected]
```

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

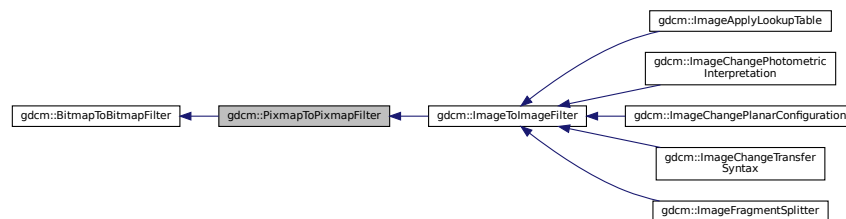
- [gdcmPixmapReader.h](#)

## 10.232 gdcm::PixmapToPixmapFilter Class Reference

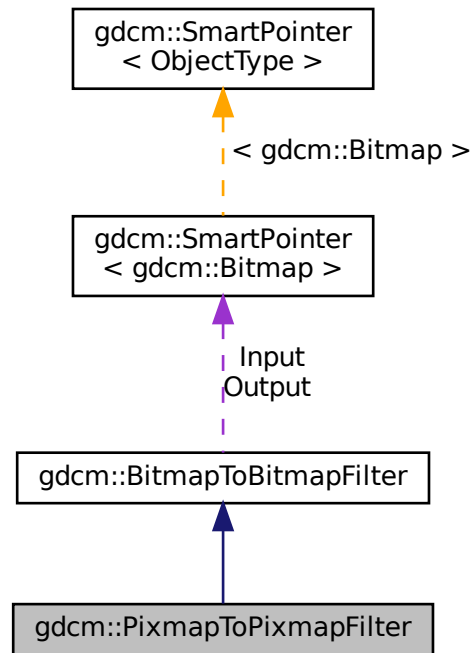
[PixmapToPixmapFilter](#) class.

```
#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

## Additional Inherited Members

### 10.232.1 Detailed Description

[PixmapToPixmapFilter](#) class.

Super class for all filter taking an image and producing an output image

## 10.232.2 Constructor & Destructor Documentation

### 10.232.2.1 PixmapToPixmapFilter()

```
gdcm::PixmapToPixmapFilter::PixmapToPixmapFilter ( )
```

### 10.232.2.2 ~PixmapToPixmapFilter()

```
gdcm::PixmapToPixmapFilter::~~PixmapToPixmapFilter ( ) [default]
```

## 10.232.3 Member Function Documentation

### 10.232.3.1 GetInput()

```
Pixmap& gdcm::PixmapToPixmapFilter::GetInput ( )
```

### 10.232.3.2 GetOutput()

```
const Pixmap& gdcm::PixmapToPixmapFilter::GetOutput ( ) const
```

Get Output image.

### 10.232.3.3 GetOutputAsPixmap()

```
const Pixmap& gdcm::PixmapToPixmapFilter::GetOutputAsPixmap ( ) const
```

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

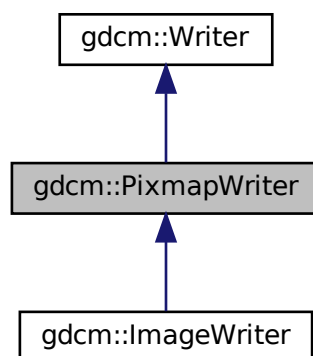
- [gdcmPixmapToPixmapFilter.h](#)

## 10.233 gdcm::PixmapWriter Class Reference

[PixmapWriter](#).

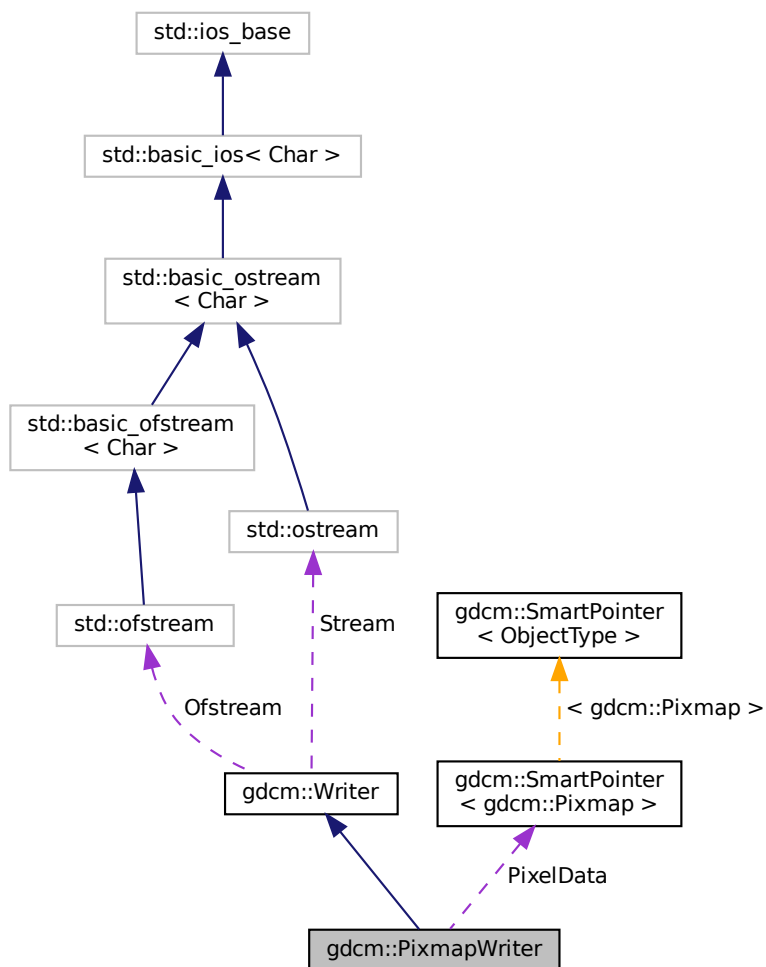
```
#include <gdcmPixmapWriter.h>
```

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.*

## Protected Member Functions

- void [DolconImage](#) ([DataSet](#) &ds, [Pixmap](#) const &image)
- bool [PrepareWrite](#) ([MediaStorage](#) const &refms)

## Protected Attributes

- [SmartPointer](#)< [Pixmap](#) > [PixelData](#)

### 10.233.1 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)

### 10.233.2 Constructor & Destructor Documentation

#### 10.233.2.1 [PixmapWriter](#)()

```
gdcm::PixmapWriter::PixmapWriter ( )
```

#### 10.233.2.2 [~PixmapWriter](#)()

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

### 10.233.3 Member Function Documentation

### 10.233.3.1 DoIconImage()

```
void gdcm::PixmapWriter::DoIconImage (
    DataSet & ds,
    Pixmap const & image ) [protected]
```

### 10.233.3.2 GetImage() [1/2]

```
virtual Pixmap& gdcm::PixmapWriter::GetImage ( ) [inline], [virtual]
```

Reimplemented in [gdcm::ImageWriter](#).

### 10.233.3.3 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](#).

### 10.233.3.4 GetPixmap() [1/2]

```
Pixmap& gdcm::PixmapWriter::GetPixmap ( ) [inline]
```

### 10.233.3.5 GetPixmap() [2/2]

```
const Pixmap& gdcm::PixmapWriter::GetPixmap ( ) const [inline]
```

### 10.233.3.6 PrepareWrite()

```
bool gdcm::PixmapWriter::PrepareWrite (
    MediaStorage const & refs ) [protected]
```

### 10.233.3.7 SetImage()

```
virtual void gdcm::PixmapWriter::SetImage (
    Pixmap const & img ) [virtual]
```

#### Examples

[CompressImage.cxx](#), [GenFakelImage.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [MergeTwoFiles.cxx](#), and [TemplateEmptyImage.cxx](#).

### 10.233.3.8 SetPixmap()

```
void gdcm::PixmapWriter::SetPixmap (
    Pixmap const & img )
```

### 10.233.3.9 Write()

```
bool gdcm::PixmapWriter::Write ( ) [override], [virtual]
```

Write.

Reimplemented from [gdcm::Writer](#).

## 10.233.4 Member Data Documentation

### 10.233.4.1 PixelData

```
SmartPointer<Pixmap> gdcm::PixmapWriter::PixelData [protected]
```

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

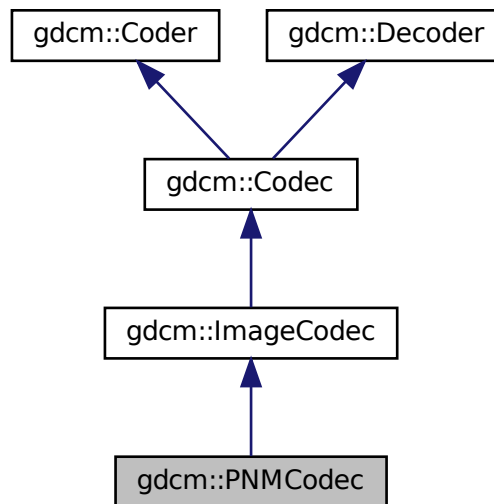
- [gdcmPixmapWriter.h](#)

## 10.234 gdcm::PNMCodec Class Reference

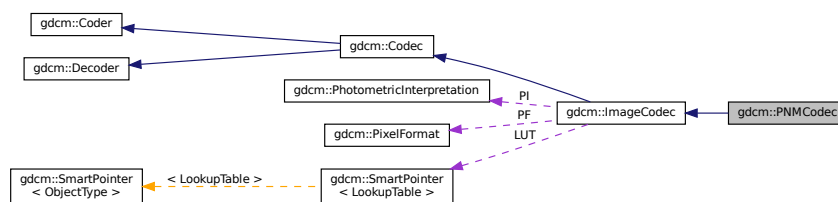
Class to do PNM.

```
#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

## Additional Inherited Members

### 10.234.1 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](#).

### 10.234.2 Constructor & Destructor Documentation

#### 10.234.2.1 PNMCodec()

```
gdcm::PNMCodec::PNMCodec ( )
```

#### 10.234.2.2 ~PNMCodec()

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

### 10.234.3 Member Function Documentation

### 10.234.3.1 CanCode()

```
bool gdcm::PNMCodec::CanCode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.234.3.2 CanDecode()

```
bool gdcm::PNMCodec::CanDecode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.234.3.3 Clone()

```
ImageCodec* gdcm::PNMCodec::Clone ( ) const [override], [virtual]
```

Implements [gdcm::ImageCodec](#).

### 10.234.3.4 GetBufferLength()

```
unsigned long gdcm::PNMCodec::GetBufferLength ( ) const [inline]
```

### 10.234.3.5 GetHeaderInfo()

```
bool gdcm::PNMCodec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.234.3.6 Read()

```
bool gdcM::PNMCodec::Read (
    const char * filename,
    DataElement & out ) const
```

### 10.234.3.7 SetBufferLength()

```
void gdcM::PNMCodec::SetBufferLength (
    unsigned long l ) [inline]
```

### 10.234.3.8 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](#)

## 10.235 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)

### 10.235.1 Detailed Description

DICOM [Preamble](#) (Part 10)

### 10.235.2 Constructor & Destructor Documentation

**10.235.2.1 Preamble()** [1/2]

```
gdcM::Preamble::Preamble ( )
```

**10.235.2.2 ~Preamble()**

```
gdcM::Preamble::~~Preamble ( )
```

**10.235.2.3 Preamble()** [2/2]

```
gdcM::Preamble::Preamble (
    Preamble const & ) [inline]
```

**10.235.3 Member Function Documentation****10.235.3.1 Clear()**

```
void gdcM::Preamble::Clear ( )
```

Clear.

**10.235.3.2 Create()**

```
void gdcM::Preamble::Create ( )
```

**10.235.3.3 GetInternal()**

```
const char* gdcM::Preamble::GetInternal ( ) const [inline]
```

Get internal pointer to preamble.

#### 10.235.3.4 GetLength()

```
VL gdcm::Preamble::GetLength ( ) const [inline]
```

Return size of [Preamble](#).

#### 10.235.3.5 IsEmpty()

```
bool gdcm::Preamble::IsEmpty ( ) const [inline]
```

Check if [Preamble](#) is empty.

#### 10.235.3.6 IsValid()

```
bool gdcm::Preamble::IsValid ( ) const [inline], [protected]
```

#### 10.235.3.7 operator=()

```
Preamble& gdcm::Preamble::operator= (
    Preamble const & ) [inline]
```

#### 10.235.3.8 Print()

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

Print [Preamble](#).

#### 10.235.3.9 Read()

```
std::istream& gdcm::Preamble::Read (
    std::istream & is )
```

Read [Preamble](#).

#### 10.235.3.10 Remove()

```
void gdcM::Preamble::Remove ( )
```

#### 10.235.3.11 Valid()

```
void gdcM::Preamble::Valid ( )
```

Set [Preamble](#) to the default one.

#### 10.235.3.12 Write()

```
std::ostream const& gdcM::Preamble::Write (
    std::ostream & os ) const
```

Write [Preamble](#).

### 10.235.4 Friends And Related Function Documentation

#### 10.235.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const Preamble & _val ) [friend]
```

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

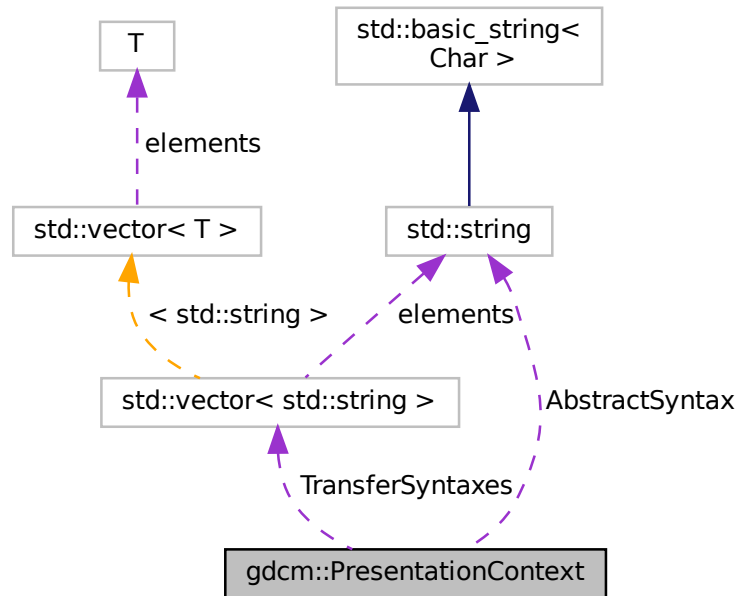
- [gdcMPreamble.h](#)

## 10.236 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](#)

### 10.236.1 Detailed Description

[PresentationContext](#).

See also

[PresentationContextAC](#) [PresentationContextRQ](#)

### 10.236.2 Member Typedef Documentation

#### 10.236.2.1 SizeType

```
typedef TransferSyntaxArrayType::size_type gdcm::PresentationContext::SizeType
```

#### 10.236.2.2 TransferSyntaxArrayType

```
typedef std::vector<std::string> gdcm::PresentationContext::TransferSyntaxArrayType
```

### 10.236.3 Constructor & Destructor Documentation

#### 10.236.3.1 PresentationContext() [1/2]

```
gdcm::PresentationContext::PresentationContext ( )
```

### 10.236.3.2 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 ).

## 10.236.4 Member Function Documentation

### 10.236.4.1 AddTransferSyntax()

```
void gdcm::PresentationContext::AddTransferSyntax (
    const char * tsstr )
```

### 10.236.4.2 GetAbstractSyntax()

```
const char* gdcm::PresentationContext::GetAbstractSyntax ( ) const [inline]
```

### 10.236.4.3 GetNumberOfTransferSyntaxes()

```
SizeType gdcm::PresentationContext::GetNumberOfTransferSyntaxes ( ) const [inline]
```

### 10.236.4.4 GetPresentationContextID()

```
uint8_t gdcm::PresentationContext::GetPresentationContextID ( ) const
```

### 10.236.4.5 GetTransferSyntax()

```
const char* gdcm::PresentationContext::GetTransferSyntax (
    SizeType i ) const [inline]
```

#### 10.236.4.6 operator==( )

```
bool gdcmm::PresentationContext::operator== (
    const PresentationContext & pc ) const [inline]
```

References AbstractSyntax, and TransferSyntaxes.

#### 10.236.4.7 Print()

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

#### 10.236.4.8 SetAbstractSyntax()

```
void gdcmm::PresentationContext::SetAbstractSyntax (
    const char * absyn ) [inline]
```

#### 10.236.4.9 SetPresentationContextID()

```
void gdcmm::PresentationContext::SetPresentationContextID (
    uint8_t id )
```

### 10.236.5 Member Data Documentation

#### 10.236.5.1 AbstractSyntax

```
std::string gdcmm::PresentationContext::AbstractSyntax [protected]
```

Referenced by operator==( ).

#### 10.236.5.2 ID

```
uint8_t gdcmm::PresentationContext::ID [protected]
```



### 10.236.5.3 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](#)

## 10.237 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

### 10.237.1 Detailed Description

[PresentationContextAC](#).

[Table 9-18](#) PRESENTATION CONTEXT ITEM FIELDS

See also

[PresentationContext](#)

### 10.237.2 Constructor & Destructor Documentation

### 10.237.2.1 PresentationContextAC()

```
gdcm::network::PresentationContextAC::PresentationContextAC ( )
```

## 10.237.3 Member Function Documentation

### 10.237.3.1 GetPresentationContextID()

```
uint8_t gdcm::network::PresentationContextAC::GetPresentationContextID ( ) const [inline]
```

### 10.237.3.2 GetReason()

```
uint8_t gdcm::network::PresentationContextAC::GetReason ( ) const [inline]
```

### 10.237.3.3 GetTransferSyntax()

```
TransferSyntaxSub const& gdcm::network::PresentationContextAC::GetTransferSyntax ( ) const [inline]
```

### 10.237.3.4 Print()

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

### 10.237.3.5 Read()

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

### 10.237.3.6 SetPresentationContextID()

```
void gdcm::network::PresentationContextAC::SetPresentationContextID (
    uint8_t id )
```

### 10.237.3.7 SetReason()

```
void gdcm::network::PresentationContextAC::SetReason (
    uint8_t r ) [inline]
```

### 10.237.3.8 SetTransferSyntax()

```
void gdcm::network::PresentationContextAC::SetTransferSyntax (
    TransferSyntaxSub const & ts )
```

### 10.237.3.9 Size()

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

### 10.237.3.10 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](#)

## 10.238 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

### 10.238.1 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](#).

## 10.238.2 Member Typedef Documentation

### 10.238.2.1 PresentationContextArrayType

```
typedef std::vector<PresentationContext> gdcm::PresentationContextGenerator::PresentationContextArrayType
```

### 10.238.2.2 SizeType

```
typedef PresentationContextArrayType::size_type gdcm::PresentationContextGenerator::SizeType
```

## 10.238.3 Constructor & Destructor Documentation

### 10.238.3.1 PresentationContextGenerator()

```
gdcm::PresentationContextGenerator::PresentationContextGenerator ( )
```

## 10.238.4 Member Function Documentation

### 10.238.4.1 AddFromFile()

```
bool gdcm::PresentationContextGenerator::AddFromFile (
    const File & file )
```

Add a single [PresentationContext](#) from a single [File](#). Call multiple times when dealing with multiple files.

### 10.238.4.2 AddPresentationContext()

```
bool gdcm::PresentationContextGenerator::AddPresentationContext (
    const char * absyn,
    const char * ts ) [protected]
```

#### 10.238.4.3 GenerateFromFileNames()

```
bool gdcm::PresentationContextGenerator::GenerateFromFileNames (
    const Directory::FilenameType & 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](#).

#### 10.238.4.4 GenerateFromUID()

```
bool gdcm::PresentationContextGenerator::GenerateFromUID (
    UIDs::TSName asname )
```

Generate the [PresentationContext](#) array from a UID (eg. VerificationSOPClass)

#### 10.238.4.5 GetDefaultTransferSyntax()

```
const char* gdcm::PresentationContextGenerator::GetDefaultTransferSyntax ( ) const [protected]
```

#### 10.238.4.6 GetPresentationContexts()

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

##### Examples

[CStoreQtProgress.cxx](#).

#### 10.238.4.7 SetDefaultTransferSyntax()

```
void gdcm::PresentationContextGenerator::SetDefaultTransferSyntax (
    const TransferSyntax & ts )
```

Not implemented for now. GDCM internally uses Implicit Little Endian.

#### 10.238.4.8 SetMergeModeToAbstractSyntax()

```
void gdcm::PresentationContextGenerator::SetMergeModeToAbstractSyntax ( )
```

#### 10.238.4.9 SetMergeModeToTransferSyntax()

```
void gdcm::PresentationContextGenerator::SetMergeModeToTransferSyntax ( )
```

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

- [gdcmPresentationContextGenerator.h](#)

## 10.239 gdcm::network::PresentationContextRQ Class Reference

[PresentationContextRQ.](#)

```
#include <gdcmPresentationContextRQ.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

### 10.239.1 Detailed Description

[PresentationContextRQ](#).

[Table 9-13 PRESENTATION CONTEXT ITEM FIELDS](#)

See also

[PresentationContextAC](#)

### 10.239.2 Member Typedef Documentation

#### 10.239.2.1 SizeType

```
typedef std::vector<TransferSyntaxSub>::size_type gdcm::network::PresentationContextRQ::SizeType
```

### 10.239.3 Constructor & Destructor Documentation

#### 10.239.3.1 PresentationContextRQ() [1/3]

```
gdcm::network::PresentationContextRQ::PresentationContextRQ ( )
```

#### 10.239.3.2 PresentationContextRQ() [2/3]

```
gdcm::network::PresentationContextRQ::PresentationContextRQ (
    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 ).

#### 10.239.3.3 PresentationContextRQ() [3/3]

```
gdcm::network::PresentationContextRQ::PresentationContextRQ (
    const PresentationContext & pc )
```



## 10.239.4 Member Function Documentation

### 10.239.4.1 AddTransferSyntax()

```
void gdcm::network::PresentationContextRQ::AddTransferSyntax (
    TransferSyntaxSub const & ts )
```

### 10.239.4.2 GetAbstractSyntax() [1/2]

```
AbstractSyntax& gdcm::network::PresentationContextRQ::GetAbstractSyntax ( ) [inline]
```

### 10.239.4.3 GetAbstractSyntax() [2/2]

```
AbstractSyntax const& gdcm::network::PresentationContextRQ::GetAbstractSyntax ( ) const [inline]
```

### 10.239.4.4 GetNumberOfTransferSyntaxes()

```
SizeType gdcm::network::PresentationContextRQ::GetNumberOfTransferSyntaxes ( ) const [inline]
```

### 10.239.4.5 GetPresentationContextID()

```
uint8_t gdcm::network::PresentationContextRQ::GetPresentationContextID ( ) const
```

### 10.239.4.6 GetTransferSyntax() [1/2]

```
TransferSyntaxSub& gdcm::network::PresentationContextRQ::GetTransferSyntax (
    SizeType i ) [inline]
```

**10.239.4.7 GetTransferSyntax() [2/2]**

```
TransferSyntaxSub const& gdcmm::network::PresentationContextRQ::GetTransferSyntax (
    SizeType i ) const [inline]
```

**10.239.4.8 GetTransferSyntaxes()**

```
std::vector<TransferSyntaxSub> const& gdcmm::network::PresentationContextRQ::GetTransferSyntaxes (
) const [inline]
```

**10.239.4.9 operator==( )**

```
bool gdcmm::network::PresentationContextRQ::operator== (
    const PresentationContextRQ & pc ) const [inline]
```

**10.239.4.10 Print()**

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

**10.239.4.11 Read()**

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

**10.239.4.12 SetAbstractSyntax()**

```
void gdcmm::network::PresentationContextRQ::SetAbstractSyntax (
    AbstractSyntax const & absyn )
```

**10.239.4.13 SetPresentationContextID()**

```
void gdcm::network::PresentationContextRQ::SetPresentationContextID (
    uint8_t id )
```

**10.239.4.14 Size()**

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

**10.239.4.15 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](#)

**10.240 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)

### 10.240.1 Detailed Description

[PresentationDataValue](#).

[Table](#) 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS

### 10.240.2 Constructor & Destructor Documentation

#### 10.240.2.1 PresentationDataValue()

```
gdcmm::network::PresentationDataValue::PresentationDataValue ( )
```

### 10.240.3 Member Function Documentation

#### 10.240.3.1 ConcatenatePDVBlobs()

```
static DataSet gdcmm::network::PresentationDataValue::ConcatenatePDVBlobs (
    const std::vector< PresentationDataValue > & inPDVs ) [static]
```

#### Warning

[DataSet](#) will be read as Implicit Little Endian TS

#### 10.240.3.2 ConcatenatePDVBlobsAsExplicit()

```
static DataSet gdcmm::network::PresentationDataValue::ConcatenatePDVBlobsAsExplicit (
    const std::vector< PresentationDataValue > & inPDVs ) [static]
```

### 10.240.3.3 GetBlob()

```
const std::string& gdcm::network::PresentationDataValue::GetBlob ( ) const
```

### 10.240.3.4 GetIsCommand()

```
bool gdcm::network::PresentationDataValue::GetIsCommand ( ) const
```

### 10.240.3.5 GetIsLastFragment()

```
bool gdcm::network::PresentationDataValue::GetIsLastFragment ( ) const
```

### 10.240.3.6 GetMessageHeader()

```
uint8_t gdcm::network::PresentationDataValue::GetMessageHeader ( ) const [inline]
```

### 10.240.3.7 GetPresentationContextID()

```
uint8_t gdcm::network::PresentationDataValue::GetPresentationContextID ( ) const [inline]
```

### 10.240.3.8 Print()

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

### 10.240.3.9 Read()

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

**10.240.3.10 ReadInto()**

```
std::istream& gdcmm::network::PresentationDataValue::ReadInto (
    std::istream & is,
    std::ostream & os )
```

**10.240.3.11 SetBlob()**

```
void gdcmm::network::PresentationDataValue::SetBlob (
    const std::string & partialblob )
```

**10.240.3.12 SetCommand()**

```
void gdcmm::network::PresentationDataValue::SetCommand (
    bool inCommand )
```

**10.240.3.13 SetDataSet()**

```
void gdcmm::network::PresentationDataValue::SetDataSet (
    const DataSet & ds )
```

Set [DataSet](#). Write [DataSet](#) in implicit.

**Warning**

size of dataset should be below maxpdusize

**10.240.3.14 SetLastFragment()**

```
void gdcmm::network::PresentationDataValue::SetLastFragment (
    bool inLast )
```

### 10.240.3.15 SetMessageHeader()

```
void gdcm::network::PresentationDataValue::SetMessageHeader (
    uint8_t messageheader ) [inline]
```

### 10.240.3.16 SetPresentationContextID()

```
void gdcm::network::PresentationDataValue::SetPresentationContextID (
    uint8_t id ) [inline]
```

### 10.240.3.17 Size()

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

### 10.240.3.18 Write()

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

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

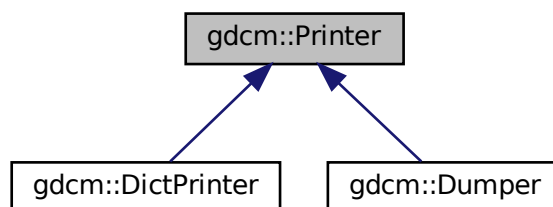
- [gdcmPresentationDataValue.h](#)

## 10.241 gdcm::Printer Class Reference

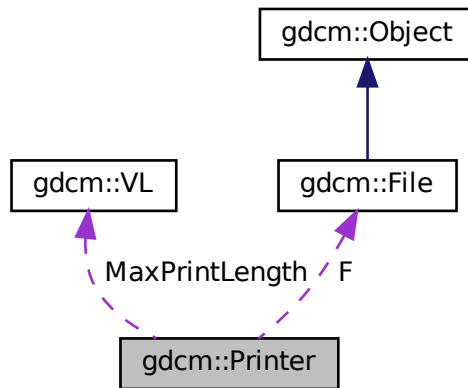
[Printer](#) class.

```
#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()`
- `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::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

- const [File](#) \* F
- [VL MaxPrintLength](#)
- [PrintStyles](#) [PrintStyle](#)

### 10.241.1 Detailed Description

[Printer](#) class.

#### Examples

[DumpSiemensBase64.cxx](#), and [DumpToshibaDTI.cxx](#).

### 10.241.2 Member Enumeration Documentation

#### 10.241.2.1 PrintStyles

enum [gdcm::Printer::PrintStyles](#)

##### Enumerator

VERBOSE_STYLE	
CONDENSED_STYLE	
XML	
CXX	

### 10.241.3 Constructor & Destructor Documentation

#### 10.241.3.1 Printer()

```
gdcm::Printer::Printer ( )
```

### 10.241.3.2 ~Printer()

```
gdcm::Printer::~~Printer ( )
```

## 10.241.4 Member Function Documentation

### 10.241.4.1 GetPrintStyle()

```
PrintStyle gdcm::Printer::GetPrintStyle ( ) const [inline]
```

Get PrintStyle value.

### 10.241.4.2 Print()

```
void gdcm::Printer::Print (
    std::ostream & os )
```

Print.

#### Examples

[DumpSiemensBase64.cxx](#).

### 10.241.4.3 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]
```

#### 10.241.4.4 PrintDataSet()

```
void gdcm::Printer::PrintDataSet (
    const DataSet & ds,
    std::ostream & os,
    const std::string & s = "" )
```

Print an individual dataset.

#### 10.241.4.5 PrintSQ()

```
void gdcm::Printer::PrintSQ (
    const SequenceOfItems * sqi,
    std::ostream & os,
    std::string const & indent ) [protected]
```

#### 10.241.4.6 SetColor()

```
void gdcm::Printer::SetColor (
    bool c )
```

Set color mode or not.

#### 10.241.4.7 SetFile()

```
void gdcm::Printer::SetFile (
    File const & f ) [inline]
```

Set file.

#### Examples

[DumpSiemensBase64.cxx](#), and [DumpToshibaDTI.cxx](#).

#### 10.241.4.8 SetStyle()

```
void gdcm::Printer::SetStyle (
    PrintStyles ps ) [inline]
```

Set PrintStyle value.

## 10.241.5 Member Data Documentation

### 10.241.5.1 F

```
const File* gdcm::Printer::F [protected]
```

### 10.241.5.2 MaxPrintLength

```
VL gdcm::Printer::MaxPrintLength [protected]
```

### 10.241.5.3 PrintStyle

```
PrintStyles gdcm::Printer::PrintStyle [protected]
```

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

- [gdcmPrinter.h](#)

## 10.242 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)

### 10.242.1 Detailed Description

Private [Dict](#).

### 10.242.2 Constructor & Destructor Documentation

#### 10.242.2.1 PrivateDict()

```
gdcm::PrivateDict::PrivateDict ( ) [default]
```

#### 10.242.2.2 ~PrivateDict()

```
gdcm::PrivateDict::~~PrivateDict ( ) [default]
```

### 10.242.3 Member Function Documentation

#### 10.242.3.1 AddDictEntry()

```
void gdcm::PrivateDict::AddDictEntry (
    const PrivateTag & tag,
    const DictEntry & de ) [inline]
```

References [gdcm::DictEntry::GetVM\(\)](#), [gdcm::DictEntry::GetVR\(\)](#), [gdcm::DictEntry::SetVM\(\)](#), [gdcm::DictEntry::SetVR\(\)](#), and [gdcm::VR::UN](#).

### 10.242.3.2 FindDictEntry()

```
bool gdcm::PrivateDict::FindDictEntry (
    const PrivateTag & tag ) const [inline]
```

### 10.242.3.3 GetDictEntry()

```
const DictEntry& gdcm::PrivateDict::GetDictEntry (
    const PrivateTag & tag ) const [inline]
```

### 10.242.3.4 IsEmpty()

```
bool gdcm::PrivateDict::IsEmpty ( ) const [inline]
```

### 10.242.3.5 LoadDefault()

```
void gdcm::PrivateDict::LoadDefault ( ) [protected]
```

### 10.242.3.6 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()`.

### 10.242.3.7 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.

## 10.242.4 Friends And Related Function Documentation

### 10.242.4.1 Dicts

```
friend class Dicts [friend]
```

### 10.242.4.2 operator<<

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

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

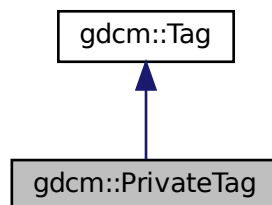
- [gdcmDict.h](#)

## 10.243 gdcm::PrivateTag Class Reference

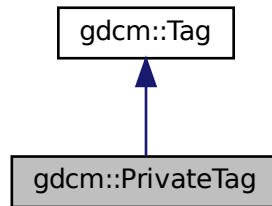
Class to represent a Private DICOM Data [Element](#) ([Attribute](#)) [Tag](#) (Group, [Element](#), Owner)

```
#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 [ReadFromCommaSeparatedString](#) (const char \*str)
- void [SetOwner](#) (const char \*owner)

## Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [PrivateTag](#) &\_val)

### 10.243.1 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](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpVisusChange.cxx](#), [ELSCINT1WaveToText.cxx](#), [MrProtocol.cxx](#), [PublicDict.cxx](#), [csa2img.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

### 10.243.2 Constructor & Destructor Documentation



### 10.243.2.1 PrivateTag() [1/2]

```
gdcmm::PrivateTag::PrivateTag (
    uint16_t group = 0,
    uint16_t element = 0,
    const char * owner = "" ) [inline]
```

### 10.243.2.2 PrivateTag() [2/2]

```
gdcmm::PrivateTag::PrivateTag (
    Tag const & t,
    const char * owner = "" ) [inline]
```

References gdcmm::Tag::GetElement().

## 10.243.3 Member Function Documentation

### 10.243.3.1 GetAsDataElement()

```
DataElement gdcmm::PrivateTag::GetAsDataElement ( ) const
```

### 10.243.3.2 GetOwner()

```
const char* gdcmm::PrivateTag::GetOwner ( ) const [inline]
```

#### Examples

[PublicDict.cxx](#).

Referenced by gdcmm::PrivateDict::PrintXML().

### 10.243.3.3 operator<()

```
bool gdcmm::PrivateTag::operator< (
    const PrivateTag & _val ) const
```

#### 10.243.3.4 ReadFromCommaSeparatedString()

```
bool gdcmm::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")

#### 10.243.3.5 SetOwner()

```
void gdcmm::PrivateTag::SetOwner (
    const char * owner ) [inline]
```

### 10.243.4 Friends And Related Function Documentation

#### 10.243.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const PrivateTag & _val ) [friend]
```

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

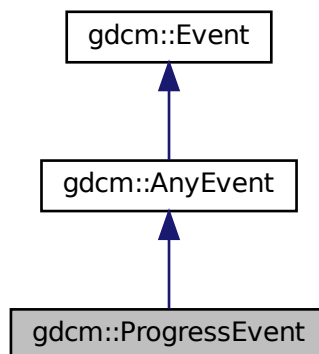
- [gdcmmPrivateTag.h](#)

## 10.244 gdcmm::ProgressEvent Class Reference

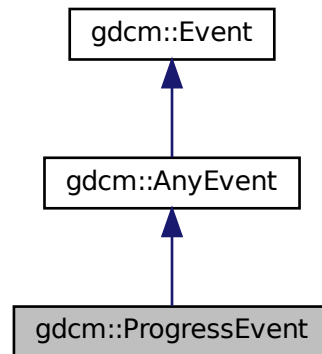
[ProgressEvent](#).

```
#include <gdcmmProgressEvent.h>
```

Inheritance diagram for gdcmm::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 ::gdcm::Event *e`) `const override`
- `const char *` [GetEventName](#) () `const override`
- `double` [GetProgress](#) () `const`
- `::gdcm::Event *` [MakeObject](#) () `const override`
- `void` [operator=](#) (`const Self &`) `=delete`
- `void` [SetProgress](#) (`double p`)

### 10.244.1 Detailed Description

[ProgressEvent](#).

Special type of event triggered during

See also

[AnyEvent](#)

## 10.244.2 Member Typedef Documentation

### 10.244.2.1 Self

```
typedef ProgressEvent gdcm::ProgressEvent::Self
```

### 10.244.2.2 Superclass

```
typedef AnyEvent gdcm::ProgressEvent::Superclass
```

## 10.244.3 Constructor & Destructor Documentation

### 10.244.3.1 ProgressEvent() [1/2]

```
gdcm::ProgressEvent::ProgressEvent (
    double p = 0 ) [inline]
```

### 10.244.3.2 ~ProgressEvent()

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

### 10.244.3.3 ProgressEvent() [2/2]

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

## 10.244.4 Member Function Documentation

#### 10.244.4.1 CheckEvent()

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

#### 10.244.4.2 GetEventName()

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

Return the StringName associated with the event.

Implements [gdcm::Event](#).

#### 10.244.4.3 GetProgress()

```
double gdcm::ProgressEvent::GetProgress ( ) const [inline]
```

#### 10.244.4.4 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](#).

#### 10.244.4.5 operator=()

```
void gdcm::ProgressEvent::operator= (
    const Self & ) [delete]
```

#### 10.244.4.6 SetProgress()

```
void gdcm::ProgressEvent::SetProgress (
    double p ) [inline]
```

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

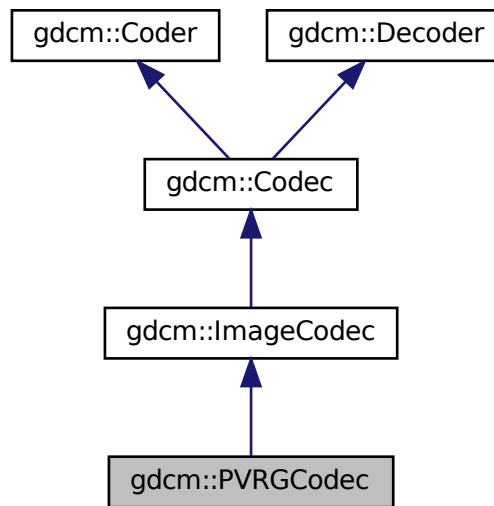
- [gdcmProgressEvent.h](#)

## 10.245 gdcm::PVRGCodec Class Reference

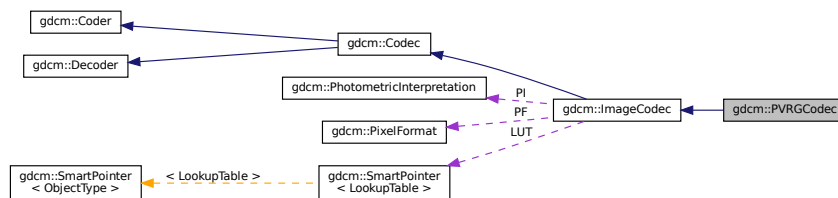
[PVRGCodec](#).

```
#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)

## Additional Inherited Members

### 10.245.1 Detailed Description

[PVRGCodec](#).

#### Note

pvrj 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\_Gyroscan-12-Jpeg\_Extended\_Process\_2\_4.dcm

we have to...

### 10.245.2 Constructor & Destructor Documentation

#### 10.245.2.1 PVRGCodec()

```
gdcm::PVRGCodec::PVRGCodec ( )
```

#### 10.245.2.2 ~PVRGCodec()

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

### 10.245.3 Member Function Documentation

#### 10.245.3.1 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](#).

#### 10.245.3.2 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](#).

#### 10.245.3.3 Clone()

```
ImageCodec* gdcm::PVRGCodec::Clone ( ) const [override], [virtual]
```

Implements [gdcm::ImageCodec](#).

#### 10.245.3.4 Code()

```
bool gdcm::PVRGCodec::Code (
    DataElement const & in_,
    DataElement & out_ ) [override], [virtual]
```

Code.

Reimplemented from [gdcm::Coder](#).



### 10.245.3.5 Decode()

```
bool gdcm::PVRGCodec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcm::ImageCodec](#).

### 10.245.3.6 SetLossyFlag()

```
void gdcm::PVRGCodec::SetLossyFlag (
    bool l )
```

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

- [gdcmPVRGCodec.h](#)

## 10.246 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)

### 10.246.1 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.

## 10.246.2 Constructor & Destructor Documentation

### 10.246.2.1 PythonFilter()

```
gdcm::PythonFilter::PythonFilter ( )
```

### 10.246.2.2 ~PythonFilter()

```
gdcm::PythonFilter::~~PythonFilter ( )
```

## 10.246.3 Member Function Documentation

### 10.246.3.1 GetFile() [1/2]

```
File& gdcm::PythonFilter::GetFile ( )
```

### 10.246.3.2 GetFile() [2/2]

```
const File& gdcm::PythonFilter::GetFile ( ) const
```

### 10.246.3.3 SetDicts()

```
void gdcm::PythonFilter::SetDicts (
    const Dicts & dicts )
```

### 10.246.3.4 SetFile()

```
void gdcm::PythonFilter::SetFile (
    const File & f )
```

### 10.246.3.5 ToPyObject()

```
PyObject* gdcm::PythonFilter::ToPyObject (
    const Tag & t ) const
```

### 10.246.3.6 UseDictAlways()

```
void gdcm::PythonFilter::UseDictAlways (
    bool ) [inline]
```

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

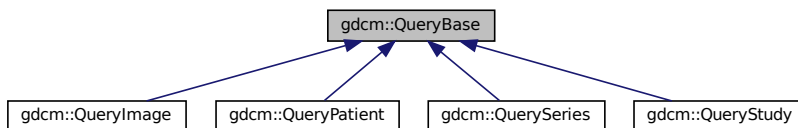
- [gdcmPythonFilter.h](#)

## 10.247 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](#) > [GetHierachicalSearchTags](#) (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

## 10.247.1 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.

## 10.247.2 Constructor & Destructor Documentation

### 10.247.2.1 ~QueryBase()

```
virtual gdcm::QueryBase::~~QueryBase ( ) [virtual], [default]
```

## 10.247.3 Member Function Documentation

### 10.247.3.1 GetAllRequiredTags()

```
std::vector<Tag> gdcm::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

### 10.247.3.2 GetAllTags()

```
std::vector<Tag> gdcm::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

### 10.247.3.3 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::QueryStudy](#), [gdcm::QuerySeries](#), [gdcm::QueryPatient](#), and [gdcm::QueryImage](#).

### 10.247.3.4 GetName()

```
virtual const char* gdcm::QueryBase::GetName ( ) const [pure virtual]
```

Implemented in [gdcm::QueryStudy](#), [gdcm::QuerySeries](#), [gdcm::QueryPatient](#), and [gdcm::QueryImage](#).

### 10.247.3.5 GetOptionalTags()

```
virtual std::vector<Tag> gdcm::QueryBase::GetOptionalTags (
    const ERootType & inRootType ) const [pure virtual]
```

Implemented in [gdcm::QueryStudy](#), [gdcm::QuerySeries](#), [gdcm::QueryPatient](#), and [gdcm::QueryImage](#).

### 10.247.3.6 GetQueryLevel()

```
virtual DataElement gdcm::QueryBase::GetQueryLevel ( ) const [pure virtual]
```

Implemented in [gdcm::QueryStudy](#), [gdcm::QuerySeries](#), [gdcm::QueryPatient](#), and [gdcm::QueryImage](#).

### 10.247.3.7 GetRequiredTags()

```
virtual std::vector<Tag> gdcM::QueryBase::GetRequiredTags (
    const ERootType & inRootType ) const [pure virtual]
```

Implemented in [gdcM::QueryStudy](#), [gdcM::QuerySeries](#), [gdcM::QueryPatient](#), and [gdcM::QueryImage](#).

### 10.247.3.8 GetUniqueTags()

```
virtual std::vector<Tag> gdcM::QueryBase::GetUniqueTags (
    const ERootType & inRootType ) const [pure virtual]
```

Implemented in [gdcM::QueryStudy](#), [gdcM::QuerySeries](#), [gdcM::QueryPatient](#), and [gdcM::QueryImage](#).

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

- [gdcMQueryBase.h](#)

## 10.248 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)

### 10.248.1 Detailed Description

QueryFactory.h.

#### Note

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

## 10.248.2 Member Function Documentation

### 10.248.2.1 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() ).

### 10.248.2.2 ListCharSets()

```
static void gdcm::QueryFactory::ListCharSets (
    std::ostream & os ) [static]
```

List all possible CharSet.

### 10.248.2.3 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

### 10.248.2.4 ProduceQuery() [1/2]

```
static BaseQuery* gdcm::QueryFactory::ProduceQuery (
    const std::string & sopInstanceUID,
    ENQueryType inQueryType ) [static]
```

### 10.248.2.5 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:

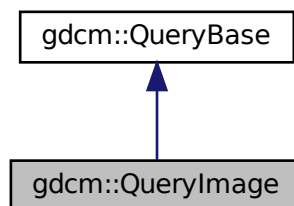
- [gdcMQueryFactory.h](#)

## 10.249 gdcM::QueryImage Class Reference

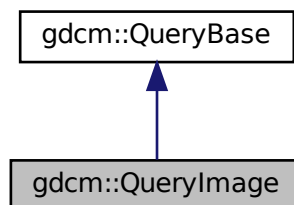
[QueryImage](#).

```
#include <gdcMQueryImage.h>
```

Inheritance diagram for gdcM::QueryImage:



Collaboration diagram for gdcM::QueryImage:





## Public Member Functions

- `std::vector< Tag > GetHierarchicalSearchTags` (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

### 10.249.1 Detailed Description

[QueryImage](#).

contains: class to construct an image-based query for C-FIND and C-MOVE

### 10.249.2 Member Function Documentation

#### 10.249.2.1 GetHierarchicalSearchTags()

```
std::vector<Tag> gdcm::QueryImage::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](#).

#### 10.249.2.2 GetName()

```
const char* gdcm::QueryImage::GetName ( ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

#### 10.249.2.3 GetOptionalTags()

```
std::vector<Tag> gdcm::QueryImage::GetOptionalTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

#### 10.249.2.4 GetQueryLevel()

```
DataElement gdcM::QueryImage::GetQueryLevel ( ) const [override], [virtual]
```

Implements [gdcM::QueryBase](#).

#### 10.249.2.5 GetRequiredTags()

```
std::vector<Tag> gdcM::QueryImage::GetRequiredTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Implements [gdcM::QueryBase](#).

#### 10.249.2.6 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:

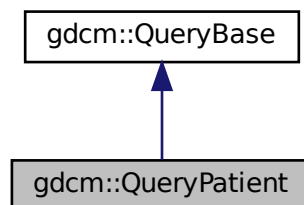
- [gdcMQueryImage.h](#)

## 10.250 gdcM::QueryPatient Class Reference

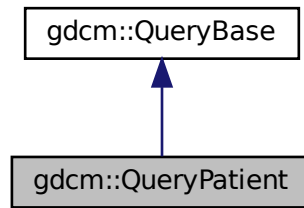
[QueryPatient](#).

```
#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

### 10.250.1 Detailed Description

[QueryPatient](#).

contains: class to construct a patient-based query for c-find and c-move

### 10.250.2 Member Function Documentation

#### 10.250.2.1 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](#).

### 10.250.2.2 GetName()

```
const char* gdcm::QueryPatient::GetName ( ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

### 10.250.2.3 GetOptionalTags()

```
std::vector<Tag> gdcm::QueryPatient::GetOptionalTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

### 10.250.2.4 GetQueryLevel()

```
DataElement gdcm::QueryPatient::GetQueryLevel ( ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

### 10.250.2.5 GetRequiredTags()

```
std::vector<Tag> gdcm::QueryPatient::GetRequiredTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

### 10.250.2.6 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:

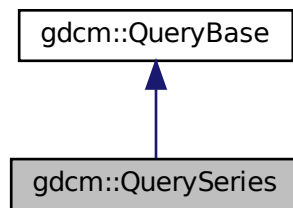
- [gdcmQueryPatient.h](#)

## 10.251 gdcm::QuerySeries Class Reference

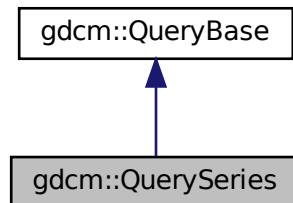
[QuerySeries](#).

```
#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

### 10.251.1 Detailed Description

[QuerySeries](#).

contains: class to construct a series-based query for c-find and c-move

### 10.251.2 Member Function Documentation

#### 10.251.2.1 GetHierarchicalSearchTags()

```
std::vector<Tag> gdc::QuerySeries::GetHierarchicalSearchTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements [gdc::QueryBase](#).

#### 10.251.2.2 GetName()

```
const char* gdc::QuerySeries::GetName ( ) const [override], [virtual]
```

Implements [gdc::QueryBase](#).

#### 10.251.2.3 GetOptionalTags()

```
std::vector<Tag> gdc::QuerySeries::GetOptionalTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Implements [gdc::QueryBase](#).

#### 10.251.2.4 GetQueryLevel()

```
DataElement gdc::QuerySeries::GetQueryLevel ( ) const [override], [virtual]
```

Implements [gdc::QueryBase](#).

### 10.251.2.5 GetRequiredTags()

```
std::vector<Tag> gdcm::QuerySeries::GetRequiredTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

### 10.251.2.6 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:

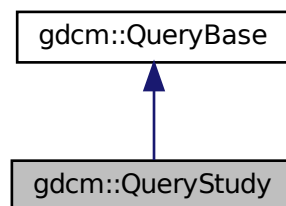
- [gdcmQuerySeries.h](#)

## 10.252 gdcm::QueryStudy Class Reference

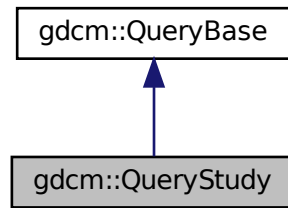
QueryStudy.h.

```
#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

### 10.252.1 Detailed Description

`QueryStudy.h`.

contains: class to construct a study-based query for C-FIND and C-MOVE

### 10.252.2 Member Function Documentation

#### 10.252.2.1 GetHierachicalSearchTags()

```
std::vector<Tag> gdcm::QueryStudy::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`.



### 10.252.2.2 GetName()

```
const char* gdcm::QueryStudy::GetName ( ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

### 10.252.2.3 GetOptionalTags()

```
std::vector<Tag> gdcm::QueryStudy::GetOptionalTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

### 10.252.2.4 GetQueryLevel()

```
DataElement gdcm::QueryStudy::GetQueryLevel ( ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

### 10.252.2.5 GetRequiredTags()

```
std::vector<Tag> gdcm::QueryStudy::GetRequiredTags (
    const ERootType & inRootType ) const [override], [virtual]
```

Implements [gdcm::QueryBase](#).

### 10.252.2.6 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:

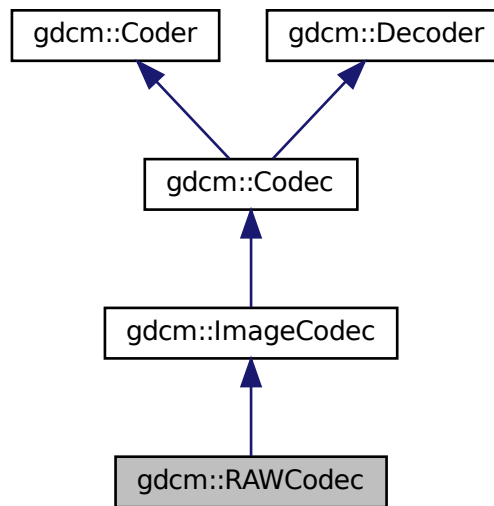
- [gdcmQueryStudy.h](#)

## 10.253 gdcm::RAWCodec Class Reference

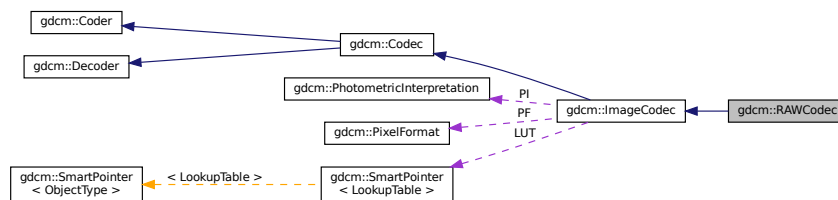
[RAWCodec](#) class.

```
#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

## Protected Member Functions

- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os) override

## Additional Inherited Members

### 10.253.1 Detailed Description

[RAWCodec](#) class.

### 10.253.2 Constructor & Destructor Documentation

#### 10.253.2.1 RAWCodec()

```
gdcm::RAWCodec::RAWCodec ( )
```

#### 10.253.2.2 ~RAWCodec()

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

### 10.253.3 Member Function Documentation

### 10.253.3.1 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](#).

### 10.253.3.2 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](#).

### 10.253.3.3 Clone()

```
ImageCodec* gdcm::RAWCodec::Clone ( ) const [override], [virtual]
```

Implements [gdcm::ImageCodec](#).

### 10.253.3.4 Code()

```
bool gdcm::RAWCodec::Code (
    DataElement const & in_,
    DataElement & out_ ) [override], [virtual]
```

Code.

Reimplemented from [gdcm::Coder](#).

### 10.253.3.5 Decode()

```
bool gdcm::RAWCodec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcm::ImageCodec](#).

### 10.253.3.6 DecodeByStreams()

```
bool gdcm::RAWCodec::DecodeByStreams (
    std::istream & is,
    std::ostream & os ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.253.3.7 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.

### 10.253.3.8 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:

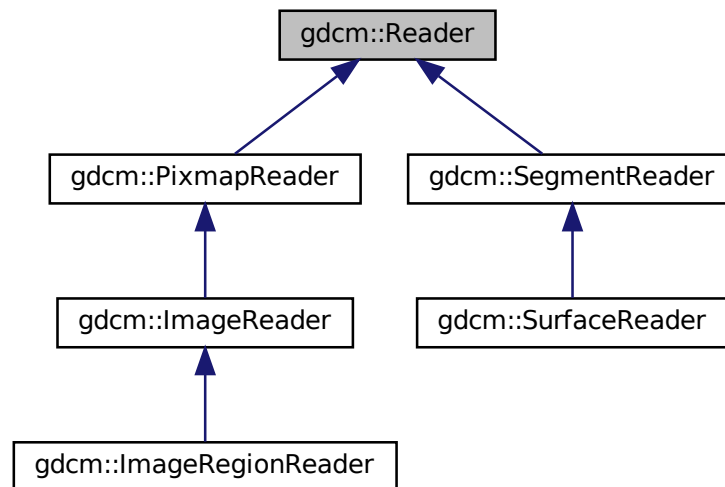
- [gdcmRAWCodec.h](#)

## 10.254 gdcm::Reader Class Reference

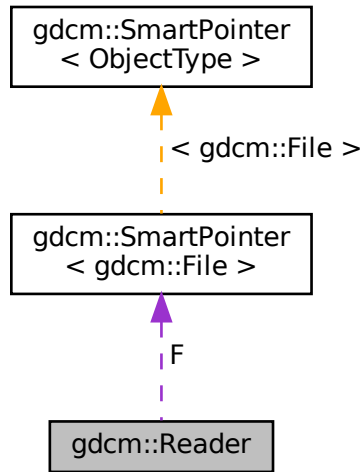
[Reader](#) ala DOM (Document [Object](#) Model)

```
#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](#)

### 10.254.1 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

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CreateFakeRTDOSE.cxx](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [FixBrokenJ2K.cxx](#), [FixOrientation.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadUTF8QtDir.cxx](#), [TestReader.cxx](#), [csa2img.cxx](#), [gdcmrionplan.cxx](#), [gdcmrtpplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).



## 10.254.2 Constructor & Destructor Documentation

### 10.254.2.1 Reader()

```
gdcm::Reader::Reader ( )
```

### 10.254.2.2 ~Reader()

```
virtual gdcm::Reader::~Reader ( ) [virtual]
```

## 10.254.3 Member Function Documentation

### 10.254.3.1 CanRead()

```
bool gdcm::Reader::CanRead ( ) const
```

Test whether this is a DICOM file

#### Warning

need to call either SetFileName or SetStream first

#### Examples

[ReadUTF8QtDir.cxx](#).

### 10.254.3.2 GetFile() [1/2]

```
File& gdcm::Reader::GetFile ( ) [inline]
```

Set/Get [File](#).

### 10.254.3.3 GetFile() [2/2]

```
const File& gdcm::Reader::GetFile ( ) const [inline]
```

Set/Get File.

#### Examples

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CompressImage.cxx](#), [CreateFakeRTDOSE.cxx](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [TestReader.cxx](#), [csa2img.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

### 10.254.3.4 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++

### 10.254.3.5 GetStreamPtr()

```
std::istream* gdcm::Reader::GetStreamPtr ( ) const [inline], [protected]
```

### 10.254.3.6 Read()

```
virtual bool gdcm::Reader::Read ( ) [virtual]
```

Main function to read a file.

Reimplemented in [gdcm::SurfaceReader](#), [gdcm::SegmentReader](#), [gdcm::PixmapReader](#), [gdcm::ImageRegionReader](#), and [gdcm::ImageReader](#).

#### Examples

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CreateFakeRTDOSE.cxx](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [FixBrokenJ2K.cxx](#), [FixOrientation.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [TestReader.cxx](#), [csa2img.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

### 10.254.3.7 ReadDataSet()

```
bool gdcm::Reader::ReadDataSet ( ) [protected]
```

### 10.254.3.8 ReadMetaInformation()

```
bool gdcm::Reader::ReadMetaInformation ( ) [protected]
```

### 10.254.3.9 ReadPreamble()

```
bool gdcm::Reader::ReadPreamble ( ) [protected]
```

### 10.254.3.10 ReadSelectedPrivateTags()

```
bool gdcm::Reader::ReadSelectedPrivateTags (
    std::set< PrivateTag > const & ptags,
    bool readvalues = true )
```

Will only read the specified selected private tags.

### 10.254.3.11 ReadSelectedTags()

```
bool gdcm::Reader::ReadSelectedTags (
    std::set< Tag > const & tags,
    bool readvalues = true )
```

Will only read the specified selected tags.

### 10.254.3.12 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](#).

**10.254.3.13 SetFile()**

```
void gdcmm::Reader::SetFile (
    File & file ) [inline]
```

Set/Get [File](#).

**10.254.3.14 SetFileName()**

```
void gdcmm::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

[BasicImageAnonymizer.cs](#), [ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CheckBigEndianBug.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CompressImage.cxx](#), [ConvertToQImage.cxx](#), [CreateFakeRTDOSE.cxx](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [PrintLUT.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadMultiTimesException.cxx](#), [ReadUTF8QtDir.cxx](#), [SimplePrintPatientName.cs](#), [TemplateEmptyImage.cxx](#), [TestReader.cxx](#), [csa2img.cxx](#), [gdcmmrionplan.cxx](#), [gdcmmrtplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), [rle2img.cxx](#), and [threadgdcmm.cxx](#).

### 10.254.3.15 SetStream()

```
void gdcm::Reader::SetStream (
    std::istream & input_stream ) [inline]
```

Set the open-ed stream directly.

#### Examples

[ReadUTF8QtDir.cxx](#).

## 10.254.4 Friends And Related Function Documentation

### 10.254.4.1 StreamImageReader

```
friend class StreamImageReader [friend]
```

## 10.254.5 Member Data Documentation

### 10.254.5.1 F

```
SmartPointer<File> gdcm::Reader::F [protected]
```

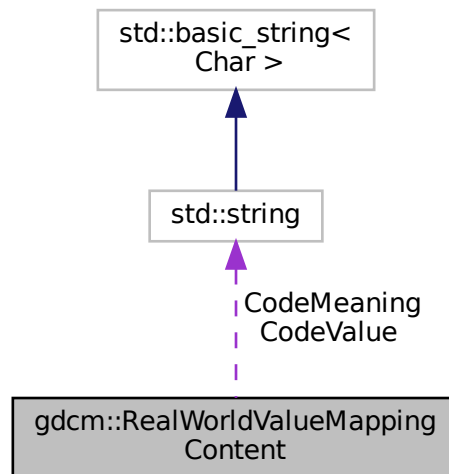
The documentation for this class was generated from the following file:

- [gdcmReader.h](#)

## 10.255 gdcm::RealWorldValueMappingContent Struct Reference

```
#include <gdcmImageHelper.h>
```

Collaboration diagram for gdcm::RealWorldValueMappingContent:



### Public Attributes

- `std::string` [CodeMeaning](#)
- `std::string` [CodeValue](#)
- `double` [RealWorldValueIntercept](#)
- `double` [RealWorldValueSlope](#)

### 10.255.1 Member Data Documentation

#### 10.255.1.1 CodeMeaning

```
std::string gdcm::RealWorldValueMappingContent::CodeMeaning
```

### 10.255.1.2 CodeValue

```
std::string gdcm::RealWorldValueMappingContent::CodeValue
```

### 10.255.1.3 RealWorldValueIntercept

```
double gdcm::RealWorldValueMappingContent::RealWorldValueIntercept
```

### 10.255.1.4 RealWorldValueSlope

```
double gdcm::RealWorldValueMappingContent::RealWorldValueSlope
```

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

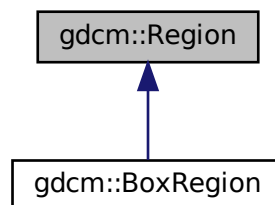
- [gdcmImageHelper.h](#)

## 10.256 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.*

### 10.256.1 Detailed Description

Class for manipulation region.

### 10.256.2 Constructor & Destructor Documentation

#### 10.256.2.1 [Region](#)()

```
gdcmm::Region::Region ( )
```

#### 10.256.2.2 [~Region](#)()

```
virtual gdcmm::Region::~~Region ( ) [virtual]
```

### 10.256.3 Member Function Documentation



### 10.256.3.1 Area()

```
virtual size_t gdcm::Region::Area ( ) const [pure virtual]
```

compute the area

Implemented in [gdcm::BoxRegion](#).

### 10.256.3.2 Clone()

```
virtual Region* gdcm::Region::Clone ( ) const [pure virtual]
```

Implemented in [gdcm::BoxRegion](#).

### 10.256.3.3 ComputeBoundingBox()

```
virtual BoxRegion gdcm::Region::ComputeBoundingBox ( ) [pure virtual]
```

Return the Axis-Aligned minimum bounding box for all regions.

Implemented in [gdcm::BoxRegion](#).

### 10.256.3.4 Empty()

```
virtual bool gdcm::Region::Empty ( ) const [pure virtual]
```

return whether this domain is empty:

Implemented in [gdcm::BoxRegion](#).

### 10.256.3.5 IsValid()

```
virtual bool gdcm::Region::IsValid ( ) const [pure virtual]
```

return whether this is valid domain

Implemented in [gdcm::BoxRegion](#).

### 10.256.3.6 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](#)

## 10.257 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)`

### 10.257.1 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](#)

### 10.257.2 Constructor & Destructor Documentation

### 10.257.2.1 Rescaler()

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

### 10.257.2.2 ~Rescaler()

```
gdcm::Rescaler::~~Rescaler ( ) [default]
```

## 10.257.3 Member Function Documentation

### 10.257.3.1 ComputeInterceptSlopePixelType()

```
PixelFormat::ScalarType gdcm::Rescaler::ComputeInterceptSlopePixelType ( )
```

Compute the Pixel Format of the output data Used for direct transformation

### 10.257.3.2 ComputePixelTypeFromMinMax()

```
PixelFormat gdcm::Rescaler::ComputePixelTypeFromMinMax ( )
```

Compute the Pixel Format of the output data Used for inverse transformation

### 10.257.3.3 GetIntercept()

```
double gdcm::Rescaler::GetIntercept ( ) const [inline]
```

### 10.257.3.4 GetSlope()

```
double gdcm::Rescaler::GetSlope ( ) const [inline]
```

### 10.257.3.5 InverseRescale()

```
bool gdcm::Rescaler::InverseRescale (
    char * out,
    const char * in,
    size_t n )
```

Inverse transform.

### 10.257.3.6 InverseRescaleFunctionIntoBestFit()

```
template<typename TIn >
void gdcm::Rescaler::InverseRescaleFunctionIntoBestFit (
    char * out,
    const TIn * in,
    size_t n ) [protected]
```

### 10.257.3.7 Rescale()

```
bool gdcm::Rescaler::Rescale (
    char * out,
    const char * in,
    size_t n )
```

Direct transform.

### 10.257.3.8 RescaleFunctionIntoBestFit()

```
template<typename TIn >
void gdcm::Rescaler::RescaleFunctionIntoBestFit (
    char * out,
    const TIn * in,
    size_t n ) [protected]
```

### 10.257.3.9 SetIntercept()

```
void gdcm::Rescaler::SetIntercept (
    double i ) [inline]
```

Set Intercept: used for both direct&inverse transformation.

#### 10.257.3.10 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

#### 10.257.3.11 SetPixelFormat()

```
void gdcm::Rescaler::SetPixelFormat (
    PixelFormat const & pf ) [inline]
```

Set Pixel Format of input data.

#### 10.257.3.12 SetSlope()

```
void gdcm::Rescaler::SetSlope (
    double s ) [inline]
```

Set Slope: user for both direct&inverse transformation.

#### 10.257.3.13 SetTargetPixelType()

```
void gdcm::Rescaler::SetTargetPixelType (
    PixelFormat const & targetst )
```

By default (when UseTargetPixelType is false), a best matching Target Pixel [Type](#) is computed. However user can override this auto selection by switching UseTargetPixelType:true and also specifying the specifix Target Pixel [Type](#)

#### 10.257.3.14 SetUseTargetPixelType()

```
void gdcm::Rescaler::SetUseTargetPixelType (
    bool b )
```

Override default behavior of Rescale.

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

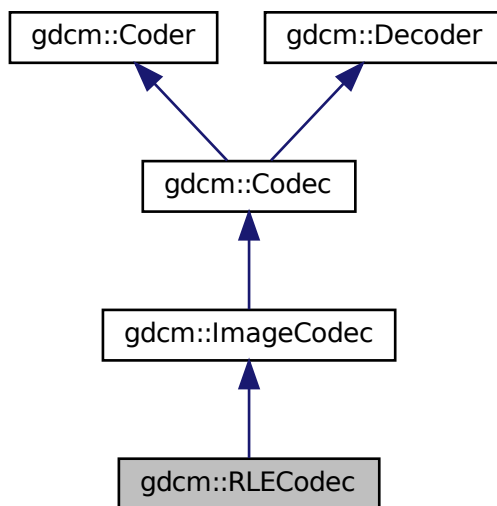
- [gdcmRescaler.h](#)

## 10.258 gdcm::RLECodec Class Reference

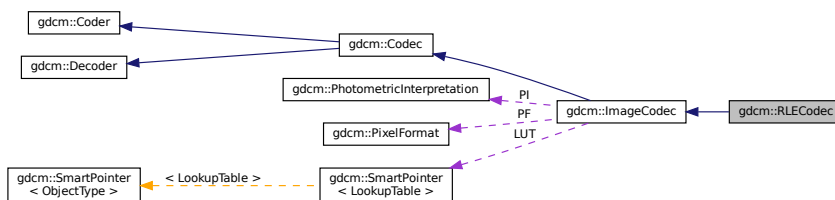
Class to do RLE.

```
#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)

## 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

## Friends

- class [ImageRegionReader](#)

## Additional Inherited Members

### 10.258.1 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).

### 10.258.2 Constructor & Destructor Documentation



### 10.258.2.1 RLECodec()

```
gdcm::RLECodec::RLECodec ( )
```

### 10.258.2.2 ~RLECodec()

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

## 10.258.3 Member Function Documentation

### 10.258.3.1 AppendFrameEncode()

```
bool gdcm::RLECodec::AppendFrameEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.258.3.2 AppendRowEncode()

```
bool gdcm::RLECodec::AppendRowEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.258.3.3 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](#).

#### 10.258.3.4 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](#).

#### 10.258.3.5 Clone()

```
ImageCodec* gdcm::RLECodec::Clone ( ) const [override], [virtual]
```

Implements [gdcm::ImageCodec](#).

#### 10.258.3.6 Code()

```
bool gdcm::RLECodec::Code (
    DataElement const & in_,
    DataElement & out_ ) [override], [virtual]
```

Code.

Reimplemented from [gdcm::Coder](#).

#### 10.258.3.7 Decode()

```
bool gdcm::RLECodec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcm::ImageCodec](#).

### 10.258.3.8 DecodeByStreams()

```
bool gdcm::RLECodec::DecodeByStreams (
    std::istream & is,
    std::ostream & os ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.258.3.9 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]
```

### 10.258.3.10 GetBufferLength()

```
unsigned long gdcm::RLECodec::GetBufferLength ( ) const [inline]
```

### 10.258.3.11 GetHeaderInfo()

```
bool gdcm::RLECodec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.258.3.12 IsFrameEncoder()

```
bool gdcm::RLECodec::IsFrameEncoder ( ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### 10.258.3.13 IsRowEncoder()

```
bool gdcm::RLECodec::IsRowEncoder ( ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### 10.258.3.14 SetBufferLength()

```
void gdcm::RLECodec::SetBufferLength (
    unsigned long l ) [inline]
```

#### 10.258.3.15 SetLength()

```
void gdcm::RLECodec::SetLength (
    unsigned long l ) [inline]
```

#### 10.258.3.16 StartEncode()

```
bool gdcm::RLECodec::StartEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### 10.258.3.17 StopEncode()

```
bool gdcm::RLECodec::StopEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.258.4 Friends And Related Function Documentation

### 10.258.4.1 ImageRegionReader

```
friend class ImageRegionReader [friend]
```

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

- [gdcmRLECodec.h](#)

## 10.259 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

### 10.259.1 Detailed Description

[RoleSelectionSub](#).

PS 3.7 [Table D.3-9](#) SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

### 10.259.2 Constructor & Destructor Documentation

#### 10.259.2.1 RoleSelectionSub()

```
gdcm::network::RoleSelectionSub::RoleSelectionSub ( )
```

### 10.259.3 Member Function Documentation

#### 10.259.3.1 Print()

```
void gdcM::network::RoleSelectionSub::Print (
    std::ostream & os ) const
```

#### 10.259.3.2 Read()

```
std::istream& gdcM::network::RoleSelectionSub::Read (
    std::istream & is )
```

#### 10.259.3.3 SetTuple()

```
void gdcM::network::RoleSelectionSub::SetTuple (
    const char * uid,
    uint8_t scurole,
    uint8_t scprole )
```

#### 10.259.3.4 Size()

```
size_t gdcM::network::RoleSelectionSub::Size ( ) const
```

#### 10.259.3.5 Write()

```
const std::ostream& gdcM::network::RoleSelectionSub::Write (
    std::ostream & os ) const
```

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

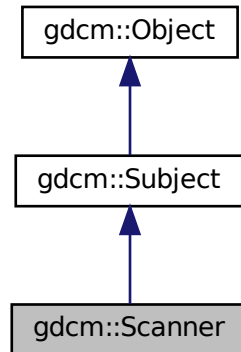
- [gdcMRoleSelectionSub.h](#)

## 10.260 gdcm::Scanner Class Reference

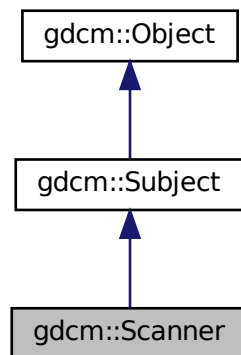
[Scanner.](#)

```
#include <gdcmScanner.h>
```

Inheritance diagram for gdcm::Scanner:



Collaboration diagram for gdcm::Scanner:



### 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

- [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::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 !*

## 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)

## Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [Scanner](#) &s)

### 10.260.1 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](#).

### 10.260.2 Member Typedef Documentation

#### 10.260.2.1 ConstIterator

```
typedef MappingType::const_iterator gdcm::Scanner::ConstIterator
```

### 10.260.2.2 MappingType

```
typedef std::map<const char *, TagToValue, ltstr> gdcm::Scanner::MappingType
```

### 10.260.2.3 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)

### 10.260.2.4 TagToValueValueType

```
typedef TagToValue::value_type gdcm::Scanner::TagToValueValueType
```

### 10.260.2.5 ValuesType

```
typedef std::set< std::string > gdcm::Scanner::ValuesType
```

## 10.260.3 Constructor & Destructor Documentation

### 10.260.3.1 Scanner()

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

### 10.260.3.2 ~Scanner()

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

## 10.260.4 Member Function Documentation

#### 10.260.4.1 AddPrivateTag()

```
void gdcm::Scanner::AddPrivateTag (
    PrivateTag const & t )
```

#### 10.260.4.2 AddSkipTag()

```
void gdcm::Scanner::AddSkipTag (
    Tag const & t )
```

Add a tag that will need to be skipped. Those are root level skip tags.

#### 10.260.4.3 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](#).

#### 10.260.4.4 Begin()

```
ConstIterator gdcm::Scanner::Begin ( ) const [inline]
```

#### 10.260.4.5 ClearSkipTags()

```
void gdcm::Scanner::ClearSkipTags ( )
```

#### 10.260.4.6 ClearTags()

```
void gdcm::Scanner::ClearTags ( )
```

#### 10.260.4.7 End()

```
ConstIterator gdcm::Scanner::End ( ) const [inline]
```

#### 10.260.4.8 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'

#### 10.260.4.9 GetFilenameFromTagToValue()

```
const char* gdcm::Scanner::GetFilenameFromTagToValue (
    Tag const & t,
    const char * valuref ) const
```

Will loop over all files and return the first file where value match the reference value 'valuref'

#### 10.260.4.10 GetFileNames()

```
Directory::FileNamesType const& gdcm::Scanner::GetFileNames ( ) const [inline]
```

#### 10.260.4.11 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](#).

#### 10.260.4.12 GetMapping()

```
TagToValue const& gdcm::Scanner::GetMapping (
    const char * filename ) const
```

Get the std::map mapping filenames to value for file 'filename'.

##### Examples

[DumpToSQLITE3.cxx](#).

#### 10.260.4.13 GetMappingFromTagToValue()

```
TagToValue const& gdcm::Scanner::GetMappingFromTagToValue (
    Tag const & t,
    const char * value ) const
```

See [GetFilenameFromTagToValue\(\)](#). This is simply GetFilenameFromTagToValue followed.

#### 10.260.4.14 GetMappings()

```
MappingType const& gdcm::Scanner::GetMappings ( ) const [inline]
```

Mappings are the mapping from a particular tag to the map, mapping filename to value:

#### 10.260.4.15 GetOrderedValues()

```
Directory::FileNamesType 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)

#### 10.260.4.16 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 !

#### 10.260.4.17 GetValues() [1/2]

```
ValueType const& gdcm::Scanner::GetValues ( ) const [inline]
```

Get all the values found (in lexicographic order)

##### Examples

[SortImage.cxx](#), and [VolumeSorter.cxx](#).

#### 10.260.4.18 GetValues() [2/2]

```
ValueType gdcm::Scanner::GetValues (
    Tag const & t ) const
```

Get all the values found (in lexicographic order) associated with [Tag](#) 't'.

#### 10.260.4.19 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](#).

#### 10.260.4.20 New()

```
static SmartPointer<Scanner> gdcm::Scanner::New ( ) [inline], [static]
```

for wrapped language: instantiate a reference counted object

#### 10.260.4.21 Print()

```
void gdcmm::Scanner::Print (
    std::ostream & os ) const [override], [virtual]
```

Print result.

Reimplemented from [gdcmm::Object](#).

#### 10.260.4.22 PrintTable()

```
void gdcmm::Scanner::PrintTable (
    std::ostream & os ) const
```

#### 10.260.4.23 ProcessPublicTag()

```
void gdcmm::Scanner::ProcessPublicTag (
    StringFilter & sf,
    const char * filename ) [protected]
```

#### 10.260.4.24 Scan()

```
bool gdcmm::Scanner::Scan (
    Directory::FilenameType const & filenames )
```

Start the scan !

#### Examples

[DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).

### 10.260.5 Friends And Related Function Documentation

### 10.260.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const Scanner & s ) [friend]
```

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

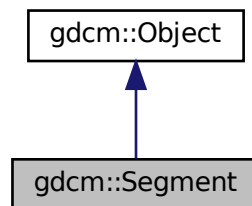
- [gdcmScanner.h](#)

## 10.261 gdcm::Segment Class Reference

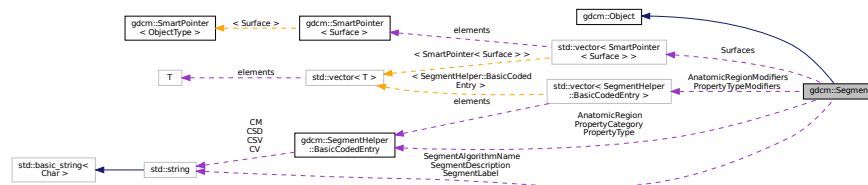
This class defines a segment.

```
#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)

## 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

### 10.261.1 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

### 10.261.2 Member Typedef Documentation

#### 10.261.2.1 BasicCodedEntryVector

```
typedef std::vector< SegmentHelper::BasicCodedEntry > gdcm::Segment::BasicCodedEntryVector
```

#### 10.261.2.2 SurfaceVector

```
typedef std::vector< SmartPointer< Surface > > gdcm::Segment::SurfaceVector
```

### 10.261.3 Member Enumeration Documentation

#### 10.261.3.1 ALGOType

```
enum gdcm::Segment::ALGOType
```

## Enumerator

AUTOMATIC	
SEMIAUTOMATIC	
MANUAL	
ALGOType_END	

## 10.261.4 Constructor & Destructor Documentation

### 10.261.4.1 Segment()

```
gdcm::Segment::Segment ( )
```

### 10.261.4.2 ~Segment()

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

## 10.261.5 Member Function Documentation

### 10.261.5.1 AddSurface()

```
void gdcm::Segment::AddSurface (
    SmartPointer< Surface > surface )
```

### 10.261.5.2 GetALGOType()

```
static ALGOType gdcm::Segment::GetALGOType (
    const char * type ) [static]
```

### 10.261.5.3 GetALGOTypeString()

```
static const char* gdcm::Segment::GetALGOTypeString (
    ALGOType type ) [static]
```

### 10.261.5.4 GetAnatomicRegion() [1/2]

```
SegmentHelper::BasicCodedEntry& gdcm::Segment::GetAnatomicRegion ( )
```

### 10.261.5.5 GetAnatomicRegion() [2/2]

```
SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetAnatomicRegion ( ) const
```

### 10.261.5.6 GetAnatomicRegionModifiers() [1/2]

```
BasicCodedEntryVector& gdcm::Segment::GetAnatomicRegionModifiers ( )
```

### 10.261.5.7 GetAnatomicRegionModifiers() [2/2]

```
BasicCodedEntryVector const& gdcm::Segment::GetAnatomicRegionModifiers ( ) const
```

### 10.261.5.8 GetPropertyCategory() [1/2]

```
SegmentHelper::BasicCodedEntry& gdcm::Segment::GetPropertyCategory ( )
```

### 10.261.5.9 GetPropertyCategory() [2/2]

```
SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetPropertyCategory ( ) const
```

**10.261.5.10 GetPropertyType() [1/2]**

```
SegmentHelper::BasicCodedEntry& gdcm::Segment::GetPropertyType ( )
```

**10.261.5.11 GetPropertyType() [2/2]**

```
SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetPropertyType ( ) const
```

**10.261.5.12 GetPropertyTypeModifiers() [1/2]**

```
BasicCodedEntryVector& gdcm::Segment::GetPropertyTypeModifiers ( )
```

**10.261.5.13 GetPropertyTypeModifiers() [2/2]**

```
BasicCodedEntryVector const& gdcm::Segment::GetPropertyTypeModifiers ( ) const
```

**10.261.5.14 GetSegmentAlgorithmName()**

```
const char* gdcm::Segment::GetSegmentAlgorithmName ( ) const
```

**10.261.5.15 GetSegmentAlgorithmType()**

```
ALGOType gdcm::Segment::GetSegmentAlgorithmType ( ) const
```

**10.261.5.16 GetSegmentDescription()**

```
const char* gdcm::Segment::GetSegmentDescription ( ) const
```

**10.261.5.17 GetSegmentLabel()**

```
const char* gdcM::Segment::GetSegmentLabel ( ) const
```

**10.261.5.18 GetSegmentNumber()**

```
unsigned short gdcM::Segment::GetSegmentNumber ( ) const
```

**10.261.5.19 GetSurface()**

```
SmartPointer< Surface > gdcM::Segment::GetSurface (
    const unsigned int idx = 0 ) const
```

**10.261.5.20 GetSurfaceCount()**

```
unsigned long gdcM::Segment::GetSurfaceCount ( )
```

**10.261.5.21 GetSurfaces() [1/2]**

```
SurfaceVector& gdcM::Segment::GetSurfaces ( )
```

**10.261.5.22 GetSurfaces() [2/2]**

```
SurfaceVector const& gdcM::Segment::GetSurfaces ( ) const
```

**10.261.5.23 SetAnatomicRegion()**

```
void gdcM::Segment::SetAnatomicRegion (
    SegmentHelper::BasicCodedEntry const & BSE )
```

#### 10.261.5.24 SetAnatomicRegionModifiers()

```
void gdcm::Segment::SetAnatomicRegionModifiers (
    BasicCodedEntryVector const & BSEV )
```

#### 10.261.5.25 SetPropertyCategory()

```
void gdcm::Segment::SetPropertyCategory (
    SegmentHelper::BasicCodedEntry const & BSE )
```

#### 10.261.5.26 SetPropertyType()

```
void gdcm::Segment::SetPropertyType (
    SegmentHelper::BasicCodedEntry const & BSE )
```

#### 10.261.5.27 SetPropertyTypeModifiers()

```
void gdcm::Segment::SetPropertyTypeModifiers (
    BasicCodedEntryVector const & BSEV )
```

#### 10.261.5.28 SetSegmentAlgorithmName()

```
void gdcm::Segment::SetSegmentAlgorithmName (
    const char * name )
```

#### 10.261.5.29 SetSegmentAlgorithmType() [1/2]

```
void gdcm::Segment::SetSegmentAlgorithmType (
    ALGOType type )
```

**10.261.5.30 SetSegmentAlgorithmType() [2/2]**

```
void gdcM::Segment::SetSegmentAlgorithmType (
    const char * typeStr )
```

**10.261.5.31 SetSegmentDescription()**

```
void gdcM::Segment::SetSegmentDescription (
    const char * description )
```

**10.261.5.32 SetSegmentLabel()**

```
void gdcM::Segment::SetSegmentLabel (
    const char * label )
```

**10.261.5.33 SetSegmentNumber()**

```
void gdcM::Segment::SetSegmentNumber (
    const unsigned short num )
```

**10.261.5.34 SetSurfaceCount()**

```
void gdcM::Segment::SetSurfaceCount (
    const unsigned long nb )
```

**10.261.6 Member Data Documentation****10.261.6.1 AnatomicRegion**

```
SegmentHelper::BasicCodedEntry gdcM::Segment::AnatomicRegion [protected]
```



### 10.261.6.2 AnatomicRegionModifiers

`BasicCodedEntryVector` `gdcm::Segment::AnatomicRegionModifiers` [protected]

### 10.261.6.3 PropertyCategory

`SegmentHelper::BasicCodedEntry` `gdcm::Segment::PropertyCategory` [protected]

### 10.261.6.4 PropertyType

`SegmentHelper::BasicCodedEntry` `gdcm::Segment::PropertyType` [protected]

### 10.261.6.5 PropertyTypeModifiers

`BasicCodedEntryVector` `gdcm::Segment::PropertyTypeModifiers` [protected]

### 10.261.6.6 SegmentAlgorithmName

`std::string` `gdcm::Segment::SegmentAlgorithmName` [protected]

### 10.261.6.7 SegmentAlgorithmType

`ALGOType` `gdcm::Segment::SegmentAlgorithmType` [protected]

### 10.261.6.8 SegmentDescription

`std::string` `gdcm::Segment::SegmentDescription` [protected]

### 10.261.6.9 SegmentLabel

```
std::string gdcM::Segment::SegmentLabel [protected]
```

### 10.261.6.10 SegmentNumber

```
unsigned short gdcM::Segment::SegmentNumber [protected]
```

### 10.261.6.11 SurfaceCount

```
unsigned long gdcM::Segment::SurfaceCount [protected]
```

### 10.261.6.12 Surfaces

```
SurfaceVector gdcM::Segment::Surfaces [protected]
```

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

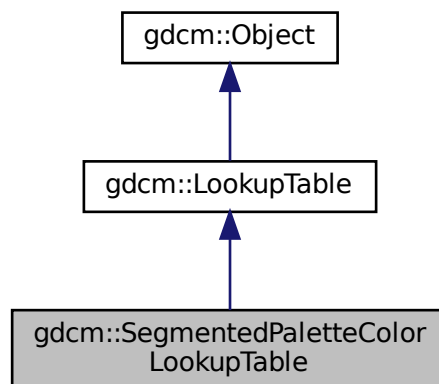
- [gdcMSegment.h](#)

## 10.262 gdcM::SegmentedPaletteColorLookupTable Class Reference

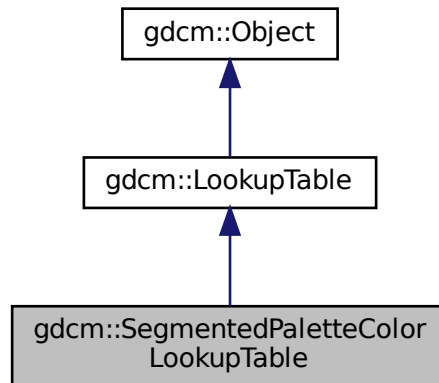
[SegmentedPaletteColorLookupTable](#) class.

```
#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](#).*

## Additional Inherited Members

### 10.262.1 Detailed Description

[SegmentedPaletteColorLookupTable](#) class.

### 10.262.2 Constructor & Destructor Documentation

#### 10.262.2.1 [SegmentedPaletteColorLookupTable](#)()

```
gdcm::SegmentedPaletteColorLookupTable::SegmentedPaletteColorLookupTable ( )
```

### 10.262.2.2 ~SegmentedPaletteColorLookupTable()

```
gdcmm::SegmentedPaletteColorLookupTable::~~SegmentedPaletteColorLookupTable ( ) [override]
```

## 10.262.3 Member Function Documentation

### 10.262.3.1 Print()

```
void gdcmm::SegmentedPaletteColorLookupTable::Print (
    std::ostream & ) const [inline], [override], [virtual]
```

Reimplemented from [gdcmm::LookupTable](#).

### 10.262.3.2 SetLUT()

```
void gdcmm::SegmentedPaletteColorLookupTable::SetLUT (
    LookupTableType type,
    const unsigned char * array,
    unsigned int length ) [override], [virtual]
```

Initialize a [SegmentedPaletteColorLookupTable](#).

Reimplemented from [gdcmm::LookupTable](#).

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

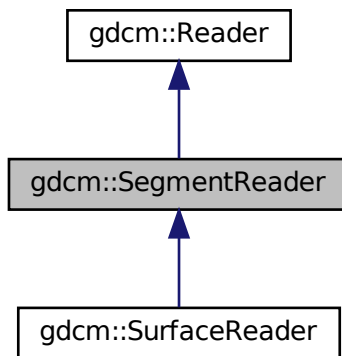
- [gdcmmSegmentedPaletteColorLookupTable.h](#)

## 10.263 gdcm::SegmentReader Class Reference

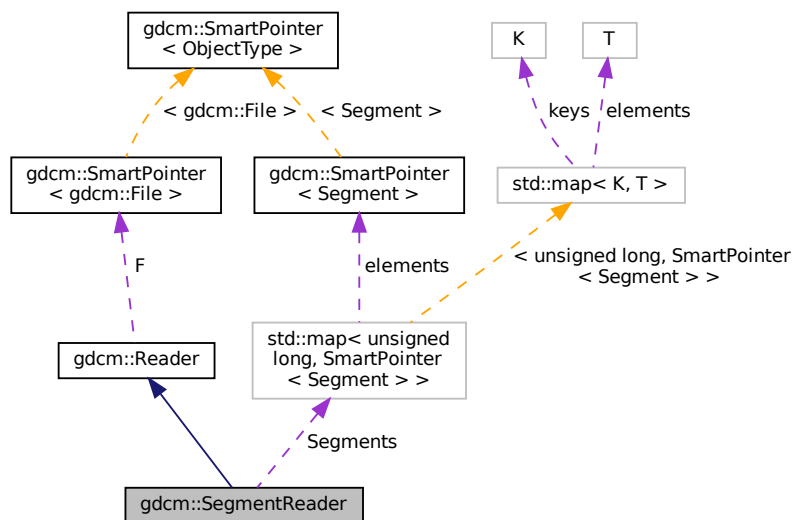
This class defines a segment reader.

```
#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](#) ()
- const [SegmentVector](#) [GetSegments](#) () const
- bool [Read](#) () override

*Read.*

## 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 Attributes

- [SegmentMap](#) [Segments](#)

### 10.263.1 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

### 10.263.2 Member Typedef Documentation

### 10.263.2.1 SegmentMap

```
typedef std::map< unsigned long, SmartPointer< Segment > > gdcm::SegmentReader::SegmentMap [protected]
```

### 10.263.2.2 SegmentVector

```
typedef std::vector< SmartPointer< Segment > > gdcm::SegmentReader::SegmentVector
```

## 10.263.3 Constructor & Destructor Documentation

### 10.263.3.1 SegmentReader()

```
gdcm::SegmentReader::SegmentReader ( )
```

### 10.263.3.2 ~SegmentReader()

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

## 10.263.4 Member Function Documentation

### 10.263.4.1 GetSegments() [1/2]

```
SegmentVector gdcm::SegmentReader::GetSegments ( )
```

### 10.263.4.2 GetSegments() [2/2]

```
const SegmentVector gdcm::SegmentReader::GetSegments ( ) const
```

#### 10.263.4.3 Read()

```
bool gdcm::SegmentReader::Read ( ) [override], [virtual]
```

Read.

Reimplemented from [gdcm::Reader](#).

Reimplemented in [gdcm::SurfaceReader](#).

#### 10.263.4.4 ReadSegment()

```
bool gdcm::SegmentReader::ReadSegment (
    const Item & segmentItem,
    const unsigned int idx ) [protected]
```

#### 10.263.4.5 ReadSegments()

```
bool gdcm::SegmentReader::ReadSegments ( ) [protected]
```

### 10.263.5 Member Data Documentation

#### 10.263.5.1 Segments

```
SegmentMap gdcm::SegmentReader::Segments [protected]
```

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

- [gdcmSegmentReader.h](#)

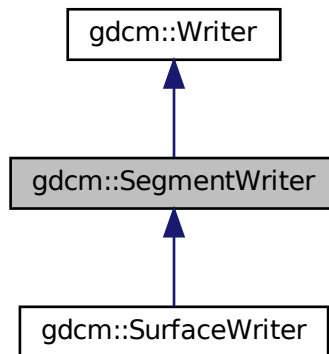


## 10.264 gdcm::SegmentWriter Class Reference

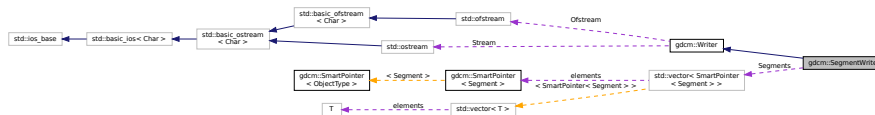
This class defines a segment writer.

```
#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.*

## Protected Member Functions

- bool [PrepareWrite](#) ()

## Protected Attributes

- [SegmentVector](#) [Segments](#)

### 10.264.1 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

### 10.264.2 Member Typedef Documentation

#### 10.264.2.1 SegmentVector

```
typedef std::vector< SmartPointer< Segment > > gdcm::SegmentWriter::SegmentVector
```

### 10.264.3 Constructor & Destructor Documentation

#### 10.264.3.1 SegmentWriter()

```
gdcm::SegmentWriter::SegmentWriter ( )
```

#### 10.264.3.2 ~SegmentWriter()

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

## 10.264.4 Member Function Documentation

### 10.264.4.1 AddSegment()

```
void gdcm::SegmentWriter::AddSegment (
    SmartPointer< Segment > segment )
```

### 10.264.4.2 GetNumberOfSegments()

```
unsigned int gdcm::SegmentWriter::GetNumberOfSegments ( ) const
```

### 10.264.4.3 GetSegment()

```
SmartPointer< Segment > gdcm::SegmentWriter::GetSegment (
    const unsigned int idx = 0 ) const
```

### 10.264.4.4 GetSegments() [1/2]

```
SegmentVector& gdcm::SegmentWriter::GetSegments ( )
```

### 10.264.4.5 GetSegments() [2/2]

```
const SegmentVector& gdcm::SegmentWriter::GetSegments ( ) const
```

### 10.264.4.6 PrepareWrite()

```
bool gdcm::SegmentWriter::PrepareWrite ( ) [protected]
```

#### 10.264.4.7 SetNumberOfSegments()

```
void gdcm::SegmentWriter::SetNumberOfSegments (
    const unsigned int size )
```

#### 10.264.4.8 SetSegments()

```
void gdcm::SegmentWriter::SetSegments (
    SegmentVector & segments )
```

#### 10.264.4.9 Write()

```
bool gdcm::SegmentWriter::Write ( ) [override], [virtual]
```

Write.

Reimplemented from [gdcm::Writer](#).

Reimplemented in [gdcm::SurfaceWriter](#).

### 10.264.5 Member Data Documentation

#### 10.264.5.1 Segments

```
SegmentVector gdcm::SegmentWriter::Segments [protected]
```

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

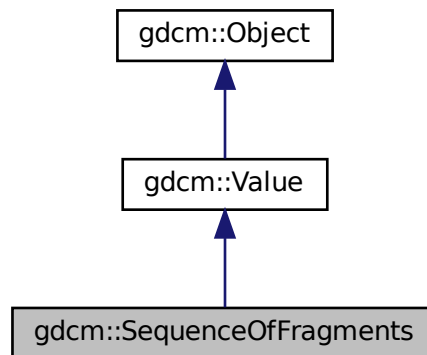
- [gdcmSegmentWriter.h](#)

## 10.265 gdcm::SequenceOfFragments Class Reference

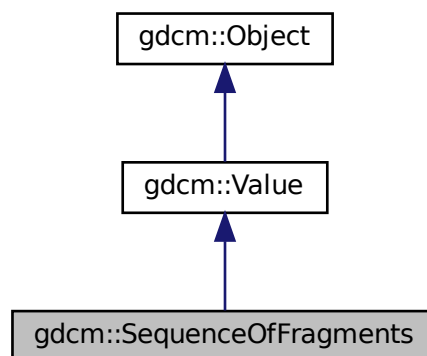
Class to represent a Sequence Of Fragments.

```
#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

## Static Public Member Functions

- static [SmartPointer](#)< [SequenceOfFragments](#) > [New](#) ()

## Additional Inherited Members

### 10.265.1 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

[FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), and [GetJPEGSamplePrecision.cxx](#).

## 10.265.2 Member Typedef Documentation

### 10.265.2.1 ConstIterator

```
typedef FragmentVector::const_iterator gdcm::SequenceOfFragments::ConstIterator
```

### 10.265.2.2 FragmentVector

```
typedef std::vector<Fragment> gdcm::SequenceOfFragments::FragmentVector
```

### 10.265.2.3 Iterator

```
typedef FragmentVector::iterator gdcm::SequenceOfFragments::Iterator
```

### 10.265.2.4 SizeType

```
typedef FragmentVector::size_type gdcm::SequenceOfFragments::SizeType
```

## 10.265.3 Constructor & Destructor Documentation

### 10.265.3.1 SequenceOfFragments()

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

constructor (UndefinedLength by default)

## 10.265.4 Member Function Documentation

#### 10.265.4.1 AddFragment()

```
void gdcm::SequenceOfFragments::AddFragment (
    Fragment const & item )
```

Appends a [Fragment](#) to the already added ones.

#### 10.265.4.2 Begin() [1/2]

```
Iterator gdcm::SequenceOfFragments::Begin ( ) [inline]
```

#### 10.265.4.3 Begin() [2/2]

```
ConstIterator gdcm::SequenceOfFragments::Begin ( ) const [inline]
```

#### 10.265.4.4 Clear()

```
void gdcm::SequenceOfFragments::Clear ( ) [override], [virtual]
```

Clear.

Implements [gdcm::Value](#).

#### 10.265.4.5 ComputeByteLength()

```
unsigned long gdcm::SequenceOfFragments::ComputeByteLength ( ) const
```

#### 10.265.4.6 ComputeLength()

```
VL gdcm::SequenceOfFragments::ComputeLength ( ) const
```



#### 10.265.4.7 End() [1/2]

```
Iterator gdcm::SequenceOfFragments::End ( ) [inline]
```

#### 10.265.4.8 End() [2/2]

```
ConstIterator gdcm::SequenceOfFragments::End ( ) const [inline]
```

#### 10.265.4.9 GetBuffer()

```
bool gdcm::SequenceOfFragments::GetBuffer (
    char * buffer,
    unsigned long length ) const
```

#### 10.265.4.10 GetFragBuffer()

```
bool gdcm::SequenceOfFragments::GetFragBuffer (
    unsigned int fragNb,
    char * buffer,
    unsigned long & length ) const
```

#### 10.265.4.11 GetFragment()

```
const Fragment& gdcm::SequenceOfFragments::GetFragment (
    SizeType num ) const
```

#### Examples

[FixBrokenJ2K.cxx](#), and [FixJAIBugJPEGLS.cxx](#).

#### 10.265.4.12 GetLength()

```
VL gdcm::SequenceOfFragments::GetLength ( ) const [inline], [override], [virtual]
```

Returns the SQ length, as read from disk.

Implements [gdcm::Value](#).

#### 10.265.4.13 GetNumberOfFragments()

```
SizeType gdcm::SequenceOfFragments::GetNumberOfFragments ( ) const
```

Examples

[FixJAIBugJPEGLS.cxx](#).

#### 10.265.4.14 GetTable() [1/2]

```
BasicOffsetTable& gdcm::SequenceOfFragments::GetTable ( ) [inline]
```

#### 10.265.4.15 GetTable() [2/2]

```
const BasicOffsetTable& gdcm::SequenceOfFragments::GetTable ( ) const [inline]
```

#### 10.265.4.16 New()

```
static SmartPointer<SequenceOfFragments> gdcm::SequenceOfFragments::New ( ) [inline], [static]
```

#### 10.265.4.17 operator==( )

```
bool gdcm::SequenceOfFragments::operator== (
    const Value & val ) const [inline], [override], [virtual]
```

Implements [gdcm::Value](#).

**10.265.4.18 Print()**

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

Reimplemented from [gdcm::Object](#).

**10.265.4.19 Read()**

```
template<typename TSwap >
std::istream& gdcm::SequenceOfFragments::Read (
    std::istream & is,
    bool readvalues = true ) [inline]
```

**10.265.4.20 ReadPreValue()**

```
template<typename TSwap >
std::istream& gdcm::SequenceOfFragments::ReadPreValue (
    std::istream & is ) [inline]
```

References [gdcmDebugMacro](#).

**10.265.4.21 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\(\)](#).

**10.265.4.22 SetLength()**

```
void gdcm::SequenceOfFragments::SetLength (
    VL length ) [inline], [override], [virtual]
```

Sets the actual SQ length.

Implements [gdcm::Value](#).

#### 10.265.4.23 Write()

```
template<typename TSwap >
std::ostream const& gdcM::SequenceOfFragments::Write (
    std::ostream & os ) const [inline]
```

References `gdcM::Tag::Write()`, and `gdcM::VL::Write()`.

#### 10.265.4.24 WriteBuffer()

```
bool gdcM::SequenceOfFragments::WriteBuffer (
    std::ostream & os ) const
```

#### Examples

[GetJPEGSamplePrecision.cxx](#).

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

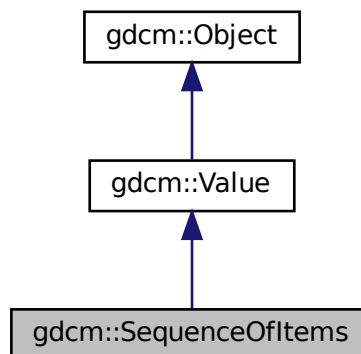
- [gdcMSequenceOfFragments.h](#)

## 10.266 gdcM::SequenceOfItems Class Reference

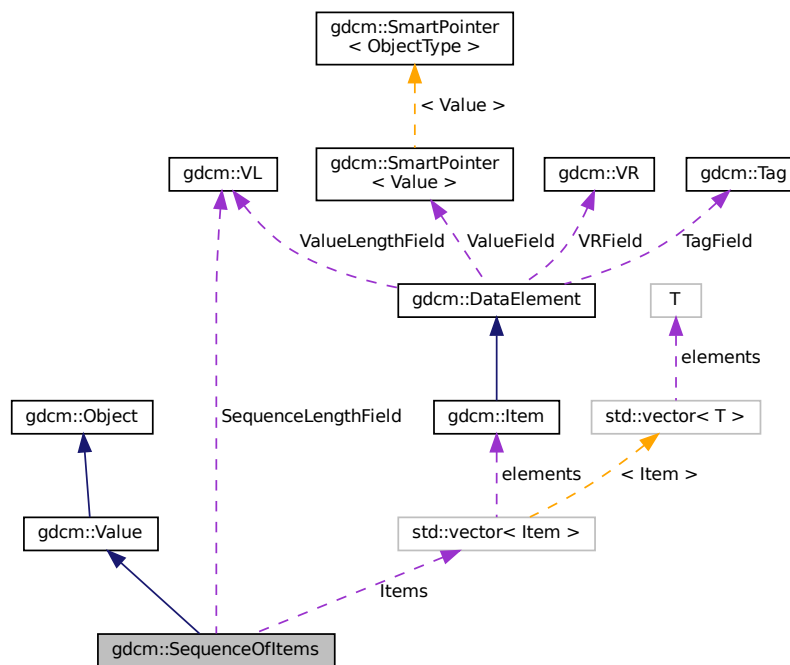
Class to represent a Sequence Of Items.

```
#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

## 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

### 10.266.1 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](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), and [GetSequenceUltrasound.cxx](#).

## 10.266.2 Member Typedef Documentation

### 10.266.2.1 ConstIterator

```
typedef ItemVector::const_iterator gdcm::SequenceOfItems::ConstIterator
```

### 10.266.2.2 ItemVector

```
typedef std::vector< Item > gdcm::SequenceOfItems::ItemVector
```

### 10.266.2.3 Iterator

```
typedef ItemVector::iterator gdcm::SequenceOfItems::Iterator
```

### 10.266.2.4 SizeType

```
typedef ItemVector::size_type gdcm::SequenceOfItems::SizeType
```

**Examples**

[DumpGEMSMovieGroup.cxx](#).

## 10.266.3 Constructor & Destructor Documentation

### 10.266.3.1 SequenceOfItems()

```
gdcmm::SequenceOfItems::SequenceOfItems ( ) [inline]
```

constructor (UndefinedLength by default)

## 10.266.4 Member Function Documentation

### 10.266.4.1 AddItem()

```
void gdcmm::SequenceOfItems::AddItem (
    Item const & item )
```

Appends an [Item](#) to the already added ones.

#### Examples

[Extracting\\_All\\_Resolution.cxx](#).

### 10.266.4.2 AddNewUndefinedLengthItem()

```
Item& gdcmm::SequenceOfItems::AddNewUndefinedLengthItem ( )
```

Appends an [Item](#) to the already added ones.

### 10.266.4.3 Begin() [1/2]

```
Iterator gdcmm::SequenceOfItems::Begin ( ) [inline]
```

### 10.266.4.4 Begin() [2/2]

```
ConstIterator gdcmm::SequenceOfItems::Begin ( ) const [inline]
```



#### 10.266.4.5 Clear()

```
void gdcm::SequenceOfItems::Clear ( ) [override], [virtual]
```

remove all items within the sequence

Implements [gdcm::Value](#).

#### 10.266.4.6 ComputeLength()

```
template<typename TDE >  
VL gdcm::SequenceOfItems::ComputeLength ( ) const
```

#### 10.266.4.7 End() [1/2]

```
Iterator gdcm::SequenceOfItems::End ( ) [inline]
```

#### 10.266.4.8 End() [2/2]

```
ConstIterator gdcm::SequenceOfItems::End ( ) const [inline]
```

#### 10.266.4.9 FindDataElement()

```
bool gdcm::SequenceOfItems::FindDataElement (   
    const Tag & t ) const
```

#### 10.266.4.10 GetItem() [1/2]

```
Item& gdcm::SequenceOfItems::GetItem (   
    SizeType position )
```

#### 10.266.4.11 GetItem() [2/2]

```
const Item& gdcM::SequenceOfItems::GetItem (
    SizeType position ) const
```

##### Examples

[DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [ExtractEncryptedContent.cxx](#), and [GetSequenceUltrasound.cxx](#).

#### 10.266.4.12 GetLength()

```
VL gdcM::SequenceOfItems::GetLength ( ) const [inline], [override], [virtual]
```

Returns the SQ length, as read from disk.

Implements [gdcM::Value](#).

#### 10.266.4.13 GetNumberOfItems()

```
SizeType gdcM::SequenceOfItems::GetNumberOfItems ( ) const [inline]
```

##### Examples

[DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [ExtractEncryptedContent.cxx](#), and [GetSequenceUltrasound.cxx](#).

#### 10.266.4.14 IsEmpty()

```
bool gdcM::SequenceOfItems::IsEmpty ( ) const [inline]
```

#### 10.266.4.15 IsUndefinedLength()

```
bool gdcM::SequenceOfItems::IsUndefinedLength ( ) const [inline]
```

return if [Value](#) Length if of undefined length

#### 10.266.4.16 New()

```
static SmartPointer<SequenceOfItems> gdcm::SequenceOfItems::New ( ) [inline], [static]
```

##### Examples

[NewSequence.cs](#).

#### 10.266.4.17 operator=()

```
SequenceOfItems& gdcm::SequenceOfItems::operator= (
    const SequenceOfItems & val ) [inline]
```

References [Items](#), and [SequenceLengthField](#).

#### 10.266.4.18 operator==()

```
bool gdcm::SequenceOfItems::operator== (
    const Value & val ) const [inline], [override], [virtual]
```

Implements [gdcm::Value](#).

References [Items](#), and [SequenceLengthField](#).

#### 10.266.4.19 Print()

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

Reimplemented from [gdcm::Object](#).

#### 10.266.4.20 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\(\)](#).

**10.266.4.21 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

**10.266.4.22 SetLength()**

```
void gdcM::SequenceOfItems::SetLength (
    VL length ) [inline], [override], [virtual]
```

Sets the actual SQ length.

Implements [gdcM::Value](#).

**10.266.4.23 SetLengthToUndefined()**

```
void gdcM::SequenceOfItems::SetLengthToUndefined ( )
```

Properly set the Sequence of [Item](#) to be undefined length.

**10.266.4.24 SetNumberOfItems()**

```
void gdcM::SequenceOfItems::SetNumberOfItems (
    SizeType n ) [inline]
```

**10.266.4.25 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\(\)](#).

**10.266.5 Member Data Documentation**

### 10.266.5.1 Items

`ItemVector` `gdcm::SequenceOfItems::Items`

Vector of Sequence Items.

Referenced by `operator=()`, and `operator==()`.

### 10.266.5.2 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:

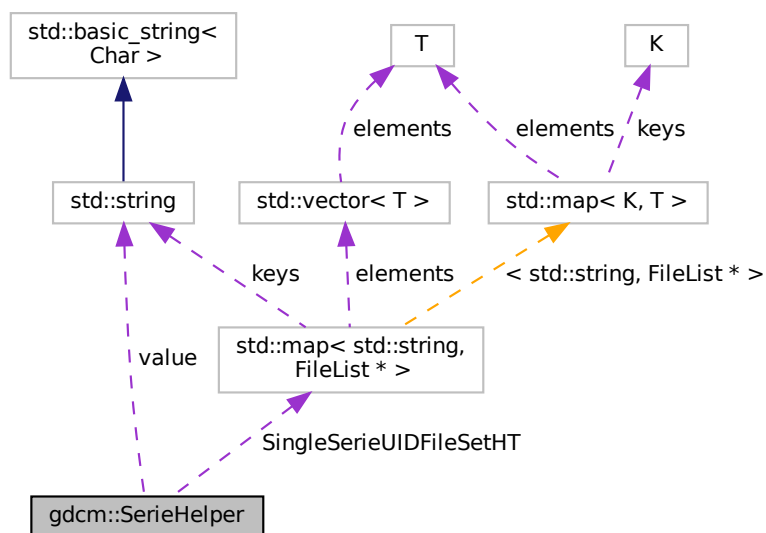
- [gdcmSequenceOfItems.h](#)

## 10.267 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](#) = struct 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](#)

### 10.267.1 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](#)

## 10.267.2 Member Typedef Documentation

### 10.267.2.1 Rule

```
using gdcm::SerieHelper::Rule = struct RuleStructure{ uint16_t group [protected]
```

### 10.267.2.2 SerieRestrictions

```
typedef std::vector<Rule> gdcm::SerieHelper::SerieRestrictions [protected]
```

### 10.267.2.3 SingleSerieUIDFileSetmap

```
typedef std::map<std::string, FileList *> gdcm::SerieHelper::SingleSerieUIDFileSetmap [protected]
```

## 10.267.3 Constructor & Destructor Documentation

### 10.267.3.1 SerieHelper()

```
gdcm::SerieHelper::SerieHelper ( )
```

### 10.267.3.2 ~SerieHelper()

```
gdcm::SerieHelper::~~SerieHelper ( )
```

## 10.267.4 Member Function Documentation

#### 10.267.4.1 AddFile()

```
bool gdcm::SerieHelper::AddFile (
    FileWithName & header ) [protected]
```

#### 10.267.4.2 AddFileName()

```
void gdcm::SerieHelper::AddFileName (
    std::string const & filename ) [protected]
```

#### 10.267.4.3 AddRestriction() [1/3]

```
void gdcm::SerieHelper::AddRestriction (
    const std::string & tag )
```

#### 10.267.4.4 AddRestriction() [2/3]

```
void gdcm::SerieHelper::AddRestriction (
    const Tag & tag ) [protected]
```

#### 10.267.4.5 AddRestriction() [3/3]

```
void gdcm::SerieHelper::AddRestriction (
    uint16_t group,
    uint16_t elem,
    std::string const & value,
    int op )
```

#### 10.267.4.6 Clear()

```
void gdcm::SerieHelper::Clear ( )
```



#### 10.267.4.7 CreateDefaultUniqueSeriesIdentifier()

```
void gdcm::SerieHelper::CreateDefaultUniqueSeriesIdentifier ( )
```

#### 10.267.4.8 CreateUniqueSeriesIdentifier()

```
std::string gdcm::SerieHelper::CreateUniqueSeriesIdentifier (
    File * inFile )
```

#### 10.267.4.9 FileNameOrdering()

```
bool gdcm::SerieHelper::FileNameOrdering (
    FileList * fileList ) [protected]
```

#### 10.267.4.10 GetFirstSingleSerieUIDFileSet()

```
FileList* gdcm::SerieHelper::GetFirstSingleSerieUIDFileSet ( )
```

#### 10.267.4.11 GetNextSingleSerieUIDFileSet()

```
FileList* gdcm::SerieHelper::GetNextSingleSerieUIDFileSet ( )
```

#### 10.267.4.12 ImageNumberOrdering()

```
bool gdcm::SerieHelper::ImageNumberOrdering (
    FileList * fileList ) [protected]
```

#### 10.267.4.13 ImagePositionPatientOrdering()

```
bool gdcm::SerieHelper::ImagePositionPatientOrdering (
    FileList * fileSet ) [protected]
```

**10.267.4.14 OrderFileList()**

```
void gdcm::SerieHelper::OrderFileList (
    FileList * fileSet )
```

**10.267.4.15 SetDirectory()**

```
void gdcm::SerieHelper::SetDirectory (
    std::string const & dir,
    bool recursive = false )
```

**10.267.4.16 SetLoadMode()**

```
void gdcm::SerieHelper::SetLoadMode (
    int ) [inline]
```

**10.267.4.17 SetUseSeriesDetails()**

```
void gdcm::SerieHelper::SetUseSeriesDetails (
    bool useSeriesDetails )
```

**10.267.4.18 UserOrdering()**

```
bool gdcm::SerieHelper::UserOrdering (
    FileList * fileSet ) [protected]
```

**10.267.5 Member Data Documentation****10.267.5.1 elem**

```
uint16_t gdcm::SerieHelper::elem [protected]
```

### 10.267.5.2 ItFileSetHt

```
SingleSerieUIDFileSetmap::iterator gdcm::SerieHelper::ItFileSetHt [protected]
```

### 10.267.5.3 op

```
int gdcm::SerieHelper::op [protected]
```

### 10.267.5.4 SingleSerieUIDFileSetHT

```
SingleSerieUIDFileSetmap gdcm::SerieHelper::SingleSerieUIDFileSetHT [protected]
```

### 10.267.5.5 value

```
std::string gdcm::SerieHelper::value [protected]
```

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

- [gdcmSerieHelper.h](#)

## 10.268 gdcm::Series Class Reference

[Series.](#)

```
#include <gdcmSeries.h>
```

### Public Member Functions

- [Series](#) ()=default

### 10.268.1 Detailed Description

[Series.](#)

## 10.268.2 Constructor & Destructor Documentation

### 10.268.2.1 Series()

```
gdcM::Series::Series ( ) [default]
```

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

- [gdcMSeries.h](#)

## 10.269 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

### 10.269.1 Detailed Description

PS 3.4 [Table B.3-1](#) SERVICE-CLASS-APPLICATION-INFORMATION (A-ASSOCIATE-RQ)

## 10.269.2 Constructor & Destructor Documentation

### 10.269.2.1 ServiceClassApplicationInformation()

```
gdcM::network::ServiceClassApplicationInformation::ServiceClassApplicationInformation ( )
```

### 10.269.3 Member Function Documentation

#### 10.269.3.1 Print()

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

#### 10.269.3.2 Read()

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

#### 10.269.3.3 SetTuple()

```
void gdcm::network::ServiceClassApplicationInformation::SetTuple (
    uint8_t levelofsupport,
    uint8_t levelofdigitalsig,
    uint8_t elementcoercion )
```

#### 10.269.3.4 Size()

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

#### 10.269.3.5 Write()

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

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

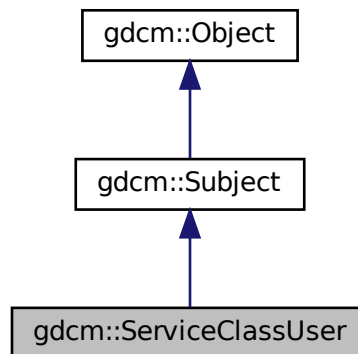
- [gdcmServiceClassApplicationInformation.h](#)

## 10.270 gdcmm::ServiceClassUser Class Reference

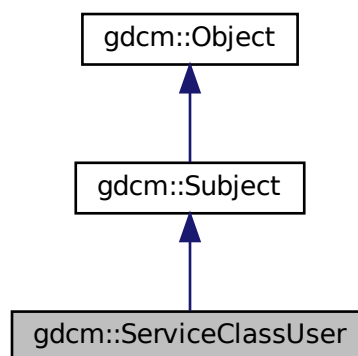
[ServiceClassUser](#).

```
#include <gdcmmServiceClassUser.h>
```

Inheritance diagram for gdcmm::ServiceClassUser:



Collaboration diagram for gdcmm::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.*

## Static Public Member Functions

- static [SmartPointer](#)< [ServiceClassUser](#) > [New](#) ()  
*for wrapped language: instanciate a reference counted object*

## Additional Inherited Members

### 10.270.1 Detailed Description

[ServiceClassUser](#).

#### Examples

[CStoreQtProgress.cxx](#).

### 10.270.2 Constructor & Destructor Documentation

#### 10.270.2.1 [ServiceClassUser](#)() [1/2]

```
gdcmm::ServiceClassUser::ServiceClassUser ( )
```

Construct a SCU with default:

- hostname = localhost
- port = 104

#### 10.270.2.2 [~ServiceClassUser](#)()

```
gdcmm::ServiceClassUser::~~ServiceClassUser ( ) [override]
```

#### 10.270.2.3 [ServiceClassUser](#)() [2/2]

```
gdcmm::ServiceClassUser::ServiceClassUser (
    const ServiceClassUser & ) [delete]
```



## 10.270.3 Member Function Documentation

### 10.270.3.1 GetAETitle()

```
const char* gdcm::ServiceClassUser::GetAETitle ( ) const
```

### 10.270.3.2 GetCalledAETitle()

```
const char* gdcm::ServiceClassUser::GetCalledAETitle ( ) const
```

### 10.270.3.3 GetTimeout()

```
double gdcm::ServiceClassUser::GetTimeout ( ) const
```

### 10.270.3.4 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](#).

### 10.270.3.5 IsPresentationContextAccepted()

```
bool gdcm::ServiceClassUser::IsPresentationContextAccepted (
    const PresentationContext & pc ) const
```

Return if the passed in presentation was accepted during association negotiation.

### 10.270.3.6 New()

```
static SmartPointer<ServiceClassUser> gdcm::ServiceClassUser::New ( ) [inline], [static]
```

for wrapped language: instantiate a reference counted object

### 10.270.3.7 operator=()

```
void gdcm::ServiceClassUser::operator= (
    const ServiceClassUser & ) [delete]
```

### 10.270.3.8 SendEcho()

```
bool gdcm::ServiceClassUser::SendEcho ( )
```

C-ECHO.

### 10.270.3.9 SendFind()

```
bool gdcm::ServiceClassUser::SendFind (
    const BaseRootQuery * query,
    std::vector< DataSet > & retDatasets )
```

C-FIND a query, return result are in retDatasets.

### 10.270.3.10 SendMove() [1/3]

```
bool gdcm::ServiceClassUser::SendMove (
    const BaseRootQuery * query,
    const char * outputdir )
```

Execute a C-MOVE, based on query, return files are written in outputdir.

**10.270.3.11 SendMove()** [2/3]

```
bool gdcm::ServiceClassUser::SendMove (
    const BaseRootQuery * query,
    std::vector< DataSet > & retDatasets )
```

Execute a C-MOVE, based on query, returned dataset are Implicit.

**10.270.3.12 SendMove()** [3/3]

```
bool gdcm::ServiceClassUser::SendMove (
    const BaseRootQuery * query,
    std::vector< File > & retFile )
```

Execute a C-MOVE, based on query, returned Files are stored in vector.

**10.270.3.13 SendStore()** [1/3]

```
bool gdcm::ServiceClassUser::SendStore (
    const char * filename )
```

Execute a C-STORE on file on disk, named filename.

**Examples**

[CStoreQtProgress.cxx](#).

**10.270.3.14 SendStore()** [2/3]

```
bool gdcm::ServiceClassUser::SendStore (
    DataSet const & ds )
```

Execute a C-STORE on a [DataSet](#), the transfer syntax used will be Implicit.

**10.270.3.15 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.

**10.270.3.16 SetAETitle()**

```
void gdcM::ServiceClassUser::SetAETitle (
    const char * aetitle )
```

set calling ae title

**10.270.3.17 SetCalledAETitle()**

```
void gdcM::ServiceClassUser::SetCalledAETitle (
    const char * aetitle )
```

set called ae title

**Examples**

[CStoreQtProgress.cxx](#).

**10.270.3.18 SetHostname()**

```
void gdcM::ServiceClassUser::SetHostname (
    const char * hostname )
```

Set the name of the called hostname (hostname or IP address)

**Examples**

[CStoreQtProgress.cxx](#).

**10.270.3.19 SetPort()**

```
void gdcM::ServiceClassUser::SetPort (
    uint16_t port )
```

Set port of remote host (called application)

**Examples**

[CStoreQtProgress.cxx](#).

### 10.270.3.20 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)

### 10.270.3.21 SetPresentationContexts()

```
void gdcm::ServiceClassUser::SetPresentationContexts (
    std::vector< PresentationContext > const & pcs )
```

Set the Presentation Context used for the Association.

#### Examples

[CStoreQtProgress.cxx](#).

### 10.270.3.22 SetTimeout()

```
void gdcm::ServiceClassUser::SetTimeout (
    double t )
```

set/get Timeout

#### Examples

[CStoreQtProgress.cxx](#).

### 10.270.3.23 StartAssociation()

```
bool gdcm::ServiceClassUser::StartAssociation ( )
```

Start the association. Need to call SetPresentationContexts before.

#### Examples

[CStoreQtProgress.cxx](#).

### 10.270.3.24 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](#)

## 10.271 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])

### 10.271.1 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

## 10.271.2 Constructor & Destructor Documentation

### 10.271.2.1 SHA1() [1/2]

```
gdcm::SHA1::SHA1 ( )
```

### 10.271.2.2 ~SHA1()

```
gdcm::SHA1::~~SHA1 ( )
```

### 10.271.2.3 SHA1() [2/2]

```
gdcm::SHA1::SHA1 (
    const SHA1 & ) [delete]
```

## 10.271.3 Member Function Documentation

### 10.271.3.1 Compute()

```
static bool gdcm::SHA1::Compute (
    const char * buffer,
    unsigned long buf_len,
    char digest_str[20 *2+1] ) [static]
```

### 10.271.3.2 ComputeFile()

```
static bool gdcm::SHA1::ComputeFile (
    const char * filename,
    char digest_str[20 *2+1] ) [static]
```

### 10.271.3.3 operator=()

```
void gdcM::SHA1::operator= (
    const SHA1 & ) [delete]
```

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

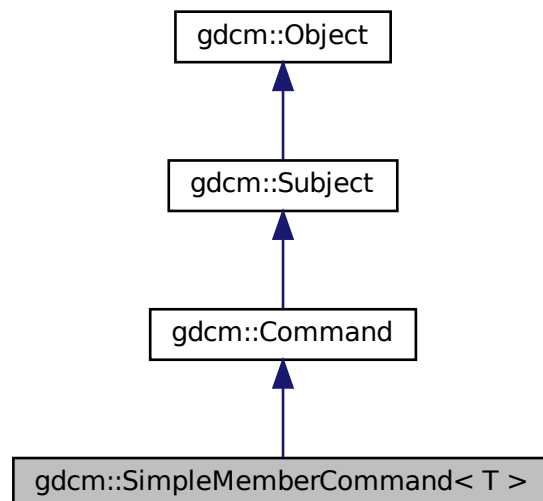
- [gdcMSHA1.h](#)

## 10.272 gdcM::SimpleMemberCommand< T > Class Template Reference

[Command](#) subclass that calls a pointer to a member function.

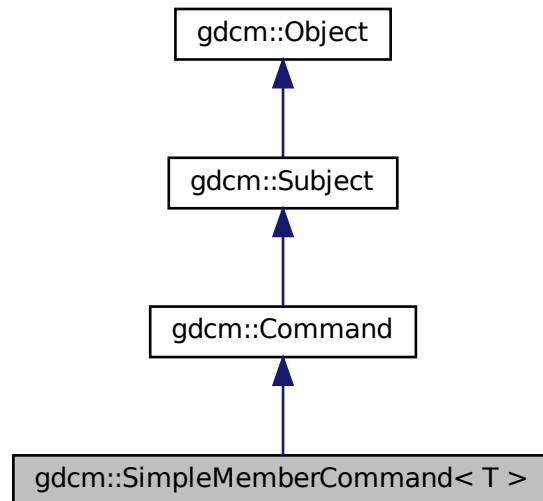
```
#include <gdcMCommand.h>
```

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`)

## Static Public Member Functions

- static `SmartPointer< SimpleMemberCommand > New` ()

## Protected Member Functions

- `SimpleMemberCommand` ()
- `~SimpleMemberCommand` () override=default

## Protected Attributes

- [TMemberFunctionPointer m\\_MemberFunction](#)
- [T \\* m\\_This](#)

### 10.272.1 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.

### 10.272.2 Member Typedef Documentation

#### 10.272.2.1 Self

```
template<typename T >
typedef SimpleMemberCommand gdcM::SimpleMemberCommand< T >::Self
```

Standard class typedefs.

#### 10.272.2.2 TMemberFunctionPointer

```
template<typename T >
typedef void(T::* gdcM::SimpleMemberCommand< T >::TMemberFunctionPointer) ()
```

A method callback.

### 10.272.3 Constructor & Destructor Documentation

#### 10.272.3.1 SimpleMemberCommand() [1/2]

```
template<typename T >
gdcM::SimpleMemberCommand< T >::SimpleMemberCommand (
    const Self & ) [delete]
```

### 10.272.3.2 SimpleMemberCommand() [2/2]

```
template<typename T >
gdcm::SimpleMemberCommand< T >::SimpleMemberCommand ( ) [inline], [protected]
```

Referenced by `gdcm::SimpleMemberCommand< T >::New()`.

### 10.272.3.3 ~SimpleMemberCommand()

```
template<typename T >
gdcm::SimpleMemberCommand< T >::~~SimpleMemberCommand ( ) [override], [protected], [default]
```

## 10.272.4 Member Function Documentation

### 10.272.4.1 Execute() [1/2]

```
template<typename T >
void gdcm::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 [gdcm::Command](#).

References `gdcm::SimpleMemberCommand< T >::m_MemberFunction`.

### 10.272.4.2 Execute() [2/2]

```
template<typename T >
void gdcm::SimpleMemberCommand< T >::Execute (
    Subject * ,
    const Event & ) [inline], [override], [virtual]
```

Invoke the callback function.

Implements [gdcm::Command](#).

References `gdcm::SimpleMemberCommand< T >::m_MemberFunction`.

### 10.272.4.3 New()

```
template<typename T >
static SmartPointer<SimpleMemberCommand> gdcM::SimpleMemberCommand< T >::New ( ) [inline],
[static]
```

Run-time type information (and related methods). Method for creation through the object factory.

References [gdcM::SimpleMemberCommand](#)< T >::SimpleMemberCommand().

### 10.272.4.4 operator=()

```
template<typename T >
void gdcM::SimpleMemberCommand< T >::operator= (
    const Self & ) [delete]
```

### 10.272.4.5 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.

## 10.272.5 Member Data Documentation

### 10.272.5.1 m\_MemberFunction

```
template<typename T >
TMemberFunctionPointer gdcM::SimpleMemberCommand< T >::m_MemberFunction [protected]
```

Referenced by [gdcM::SimpleMemberCommand](#)< T >::Execute(), and [gdcM::SimpleMemberCommand](#)< T >::Set↵  
CallbackFunction().

### 10.272.5.2 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](#)

## 10.273 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](#) ()

### 10.273.1 Detailed Description

[SimpleSubjectWatcher](#).

This is a typical [Subject](#) Watcher class. It will observe all events.

#### Examples

[SimpleScanner.cxx](#).

## 10.273.2 Constructor & Destructor Documentation

### 10.273.2.1 SimpleSubjectWatcher() [1/2]

```
gdcmm::SimpleSubjectWatcher::SimpleSubjectWatcher (
    Subject * s,
    const char * comment = "" )
```

### 10.273.2.2 ~SimpleSubjectWatcher()

```
virtual gdcmm::SimpleSubjectWatcher::~~SimpleSubjectWatcher ( ) [virtual]
```

### 10.273.2.3 SimpleSubjectWatcher() [2/2]

```
gdcmm::SimpleSubjectWatcher::SimpleSubjectWatcher (
    const SimpleSubjectWatcher & ) [delete]
```

## 10.273.3 Member Function Documentation

### 10.273.3.1 EndFilter()

```
virtual void gdcmm::SimpleSubjectWatcher::EndFilter ( ) [protected], [virtual]
```

### 10.273.3.2 operator=()

```
void gdcmm::SimpleSubjectWatcher::operator= (
    const SimpleSubjectWatcher & ) [delete]
```

### 10.273.3.3 ShowAbort()

```
virtual void gdcm::SimpleSubjectWatcher::ShowAbort ( ) [protected], [virtual]
```

### 10.273.3.4 ShowAnonymization()

```
virtual void gdcm::SimpleSubjectWatcher::ShowAnonymization (
    Subject * caller,
    const Event & evt ) [protected], [virtual]
```

### 10.273.3.5 ShowData()

```
virtual void gdcm::SimpleSubjectWatcher::ShowData (
    Subject * caller,
    const Event & evt ) [protected], [virtual]
```

### 10.273.3.6 ShowDataSet()

```
virtual void gdcm::SimpleSubjectWatcher::ShowDataSet (
    Subject * caller,
    const Event & evt ) [protected], [virtual]
```

### 10.273.3.7 ShowFileName()

```
virtual void gdcm::SimpleSubjectWatcher::ShowFileName (
    Subject * caller,
    const Event & evt ) [protected], [virtual]
```

### Examples

[SimpleScanner.cxx](#).

#### 10.273.3.8 ShowIteration()

```
virtual void gdcM::SimpleSubjectWatcher::ShowIteration ( ) [protected], [virtual]
```

#### 10.273.3.9 ShowProgress()

```
virtual void gdcM::SimpleSubjectWatcher::ShowProgress (
    Subject * caller,
    const Event & evt ) [protected], [virtual]
```

#### 10.273.3.10 StartFilter()

```
virtual void gdcM::SimpleSubjectWatcher::StartFilter ( ) [protected], [virtual]
```

#### 10.273.3.11 TestAbortOff()

```
void gdcM::SimpleSubjectWatcher::TestAbortOff ( ) [protected]
```

#### 10.273.3.12 TestAbortOn()

```
void gdcM::SimpleSubjectWatcher::TestAbortOn ( ) [protected]
```

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

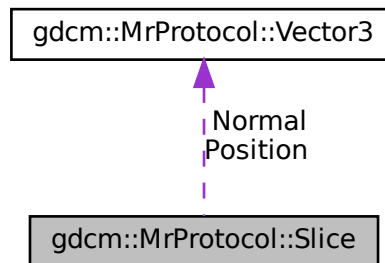
- [gdcMSimpleSubjectWatcher.h](#)



## 10.274 gdcm::MrProtocol::Slice Struct Reference

```
#include <gdcmMrProtocol.h>
```

Collaboration diagram for gdcm::MrProtocol::Slice:



### Public Attributes

- [Vector3 Normal](#)
- [Vector3 Position](#)

### 10.274.1 Member Data Documentation

#### 10.274.1.1 Normal

```
Vector3 gdcm::MrProtocol::Slice::Normal
```

#### 10.274.1.2 Position

```
Vector3 gdcm::MrProtocol::Slice::Position
```

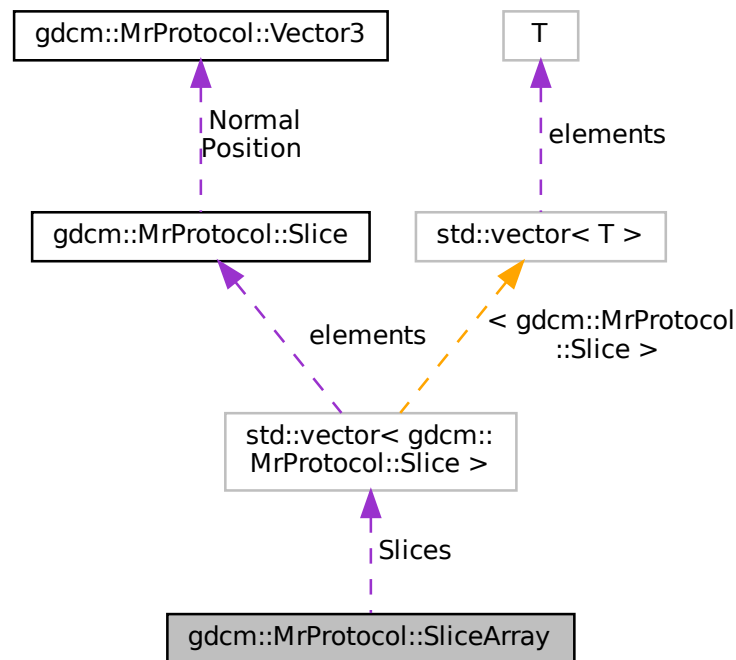
The documentation for this struct was generated from the following file:

- [gdcmMrProtocol.h](#)

## 10.275 gdcm::MrProtocol::SliceArray Struct Reference

```
#include <gdcmMrProtocol.h>
```

Collaboration diagram for gdcm::MrProtocol::SliceArray:



### Public Attributes

- `std::vector< Slice > Slices`

### 10.275.1 Member Data Documentation

#### 10.275.1.1 Slices

```
std::vector< Slice > gdcm::MrProtocol::SliceArray::Slices
```

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

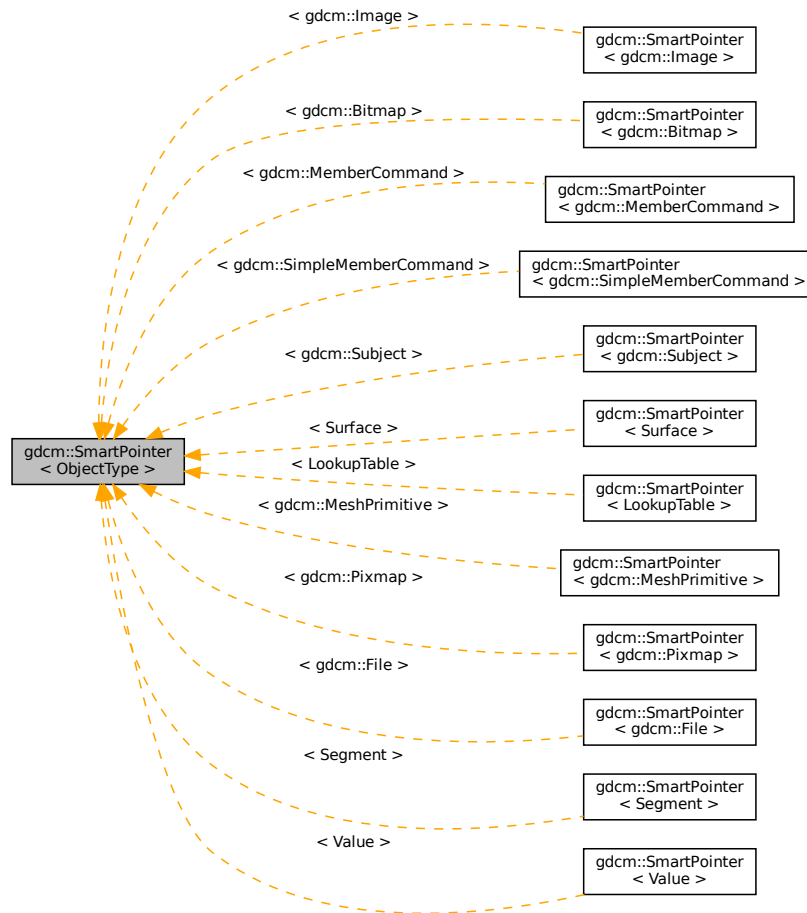
- [gdcmMrProtocol.h](#)

## 10.276 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.*

### 10.276.1 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/smarty.htm>

and `itk::SmartPointer`

#### Examples

[CStoreQtProgress.cxx](#), [ChangeSequenceUltrasound.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpToshibaDTI.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [LargeVRDSExplicit.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [SimpleScanner.cxx](#), [gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

### 10.276.2 Constructor & Destructor Documentation

**10.276.2.1 SmartPointer() [1/4]**

```
template<class ObjectType >  
gdcmm::SmartPointer< ObjectType >::SmartPointer ( ) [inline]
```

**10.276.2.2 SmartPointer() [2/4]**

```
template<class ObjectType >  
gdcmm::SmartPointer< ObjectType >::SmartPointer (   
    const SmartPointer< ObjectType > & p ) [inline]
```

**10.276.2.3 SmartPointer() [3/4]**

```
template<class ObjectType >  
gdcmm::SmartPointer< ObjectType >::SmartPointer (   
    ObjectType * p ) [inline]
```

**10.276.2.4 SmartPointer() [4/4]**

```
template<class ObjectType >  
gdcmm::SmartPointer< ObjectType >::SmartPointer (   
    ObjectType const & p ) [inline]
```

**10.276.2.5 ~SmartPointer()**

```
template<class ObjectType >  
gdcmm::SmartPointer< ObjectType >::~~SmartPointer ( ) [inline]
```

**10.276.3 Member Function Documentation**

### 10.276.3.1 GetPointer()

```
template<class ObjectType >
ObjectType* gdcm::SmartPointer< ObjectType >::GetPointer ( ) const [inline]
```

Explicit function to retrieve the pointer.

### 10.276.3.2 operator ObjectType \*()

```
template<class ObjectType >
gdcm::SmartPointer< ObjectType >::operator ObjectType * ( ) const [inline]
```

Return pointer to object.

### 10.276.3.3 operator\*()

```
template<class ObjectType >
ObjectType& gdcm::SmartPointer< ObjectType >::operator* ( ) const [inline]
```

### 10.276.3.4 operator->()

```
template<class ObjectType >
ObjectType* gdcm::SmartPointer< ObjectType >::operator-> ( ) const [inline]
```

Overload operator ->

### 10.276.3.5 operator=( ) [1/3]

```
template<class ObjectType >
SmartPointer& gdcm::SmartPointer< ObjectType >::operator= (
    ObjectType * r ) [inline]
```

Overload operator assignment.

**10.276.3.6 operator=()** [2/3]

```
template<class ObjectType >
SmartPointer& gdcm::SmartPointer< ObjectType >::operator= (
    ObjectType const & r ) [inline]
```

References `gdcm::SmartPointer< ObjectType >::operator=()`.

**10.276.3.7 operator=()** [3/3]

```
template<class ObjectType >
SmartPointer& gdcm::SmartPointer< ObjectType >::operator= (
    SmartPointer< ObjectType > const & r ) [inline]
```

Overload operator assignment.

Referenced by `gdcm::SmartPointer< ObjectType >::operator=()`.

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

- [gdcmObject.h](#)
- [gdcmSmartPointer.h](#)

## 10.277 gdcm::network::SOPClassExtendedNegotiationSub Class Reference

[SOPClassExtendedNegotiationSub](#).

```
#include <gdcmSOPClassExtendedNegotiationSub.h>
```

### Public Member Functions

- [SOPClassExtendedNegotiationSub](#) ()
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [SetTuple](#) (const char \*uid, uint8\_t levelofsupport=3, uint8\_t levelofdigitalsig=0, uint8\_t elementcoercion=2)
- size\_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

### 10.277.1 Detailed Description

[SOPClassExtendedNegotiationSub](#).

PS 3.7 [Table](#) D.3-11 SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-↔ ASSOCIATE-AC)

## 10.277.2 Constructor & Destructor Documentation

### 10.277.2.1 SOPClassExtendedNegociationSub()

```
gdcm::network::SOPClassExtendedNegociationSub::SOPClassExtendedNegociationSub ( )
```

## 10.277.3 Member Function Documentation

### 10.277.3.1 Print()

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

### 10.277.3.2 Read()

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

### 10.277.3.3 SetTuple()

```
void gdcm::network::SOPClassExtendedNegociationSub::SetTuple (
    const char * uid,
    uint8_t levelofsupport = 3,
    uint8_t levelofdigitalsig = 0,
    uint8_t elementcoercion = 2 )
```

### 10.277.3.4 Size()

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



### 10.277.3.5 Write()

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

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

- [gdcmSOPClassExtendedNegociationSub.h](#)

## 10.278 gdcm::SOPClassUIDToIOD Class Reference

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

```
#include <gdcmSOPClassUIDToIOD.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](#) ()

### 10.278.1 Detailed Description

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

Reference PS 3.4 [Table](#) B.5-1 STANDARD SOP CLASSES

### 10.278.2 Member Typedef Documentation

#### 10.278.2.1 const

```
typedef const char* gdcm::SOPClassUIDToIOD::const (SOPClassUIDToIODType) [2]
```

## 10.278.3 Member Function Documentation

### 10.278.3.1 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](#).

### 10.278.3.2 GetIODFromSOPClassUID()

```
static const char* gdcm::SOPClassUIDToIOD::GetIODFromSOPClassUID (
    const char * sopcuid ) [static]
```

### 10.278.3.3 GetNumberOfSOPClassToIOD()

```
static unsigned int gdcm::SOPClassUIDToIOD::GetNumberOfSOPClassToIOD ( ) [static]
```

Return the number of SOP Class UID listed internally.

### 10.278.3.4 GetSOPClassUIDFromIOD()

```
static const char* gdcm::SOPClassUIDToIOD::GetSOPClassUIDFromIOD (
    const char * iod ) [static]
```

### 10.278.3.5 GetSOPClassUIDToIOD()

```
static SOPClassUIDToIODType& gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIOD (
    unsigned int i ) [static]
```

### 10.278.3.6 GetSOPClassUIDToIODs()

```
static SOPClassUIDToIODType* gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIODs ( ) [static]
```

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

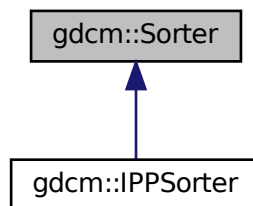
- [gdcmSOPClassUIDToIOD.h](#)

## 10.279 gdcm::Sorter Class Reference

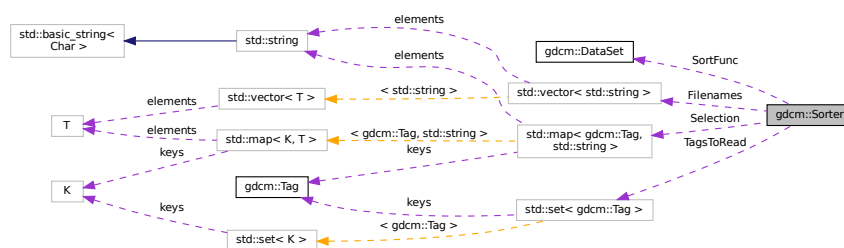
[Sorter.](#)

```
#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)

## 10.279.1 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 everytime you call Sort, all files specified as input paramater are *read*

### See also

[Scanner](#)

### Examples

[SortImage.cxx](#), and [VolumeSorter.cxx](#).

## 10.279.2 Member Typedef Documentation

### 10.279.2.1 SelectionMap

```
typedef std::map<Tag, std::string> gdcm::Sorter::SelectionMap [protected]
```

### 10.279.2.2 SortFunction

```
typedef bool(* gdcm::Sorter::SortFunction) (DataSet const &, DataSet const &)
```

Set the sort function which compares one dataset to the other.

## 10.279.3 Constructor & Destructor Documentation

### 10.279.3.1 Sorter()

```
gdcm::Sorter::Sorter ( )
```

### 10.279.3.2 ~Sorter()

```
virtual gdcm::Sorter::~Sorter ( ) [virtual]
```

## 10.279.4 Member Function Documentation

### 10.279.4.1 AddSelect()

```
bool gdcm::Sorter::AddSelect (
    Tag const & tag,
    const char * value )
```

UNSUPPORTED FOR NOW.

#### 10.279.4.2 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](#).

#### 10.279.4.3 Print()

```
void gdcM::Sorter::Print (
    std::ostream & os ) const
```

Print.

##### Examples

[SortImage.cxx](#), [VolumeSorter.cxx](#), and [gdcMOrthoplanes.cxx](#).

#### 10.279.4.4 SetSortFunction()

```
void gdcM::Sorter::SetSortFunction (
    SortFunction f )
```

##### Examples

[SortImage.cxx](#), and [VolumeSorter.cxx](#).

#### 10.279.4.5 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.

#### 10.279.4.6 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](#).

#### 10.279.4.7 StableSort()

```
virtual bool gdcm::Sorter::StableSort (
    std::vector< std::string > const & filenames ) [virtual]
```

##### Examples

[SortImage.cxx](#), and [VolumeSorter.cxx](#).

### 10.279.5 Friends And Related Function Documentation

#### 10.279.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const Sorter & s ) [friend]
```

### 10.279.6 Member Data Documentation

#### 10.279.6.1 Filenames

```
std::vector<std::string> gdcm::Sorter::Filenames [protected]
```

### 10.279.6.2 Selection

```
std::map<Tag, std::string> gdcm::Sorter::Selection [protected]
```

### 10.279.6.3 SortFunc

```
SortFunction gdcm::Sorter::SortFunc [protected]
```

### 10.279.6.4 TagsToRead

```
std::set<Tag> gdcm::Sorter::TagsToRead [protected]
```

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

- [gdcmSorter.h](#)

## 10.280 gdcm::Spacing Class Reference

Class for [Spacing](#).

```
#include <gdcmSpacing.h>
```

### Public Types

- enum [SpacingType](#) {  
    [DETECTOR](#) = 0 ,  
    [MAGNIFIED](#) ,  
    [CALIBRATED](#) ,  
    [UNKNOWN](#) }

### Public Member Functions

- [Spacing](#) ()
- [~Spacing](#) ()

### Static Public Member Functions

- static [Attribute](#)< 0x28, 0x34 > [ComputePixelAspectRatioFromPixelSpacing](#) (const [Attribute](#)< 0x28, 0x30 > &pixelspacing)



## 10.280.1 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)

## 10.280.2 Member Enumeration Documentation

### 10.280.2.1 SpacingType

enum [gdcm::Spacing::SpacingType](#)

## Enumerator

DETECTOR	
MAGNIFIED	
CALIBRATED	
UNKNOWN	

**10.280.3 Constructor & Destructor Documentation****10.280.3.1 Spacing()**

```
gdcM::Spacing::Spacing ( )
```

**10.280.3.2 ~Spacing()**

```
gdcM::Spacing::~Spacing ( )
```

**10.280.4 Member Function Documentation****10.280.4.1 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](#)

**10.281 gdcM::Spectroscopy Class Reference**

[Spectroscopy](#) class.

```
#include <gdcMSpectroscopy.h>
```

## Public Member Functions

- [Spectroscopy](#) ()=default

### 10.281.1 Detailed Description

[Spectroscopy](#) class.

### 10.281.2 Constructor & Destructor Documentation

#### 10.281.2.1 Spectroscopy()

```
gdcm::Spectroscopy::Spectroscopy ( ) [default]
```

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

- [gdcmSpectroscopy.h](#)

## 10.282 gdcm::SplitMosaicFilter Class Reference

[SplitMosaicFilter](#) class.

```
#include <gdcmSplitMosaicFilter.h>
```

## Public Member Functions

- [SplitMosaicFilter](#) ()
- [~SplitMosaicFilter](#) ()
- bool [ComputeMOSAICDimensions](#) (unsigned int dims[3])
- bool [ComputeMOSAICSliceNormal](#) (double dims[3], bool &inverted)  
*Extract the value for SliceNormalVector (CSA header)*
- bool [ComputeMOSAICSlicePosition](#) (double pos[3], bool inverted)  
*Extract the value for ImagePositionPatient (requires inverted flag)*
- [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.*

### 10.282.1 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).

### 10.282.2 Constructor & Destructor Documentation

#### 10.282.2.1 SplitMosaicFilter()

```
gdcm::SplitMosaicFilter::SplitMosaicFilter ( )
```

#### 10.282.2.2 ~SplitMosaicFilter()

```
gdcm::SplitMosaicFilter::~~SplitMosaicFilter ( )
```

### 10.282.3 Member Function Documentation

#### 10.282.3.1 ComputeMOSAICDimensions()

```
bool gdcm::SplitMosaicFilter::ComputeMOSAICDimensions (
    unsigned int dims[3] )
```

Compute the new dimensions according to private information stored in the MOSAIC header.

### 10.282.3.2 ComputeMOSAICSliceNormal()

```
bool gdcm::SplitMosaicFilter::ComputeMOSAICSliceNormal (
    double dims[3],
    bool & inverted )
```

Extract the value for SliceNormalVector (CSA header)

### 10.282.3.3 ComputeMOSAICSlicePosition()

```
bool gdcm::SplitMosaicFilter::ComputeMOSAICSlicePosition (
    double pos[3],
    bool inverted )
```

Extract the value for ImagePositionPatient (requires inverted flag)

### 10.282.3.4 GetAcquisitionSize()

```
static bool gdcm::SplitMosaicFilter::GetAcquisitionSize (
    unsigned int size[2],
    DataSet const & ds ) [static]
```

Get the Acquisition Matrix (non zero value):

### 10.282.3.5 GetFile() [1/2]

```
File& gdcm::SplitMosaicFilter::GetFile ( ) [inline]
```

### 10.282.3.6 GetFile() [2/2]

```
const File& gdcm::SplitMosaicFilter::GetFile ( ) const [inline]
```

### 10.282.3.7 GetImage() [1/2]

```
Image& gdcm::SplitMosaicFilter::GetImage ( ) [inline]
```

### 10.282.3.8 GetImage() [2/2]

```
const Image& gdcm::SplitMosaicFilter::GetImage ( ) const [inline]
```

### 10.282.3.9 GetNumberOfImagesInMosaic()

```
static unsigned int gdcm::SplitMosaicFilter::GetNumberOfImagesInMosaic (
    File const & file ) [static]
```

Return the value for NumberOfImagesInMosaic, or compute it from Acquisition Size.

### 10.282.3.10 SetFile()

```
void gdcm::SplitMosaicFilter::SetFile (
    const File & f ) [inline]
```

### 10.282.3.11 SetImage()

```
void gdcm::SplitMosaicFilter::SetImage (
    const Image & image )
```

### 10.282.3.12 Split()

```
bool gdcm::SplitMosaicFilter::Split ( )
```

Split the SIEMENS MOSAIC image.

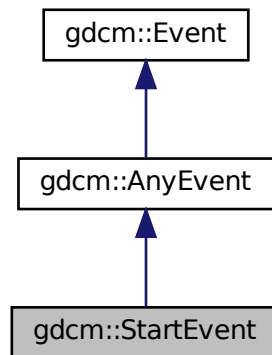
The documentation for this class was generated from the following file:

- [gdcmSplitMosaicFilter.h](#)

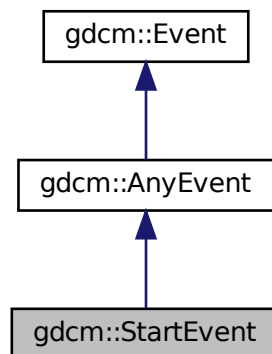
## 10.283 gdcm::StartEvent Class Reference

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

Inheritance diagram for gdcm::StartEvent:



Collaboration diagram for gdcm::StartEvent:



### Additional Inherited Members

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

- [gdcmEvent.h](#)

## 10.284 `gdcm::static_assert_test< x >` Struct Template Reference

```
#include <gdcmStaticAssert.h>
```

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

- [gdcmStaticAssert.h](#)

## 10.285 `gdcm::STATIC_ASSERTION_FAILURE< x >` Struct Template Reference

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

- [gdcmStaticAssert.h](#)

## 10.286 `gdcm::STATIC_ASSERTION_FAILURE< true >` Struct Reference

```
#include <gdcmStaticAssert.h>
```

### Public Types

- enum { `value` = 1 }

### 10.286.1 Member Enumeration Documentation

#### 10.286.1.1 anonymous enum

```
anonymous enum
```

#### Enumerator

value	
-------	--

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

- [gdcmStaticAssert.h](#)



## 10.287 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)

### 10.287.1 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

[StreamImageReaderTest.cxx](#).

### 10.287.2 Constructor & Destructor Documentation

### 10.287.2.1 StreamImageReader()

```
gdcM::StreamImageReader::StreamImageReader ( )
```

### 10.287.2.2 ~StreamImageReader()

```
virtual gdcM::StreamImageReader::~~StreamImageReader ( ) [virtual]
```

## 10.287.3 Member Function Documentation

### 10.287.3.1 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](#).

### 10.287.3.2 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

[StreamImageReaderTest.cxx](#).

### 10.287.3.3 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

[StreamImageReaderTest.cxx](#).

### 10.287.3.4 GetDimensionsValueForResolution()

```
std::vector<unsigned int> gdcm::StreamImageReader::GetDimensionsValueForResolution (
    unsigned int )
```

### 10.287.3.5 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

[StreamImageReaderTest.cxx](#).

### 10.287.3.6 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 metaimageio in itk MUST have an extent defined, or else Read will return false. If no particular extent is required, use [ImageReader](#) instead.

#### Examples

[StreamImageReaderTest.cxx](#).

### 10.287.3.7 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

[StreamImageReaderTest.cxx](#).

### 10.287.3.8 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](#).

### 10.287.3.9 SetStream()

```
void gdcM::StreamImageReader::SetStream (
    std::istream & inStream )
```

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

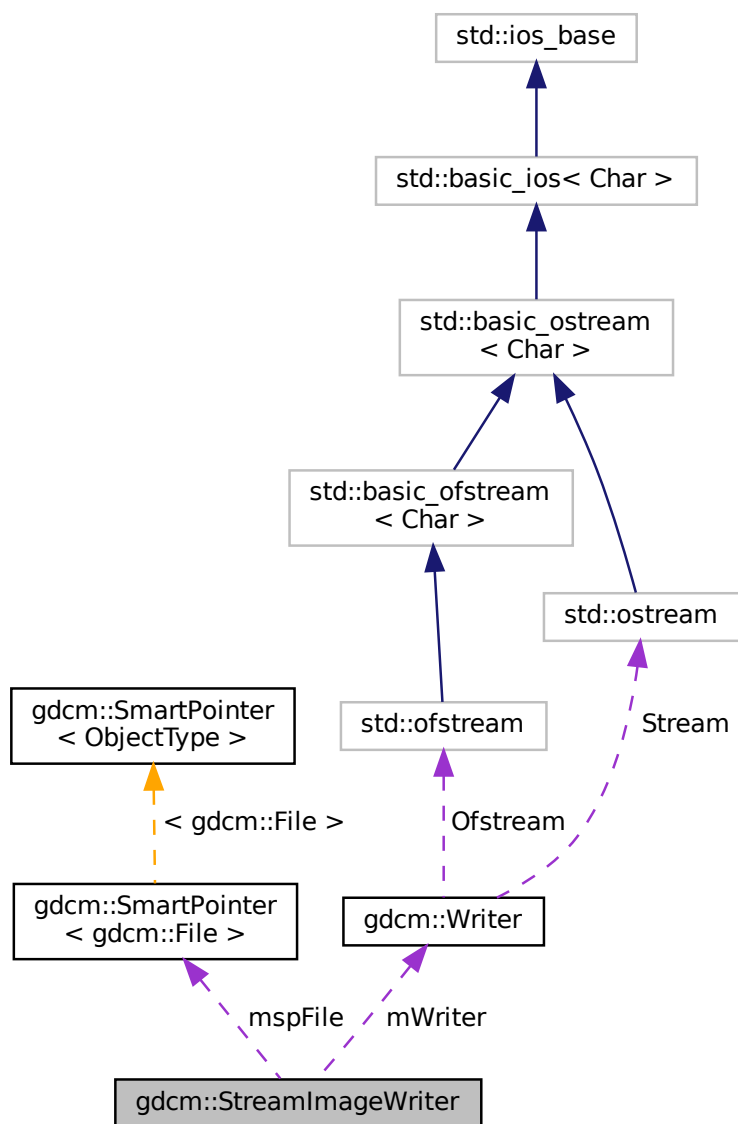
- [gdcMStreamImageReader.h](#)

## 10.288 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](#)

### 10.288.1 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](#).

## 10.288.2 Constructor & Destructor Documentation

### 10.288.2.1 StreamImageWriter()

```
gdcm::StreamImageWriter::StreamImageWriter ( )
```

### 10.288.2.2 ~StreamImageWriter()

```
virtual gdcm::StreamImageWriter::~~StreamImageWriter ( ) [virtual]
```

## 10.288.3 Member Function Documentation

### 10.288.3.1 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](#).

### 10.288.3.2 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](#).

### 10.288.3.3 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](#).

### 10.288.3.4 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](#).

### 10.288.3.5 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.

### 10.288.3.6 SetStream()

```
void gdcM::StreamImageWriter::SetStream (
    std::ostream & inStream )
```

#### Examples

[Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).



### 10.288.3.7 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 accomodate 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](#).

### 10.288.3.8 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](#).

### 10.288.3.9 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

### 10.288.3.10 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.

## 10.288.4 Member Data Documentation

### 10.288.4.1 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.

### 10.288.4.2 mElementOffsets1

```
int gdcm::StreamImageWriter::mElementOffsets1 [protected]
```

### 10.288.4.3 mspFile

```
SmartPointer<File> gdcm::StreamImageWriter::mspFile [protected]
```

### 10.288.4.4 mWriter

```
Writer gdcm::StreamImageWriter::mWriter [protected]
```

#### 10.288.4.5 mXMax

uint16\_t gdcm::StreamImageWriter::mXMax [protected]

#### 10.288.4.6 mXMin

uint16\_t gdcm::StreamImageWriter::mXMin [protected]

#### 10.288.4.7 mYMax

uint16\_t gdcm::StreamImageWriter::mYMax [protected]

#### 10.288.4.8 mYMin

uint16\_t gdcm::StreamImageWriter::mYMin [protected]

#### 10.288.4.9 mZMax

uint16\_t gdcm::StreamImageWriter::mZMax [protected]

#### 10.288.4.10 mZMin

uint16\_t gdcm::StreamImageWriter::mZMin [protected]

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

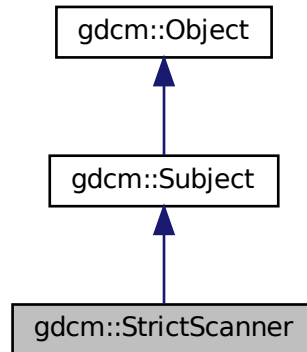
- [gdcmStreamImageWriter.h](#)

## 10.289 gdcM::StrictScanner Class Reference

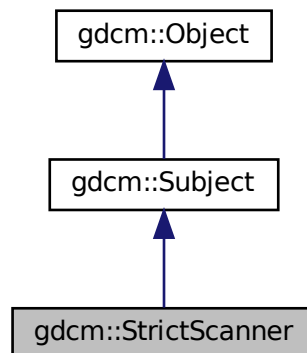
[StrictScanner](#).

```
#include <gdcMStrictScanner.h>
```

Inheritance diagram for gdcM::StrictScanner:



Collaboration diagram for gdcM::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 !*

## 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)

## Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [StrictScanner](#) &s)

### 10.289.1 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

[SimpleScanner.cxx](#).

### 10.289.2 Member Typedef Documentation

#### 10.289.2.1 ConstIterator

```
typedef MappingType::const_iterator gdcm::StrictScanner::ConstIterator
```

### 10.289.2.2 MappingType

```
typedef std::map<const char *, TagToValue, ltstr> gdcm::StrictScanner::MappingType
```

### 10.289.2.3 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)

### 10.289.2.4 TagToValueValueType

```
typedef TagToValue::value_type gdcm::StrictScanner::TagToValueValueType
```

### 10.289.2.5 ValuesType

```
typedef std::set< std::string > gdcm::StrictScanner::ValuesType
```

## 10.289.3 Constructor & Destructor Documentation

### 10.289.3.1 StrictScanner()

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

### 10.289.3.2 ~StrictScanner()

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

## 10.289.4 Member Function Documentation

#### 10.289.4.1 AddPrivateTag()

```
void gdcm::StrictScanner::AddPrivateTag (
    PrivateTag const & t )
```

#### 10.289.4.2 AddSkipTag()

```
void gdcm::StrictScanner::AddSkipTag (
    Tag const & t )
```

Add a tag that will need to be skipped. Those are root level skip tags.

#### 10.289.4.3 AddTag()

```
void gdcm::StrictScanner::AddTag (
    Tag const & t )
```

Add a tag that will need to be read. Those are root level skip tags.

#### Examples

[SimpleScanner.cxx](#).

#### 10.289.4.4 Begin()

```
ConstIterator gdcm::StrictScanner::Begin ( ) const [inline]
```

#### 10.289.4.5 ClearSkipTags()

```
void gdcm::StrictScanner::ClearSkipTags ( )
```

#### 10.289.4.6 ClearTags()

```
void gdcm::StrictScanner::ClearTags ( )
```



#### 10.289.4.7 End()

```
ConstIterator gdcm::StrictScanner::End ( ) const [inline]
```

#### 10.289.4.8 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'

#### 10.289.4.9 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'

#### 10.289.4.10 GetFileNames()

```
Directory::FileNamesType const& gdcm::StrictScanner::GetFileNames ( ) const [inline]
```

#### 10.289.4.11 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

#### 10.289.4.12 GetMapping()

```
TagToValue const& gdcm::StrictScanner::GetMapping (
    const char * filename ) const
```

Get the std::map mapping filenames to value for file 'filename'.

#### Examples

[SimpleScanner.cxx](#).

#### 10.289.4.13 GetMappingFromTagToValue()

```
TagToValue const& gdcm::StrictScanner::GetMappingFromTagToValue (
    Tag const & t,
    const char * value ) const
```

See [GetFilenameFromTagToValue\(\)](#). This is simply GetFilenameFromTagToValue followed.

#### 10.289.4.14 GetMappings()

```
MappingType const& gdcm::StrictScanner::GetMappings ( ) const [inline]
```

Mappings are the mapping from a particular tag to the map, mapping filename to value:

#### 10.289.4.15 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)

#### 10.289.4.16 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 !

#### 10.289.4.17 GetValues() [1/2]

```
ValuesType const& gdcm::StrictScanner::GetValues ( ) const [inline]
```

Get all the values found (in lexicographic order)

#### 10.289.4.18 GetValues() [2/2]

```
ValueType gdcm::StrictScanner::GetValues (
    Tag const & t ) const
```

Get all the values found (in lexicographic order) associated with Tag 't'.

#### 10.289.4.19 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

[SimpleScanner.cxx](#).

#### 10.289.4.20 New()

```
static SmartPointer<StrictScanner> gdcm::StrictScanner::New ( ) [inline], [static]
```

for wrapped language: instantiate a reference counted object

#### 10.289.4.21 Print()

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

Print result.

Reimplemented from [gdcm::Object](#).

#### 10.289.4.22 PrintTable()

```
void gdcm::StrictScanner::PrintTable (
    std::ostream & os ) const
```

#### 10.289.4.23 ProcessPublicTag()

```
void gdcM::StrictScanner::ProcessPublicTag (
    StringFilter & sf,
    const char * filename )    [protected]
```

#### 10.289.4.24 Scan()

```
bool gdcM::StrictScanner::Scan (
    Directory::FileNamesType const & filenames )
```

Start the scan !

#### Examples

[SimpleScanner.cxx](#).

### 10.289.5 Friends And Related Function Documentation

#### 10.289.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const StrictScanner & s )    [friend]
```

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

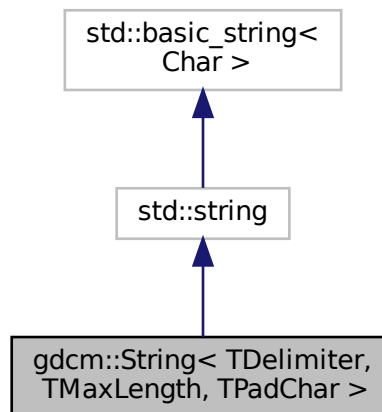
- [gdcMStrictScanner.h](#)

## 10.290 gdcm::String< TDelimiter, TMaxLength, TPadChar > Class Template Reference

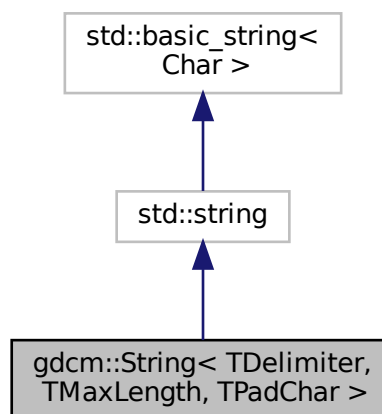
[String](#).

```
#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=npos)
- [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
- [gdcmm::String](#)< TDelimiter, TMaxLength, TPadChar > [Truncate](#) () const

## Static Public Member Functions

- static std::string [Trim](#) (const char \*input)

### 10.290.1 Detailed Description

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
class gdcmm::String< TDelimiter, TMaxLength, TPadChar >
```

[String](#).

#### Note

TDelimiter template parameter is used to separate multiple [String](#) (VM1 >) TMaxLength is only a hint. Noone actually respect the max length TPadChar is the string padding (0 or space)

#### Examples

[TemplateEmptyImage.cxx](#).

## 10.290.2 Member Typedef Documentation

### 10.290.2.1 const\_iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::const_iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::const_iterator
```

### 10.290.2.2 const\_reference

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::const_reference gdcm::String< TDelimiter, TMaxLength, TPadChar >::const_reference
```

### 10.290.2.3 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
```

### 10.290.2.4 difference\_type

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::difference_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::difference_type
```

### 10.290.2.5 iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::iterator
```

### 10.290.2.6 pointer

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::pointer gdcM::String< TDelimiter, TMaxLength, TPadChar >::pointer
```

### 10.290.2.7 reference

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::reference gdcM::String< TDelimiter, TMaxLength, TPadChar >::reference
```

### 10.290.2.8 reverse\_iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::reverse_iterator gdcM::String< TDelimiter, TMaxLength, TPadChar >::reverse\_iterator
```

### 10.290.2.9 size\_type

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::size_type gdcM::String< TDelimiter, TMaxLength, TPadChar >::size\_type
```

### 10.290.2.10 value\_type

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::value_type gdcM::String< TDelimiter, TMaxLength, TPadChar >::value\_type
```

## 10.290.3 Constructor & Destructor Documentation

### 10.290.3.1 String() [1/4]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcM::String< TDelimiter, TMaxLength, TPadChar >::String ( ) [inline]
```

[String](#) constructors.



**10.290.3.2 String() [2/4]**

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (
    const value_type * s ) [inline]
```

**10.290.3.3 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]
```

**10.290.3.4 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]
```

**10.290.4 Member Function Documentation****10.290.4.1 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().

**10.290.4.2 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:

#### 10.290.4.3 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](#), and [DumpPhilipsECHO.cxx](#).

#### 10.290.4.4 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]
```

#### 10.290.4.5 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](#)

## 10.291 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

### 10.291.1 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](#), and [ReadAndPrintAttributes.cxx](#).

### 10.291.2 Constructor & Destructor Documentation

#### 10.291.2.1 StringFilter()

```
gdcm::StringFilter::StringFilter ( )
```

### 10.291.2.2 ~StringFilter()

```
gdcM::StringFilter::~~StringFilter ( )
```

## 10.291.3 Member Function Documentation

### 10.291.3.1 ExecuteQuery() [1/2]

```
bool gdcM::StringFilter::ExecuteQuery (
    std::string const & query,
    DataSet const & ds,
    std::string & value ) const [protected]
```

### 10.291.3.2 ExecuteQuery() [2/2]

```
bool gdcM::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 syntatically correct

### 10.291.3.3 FromString()

```
std::string gdcM::StringFilter::FromString (
    const Tag & t,
    const char * value,
    size_t len )
```

Convert to string the char array defined by the pair (value,len)

### 10.291.3.4 GetFile() [1/2]

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

### 10.291.3.5 GetFile() [2/2]

```
const File& gdcm::StringFilter::GetFile ( ) const [inline]
```

### 10.291.3.6 SetDicts()

```
void gdcm::StringFilter::SetDicts (
    const Dicts & dicts )
```

Allow user to pass in there own dicts.

### 10.291.3.7 SetFile()

```
void gdcm::StringFilter::SetFile (
    const File & f ) [inline]
```

Set/Get [File](#).

#### Examples

[DumpVisusChange.cxx](#), [ReadAndPrintAttributes.cxx](#), and [SimplePrintPatientName.cs](#).

### 10.291.3.8 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](#), and [ReadAndPrintAttributes.cxx](#).

### 10.291.3.9 ToString() [2/3]

```
std::string gdcm::StringFilter::ToString (
    const PrivateTag & t ) const
```

**10.291.3.10 ToString()** [3/3]

```
std::string gdcm::StringFilter::ToString (
    const Tag & t ) const
```

Directly from a [Tag](#):

**10.291.3.11 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](#) pair.second : the value encoded into a string (US,UL...) are properly converted

**Examples**

[ReadAndPrintAttributes.cxx](#).

**10.291.3.12 ToStringPair()** [2/3]

```
std::pair<std::string, std::string> gdcm::StringFilter::ToStringPair (
    const Tag & t ) const
```

Directly from a [Tag](#):

**10.291.3.13 ToStringPair()** [3/3]

```
std::pair<std::string, std::string> gdcm::StringFilter::ToStringPair (
    const Tag & t,
    DataSet const & ds ) const [protected]
```

**10.291.3.14 UseDictAlways()**

```
void gdcm::StringFilter::UseDictAlways (
    bool ) [inline]
```

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

- [gdcmStringFilter.h](#)

## 10.292 gdcm::Study Class Reference

[Study.](#)

```
#include <gdcmStudy.h>
```

### Public Member Functions

- [Study](#) ()=default

### 10.292.1 Detailed Description

[Study.](#)

### 10.292.2 Constructor & Destructor Documentation

#### 10.292.2.1 Study()

```
gdcm::Study::Study ( ) [default]
```

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

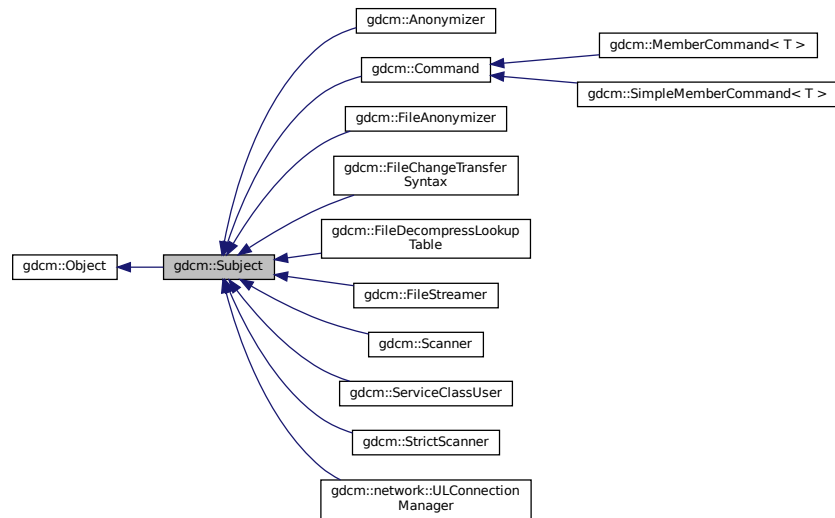
- [gdcmStudy.h](#)

## 10.293 gdcm::Subject Class Reference

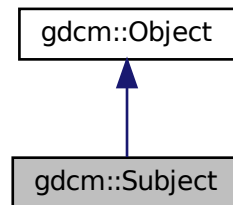
[Subject.](#)

```
#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)`



## Additional Inherited Members

### 10.293.1 Detailed Description

[Subject](#).

See also

[Command Event](#)

Examples

[SimpleScanner.cxx](#).

### 10.293.2 Constructor & Destructor Documentation

#### 10.293.2.1 Subject()

```
gdcmm::Subject::Subject ( )
```

#### 10.293.2.2 ~Subject()

```
gdcmm::Subject::~~Subject ( ) [override]
```

### 10.293.3 Member Function Documentation

#### 10.293.3.1 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

### 10.293.3.2 AddObserver() [2/2]

```
unsigned long gdcmm::Subject::AddObserver (
    const Event & event,
    Command * ) const
```

### 10.293.3.3 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.

### 10.293.3.4 HasObserver()

```
bool gdcmm::Subject::HasObserver (
    const Event & event ) const
```

Return true if an observer is registered for this event.

### 10.293.3.5 InvokeEvent() [1/2]

```
void gdcmm::Subject::InvokeEvent (
    const Event & )
```

Call Execute on all the Commands observing this event id.

### 10.293.3.6 InvokeEvent() [2/2]

```
void gdcmm::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.

### 10.293.3.7 RemoveAllObservers()

```
void gdcmm::Subject::RemoveAllObservers ( )
```

Remove all observers .

### 10.293.3.8 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:

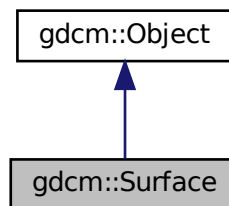
- [gdcmSubject.h](#)

## 10.294 gdcm::Surface Class Reference

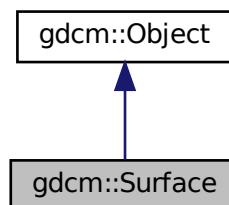
This class defines a SURFACE IE.

```
#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)

## 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

### 10.294.1 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

## 10.294.2 Member Enumeration Documentation

### 10.294.2.1 STATES

enum `gdcm::Surface::STATES`

#### Enumerator

NO	
YES	
UNKNOWN	
STATES_END	

### 10.294.2.2 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	

## 10.294.3 Constructor & Destructor Documentation

### 10.294.3.1 Surface()

`gdcm::Surface::Surface ( )`

### 10.294.3.2 ~Surface()

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

## 10.294.4 Member Function Documentation

### 10.294.4.1 GetAlgorithmFamily() [1/2]

```
SegmentHelper::BasicCodedEntry& gdcm::Surface::GetAlgorithmFamily ( )
```

### 10.294.4.2 GetAlgorithmFamily() [2/2]

```
SegmentHelper::BasicCodedEntry const& gdcm::Surface::GetAlgorithmFamily ( ) const
```

### 10.294.4.3 GetAlgorithmName()

```
const char* gdcm::Surface::GetAlgorithmName ( ) const
```

### 10.294.4.4 GetAlgorithmVersion()

```
const char* gdcm::Surface::GetAlgorithmVersion ( ) const
```

### 10.294.4.5 GetAxisOfRotation()

```
const float* gdcm::Surface::GetAxisOfRotation ( ) const
```

#### Note

Pointer is null if undefined

#### 10.294.4.6 GetCenterOfRotation()

```
const float* gdcm::Surface::GetCenterOfRotation ( ) const
```

##### Note

Pointer is null if undefined

#### 10.294.4.7 GetFiniteVolume()

```
STATES gdcm::Surface::GetFiniteVolume ( ) const
```

#### 10.294.4.8 GetManifold()

```
STATES gdcm::Surface::GetManifold ( ) const
```

#### 10.294.4.9 GetMaximumPointDistance()

```
float gdcm::Surface::GetMaximumPointDistance ( ) const
```

#### 10.294.4.10 GetMeanPointDistance()

```
float gdcm::Surface::GetMeanPointDistance ( ) const
```

#### 10.294.4.11 GetMeshPrimitive() [1/2]

```
MeshPrimitive& gdcm::Surface::GetMeshPrimitive ( )
```



**10.294.4.12 GetMeshPrimitive() [2/2]**

```
MeshPrimitive const& gdcm::Surface::GetMeshPrimitive ( ) const
```

**10.294.4.13 GetNumberOfSurfacePoints()**

```
unsigned long gdcm::Surface::GetNumberOfSurfacePoints ( ) const
```

**10.294.4.14 GetNumberOfVectors()**

```
unsigned long gdcm::Surface::GetNumberOfVectors ( ) const
```

**10.294.4.15 GetPointCoordinatesData() [1/2]**

```
DataElement& gdcm::Surface::GetPointCoordinatesData ( )
```

**10.294.4.16 GetPointCoordinatesData() [2/2]**

```
const DataElement& gdcm::Surface::GetPointCoordinatesData ( ) const
```

**10.294.4.17 GetPointPositionAccuracy()**

```
const float* gdcm::Surface::GetPointPositionAccuracy ( ) const
```

**Note**

Pointer is null if undefined

**10.294.4.18 GetPointsBoundingBoxCoordinates()**

```
const float* gdcm::Surface::GetPointsBoundingBoxCoordinates ( ) const
```

**Note**

Pointer is null if undefined

**10.294.4.19 GetProcessingAlgorithm() [1/2]**

```
SegmentHelper::BasicCodedEntry& gdcm::Surface::GetProcessingAlgorithm ( )
```

**10.294.4.20 GetProcessingAlgorithm() [2/2]**

```
SegmentHelper::BasicCodedEntry const& gdcm::Surface::GetProcessingAlgorithm ( ) const
```

**10.294.4.21 GetRecommendedDisplayCIELabValue() [1/2]**

```
const unsigned short* gdcm::Surface::GetRecommendedDisplayCIELabValue ( ) const
```

**10.294.4.22 GetRecommendedDisplayCIELabValue() [2/2]**

```
unsigned short gdcm::Surface::GetRecommendedDisplayCIELabValue (
    const unsigned int idx ) const
```

**10.294.4.23 GetRecommendedDisplayGrayscaleValue()**

```
unsigned short gdcm::Surface::GetRecommendedDisplayGrayscaleValue ( ) const
```

#### 10.294.4.24 GetRecommendedPresentationOpacity()

```
float gdcm::Surface::GetRecommendedPresentationOpacity ( ) const
```

#### 10.294.4.25 GetRecommendedPresentationType()

```
VIEWType gdcm::Surface::GetRecommendedPresentationType ( ) const
```

#### 10.294.4.26 GetSTATES()

```
static STATES gdcm::Surface::GetSTATES (
    const char * state ) [static]
```

#### 10.294.4.27 GetSTATESString()

```
static const char* gdcm::Surface::GetSTATESString (
    STATES state ) [static]
```

#### 10.294.4.28 GetSurfaceComments()

```
const char* gdcm::Surface::GetSurfaceComments ( ) const
```

#### 10.294.4.29 GetSurfaceNumber()

```
unsigned long gdcm::Surface::GetSurfaceNumber ( ) const
```

#### 10.294.4.30 GetSurfaceProcessing()

```
bool gdcm::Surface::GetSurfaceProcessing ( ) const
```

**10.294.4.31 GetSurfaceProcessingDescription()**

```
const char* gdcm::Surface::GetSurfaceProcessingDescription ( ) const
```

**10.294.4.32 GetSurfaceProcessingRatio()**

```
float gdcm::Surface::GetSurfaceProcessingRatio ( ) const
```

**10.294.4.33 GetVectorAccuracy()**

```
const float* gdcm::Surface::GetVectorAccuracy ( ) const
```

**10.294.4.34 GetVectorCoordinateData() [1/2]**

```
DataElement& gdcm::Surface::GetVectorCoordinateData ( )
```

**10.294.4.35 GetVectorCoordinateData() [2/2]**

```
const DataElement& gdcm::Surface::GetVectorCoordinateData ( ) const
```

**10.294.4.36 GetVectorDimensionality()**

```
unsigned short gdcm::Surface::GetVectorDimensionality ( ) const
```

**10.294.4.37 GetVIEWType()**

```
static VIEWType gdcm::Surface::GetVIEWType (
    const char * type ) [static]
```

#### 10.294.4.38 GetVIEWTypeString()

```
static const char* gdcm::Surface::GetVIEWTypeString (
    VIEWType type ) [static]
```

#### 10.294.4.39 SetAlgorithmFamily()

```
void gdcm::Surface::SetAlgorithmFamily (
    SegmentHelper::BasicCodedEntry const & BSE )
```

#### 10.294.4.40 SetAlgorithmName()

```
void gdcm::Surface::SetAlgorithmName (
    const char * str )
```

#### 10.294.4.41 SetAlgorithmVersion()

```
void gdcm::Surface::SetAlgorithmVersion (
    const char * str )
```

#### 10.294.4.42 SetAxisOfRotation()

```
void gdcm::Surface::SetAxisOfRotation (
    const float * axis )
```

#### 10.294.4.43 SetCenterOfRotation()

```
void gdcm::Surface::SetCenterOfRotation (
    const float * center )
```

**10.294.4.44 SetFiniteVolume()**

```
void gdcM::Surface::SetFiniteVolume (
    STATES state )
```

**10.294.4.45 SetManifold()**

```
void gdcM::Surface::SetManifold (
    STATES state )
```

**10.294.4.46 SetMaximumPointDistance()**

```
void gdcM::Surface::SetMaximumPointDistance (
    float maximum )
```

**10.294.4.47 SetMeanPointDistance()**

```
void gdcM::Surface::SetMeanPointDistance (
    float average )
```

**10.294.4.48 SetMeshPrimitive()**

```
void gdcM::Surface::SetMeshPrimitive (
    MeshPrimitive & mp )
```

**10.294.4.49 SetNumberOfSurfacePoints()**

```
void gdcM::Surface::SetNumberOfSurfacePoints (
    const unsigned long nb )
```

**10.294.4.50 SetNumberOfVectors()**

```
void gdcm::Surface::SetNumberOfVectors (
    const unsigned long nb )
```

**10.294.4.51 SetPointCoordinatesData()**

```
void gdcm::Surface::SetPointCoordinatesData (
    DataElement const & de )
```

**10.294.4.52 SetPointPositionAccuracy()**

```
void gdcm::Surface::SetPointPositionAccuracy (
    const float * accuracies )
```

**10.294.4.53 SetPointsBoundingBoxCoordinates()**

```
void gdcm::Surface::SetPointsBoundingBoxCoordinates (
    const float * coordinates )
```

**10.294.4.54 SetProcessingAlgorithm()**

```
void gdcm::Surface::SetProcessingAlgorithm (
    SegmentHelper::BasicCodedEntry const & BSE )
```

**10.294.4.55 SetRecommendedDisplayCIELabValue()** [1/3]

```
void gdcm::Surface::SetRecommendedDisplayCIELabValue (
    const std::vector< unsigned short > & vl )
```

**10.294.4.56 SetRecommendedDisplayCIELabValue()** [2/3]

```
void gdcM::Surface::SetRecommendedDisplayCIELabValue (
    const unsigned short vl,
    const unsigned int idx = 0 )
```

**10.294.4.57 SetRecommendedDisplayCIELabValue()** [3/3]

```
void gdcM::Surface::SetRecommendedDisplayCIELabValue (
    const unsigned short vl[3] )
```

**10.294.4.58 SetRecommendedDisplayGrayscaleValue()**

```
void gdcM::Surface::SetRecommendedDisplayGrayscaleValue (
    const unsigned short vl )
```

**10.294.4.59 SetRecommendedPresentationOpacity()**

```
void gdcM::Surface::SetRecommendedPresentationOpacity (
    const float opacity )
```

**10.294.4.60 SetRecommendedPresentationType()**

```
void gdcM::Surface::SetRecommendedPresentationType (
    VIEWType type )
```

**10.294.4.61 SetSurfaceComments()**

```
void gdcM::Surface::SetSurfaceComments (
    const char * comment )
```



**10.294.4.62 SetSurfaceNumber()**

```
void gdcmm::Surface::SetSurfaceNumber (
    const unsigned long nb )
```

**10.294.4.63 SetSurfaceProcessing()**

```
void gdcmm::Surface::SetSurfaceProcessing (
    bool b )
```

**10.294.4.64 SetSurfaceProcessingDescription()**

```
void gdcmm::Surface::SetSurfaceProcessingDescription (
    const char * description )
```

**10.294.4.65 SetSurfaceProcessingRatio()**

```
void gdcmm::Surface::SetSurfaceProcessingRatio (
    const float ratio )
```

**10.294.4.66 SetVectorAccuracy()**

```
void gdcmm::Surface::SetVectorAccuracy (
    const float * accuracy )
```

**10.294.4.67 SetVectorCoordinateData()**

```
void gdcmm::Surface::SetVectorCoordinateData (
    DataElement const & de )
```

#### 10.294.4.68 SetVectorDimensionality()

```
void gdcmm::Surface::SetVectorDimensionality (
    const unsigned short dim )
```

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

- [gdcmmSurface.h](#)

### 10.295 gdcmm::SurfaceHelper Class Reference

[SurfaceHelper](#).

```
#include <gdcmmSurfaceHelper.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).*

#### 10.295.1 Detailed Description

[SurfaceHelper](#).

Helper class for [Surface](#) object

## 10.295.2 Member Typedef Documentation

### 10.295.2.1 ColorArray

```
typedef std::vector< unsigned short > gdcm::SurfaceHelper::ColorArray
```

## 10.295.3 Member Function Documentation

### 10.295.3.1 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>	Type of CIELab components.
<i>U</i>	Type of rangeMax value.

### 10.295.3.2 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.
----------	-------------------------

### 10.295.3.3 RGBToRecommendedDisplayCIELab()

```
template<typename T , typename U >
SurfaceHelper::ColorArray gdc::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.

### 10.295.3.4 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:

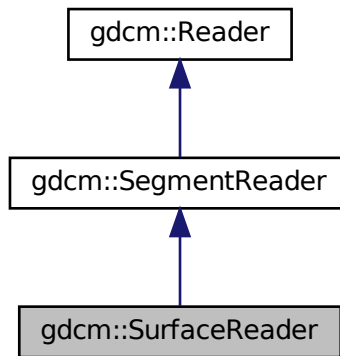
- [gdcmSurfaceHelper.h](#)

## 10.296 gdcm::SurfaceReader Class Reference

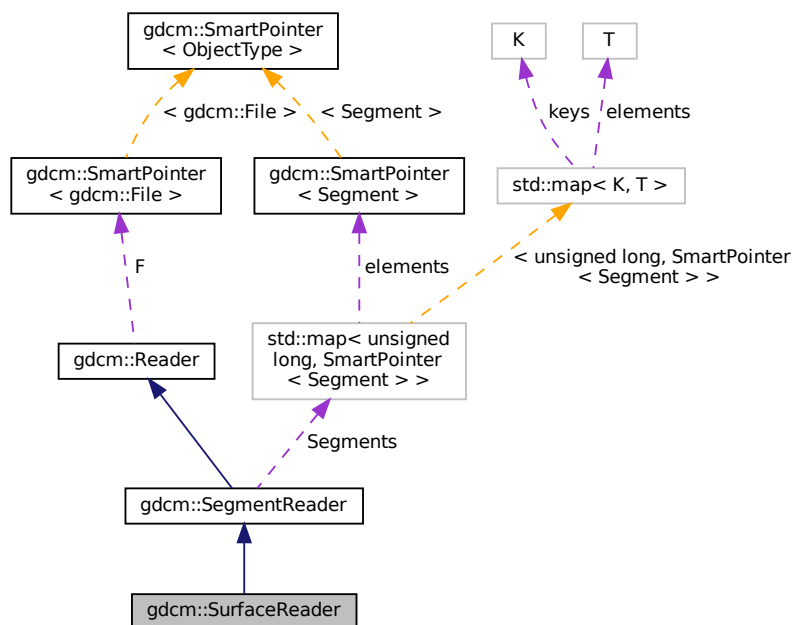
This class defines a SURFACE IE reader.

```
#include <gdcmSurfaceReader.h>
```

Inheritance diagram for `gdc::SurfaceReader`:



Collaboration diagram for `gdc::SurfaceReader`:



## Public Member Functions

- [SurfaceReader](#) ()

- [~SurfaceReader](#) () override
  - unsigned long [GetNumberOfSurfaces](#) () const
  - bool [Read](#) () override
- Read.*

## Protected Member Functions

- bool [ReadPointMacro](#) ([SmartPointer](#)< [Surface](#) > surface, const [DataSet](#) &surfaceDS)
- bool [ReadSurface](#) (const [Item](#) &surfItem, const unsigned long idx)
- bool [ReadSurfaces](#) ()

## Additional Inherited Members

### 10.296.1 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

### 10.296.2 Constructor & Destructor Documentation

#### 10.296.2.1 SurfaceReader()

```
gdcm::SurfaceReader::SurfaceReader ( )
```

#### 10.296.2.2 ~SurfaceReader()

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

### 10.296.3 Member Function Documentation

### 10.296.3.1 GetNumberOfSurfaces()

```
unsigned long gdcm::SurfaceReader::GetNumberOfSurfaces ( ) const
```

### 10.296.3.2 Read()

```
bool gdcm::SurfaceReader::Read ( ) [override], [virtual]
```

Read.

Reimplemented from [gdcm::SegmentReader](#).

### 10.296.3.3 ReadPointMacro()

```
bool gdcm::SurfaceReader::ReadPointMacro (
    SmartPointer< Surface > surface,
    const DataSet & surfaceDS ) [protected]
```

### 10.296.3.4 ReadSurface()

```
bool gdcm::SurfaceReader::ReadSurface (
    const Item & surfaceItem,
    const unsigned long idx ) [protected]
```

### 10.296.3.5 ReadSurfaces()

```
bool gdcm::SurfaceReader::ReadSurfaces ( ) [protected]
```

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

- [gdcmSurfaceReader.h](#)





## Protected Attributes

- unsigned long [NumberOfSurfaces](#)

## Additional Inherited Members

### 10.297.1 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

### 10.297.2 Constructor & Destructor Documentation

#### 10.297.2.1 SurfaceWriter()

```
gdcM::SurfaceWriter::SurfaceWriter ( )
```

#### 10.297.2.2 ~SurfaceWriter()

```
gdcM::SurfaceWriter::~~SurfaceWriter ( ) [override]
```

### 10.297.3 Member Function Documentation

#### 10.297.3.1 ComputeNumberOfSurfaces()

```
void gdcM::SurfaceWriter::ComputeNumberOfSurfaces ( ) [protected]
```

### 10.297.3.2 GetNumberOfSurfaces()

```
unsigned long gdcm::SurfaceWriter::GetNumberOfSurfaces ( )
```

### 10.297.3.3 PrepareWrite()

```
bool gdcm::SurfaceWriter::PrepareWrite ( ) [protected]
```

### 10.297.3.4 PrepareWritePointMacro()

```
bool gdcm::SurfaceWriter::PrepareWritePointMacro (
    SmartPointer< Surface > surface,
    DataSet & surfaceDS,
    const TransferSyntax & ts ) [protected]
```

### 10.297.3.5 SetNumberOfSurfaces()

```
void gdcm::SurfaceWriter::SetNumberOfSurfaces (
    const unsigned long nb )
```

### 10.297.3.6 Write()

```
bool gdcm::SurfaceWriter::Write ( ) [override], [virtual]
```

Write.

Reimplemented from [gdcm::SegmentWriter](#).

## 10.297.4 Member Data Documentation

#### 10.297.4.1 NumberOfSurfaces

```
unsigned long gdcM::SurfaceWriter::NumberOfSurfaces [protected]
```

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

- [gdcMSurfaceWriter.h](#)

### 10.298 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)

## 10.298.1 Detailed Description

[SwapCode](#) representation.

Examples

[TestByteSwap.cxx](#).

## 10.298.2 Member Enumeration Documentation

### 10.298.2.1 SwapCodeType

```
enum gdcm::SwapCode::SwapCodeType
```

Enumerator

Unknown	
LittleEndian	
BigEndian	
BadLittleEndian	
BadBigEndian	

## 10.298.3 Constructor & Destructor Documentation

### 10.298.3.1 SwapCode()

```
gdcm::SwapCode::SwapCode (
    SwapCodeType sc = Unknown ) [inline]
```

## 10.298.4 Member Function Documentation

### 10.298.4.1 GetIndex()

```
static int gdcm::SwapCode::GetIndex (
    SwapCode const & sc ) [static], [protected]
```

#### 10.298.4.2 GetSwapCodeString()

```
static const char* gdcM::SwapCode::GetSwapCodeString (
    SwapCode const & sc ) [static]
```

#### 10.298.4.3 operator SwapCode::SwapCodeType()

```
gdcM::SwapCode::operator SwapCode::SwapCodeType ( ) const [inline]
```

### 10.298.5 Friends And Related Function Documentation

#### 10.298.5.1 operator<<

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

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

- [gdcMSwapCode.h](#)

## 10.299 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)

#### 10.299.1 Member Function Documentation

### 10.299.1.1 Swap()

```
template<typename T >
static T gdcm::SwapperDoOp::Swap (
    T val ) [static]
```

### 10.299.1.2 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](#)

## 10.300 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)

### 10.300.1 Detailed Description

#### Examples

[DumpSiemensBase64.cxx](#), and [DumpToshibaDTI.cxx](#).

### 10.300.2 Member Function Documentation

### 10.300.2.1 Swap()

```
template<typename T >
static T gdcm::SwapperNoOp::Swap (
    T val ) [inline], [static]
```

### 10.300.2.2 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](#)

## 10.301 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 sytem.*
- 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)

### 10.301.1 Detailed Description

Class to do system operation.

OS independent functionalities

### 10.301.2 Member Function Documentation

#### 10.301.2.1 ConvertToUNC()

```
static std::wstring gdcmm::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.

### 10.301.2.2 DeleteDirectory()

```
static bool gdcM::System::DeleteDirectory (
    const char * source ) [static]
```

remove a directory named source

### 10.301.2.3 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

### 10.301.2.4 FileExists()

```
static bool gdcM::System::FileExists (
    const char * filename ) [static]
```

Check whether the specified file exist on the sytem.

#### Examples

[DumpVisusChange.cxx](#), [EncapsulateFileInRawData.cxx](#), [MagnifyFile.cxx](#), and [gdcMorthoplanes.cxx](#).

### 10.301.2.5 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](#).

### 10.301.2.6 FileIsSymlink()

```
static bool gdcm::System::FileIsSymlink (
    const char * name ) [static]
```

Check whether name is a symlink.

### 10.301.2.7 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](#).

### 10.301.2.8 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

### 10.301.2.9 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]

**10.301.2.10 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](#).

**10.301.2.11 GetCurrentModuleFileName()**

```
static const char* gdcm::System::GetCurrentModuleFileName ( ) [static]
```

Return the directory the current module is located: NOT THREAD SAFE

**10.301.2.12 GetCurrentProcessFileName()**

```
static const char* gdcm::System::GetCurrentProcessFileName ( ) [static]
```

Return the directory the current process (executable) is located: NOT THREAD SAFE

**10.301.2.13 GetCurrentResourcesDirectory()**

```
static const char* gdcm::System::GetCurrentResourcesDirectory ( ) [static]
```

On some system (Apple) return the path to the current bundled 'Resources' directory NOT THREAD SAFE

**10.301.2.14 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

**10.301.2.15 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

#### 10.301.2.16 GetLastSystemError()

```
static const char* gdcm::System::GetLastSystemError ( ) [static]
```

Return the last error.

#### 10.301.2.17 GetLocaleCharset()

```
static const char* gdcm::System::GetLocaleCharset ( ) [static]
```

return locale charmap

#### 10.301.2.18 GetPermissions()

```
static bool gdcm::System::GetPermissions (
    const char * file,
    unsigned short & mode ) [static], [protected]
```

NOT THREAD SAFE.

#### 10.301.2.19 GetTimezoneOffsetFromUTC()

```
static const char* gdcm::System::GetTimezoneOffsetFromUTC ( ) [static]
```

Return the value for Timezone Offset From UTC as string.

##### Warning

not thread safe

#### 10.301.2.20 MakeDirectory()

```
static bool gdcm::System::MakeDirectory (
    const char * path ) [static]
```

Create a directory name path.

**10.301.2.21 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)

**10.301.2.22 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](#)

**10.301.2.23 RemoveFile()**

```
static bool gdcm::System::RemoveFile (
    const char * source ) [static]
```

remove a file named `source`

**10.301.2.24 SetPermissions()**

```
static bool gdcm::System::SetPermissions (
    const char * file,
    unsigned short mode ) [static], [protected]
```

### 10.301.2.25 StrCaseCmp()

```
static int gdcm::System::StrCaseCmp (
    const char * s1,
    const char * s2 ) [static]
```

consistent func for C99 spec of strcasecmp/strncasecmp

### 10.301.2.26 StrNCaseCmp()

```
static int gdcm::System::StrNCaseCmp (
    const char * s1,
    const char * s2,
    size_t n ) [static]
```

#### Precondition

$n \neq 0$

### 10.301.2.27 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

### 10.301.2.28 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:

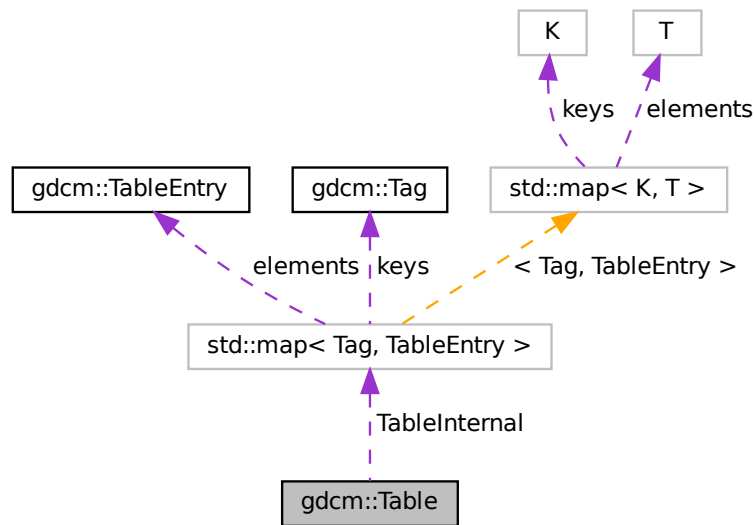
- [gdcmSystem.h](#)

## 10.302 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)`

### 10.302.1 Detailed Description

[Table](#).

### 10.302.2 Member Typedef Documentation

#### 10.302.2.1 MapTableEntry

```
typedef std::map<Tag, TableEntry> gdcm::Table::MapTableEntry
```

### 10.302.3 Constructor & Destructor Documentation

#### 10.302.3.1 [Table\(\)](#) [1/2]

```
gdcm::Table::Table ( ) [default]
```

#### 10.302.3.2 [~Table\(\)](#)

```
gdcm::Table::~~Table ( ) [default]
```

#### 10.302.3.3 [Table\(\)](#) [2/2]

```
gdcm::Table::Table (  
    const Table & _val ) [delete]
```

### 10.302.4 Member Function Documentation

#### 10.302.4.1 GetTableEntry()

```
const TableEntry& gdcM::Table::GetTableEntry (
    const Tag & tag ) const [inline]
```

References [TableInternal](#).

#### 10.302.4.2 InsertEntry()

```
void gdcM::Table::InsertEntry (
    Tag const & tag,
    TableEntry const & te ) [inline]
```

References [TableInternal](#).

#### 10.302.4.3 operator=()

```
Table& gdcM::Table::operator= (
    const Table & _val ) [delete]
```

### 10.302.5 Friends And Related Function Documentation

#### 10.302.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const Table & _val ) [friend]
```

### 10.302.6 Member Data Documentation

#### 10.302.6.1 TableInternal

[MapTableEntry](#) gdcM::Table::TableInternal

Referenced by [GetTableEntry\(\)](#), and [InsertEntry\(\)](#).

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

- [gdcMTable.h](#)

## 10.303 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

### 10.303.1 Detailed Description

[TableEntry.](#)

### 10.303.2 Constructor & Destructor Documentation

#### 10.303.2.1 TableEntry()

```
gdcm::TableEntry::TableEntry (  
    const char * attribute = nullptr,  
    Type const & type = Type(),  
    const char * des = nullptr ) [inline]
```

#### 10.303.2.2 ~TableEntry()

```
gdcm::TableEntry::~~TableEntry ( ) [default]
```

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

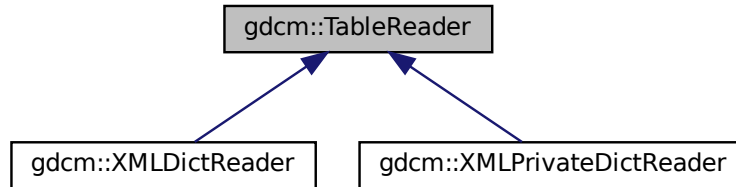
- [gdcmTableEntry.h](#)

## 10.304 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)

### 10.304.1 Detailed Description

Class for representing a [TableReader](#).

#### Note

This class is an empty shell meant to be derived

## 10.304.2 Constructor & Destructor Documentation

### 10.304.2.1 TableReader()

```
gdcm::TableReader::TableReader (
    Defs & defs ) [inline]
```

### 10.304.2.2 ~TableReader()

```
virtual gdcm::TableReader::~~TableReader ( ) [virtual], [default]
```

## 10.304.3 Member Function Documentation

### 10.304.3.1 CharacterDataHandler()

```
virtual void gdcm::TableReader::CharacterDataHandler (
    const char * data,
    int length ) [virtual]
```

Reimplemented in [gdcm::XMLPrivateDictReader](#), and [gdcm::XMLDictReader](#).

### 10.304.3.2 EndElement()

```
virtual void gdcm::TableReader::EndElement (
    const char * name ) [virtual]
```

Reimplemented in [gdcm::XMLPrivateDictReader](#), and [gdcm::XMLDictReader](#).

### 10.304.3.3 GetDefs()

```
const Defs& gdcm::TableReader::GetDefs ( ) const [inline]
```

#### 10.304.3.4 GetFilename()

```
const char* gdcM::TableReader::GetFilename ( ) [inline]
```

#### 10.304.3.5 HandleIOD()

```
void gdcM::TableReader::HandleIOD (
    const char ** atts )
```

#### 10.304.3.6 HandleIODEntry()

```
void gdcM::TableReader::HandleIODEntry (
    const char ** atts )
```

#### 10.304.3.7 HandleMacro()

```
void gdcM::TableReader::HandleMacro (
    const char ** atts )
```

#### 10.304.3.8 HandleMacroEntry()

```
void gdcM::TableReader::HandleMacroEntry (
    const char ** atts )
```

#### 10.304.3.9 HandleMacroEntryDescription()

```
void gdcM::TableReader::HandleMacroEntryDescription (
    const char ** atts )
```

#### 10.304.3.10 HandleModule()

```
void gdcm::TableReader::HandleModule (
    const char ** atts )
```

#### 10.304.3.11 HandleModuleEntry()

```
void gdcm::TableReader::HandleModuleEntry (
    const char ** atts )
```

#### 10.304.3.12 HandleModuleEntryDescription()

```
void gdcm::TableReader::HandleModuleEntryDescription (
    const char ** atts )
```

#### 10.304.3.13 HandleModuleInclude()

```
void gdcm::TableReader::HandleModuleInclude (
    const char ** atts )
```

#### 10.304.3.14 Read()

```
int gdcm::TableReader::Read ( )
```

#### 10.304.3.15 SetFilename()

```
void gdcm::TableReader::SetFilename (
    const char * filename ) [inline]
```

### 10.304.3.16 StartElement()

```
virtual void gdcM::TableReader::StartElement (
    const char * name,
    const char ** atts ) [virtual]
```

Reimplemented in [gdcM::XMLPrivateDictReader](#), and [gdcM::XMLDictReader](#).

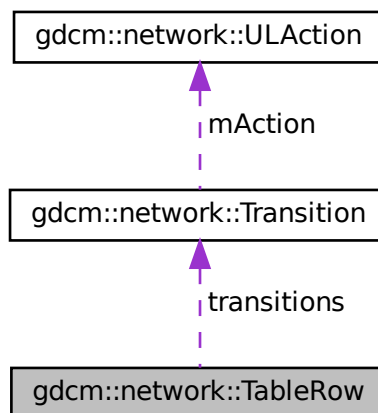
The documentation for this class was generated from the following file:

- [gdcMTableReader.h](#)

## 10.305 gdcM::network::TableRow Class Reference

```
#include <gdcMULTransitionTable.h>
```

Collaboration diagram for gdcM::network::TableRow:



### Public Member Functions

- [TableRow](#) ()
- [~TableRow](#) ()

### Public Attributes

- [Transition](#) \* `transitions` [`cMaxStateID`]



## 10.305.1 Constructor & Destructor Documentation

### 10.305.1.1 TableRow()

```
gdcm::network::TableRow::TableRow ( ) [inline]
```

References `gdcm::network::cMaxStateID`, and `transitions`.

### 10.305.1.2 ~TableRow()

```
gdcm::network::TableRow::~~TableRow ( ) [inline]
```

References `gdcm::network::cMaxStateID`, and `transitions`.

## 10.305.2 Member Data Documentation

### 10.305.2.1 transitions

```
Transition* gdcm::network::TableRow::transitions[cMaxStateID]
```

Referenced by `TableRow()`, and `~TableRow()`.

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

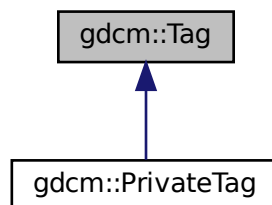
- [gdcmULTransitionTable.h](#)

## 10.306 gdcm::Tag Class Reference

Class to represent a DICOM Data [Element](#) ([Attribute](#)) [Tag](#) (Group, [Element](#)).

```
#include <gdcmTag.h>
```

Inheritance diagram for `gdcm::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)

## 10.306.1 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

[ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [DeriveSeries.cxx](#), [DumpToSQLITE3.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_V](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenAIIVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [MergeTwoFiles.cxx](#), [PatchFile.cxx](#), [PublicDict.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2](#), [ReadExplicitLengthSQIVR.cxx](#), [SimpleScanner.cxx](#), [SortImage.cxx](#), [StreamImageReaderTest.cxx](#), [VolumeSorter.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

## 10.306.2 Constructor & Destructor Documentation

#### 10.306.2.1 Tag() [1/3]

```
gdcm::Tag::Tag (
    uint16_t group,
    uint16_t element ) [inline]
```

Constructor with 2\*uint16\_t.

#### 10.306.2.2 Tag() [2/3]

```
gdcm::Tag::Tag (
    uint32_t tag = 0 ) [inline]
```

Constructor with 1\*uint32\_t Prefer the ctor that takes two uint16\_t.

#### 10.306.2.3 Tag() [3/3]

```
gdcm::Tag::Tag (
    const Tag & _val ) [inline]
```

References tag.

### 10.306.3 Member Function Documentation

#### 10.306.3.1 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().

### 10.306.3.2 GetElementTag()

```
uint32_t gdcm::Tag::GetElementTag ( ) const [inline]
```

Returns the full tag value of the given [Tag](#).

### 10.306.3.3 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()`.

### 10.306.3.4 GetLength()

```
uint32_t gdcm::Tag::GetLength ( ) const [inline]
```

return the length of tag (read: size on disk)

### 10.306.3.5 GetPrivateCreator()

```
Tag gdcm::Tag::GetPrivateCreator ( ) const [inline]
```

Return the Private Creator Data [Element](#) tag of a private data element.

References `SetElement()`.

### 10.306.3.6 IsGroupLength()

```
bool gdcm::Tag::IsGroupLength ( ) const [inline]
```

return whether the tag correspond to a group length tag:

### 10.306.3.7 IsGroupXX()

```
bool gdcM::Tag::IsGroupXX (
    const Tag & t ) const [inline]
```

e.g 6002,3000 belong to groupXX: 6000,3000

References GetElement(), GetGroup(), and IsPrivate().

### 10.306.3.8 IsIllegal()

```
bool gdcM::Tag::IsIllegal ( ) const [inline]
```

return if the tag is considered to be an illegal tag

### 10.306.3.9 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().

### 10.306.3.10 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](#).

### 10.306.3.11 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...

### 10.306.3.12 operator!=(())

```
bool gdcm::Tag::operator!= (
    const Tag & _val ) const [inline]
```

References tag.

### 10.306.3.13 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.

### 10.306.3.14 operator<=()

```
bool gdcm::Tag::operator<= (
    const Tag & t2 ) const [inline]
```

### 10.306.3.15 operator=()

```
Tag& gdcm::Tag::operator= (
    const Tag & _val ) [inline]
```

References tag.

### 10.306.3.16 operator==( )

```
bool gdcm::Tag::operator==(
    const Tag & _val ) const [inline]
```

References tag.

### 10.306.3.17 operator[]() [1/2]

```
uint16_t& gdcm::Tag::operator[] (
    const unsigned int & _id ) [inline]
```

Returns the Group or [Element](#) of the given [Tag](#), depending on id (0/1)

### 10.306.3.18 operator[]() [2/2]

```
const uint16_t& gdcm::Tag::operator[] (
    const unsigned int & _id ) const [inline]
```

Returns the Group or [Element](#) of the given [Tag](#), depending on id (0/1)

### 10.306.3.19 PrintAsContinuousString()

```
std::string gdcm::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)

### 10.306.3.20 PrintAsContinuousUpperCaseString()

```
std::string gdcm::Tag::PrintAsContinuousUpperCaseString ( ) const
```

Same as PrintAsContinuousString, but hexadecimal [a-f] are printed using upper case.



### 10.306.3.21 PrintAsPipeSeparatedString()

```
std::string gdcm::Tag::PrintAsPipeSeparatedString ( ) const
```

Print as a pipe separated string (GDCM 1.x compat only). Do not use in newer code

See also

[ReadFromPipeSeparatedString](#)

### 10.306.3.22 Read()

```
template<typename TSwap >  
std::istream& gdcm::Tag::Read (   
    std::istream & is ) [inline]
```

Read a tag from binary representation.

### 10.306.3.23 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

### 10.306.3.24 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)

### 10.306.3.25 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](#)

#### 10.306.3.26 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\(\)](#).

#### 10.306.3.27 SetElementTag() [1/2]

```
void gdcm::Tag::SetElementTag (
    uint16_t group,
    uint16_t element ) [inline]
```

Sets the 'Group number' & '[Element](#) number' of the given [Tag](#).

#### 10.306.3.28 SetElementTag() [2/2]

```
void gdcm::Tag::SetElementTag (
    uint32_t tag ) [inline]
```

Sets the full tag value of the given [Tag](#).

#### 10.306.3.29 SetGroup()

```
void gdcm::Tag::SetGroup (
    uint16_t group ) [inline]
```

Sets the 'Group number' of the given [Tag](#).

### 10.306.3.30 SetPrivateCreator()

```
void gdcm::Tag::SetPrivateCreator (
    Tag const & t ) [inline]
```

Set private creator:

#### Examples

[DuplicatePCDE.cxx](#).

References `GetElement()`, `GetGroup()`, and `IsPrivate()`.

### 10.306.3.31 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()`.

## 10.306.4 Friends And Related Function Documentation

### 10.306.4.1 operator<<

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

### 10.306.4.2 operator>>

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

## 10.306.5 Member Data Documentation

### 10.306.5.1 bytes

```
char gdcM::Tag::bytes[4]
```

### 10.306.5.2 tag

```
uint32_t gdcM::Tag::tag
```

Referenced by Tag(), operator!=(), operator<(), operator=(), and operator==().

### 10.306.5.3 tags

```
uint16_t gdcM::Tag::tags[2]
```

Referenced by operator<().

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

- [gdcMTag.h](#)

## 10.307 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.*

### 10.307.1 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)

### 10.307.2 Constructor & Destructor Documentation

#### 10.307.2.1 TagPath()

```
gdcm::TagPath::TagPath ( )
```

#### 10.307.2.2 ~TagPath()

```
gdcm::TagPath::~~TagPath ( )
```

### 10.307.3 Member Function Documentation

#### 10.307.3.1 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

### 10.307.3.2 ConstructFromTagList()

```
bool gdcmm::TagPath::ConstructFromTagList (
    Tag const * l,
    unsigned int n )
```

Construct from a list of tags.

### 10.307.3.3 IsValid()

```
static bool gdcmm::TagPath::IsValid (
    const char * path ) [static]
```

Return if path is valid or not.

### 10.307.3.4 Print()

```
void gdcmm::TagPath::Print (
    std::ostream & ) const
```

### 10.307.3.5 Push() [1/2]

```
bool gdcmm::TagPath::Push (
    Tag const & t )
```

### 10.307.3.6 Push() [2/2]

```
bool gdcmm::TagPath::Push (
    unsigned int itemnum )
```

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

- [gdcmmTagPath.h](#)

## 10.308 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.*

### 10.308.1 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

### 10.308.2 Member Typedef Documentation

#### 10.308.2.1 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

#### 10.308.2.2 MediaStorageDataFilesType

```
typedef const char* const(* gdcm::Testing::MediaStorageDataFilesType) [2]
```

return the table that map the media storage (as string) of a filename (gdcmData)

### 10.308.3 Constructor & Destructor Documentation

#### 10.308.3.1 Testing()

```
gdcm::Testing::Testing ( ) [default]
```



### 10.308.3.2 ~Testing()

```
gdcm::Testing::~~Testing ( ) [default]
```

## 10.308.4 Member Function Documentation

### 10.308.4.1 ComputeFileMD5()

```
static bool gdcm::Testing::ComputeFileMD5 (
    const char * filename,
    char digest_str[33] ) [static]
```

#### Examples

[MetalImageMD5Activiz.cs](#).

### 10.308.4.2 ComputeMD5()

```
static bool gdcm::Testing::ComputeMD5 (
    const char * buffer,
    size_t buf_len,
    char digest_str[33] ) [static]
```

[MD5](#) stuff digest\_str needs to be at least : strlen = [2\*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 [gdcm::MD5](#) API when doing md5 computation.

### 10.308.4.3 GetDataExtraRoot()

```
static const char* gdcm::Testing::GetDataExtraRoot ( ) [static]
```

Return the GDCM DATA EXTRA ROOT.

#### Examples

[DiscriminateVolume.cxx](#), [VolumeSorter.cxx](#), and [reslicesphere.cxx](#).

#### 10.308.4.4 GetDataRoot()

```
static const char* gdcM::Testing::GetDataRoot ( ) [static]
```

Return the GDCM DATA ROOT.

##### Examples

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), and [MagnifyFile.cxx](#).

#### 10.308.4.5 GetFileName()

```
static const char* gdcM::Testing::GetFileName (
    unsigned int file ) [static]
```

##### Examples

[MetaImageMD5Activiz.cs](#).

#### 10.308.4.6 GetFileNames()

```
static const char* const* gdcM::Testing::GetFileNames ( ) [static]
```

return the table of fullpath to gdcMData DICOM files:

##### Examples

[TestReader.cxx](#).

#### 10.308.4.7 GetLossyFlagFromFile()

```
static int gdcM::Testing::GetLossyFlagFromFile (
    const char * filepath ) [static]
```

Return the lossy flag of the given filename -1 -> Error 0 -> Lossless 1 -> Lossy

#### 10.308.4.8 GetMD5DataImage()

```
static const char* const* gdcm::Testing::GetMD5DataImage (
    unsigned int file ) [static]
```

#### 10.308.4.9 GetMD5DataImages()

```
static MD5DataImagesType gdcm::Testing::GetMD5DataImages ( ) [static]
```

#### 10.308.4.10 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.

#### 10.308.4.11 GetMD5FromFile()

```
static const char* gdcm::Testing::GetMD5FromFile (
    const char * filepath ) [static]
```

#### 10.308.4.12 GetMediaStorageDataFile()

```
static const char* const* gdcm::Testing::GetMediaStorageDataFile (
    unsigned int file ) [static]
```

#### 10.308.4.13 GetMediaStorageDataFiles()

```
static MediaStorageDataFilesType gdcm::Testing::GetMediaStorageDataFiles ( ) [static]
```

**10.308.4.14 GetMediaStorageFromFile()**

```
static const char* gdcM::Testing::GetMediaStorageFromFile (
    const char * filepath ) [static]
```

**Examples**

[MetaImageMD5Activiz.cs](#), and [TestReader.cxx](#).

**10.308.4.15 GetNumberOfFileNames()**

```
static unsigned int gdcM::Testing::GetNumberOfFileNames ( ) [static]
```

**Examples**

[MetaImageMD5Activiz.cs](#).

**10.308.4.16 GetNumberOfMD5DataImages()**

```
static unsigned int gdcM::Testing::GetNumberOfMD5DataImages ( ) [static]
```

**10.308.4.17 GetNumberOfMediaStorageDataFiles()**

```
static unsigned int gdcM::Testing::GetNumberOfMediaStorageDataFiles ( ) [static]
```

**10.308.4.18 GetPixelSpacingDataRoot()**

```
static const char* gdcM::Testing::GetPixelSpacingDataRoot ( ) [static]
```

Return the GDCM PIXEL SPACING DATA ROOT (See David Clunie website for dataset)

#### 10.308.4.19 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

#### 10.308.4.20 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

#### 10.308.4.21 GetSourceDirectory()

```
static const char* gdcm::Testing::GetSourceDirectory ( ) [static]
```

##### Examples

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

#### 10.308.4.22 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

#### 10.308.4.23 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

[MetaImageMD5Activiz.cs](#).

#### 10.308.4.24 GetTempDirectoryW()

```
static const wchar_t* gdcm::Testing::GetTempDirectoryW (
    const wchar_t * subdir = nullptr ) [static]
```

NOT THREAD SAFE.

#### 10.308.4.25 GetTempFilename()

```
static const char* gdcm::Testing::GetTempFilename (
    const char * filename,
    const char * subdir = nullptr ) [static]
```

NOT THREAD SAFE.

#### Examples

[MetaImageMD5Activiz.cs](#).

#### 10.308.4.26 GetTempFilenameW()

```
static const wchar_t* gdcm::Testing::GetTempFilenameW (
    const wchar_t * filename,
    const wchar_t * subdir = nullptr ) [static]
```

NOT THREAD SAFE.

#### 10.308.4.27 Print()

```
void gdcm::Testing::Print (
    std::ostream & os = std::cout )
```

Print.

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

- [gdcmTesting.h](#)

## 10.309 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](#) ()

### 10.309.1 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.

### 10.309.2 Constructor & Destructor Documentation

#### 10.309.2.1 `Trace()`

```
gdcM::Trace::Trace ( )
```

#### 10.309.2.2 `~Trace()`

```
gdcM::Trace::~~Trace ( )
```

### 10.309.3 Member Function Documentation

#### 10.309.3.1 `DebugOff()`

```
static void gdcM::Trace::DebugOff ( ) [static]
```

#### Examples

[MetalImageMD5Activiz.cs](#), and [TestReader.cxx](#).



### 10.309.3.2 DebugOn()

```
static void gdcm::Trace::DebugOn ( ) [static]
```

#### Examples

[CreateFakePET.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

### 10.309.3.3 ErrorOff()

```
static void gdcm::Trace::ErrorOff ( ) [static]
```

#### Examples

[MetaImageMD5Activiz.cs](#).

### 10.309.3.4 ErrorOn()

```
static void gdcm::Trace::ErrorOn ( ) [static]
```

### 10.309.3.5 GetDebugFlag()

```
static bool gdcm::Trace::GetDebugFlag ( ) [static]
```

### 10.309.3.6 GetDebugStream()

```
static std::ostream& gdcm::Trace::GetDebugStream ( ) [static]
```

### 10.309.3.7 GetErrorFlag()

```
static bool gdcm::Trace::GetErrorFlag ( ) [static]
```

#### 10.309.3.8 GetErrorStream()

```
static std::ostream& gdcm::Trace::GetErrorStream ( ) [static]
```

#### 10.309.3.9 GetStream()

```
static std::ostream& gdcm::Trace::GetStream ( ) [static]
```

#### 10.309.3.10 GetWarningFlag()

```
static bool gdcm::Trace::GetWarningFlag ( ) [static]
```

#### 10.309.3.11 GetWarningStream()

```
static std::ostream& gdcm::Trace::GetWarningStream ( ) [static]
```

#### 10.309.3.12 SetDebug()

```
static void gdcm::Trace::SetDebug (
    bool debug ) [static]
```

Turn debug messages on (default: false)

#### Examples

[DumpToSQLITE3.cxx](#).

#### 10.309.3.13 SetDebugStream()

```
static void gdcm::Trace::SetDebugStream (
    std::ostream & os ) [static]
```

Explicitly set the stream which receive Debug messages:

#### 10.309.3.14 SetError()

```
static void gdcm::Trace::SetError (
    bool debug ) [static]
```

Turn error messages on (default: true)

#### 10.309.3.15 SetErrorStream()

```
static void gdcm::Trace::SetErrorStream (
    std::ostream & os ) [static]
```

Explicitly set the stream which receive Error messages:

##### Examples

[CStoreQtProgress.cxx](#).

#### 10.309.3.16 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:

#### 10.309.3.17 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)

#### 10.309.3.18 SetWarning()

```
static void gdcm::Trace::SetWarning (
    bool debug ) [static]
```

Turn warning messages on (default: true)

##### Examples

[DumpToSQLITE3.cxx](#).

### 10.309.3.19 SetWarningStream()

```
static void gdcM::Trace::SetWarningStream (
    std::ostream & os ) [static]
```

Explicitly set the stream which receive Warning messages:

### 10.309.3.20 WarningOff()

```
static void gdcM::Trace::WarningOff ( ) [static]
```

#### Examples

[MetalImageMD5Activiz.cs](#), and [TestReader.cxx](#).

### 10.309.3.21 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](#)

## 10.310 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](#) ,  
[JPEGsSpectralSelectionProcess6\\_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)

### 10.310.1 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

[GetJPEGSamplePrecision.cxx](#), [LargeVRDSExplicit.cxx](#), and [MakeTemplate.cxx](#).

### 10.310.2 Member Enumeration Documentation

#### 10.310.2.1 NegotiatedType

```
enum gdcm::TransferSyntax::NegociatedType
```

#### Enumerator

Unknown	
Explicit	
Implicit	

### 10.310.2.2 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	
JPIPReferenced	
MPEG2MainProfileHighLevel	
MPEG4AVCH264HighProfileLevel4_1	
MPEG4AVCH264BDcompatibleHighProfileLevel4↔ _1	
TS_END	

### 10.310.3 Constructor & Destructor Documentation

### 10.310.3.1 TransferSyntax()

```
gdcm::TransferSyntax::TransferSyntax (
    TSType type = ImplicitVRLittleEndian ) [inline]
```

## 10.310.4 Member Function Documentation

### 10.310.4.1 CanStoreLossy()

```
bool gdcm::TransferSyntax::CanStoreLossy ( ) const
```

return true if TransFer Syntax Allow storing of Lossy Pixel Data

### 10.310.4.2 GetNegociatedType()

```
NegociatedType gdcm::TransferSyntax::GetNegociatedType ( ) const
```

### 10.310.4.3 GetString()

```
const char* gdcm::TransferSyntax::GetString ( ) const [inline]
```

References GetTSString().

### 10.310.4.4 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.



#### 10.310.4.5 GetTSString()

```
static const char* gdcm::TransferSyntax::GetTSString (
    TSType ts ) [static]
```

##### Examples

[LargeVRDSExplicit.cxx](#).

Referenced by GetString().

#### 10.310.4.6 GetTSType()

```
static TSType gdcm::TransferSyntax::GetTSType (
    const char * str ) [static]
```

#### 10.310.4.7 IsEncapsulated()

```
bool gdcm::TransferSyntax::IsEncapsulated ( ) const
```

##### Examples

[ExtractIconFromFile.cxx](#).

#### 10.310.4.8 IsEncoded()

```
bool gdcm::TransferSyntax::IsEncoded ( ) const
```

#### 10.310.4.9 IsExplicit()

```
bool gdcm::TransferSyntax::IsExplicit ( ) const
```

#### 10.310.4.10 IsImplicit()

```
bool gdcM::TransferSyntax::IsImplicit ( ) const
```

#### 10.310.4.11 IsLossless()

```
bool gdcM::TransferSyntax::IsLossless ( ) const
```

Return true if the transfer syntax algorithm is a lossless algorithm

#### 10.310.4.12 IsLossy()

```
bool gdcM::TransferSyntax::IsLossy ( ) const
```

Return true if the transfer syntax algorithm is a lossy algorithm

#### 10.310.4.13 IsValid()

```
bool gdcM::TransferSyntax::IsValid ( ) const [inline]
```

#### 10.310.4.14 operator TType()

```
gdcM::TransferSyntax::operator TType ( ) const [inline]
```

### 10.310.5 Friends And Related Function Documentation

#### 10.310.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const TransferSyntax & ts ) [friend]
```

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

- [gdcMTransferSyntax.h](#)

## 10.311 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

### 10.311.1 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

### 10.311.2 Constructor & Destructor Documentation

#### 10.311.2.1 TransferSyntaxSub()

```
gdcm::network::TransferSyntaxSub::TransferSyntaxSub ( )
```

### 10.311.3 Member Function Documentation

**10.311.3.1 GetName()**

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

**10.311.3.2 operator==()**

```
bool gdcm::network::TransferSyntaxSub::operator== (
    const TransferSyntaxSub & ts ) const [inline]
```

**10.311.3.3 Print()**

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

**10.311.3.4 Read()**

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

**10.311.3.5 SetName()**

```
void gdcm::network::TransferSyntaxSub::SetName (
    const char * name )
```

**10.311.3.6 SetNameFromUID()**

```
void gdcm::network::TransferSyntaxSub::SetNameFromUID (
    UIDs::TSName tsname )
```

### 10.311.3.7 Size()

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

### 10.311.3.8 Write()

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

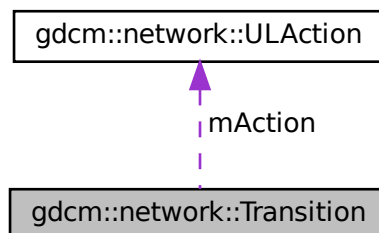
The documentation for this class was generated from the following file:

- [gdcmTransferSyntaxSub.h](#)

## 10.312 gdcm::network::Transition Struct Reference

```
#include <gdcmULTransitionTable.h>
```

Collaboration diagram for gdcm::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](#)

## 10.312.1 Constructor & Destructor Documentation

### 10.312.1.1 Transition() [1/2]

```
gdcmm::network::Transition::Transition ( ) [inline]
```

References [gdcmm::network::eStaDoesNotExist](#), [mAction](#), and [mEnd](#).

Referenced by [MakeNew\(\)](#).

### 10.312.1.2 ~Transition()

```
gdcmm::network::Transition::~~Transition ( ) [inline]
```

References [mAction](#).

### 10.312.1.3 Transition() [2/2]

```
gdcmm::network::Transition::Transition (
    int inEndState,
    ULAction * inAction ) [inline]
```

References [mAction](#), and [mEnd](#).

## 10.312.2 Member Function Documentation

### 10.312.2.1 MakeNew()

```
static Transition* gdcmm::network::Transition::MakeNew (
    int inEndState,
    ULAction * inAction ) [inline], [static]
```

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

### 10.312.3 Member Data Documentation

#### 10.312.3.1 mAction

[ULAction\\*](#) gdcm::network::Transition::mAction

Referenced by [Transition\(\)](#), and [~Transition\(\)](#).

#### 10.312.3.2 mEnd

int gdcm::network::Transition::mEnd

Referenced by [Transition\(\)](#).

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

- [gdcmULTransitionTable.h](#)

## 10.313 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)

### 10.313.1 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.

### 10.313.2 Member Enumeration Documentation

#### 10.313.2.1 TypeType

```
enum gdcm::Type::TypeType
```

#### Enumerator

T1	
T1C	
T2	
T2C	
T3	
UNKNOWN	

### 10.313.3 Constructor & Destructor Documentation



### 10.313.3.1 Type()

```
gdcm::Type::Type (
    TypeType type = UNKNOWN ) [inline]
```

## 10.313.4 Member Function Documentation

### 10.313.4.1 GetTypeString()

```
static const char* gdcm::Type::GetTypeString (
    TypeType type ) [static]
```

### 10.313.4.2 GetTypeType()

```
static TypeType gdcm::Type::GetTypeType (
    const char * type ) [static]
```

Referenced by `gdcm::ModuleEntry::ModuleEntry()`.

### 10.313.4.3 operator TypeType()

```
gdcm::Type::operator TypeType ( ) const [inline]
```

## 10.313.5 Friends And Related Function Documentation

### 10.313.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const Type & vr ) [friend]
```

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

- [gdcmType.h](#)

## 10.314 gdcm::UI Struct Reference

```
#include <gdcmVR.h>
```

### Public Attributes

- char [Internal](#) [64+1]

### Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [UI](#) &\_val)

## 10.314.1 Friends And Related Function Documentation

### 10.314.1.1 operator<<

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

## 10.314.2 Member Data Documentation

### 10.314.2.1 Internal

```
char gdcm::UI::Internal[64+1]
```

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

- [gdcmVR.h](#)

## 10.315 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)

### 10.315.1 Detailed Description

Class for generating unique UID.

#### Note

bla [Usage](#): 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](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), and [uid\\_unique.cxx](#).

### 10.315.2 Constructor & Destructor Documentation

#### 10.315.2.1 UIDGenerator()

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

By default the root of a UID is a GDCM Root...

### 10.315.3 Member Function Documentation

#### 10.315.3.1 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.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), and [uid\\_unique.cxx](#).

#### 10.315.3.2 GenerateUUID()

```
static bool gdcm::UIDGenerator::GenerateUUID (
    unsigned char * uuid_data ) [static], [protected]
```

#### 10.315.3.3 GetGDCMUID()

```
static const char* gdcm::UIDGenerator::GetGDCMUID ( ) [static]
```

Return the default (GDCM) root UID:

#### 10.315.3.4 GetRoot()

```
static const char* gdcm::UIDGenerator::GetRoot ( ) [static]
```

##### Examples

[ClinicalTrialIdentificationWorkflow.cs](#), [ReformatFile.cs](#), and [StandardizeFiles.cs](#).

### 10.315.3.5 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)

### 10.315.3.6 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 responsablility 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](#)

## 10.316 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 ,
  - [JPIPReferenced](#) = 30 ,
  - [JPIPReferencedDeflate](#) = 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 ,
  - [SPM2CSFFrameofReference](#) = 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 ,  
[MayoClinicNonradiologicalImagesSBSSAnatomicalSurfaceRegionGuide](#) = 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 ,  
[OphthalmicOpticalCoherenceTomographyEnFacImageStorage](#) = 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)

### 10.316.1 Detailed Description

all known uids

Examples

[GenerateStandardSOPClasses.cxx](#).

### 10.316.2 Member Typedef Documentation

#### 10.316.2.1 TransferSyntaxStringsType

```
typedef const char* const (* gdcm::UIDs::TransferSyntaxStringsType) [2]
```

### 10.316.3 Member Enumeration Documentation

#### 10.316.3.1 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	
JPIPReferencedDeflate	
MPEG2MainProfileMainLevel	
RLELossless	
RFC2557MIMEencapsulation	
XMLEncoding	

## Enumerator

MediaStorageDirectoryStorage	
TalairachBrainAtlasFrameofReference	
SPM2T1FrameofReference	
SPM2T2FrameofReference	
SPM2PDFFrameofReference	
SPM2EPIFrameofReference	
SPM2FILT1FrameofReference	
SPM2PETFrameofReference	
SPM2TRANSMFrameofReference	
SPM2SPECTFrameofReference	
SPM2GRAYFrameofReference	
SPM2WHITEFrameofReference	
SPM2CSFFrameofReference	
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	
ModalityPerformedProcedureStepNotificationSOPClass	
DetachedResultsManagementSOPClassRetired	
DetachedResultsManagementMetaSOPClassRetired	
DetachedStudyManagementMetaSOPClassRetired	

## Enumerator

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	
UltrasoundMultiframeImageStorage	
MRImageStorage	
EnhancedMRImageStorage	
MRSpectroscopyStorage	



## Enumerator

NuclearMedicineImageStorageRetired	
UltrasoundImageStorageRetired	
UltrasoundImageStorage	
SecondaryCaptureImageStorage	
MultiframeSingleBitSecondaryCaptureImageStorage	
MultiframeGrayscaleByteSecondaryCaptureImageStorage	
MultiframeGrayscaleWordSecondaryCaptureImageStorage	
MultiframeTrueColorSecondaryCaptureImageStorage	
StandaloneOverlayStorageRetired	
StandaloneCurveStorageRetired	
WaveformStorageTrialRetired	
ECG12leadWaveformStorage	
GeneralECGWaveformStorage	
AmbulatoryECGWaveformStorage	
HemodynamicWaveformStorage	
CardiacElectrophysiologyWaveformStorage	
BasicVoiceAudioWaveformStorage	
StandaloneModalityLUTStorageRetired	
StandaloneVOILUTStorageRetired	
GrayscaleSoftcopyPresentationStateStorageSOPClass	
ColorSoftcopyPresentationStateStorageSOPClass	
PseudoColorSoftcopyPresentationStateStorageSOPClass	
BlendingSoftcopyPresentationStateStorageSOPClass	
XRayAngiographicImageStorage	
EnhancedXAImageStorage	
XRayRadiofluoroscopicImageStorage	
EnhancedXRFImageStorage	
XRay3DAngiographicImageStorage	
XRay3DCraniofacialImageStorage	
XRayAngiographicBiPlaneImageStorageRetired	
NuclearMedicineImageStorage	
RawDataStorage	
SpatialRegistrationStorage	
SpatialFiducialsStorage	
DeformableSpatialRegistrationStorage	
SegmentationStorage	
RealWorldValueMappingStorage	
VLImageStorageTrialRetired	
VLMultiframeImageStorageTrialRetired	
VLEndoscopicImageStorage	
VideoEndoscopicImageStorage	
VLMicroscopicImageStorage	
VideoMicroscopicImageStorage	

## Enumerator

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	
PatientStudyOnlyQueryRetrieveInformationModelMOVERetired	
PatientStudyOnlyQueryRetrieveInformationModelGETRetired	
ModalityWorklistInformationModelFIND	
GeneralPurposeWorklistInformationModelFIND	

## Enumerator

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	
dicomTLSCyphersuite	
dicomAuthorizedNodeCertificateReference	
dicomThisNodeCertificateReference	
dicomInstalled	

## Enumerator

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
NewYorkUniversityMelanomaClinicalCooperativeGroup
MayoClinicNonradiologicalImagesSBSAnatomicalSurfaceRegionGuide
ImageBiomarkerStandardisationInitiative
RadiomicsOntology

## Enumerator

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	
IntraocularLensCalculationsStorage	
MacularGridThicknessandVolumeReportStorage	
OphthalmicVisualFieldStaticPerimetryMeasurementsStorage	
OphthalmicThicknessMapStorage	

## Enumerator

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	
UnifiedWorklistandProcedureStepServiceClass1	
UnifiedProcedureStepPushSOPClass1	
UnifiedProcedureStepWatchSOPClass1	
UnifiedProcedureStepPullSOPClass1	

## Enumerator

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	

## 10.316.3.2 TSType

```
enum gdcmm::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	
uid_1_2_840_10008_1_2_4_50	
uid_1_2_840_10008_1_2_4_51	

## Enumerator

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	
uid_1_2_840_10008_1_4_1_16	
uid_1_2_840_10008_1_4_1_17	



## Enumerator

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	
uid_1_2_840_10008_5_1_1_23	
uid_1_2_840_10008_5_1_1_24	

## Enumerator

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	
uid_1_2_840_10008_5_1_4_1_1_11_3	
uid_1_2_840_10008_5_1_4_1_1_11_4	

## Enumerator

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	
uid_1_2_840_10008_5_1_4_1_1_88_67	
uid_1_2_840_10008_5_1_4_1_1_104_1	

## Enumerator

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	
uid_1_2_840_10008_15_0_3_1	
uid_1_2_840_10008_15_0_3_2	

## Enumerator

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	
uid_1_2_840_10008_1_2_4_101	
uid_1_2_840_10008_1_2_4_102	

## Enumerator

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	
uid_1_2_840_10008_5_1_4_1_1_66_6	
uid_1_2_840_10008_5_1_4_1_1_68_1	

## Enumerator

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	
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	

## Enumerator

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	



## 10.316.4 Member Function Documentation

### 10.316.4.1 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<<()`.

### 10.316.4.2 GetNumberOfTransferSyntaxStrings()

```
static unsigned int gdcm::UIDs::GetNumberOfTransferSyntaxStrings ( ) [static]
```

### 10.316.4.3 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<<()`.

### 10.316.4.4 GetTransferSyntaxString()

```
static const char* const* gdcm::UIDs::GetTransferSyntaxString (
    unsigned int ts ) [static]
```

#### 10.316.4.5 GetTransferSyntaxStrings()

```
static TransferSyntaxStringsType gdcM::UIDs::GetTransferSyntaxStrings ( ) [static]
```

#### 10.316.4.6 GetUIDName()

```
static const char* gdcM::UIDs::GetUIDName (
    unsigned int ts ) [static]
```

#### 10.316.4.7 GetUIDString()

```
static const char* gdcM::UIDs::GetUIDString (
    unsigned int ts ) [static]
```

#### 10.316.4.8 operator TSType()

```
gdcM::UIDs::operator TSType ( ) const [inline]
```

#### 10.316.4.9 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:

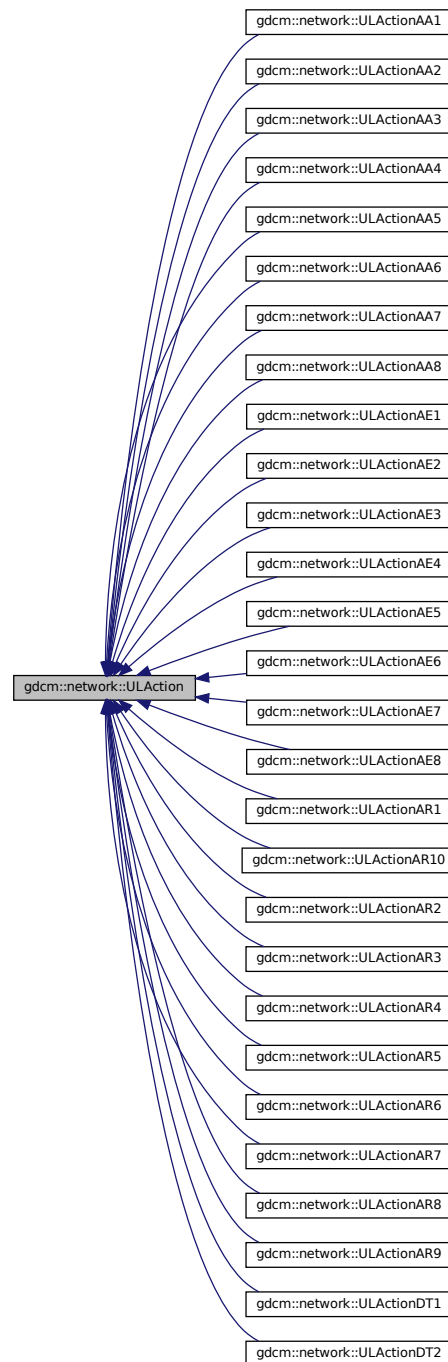
- [gdcMUIDs.h](#)

## 10.317 gdcm::network::ULAction Class Reference

[ULAction.](#)

```
#include <gdcmULAction.h>
```

Inheritance diagram for gdcm::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

### 10.317.1 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 thos 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.

### 10.317.2 Constructor & Destructor Documentation

#### 10.317.2.1 [ULAction](#)() [1/2]

```
gdcmm::network::ULAction::ULAction ( ) [default]
```

### 10.317.2.2 ~ULAction()

```
virtual gdcm::network::ULAction::~~ULAction ( ) [virtual], [default]
```

### 10.317.2.3 ULAction() [2/2]

```
gdcm::network::ULAction::ULAction (
    const ULAction & inAction ) [delete]
```

## 10.317.3 Member Function Documentation

### 10.317.3.1 operator=()

```
void gdcm::network::ULAction::operator= (
    const ULAction & ) [delete]
```

### 10.317.3.2 PerformAction()

```
virtual EStateID gdcm::network::ULAction::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [pure virtual]
```

Implemented in [gdcm::network::ULActionDT2](#), [gdcm::network::ULActionDT1](#), [gdcm::network::ULActionAR10](#), [gdcm::network::ULActionAR9](#), [gdcm::network::ULActionAR8](#), [gdcm::network::ULActionAR7](#), [gdcm::network::ULActionAR6](#), [gdcm::network::ULActionAR5](#), [gdcm::network::ULActionAR4](#), [gdcm::network::ULActionAR3](#), [gdcm::network::ULActionAR2](#), [gdcm::network::ULActionAR1](#), [gdcm::network::ULActionAE8](#), [gdcm::network::ULActionAE7](#), [gdcm::network::ULActionAE6](#), [gdcm::network::ULActionAE5](#), [gdcm::network::ULActionAE4](#), [gdcm::network::ULActionAE3](#), [gdcm::network::ULActionAE2](#), [gdcm::network::ULActionAE1](#), [gdcm::network::ULActionAA8](#), [gdcm::network::ULActionAA7](#), [gdcm::network::ULActionAA6](#), [gdcm::network::ULActionAA5](#), [gdcm::network::ULActionAA4](#), [gdcm::network::ULActionAA3](#), [gdcm::network::ULActionAA2](#), and [gdcm::network::ULActionAA1](#).

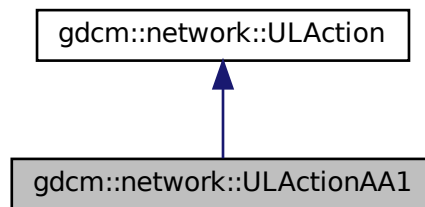
The documentation for this class was generated from the following file:

- [gdcmULAction.h](#)

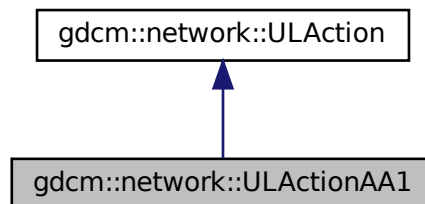
## 10.318 gdcm::network::ULActionAA1 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA1:



Collaboration diagram for gdcm::network::ULActionAA1:



### Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

### 10.318.1 Member Function Documentation

### 10.318.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA1::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

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

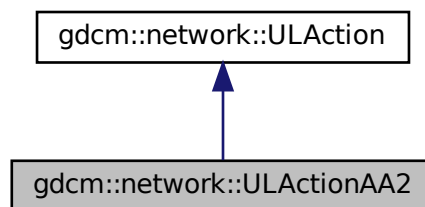
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

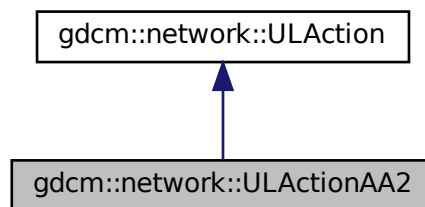
## 10.319 gdcm::network::ULActionAA2 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA2:



Collaboration diagram for gdcm::network::ULActionAA2:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

### 10.319.1 Member Function Documentation

#### 10.319.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAA2::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcmm::network::ULAction](#).

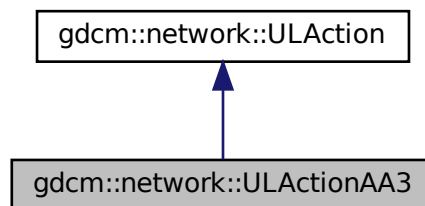
The documentation for this class was generated from the following file:

- [gdcmmULActionAA.h](#)

## 10.320 gdcmm::network::ULActionAA3 Class Reference

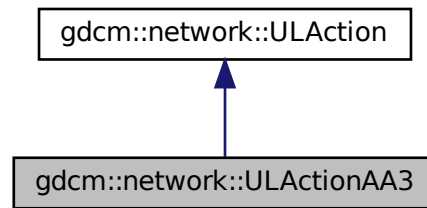
```
#include <gdcmmULActionAA.h>
```

Inheritance diagram for gdcmm::network::ULActionAA3:





Collaboration diagram for gdcm::network::ULActionAA3:



## Public Member Functions

- `EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)` override

### 10.320.1 Member Function Documentation

#### 10.320.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA3::PerformAction (  
    Subject * s,  
    ULEvent & inEvent,  
    ULConnection & inConnection,  
    bool & outWaitingForEvent,  
    EEventID & outRaisedEvent ) [override], [virtual]
```

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

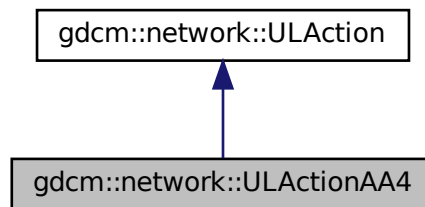
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

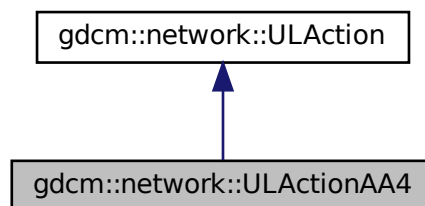
## 10.321 gdcm::network::ULActionAA4 Class Reference

```
#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

### 10.321.1 Member Function Documentation

### 10.321.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA4::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

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

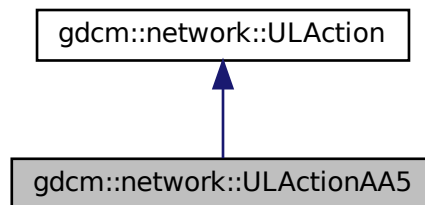
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

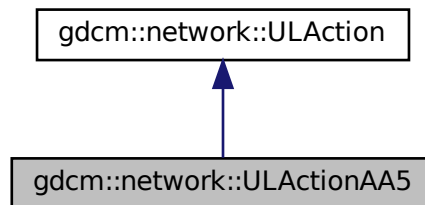
## 10.322 gdcm::network::ULActionAA5 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA5:



Collaboration diagram for gdcm::network::ULActionAA5:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

### 10.322.1 Member Function Documentation

#### 10.322.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAA5::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcmm::network::ULAction](#).

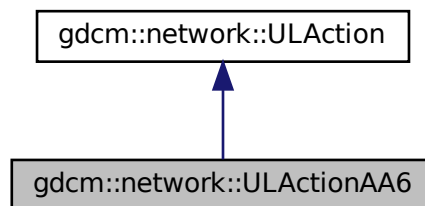
The documentation for this class was generated from the following file:

- [gdcmmULActionAA.h](#)

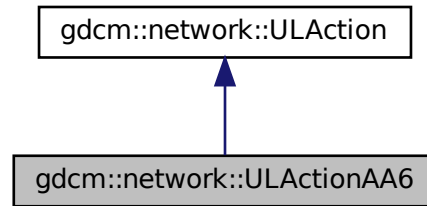
### 10.323 gdcmm::network::ULActionAA6 Class Reference

```
#include <gdcmmULActionAA.h>
```

Inheritance diagram for gdcmm::network::ULActionAA6:



Collaboration diagram for gdcm::network::ULActionAA6:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

### 10.323.1 Member Function Documentation

#### 10.323.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA6::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

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

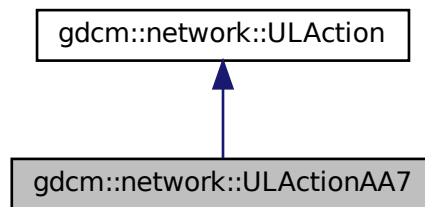
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

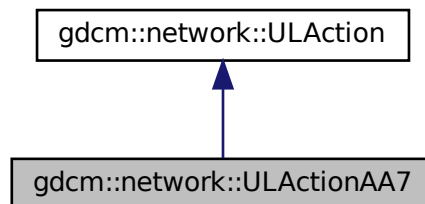
## 10.324 gdcm::network::ULActionAA7 Class Reference

```
#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

### 10.324.1 Member Function Documentation

### 10.324.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA7::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

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

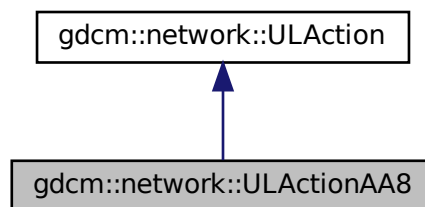
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

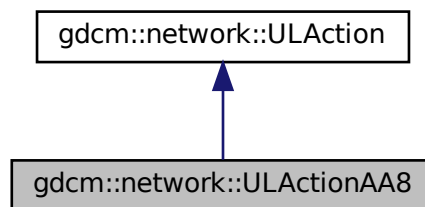
## 10.325 gdcm::network::ULActionAA8 Class Reference

```
#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

### 10.325.1 Member Function Documentation

#### 10.325.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAA8::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcmm::network::ULAction](#).

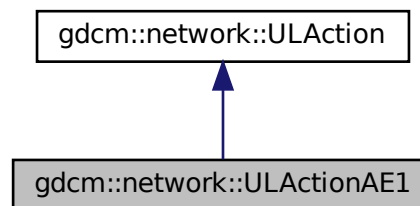
The documentation for this class was generated from the following file:

- [gdcmmULActionAA.h](#)

### 10.326 gdcmm::network::ULActionAE1 Class Reference

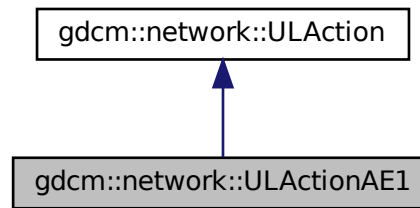
```
#include <gdcmmULActionAE.h>
```

Inheritance diagram for gdcmm::network::ULActionAE1:





Collaboration diagram for gdcm::network::ULActionAE1:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

### 10.326.1 Member Function Documentation

#### 10.326.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE1::PerformAction (  
    Subject * s,  
    ULEvent & inEvent,  
    ULConnection & inConnection,  
    bool & outWaitingForEvent,  
    EEventID & outRaisedEvent ) [override], [virtual]
```

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

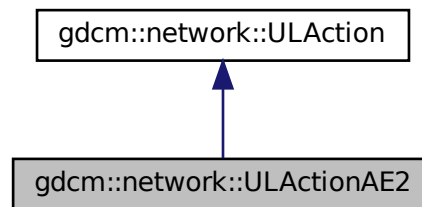
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

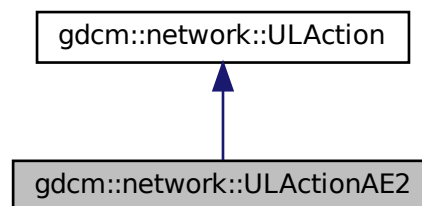
## 10.327 gdcm::network::ULActionAE2 Class Reference

```
#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`

### 10.327.1 Member Function Documentation

### 10.327.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE2::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

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

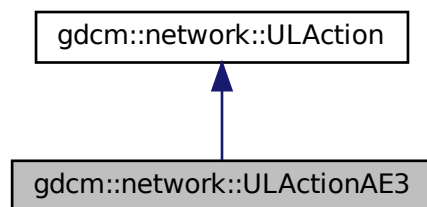
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

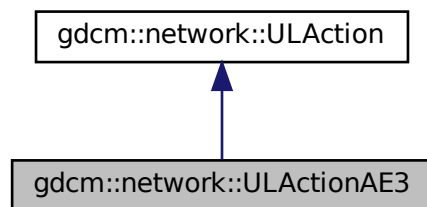
## 10.328 gdcm::network::ULActionAE3 Class Reference

```
#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

### 10.328.1 Member Function Documentation

#### 10.328.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAE3::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcmm::network::ULAction](#).

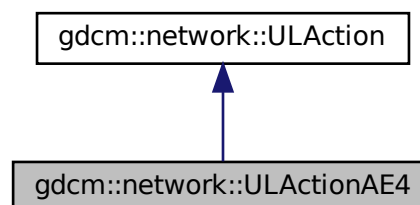
The documentation for this class was generated from the following file:

- [gdcmmULActionAE.h](#)

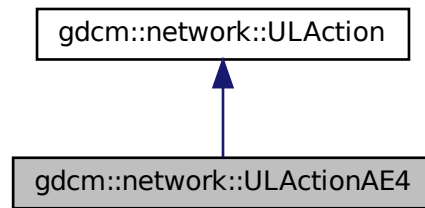
### 10.329 gdcmm::network::ULActionAE4 Class Reference

```
#include <gdcmmULActionAE.h>
```

Inheritance diagram for [gdcmm::network::ULActionAE4](#):



Collaboration diagram for gdcm::network::ULActionAE4:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

### 10.329.1 Member Function Documentation

#### 10.329.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE4::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

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

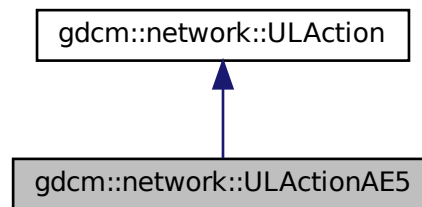
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

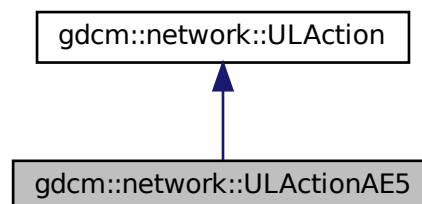
## 10.330 gdcm::network::ULActionAE5 Class Reference

```
#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

### 10.330.1 Member Function Documentation

### 10.330.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE5::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

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

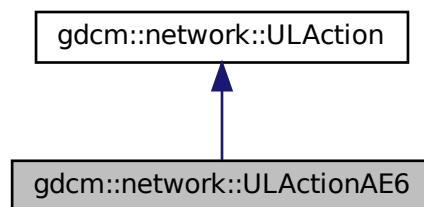
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

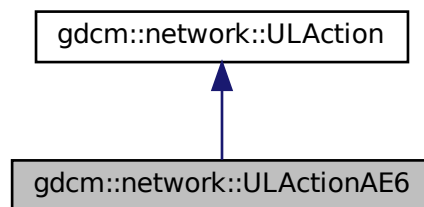
## 10.331 gdcm::network::ULActionAE6 Class Reference

```
#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

### 10.331.1 Member Function Documentation

#### 10.331.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAE6::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcmm::network::ULAction](#).

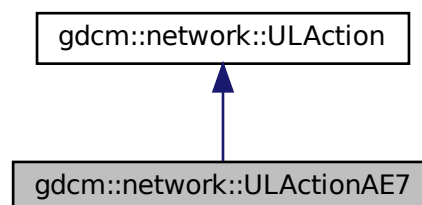
The documentation for this class was generated from the following file:

- [gdcmmULActionAE.h](#)

## 10.332 gdcmm::network::ULActionAE7 Class Reference

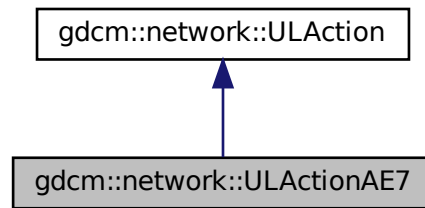
```
#include <gdcmmULActionAE.h>
```

Inheritance diagram for gdcmm::network::ULActionAE7:





Collaboration diagram for gdcm::network::ULActionAE7:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

### 10.332.1 Member Function Documentation

#### 10.332.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE7::PerformAction (  
    Subject * s,  
    ULEvent & inEvent,  
    ULConnection & inConnection,  
    bool & outWaitingForEvent,  
    EEventID & outRaisedEvent ) [override], [virtual]
```

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

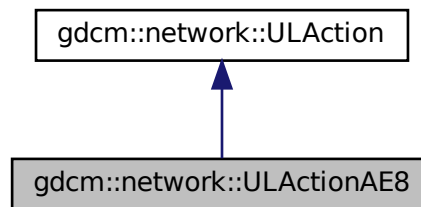
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

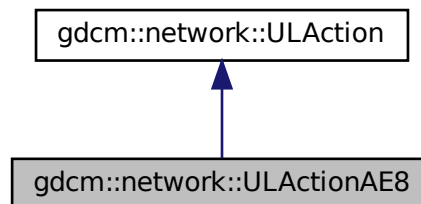
### 10.333 gdcm::network::ULActionAE8 Class Reference

```
#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

#### 10.333.1 Member Function Documentation

### 10.333.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE8::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

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

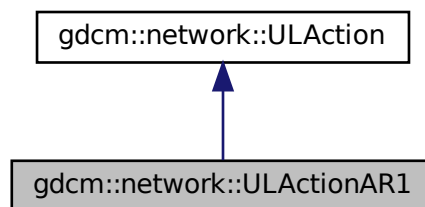
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

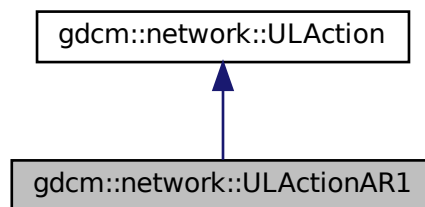
## 10.334 gdcm::network::ULActionAR1 Class Reference

```
#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

### 10.334.1 Member Function Documentation

#### 10.334.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAR1::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcmm::network::ULAction](#).

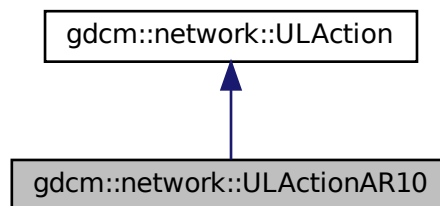
The documentation for this class was generated from the following file:

- [gdcmmULActionAR.h](#)

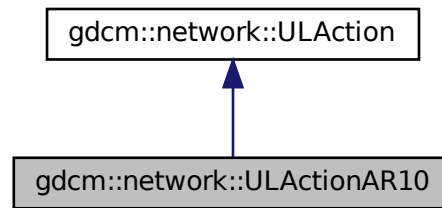
### 10.335 gdcmm::network::ULActionAR10 Class Reference

```
#include <gdcmmULActionAR.h>
```

Inheritance diagram for gdcmm::network::ULActionAR10:



Collaboration diagram for gdcm::network::ULActionAR10:



## Public Member Functions

- `EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) override`

### 10.335.1 Member Function Documentation

#### 10.335.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR10::PerformAction (  
    Subject * s,  
    ULEvent & inEvent,  
    ULConnection & inConnection,  
    bool & outWaitingForEvent,  
    EEventID & outRaisedEvent ) [override], [virtual]
```

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

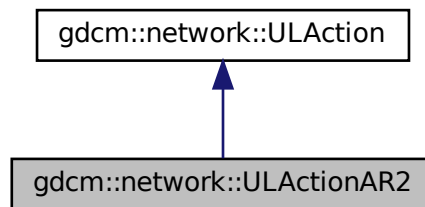
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

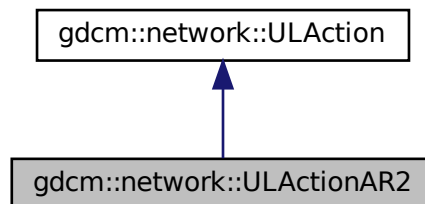
## 10.336 gdcm::network::ULActionAR2 Class Reference

```
#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

### 10.336.1 Member Function Documentation

### 10.336.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR2::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

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

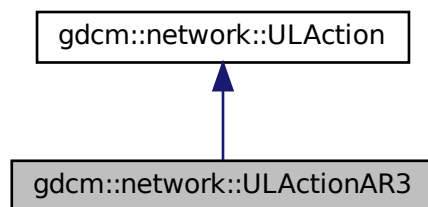
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

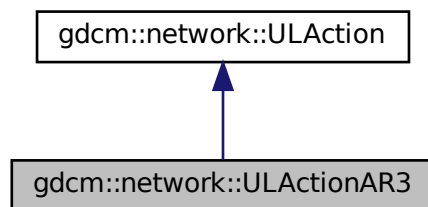
## 10.337 gdcm::network::ULActionAR3 Class Reference

```
#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

### 10.337.1 Member Function Documentation

#### 10.337.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAR3::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcmm::network::ULAction](#).

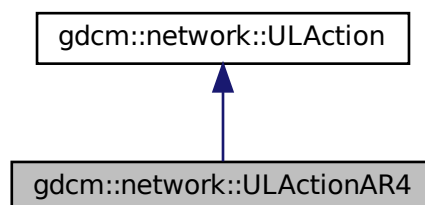
The documentation for this class was generated from the following file:

- [gdcmmULActionAR.h](#)

### 10.338 gdcmm::network::ULActionAR4 Class Reference

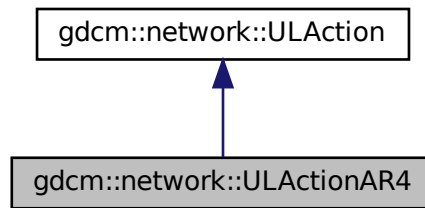
```
#include <gdcmmULActionAR.h>
```

Inheritance diagram for gdcmm::network::ULActionAR4:





Collaboration diagram for gdcm::network::ULActionAR4:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

### 10.338.1 Member Function Documentation

#### 10.338.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR4::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

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

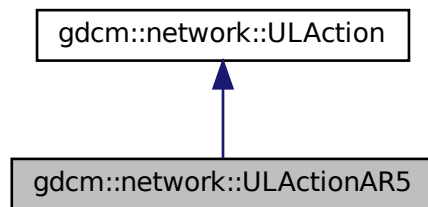
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

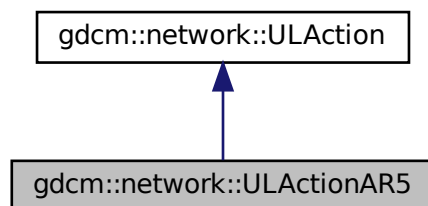
## 10.339 gdcm::network::ULActionAR5 Class Reference

```
#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

### 10.339.1 Member Function Documentation

### 10.339.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR5::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

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

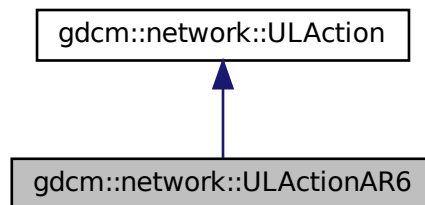
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

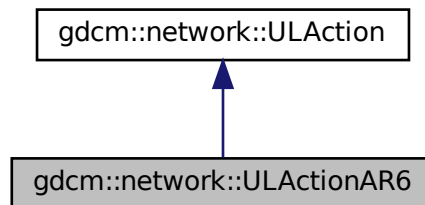
## 10.340 gdcm::network::ULActionAR6 Class Reference

```
#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

### 10.340.1 Member Function Documentation

#### 10.340.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAR6::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcmm::network::ULAction](#).

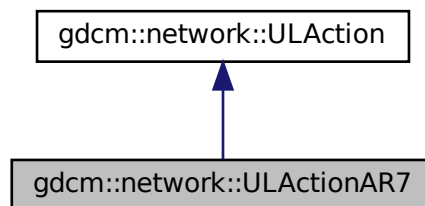
The documentation for this class was generated from the following file:

- [gdcmmULActionAR.h](#)

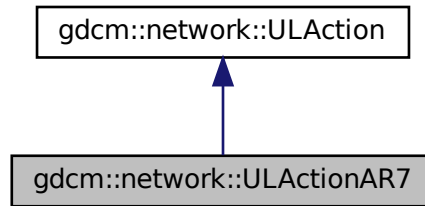
## 10.341 gdcmm::network::ULActionAR7 Class Reference

```
#include <gdcmmULActionAR.h>
```

Inheritance diagram for gdcmm::network::ULActionAR7:



Collaboration diagram for gdcm::network::ULActionAR7:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

### 10.341.1 Member Function Documentation

#### 10.341.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR7::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

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

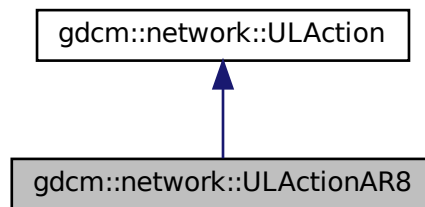
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

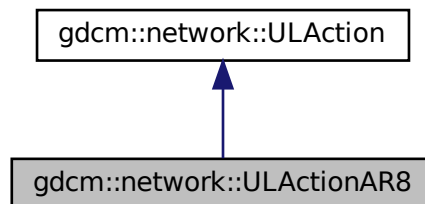
## 10.342 gdcm::network::ULActionAR8 Class Reference

```
#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

### 10.342.1 Member Function Documentation

### 10.342.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR8::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

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

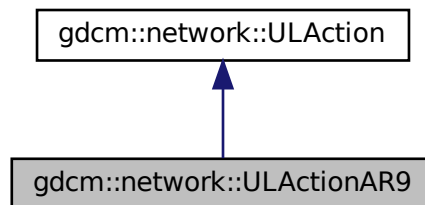
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

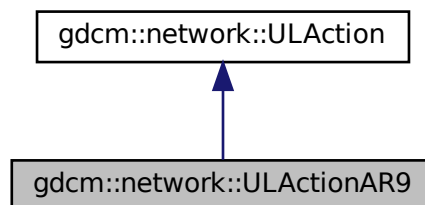
## 10.343 gdcm::network::ULActionAR9 Class Reference

```
#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

### 10.343.1 Member Function Documentation

#### 10.343.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAR9::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

Implements [gdcmm::network::ULAction](#).

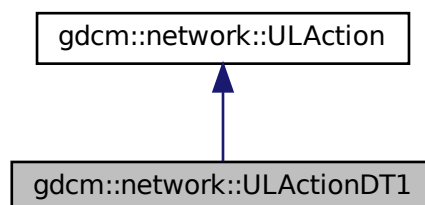
The documentation for this class was generated from the following file:

- [gdcmmULActionAR.h](#)

### 10.344 gdcmm::network::ULActionDT1 Class Reference

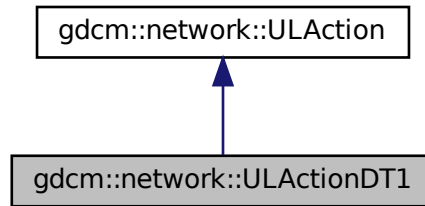
```
#include <gdcmmULActionDT.h>
```

Inheritance diagram for gdcmm::network::ULActionDT1:





Collaboration diagram for gdcm::network::ULActionDT1:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

### 10.344.1 Member Function Documentation

#### 10.344.1.1 PerformAction()

```
EStateID gdcm::network::ULActionDT1::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [override], [virtual]
```

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

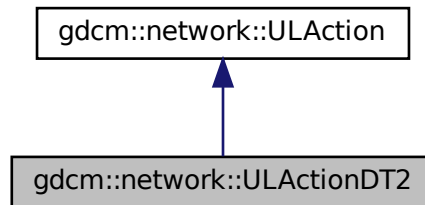
The documentation for this class was generated from the following file:

- [gdcmULActionDT.h](#)

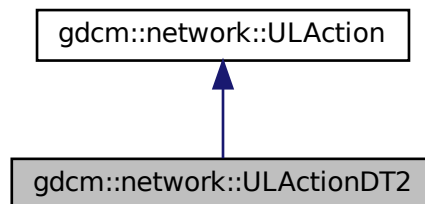
## 10.345 gdcm::network::ULActionDT2 Class Reference

```
#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

### 10.345.1 Member Function Documentation

### 10.345.1.1 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:

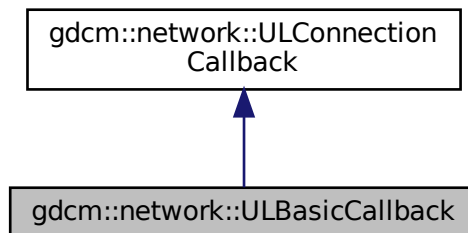
- [gdcmULActionDT.h](#)

## 10.346 gdcm::network::ULBasicCallback Class Reference

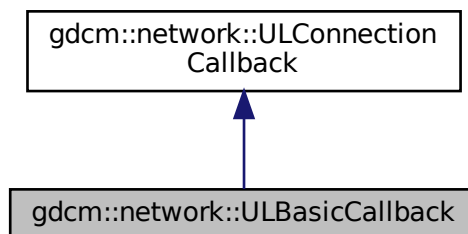
[ULBasicCallback](#).

```
#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`

## Additional Inherited Members

### 10.346.1 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](#).

### 10.346.2 Constructor & Destructor Documentation

#### 10.346.2.1 [ULBasicCallback\(\)](#)

```
gdcmm::network::ULBasicCallback::ULBasicCallback ( ) [default]
```

#### 10.346.2.2 [~ULBasicCallback\(\)](#)

```
gdcmm::network::ULBasicCallback::~~ULBasicCallback ( ) [override], [default]
```

### 10.346.3 Member Function Documentation

#### 10.346.3.1 [GetDataSets\(\)](#)

```
std::vector<DataSet> const& gdcmm::network::ULBasicCallback::GetDataSets ( ) const
```

### 10.346.3.2 GetResponses()

```
std::vector<DataSet> const& gdcm::network::ULBasicCallback::GetResponses ( ) const
```

### 10.346.3.3 HandleDataSet()

```
void gdcm::network::ULBasicCallback::HandleDataSet (
    const DataSet & inDataSet ) [override], [virtual]
```

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

### 10.346.3.4 HandleResponse()

```
void gdcm::network::ULBasicCallback::HandleResponse (
    const DataSet & inDataSet ) [override], [virtual]
```

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

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

- [gdcmULBasicCallback.h](#)

## 10.347 gdcm::network::ULConnection Class Reference

[ULConnection](#).

```
#include <gdcmULConnection.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](#)

## 10.347.1 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 [gdcmm](#) 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.

## 10.347.2 Constructor & Destructor Documentation

### 10.347.2.1 ULConnection() [1/2]

```
gdcm::network::ULConnection::ULConnection (
    const ULConnectionInfo & inUserInfo )
```

### 10.347.2.2 ~ULConnection()

```
virtual gdcm::network::ULConnection::~~ULConnection ( ) [virtual]
```

### 10.347.2.3 ULConnection() [2/2]

```
gdcm::network::ULConnection::ULConnection (
    const ULConnection & ) [delete]
```

## 10.347.3 Member Function Documentation

### 10.347.3.1 AddAcceptedPresentationContext()

```
void gdcm::network::ULConnection::AddAcceptedPresentationContext (
    const PresentationContextAC & inPC )
```

### 10.347.3.2 FindContext()

```
PresentationContextRQ gdcm::network::ULConnection::FindContext (
    const DataElement & de ) const
```

**10.347.3.3 GetAcceptedPresentationContexts() [1/2]**

```
std::vector<PresentationContextAC>& gdc::network::ULConnection::GetAcceptedPresentationContexts  
( )
```

**10.347.3.4 GetAcceptedPresentationContexts() [2/2]**

```
std::vector<PresentationContextAC> const& gdc::network::ULConnection::GetAcceptedPresentation↵  
Contexts ( ) const
```

**10.347.3.5 GetConnectionInfo()**

```
const ULConnectionInfo& gdc::network::ULConnection::GetConnectionInfo ( ) const
```

**10.347.3.6 GetMaxPDUSize()**

```
uint32_t gdc::network::ULConnection::GetMaxPDUSize ( ) const
```

**10.347.3.7 GetPresentationContextACByID()**

```
const PresentationContextAC* gdc::network::ULConnection::GetPresentationContextACByID (   
    uint8_t id ) const
```

**10.347.3.8 GetPresentationContextIDFromPresentationContext()**

```
uint8_t gdc::network::ULConnection::GetPresentationContextIDFromPresentationContext (   
    PresentationContextRQ const & pc ) const
```

return 0 upon error



### 10.347.3.9 GetPresentationContextRQByID()

```
const PresentationContextRQ* gdcm::network::ULConnection::GetPresentationContextRQByID (
    uint8_t id ) const
```

### 10.347.3.10 GetPresentationContexts()

```
std::vector<PresentationContextRQ> const& gdcm::network::ULConnection::GetPresentationContexts (
) const
```

### 10.347.3.11 GetProtocol()

```
std::iostream* gdcm::network::ULConnection::GetProtocol ( )
```

### 10.347.3.12 GetState()

```
EStateID gdcm::network::ULConnection::GetState ( ) const
```

### 10.347.3.13 GetTimer()

```
ARTIMTimer& gdcm::network::ULConnection::GetTimer ( )
```

### 10.347.3.14 InitializeConnection()

```
bool gdcm::network::ULConnection::InitializeConnection ( )
```

used to establish scu connections

**10.347.3.15 InitializeIncomingConnection()**

```
bool gdcm::network::ULConnection::InitializeIncomingConnection ( )
```

used to establish scp connections

**10.347.3.16 operator=()**

```
void gdcm::network::ULConnection::operator= (
    const ULConnection & ) [delete]
```

**10.347.3.17 SetMaxPDUSize()**

```
void gdcm::network::ULConnection::SetMaxPDUSize (
    uint32_t inSize )
```

**10.347.3.18 SetPresentationContexts() [1/2]**

```
void gdcm::network::ULConnection::SetPresentationContexts (
    const std::vector< PresentationContext > & inContexts )
```

**10.347.3.19 SetPresentationContexts() [2/2]**

```
void gdcm::network::ULConnection::SetPresentationContexts (
    const std::vector< PresentationContextRQ > & inContexts )
```

**10.347.3.20 SetState()**

```
void gdcm::network::ULConnection::SetState (
    const EStateID & inState )
```

### 10.347.3.21 StopProtocol()

```
void gdcm::network::ULConnection::StopProtocol ( )
```

## 10.347.4 Friends And Related Function Documentation

### 10.347.4.1 ULActionAE6

```
friend class ULActionAE6 [friend]
```

### 10.347.4.2 ULConnectionManager

```
friend class ULConnectionManager [friend]
```

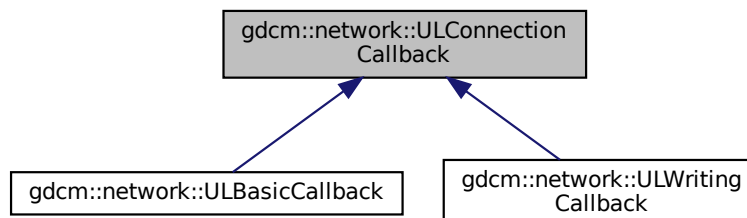
The documentation for this class was generated from the following file:

- [gdcmULConnection.h](#)

## 10.348 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](#)

### 10.348.1 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](#) ← Set to true. otherwise, the cmove event loop handler will not know data was received, and proceed to end the loop prematurely.

### 10.348.2 Constructor & Destructor Documentation

#### 10.348.2.1 [ULConnectionCallback](#)()

```
gdcm::network::ULConnectionCallback::ULConnectionCallback ( ) [inline]
```

#### 10.348.2.2 [~ULConnectionCallback](#)()

```
virtual gdcm::network::ULConnectionCallback::~~ULConnectionCallback ( ) [virtual], [default]
```

## 10.348.3 Member Function Documentation

### 10.348.3.1 DataSetHandled()

```
void gdcm::network::ULConnectionCallback::DataSetHandled ( ) [inline], [protected]
```

### 10.348.3.2 DataSetHandles()

```
bool gdcm::network::ULConnectionCallback::DataSetHandles ( ) const [inline]
```

### 10.348.3.3 HandleDataSet()

```
virtual void gdcm::network::ULConnectionCallback::HandleDataSet (
    const DataSet & inDataSet ) [pure virtual]
```

Implemented in [gdcm::network::ULWritingCallback](#), and [gdcm::network::ULBasicCallback](#).

### 10.348.3.4 HandleResponse()

```
virtual void gdcm::network::ULConnectionCallback::HandleResponse (
    const DataSet & inDataSet ) [pure virtual]
```

Implemented in [gdcm::network::ULWritingCallback](#), and [gdcm::network::ULBasicCallback](#).

### 10.348.3.5 ResetHandledDataSet()

```
void gdcm::network::ULConnectionCallback::ResetHandledDataSet ( ) [inline]
```

### 10.348.3.6 SetImplicitFlag()

```
void gdcm::network::ULConnectionCallback::SetImplicitFlag (
    const bool imp ) [inline]
```

## 10.348.4 Member Data Documentation

### 10.348.4.1 mImplicit

```
bool gdcml::network::ULConnectionCallback::mImplicit [protected]
```

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

- [gdcmlULConnectionCallback.h](#)

## 10.349 gdcml::network::ULConnectionInfo Class Reference

[ULConnectionInfo](#).

```
#include <gdcmlULConnectionInfo.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](#) ([UserInfo](#) const &inUserInfo, const char \*inCalledAETitle, const char \*inCallingAETitle, unsigned long inCalledIPAddress, int inCalledIPPort, std::string inCalledComputerName)
- void [SetMaxPDULength](#) (unsigned long inMaxPDULength)

### 10.349.1 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.

### 10.349.2 Constructor & Destructor Documentation

### 10.349.2.1 ULConnectionInfo()

```
gdcm::network::ULConnectionInfo::ULConnectionInfo ( )
```

## 10.349.3 Member Function Documentation

### 10.349.3.1 GetCalledAETitle()

```
const char* gdcm::network::ULConnectionInfo::GetCalledAETitle ( ) const
```

### 10.349.3.2 GetCalledComputerName()

```
std::string gdcm::network::ULConnectionInfo::GetCalledComputerName ( ) const
```

### 10.349.3.3 GetCalledIPAddress()

```
unsigned long gdcm::network::ULConnectionInfo::GetCalledIPAddress ( ) const
```

### 10.349.3.4 GetCalledIPPort()

```
int gdcm::network::ULConnectionInfo::GetCalledIPPort ( ) const
```

### 10.349.3.5 GetCallingAETitle()

```
const char* gdcm::network::ULConnectionInfo::GetCallingAETitle ( ) const
```

### 10.349.3.6 GetMaxPDULength()

```
unsigned long gdcm::network::ULConnectionInfo::GetMaxPDULength ( ) const
```

### 10.349.3.7 Initialize()

```
bool gdcmm::network::ULConnectionInfo::Initialize (
    UserInformation const & inUserInformation,
    const char * inCalledAETitle,
    const char * inCallingAETitle,
    unsigned long inCalledIPAddress,
    int inCalledIPPort,
    std::string inCalledComputerName )
```

### 10.349.3.8 SetMaxPDULength()

```
void gdcmm::network::ULConnectionInfo::SetMaxPDULength (
    unsigned long inMaxPDULength )
```

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

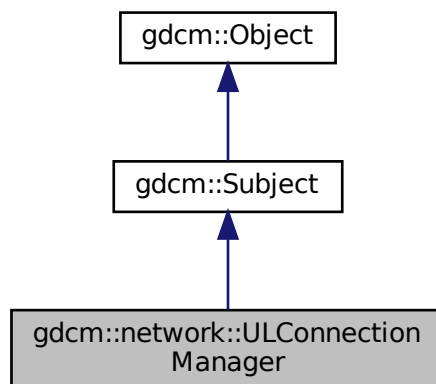
- [gdcmmULConnectionInfo.h](#)

## 10.350 gdcmm::network::ULConnectionManager Class Reference

[ULConnectionManager](#).

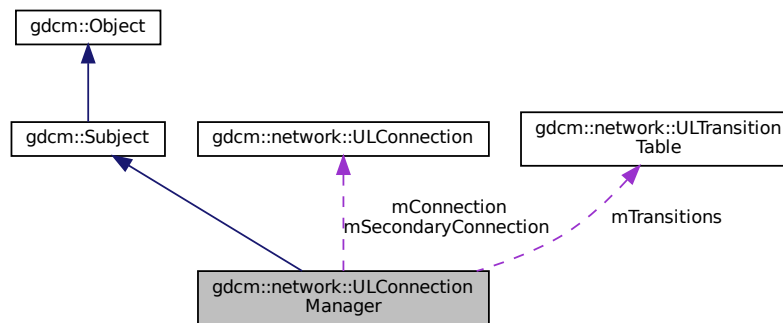
```
#include <gdcmmULConnectionManager.h>
```

Inheritance diagram for gdcmm::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*

## 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 Attributes

- [ULConnection](#) \* mConnection
- [ULConnection](#) \* mSecondaryConnection
- [ULTransitionTable](#) mTransitions

### 10.350.1 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.

### 10.350.2 Constructor & Destructor Documentation

#### 10.350.2.1 [ULConnectionManager\(\)](#) [1/2]

```
gdcmm::network::ULConnectionManager::ULConnectionManager (
    const ULConnectionManager & inCM ) [protected]
```

#### 10.350.2.2 [ULConnectionManager\(\)](#) [2/2]

```
gdcmm::network::ULConnectionManager::ULConnectionManager ( )
```

#### 10.350.2.3 [~ULConnectionManager\(\)](#)

```
gdcmm::network::ULConnectionManager::~~ULConnectionManager ( ) [override]
```

### 10.350.3 Member Function Documentation

#### 10.350.3.1 BreakConnection()

```
bool gdcm::network::ULConnectionManager::BreakConnection (
    const double & inTimeout )
```

#### 10.350.3.2 BreakConnectionNow()

```
void gdcm::network::ULConnectionManager::BreakConnectionNow ( )
```

#### 10.350.3.3 EstablishConnection()

```
bool gdcm::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.

#### 10.350.3.4 EstablishConnectionMove()

```
bool gdcm::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

#### 10.350.3.5 RunEventLoop()

```
EStateID gdcmm::network::ULConnectionManager::RunEventLoop (
    ULEvent & inEvent,
    ULConnection * inWhichConnection,
    ULConnectionCallback * inCallback,
    const bool & startWaiting ) [protected]
```

#### 10.350.3.6 RunMoveEventLoop()

```
EStateID gdcmm::network::ULConnectionManager::RunMoveEventLoop (
    ULEvent & inEvent,
    ULConnectionCallback * inCallback ) [protected]
```

#### 10.350.3.7 SendEcho()

```
std::vector<PresentationDataValue> gdcmm::network::ULConnectionManager::SendEcho ( )
```

#### 10.350.3.8 SendFind() [1/2]

```
std::vector<DataSet> gdcmm::network::ULConnectionManager::SendFind (
    const BaseRootQuery * inRootQuery )
```

#### 10.350.3.9 SendFind() [2/2]

```
void gdcmm::network::ULConnectionManager::SendFind (
    const BaseRootQuery * inRootQuery,
    ULConnectionCallback * inCallback )
```

#### 10.350.3.10 SendMove() [1/2]

```
std::vector<DataSet> gdcmm::network::ULConnectionManager::SendMove (
    const BaseRootQuery * inRootQuery )
```

**10.350.3.11 SendMove()** [2/2]

```
bool gdcm::network::ULConnectionManager::SendMove (
    const BaseRootQuery * inRootQuery,
    ULConnectionCallback * inCallback )
```

return false upon error

**10.350.3.12 SendNAction()** [1/2]

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendNAction (
    const BaseQuery * inQuery )
```

**10.350.3.13 SendNAction()** [2/2]

```
void gdcm::network::ULConnectionManager::SendNAction (
    const BaseQuery * inQuery,
    ULConnectionCallback * inCallback )
```

**10.350.3.14 SendNCreate()** [1/2]

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendNCreate (
    const BaseQuery * inQuery )
```

**10.350.3.15 SendNCreate()** [2/2]

```
void gdcm::network::ULConnectionManager::SendNCreate (
    const BaseQuery * inQuery,
    ULConnectionCallback * inCallback )
```

**10.350.3.16 SendNDelete()** [1/2]

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendNDelete (
    const BaseQuery * inQuery )
```

**10.350.3.17 SendNDelete() [2/2]**

```
void gdcm::network::ULConnectionManager::SendNDelete (
    const BaseQuery * inQuery,
    ULConnectionCallback * inCallback )
```

**10.350.3.18 SendNEventReport() [1/2]**

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendNEventReport (
    const BaseQuery * inQuery )
```

**10.350.3.19 SendNEventReport() [2/2]**

```
void gdcm::network::ULConnectionManager::SendNEventReport (
    const BaseQuery * inQuery,
    ULConnectionCallback * inCallback )
```

**10.350.3.20 SendNGet() [1/2]**

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendNGet (
    const BaseQuery * inQuery )
```

**10.350.3.21 SendNGet() [2/2]**

```
void gdcm::network::ULConnectionManager::SendNGet (
    const BaseQuery * inQuery,
    ULConnectionCallback * inCallback )
```

**10.350.3.22 SendNSet() [1/2]**

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendNSet (
    const BaseQuery * inQuery )
```

### 10.350.3.23 SendNSet() [2/2]

```
void gdcm::network::ULConnectionManager::SendNSet (
    const BaseQuery * inQuery,
    ULConnectionCallback * inCallback )
```

### 10.350.3.24 SendStore() [1/2]

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendStore (
    const File & file,
    std::istream * pStream = nullptr,
    std::streampos dataSetOffset = 0 )
```

### 10.350.3.25 SendStore() [2/2]

```
void gdcm::network::ULConnectionManager::SendStore (
    const File & file,
    ULConnectionCallback * inCallback,
    std::istream * pStream = nullptr,
    std::streampos dataSetOffset = 0 )
```

callback based API

## 10.350.4 Member Data Documentation

### 10.350.4.1 mConnection

`ULConnection*` gdcm::network::ULConnectionManager::mConnection [protected]

### 10.350.4.2 mSecondaryConnection

`ULConnection*` gdcm::network::ULConnectionManager::mSecondaryConnection [protected]

### 10.350.4.3 mTransitions

```
ULTransitionTable gdcmm::network::ULConnectionManager::mTransitions [protected]
```

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

- [gdcmmULConnectionManager.h](#)

## 10.351 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 \* [GetIStream](#) () const
- std::vector< [BasePDU](#) \* > const & [GetPDUs](#) () const
- void [SetEvent](#) (const [EEventID](#) &inEvent)
- void [SetPDU](#) (std::vector< [BasePDU](#) \* > const &inPDU)

### 10.351.1 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!

### 10.351.2 Constructor & Destructor Documentation



### 10.351.2.1 ULEvent() [1/2]

```
gdcmm::network::ULEvent::ULEvent (
    const EEventID & inEventID,
    std::vector< BasePDU * > inBasePDU,
    std::istream * iStream = nullptr,
    std::streampos posDataSet = 0 ) [inline]
```

### 10.351.2.2 ULEvent() [2/2]

```
gdcmm::network::ULEvent::ULEvent (
    const EEventID & inEventID,
    BasePDU * inBasePDU,
    std::istream * iStream = nullptr,
    std::streampos posDataSet = 0 ) [inline]
```

### 10.351.2.3 ~ULEvent()

```
gdcmm::network::ULEvent::~~ULEvent ( ) [inline]
```

## 10.351.3 Member Function Documentation

### 10.351.3.1 GetDataSetPos()

```
std::streampos gdcmm::network::ULEvent::GetDataSetPos ( ) const [inline]
```

### 10.351.3.2 GetEvent()

```
EEventID gdcmm::network::ULEvent::GetEvent ( ) const [inline]
```

### 10.351.3.3 GetIStream()

```
std::istream* gdcmm::network::ULEvent::GetIStream ( ) const [inline]
```

### 10.351.3.4 GetPDUs()

```
std::vector<BasePDU*> const& gdcmm::network::ULEvent::GetPDUs ( ) const [inline]
```

### 10.351.3.5 SetEvent()

```
void gdcmm::network::ULEvent::SetEvent (
    const EEventID & inEvent ) [inline]
```

### 10.351.3.6 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](#)

## 10.352 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

### 10.352.1 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.

## 10.352.2 Constructor & Destructor Documentation

### 10.352.2.1 ULTransitionTable()

```
gdcm::network::ULTransitionTable::ULTransitionTable ( )
```

## 10.352.3 Member Function Documentation

### 10.352.3.1 HandleEvent()

```
void gdcm::network::ULTransitionTable::HandleEvent (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) const
```

### 10.352.3.2 PrintTable()

```
void gdcm::network::ULTransitionTable::PrintTable ( ) const
```

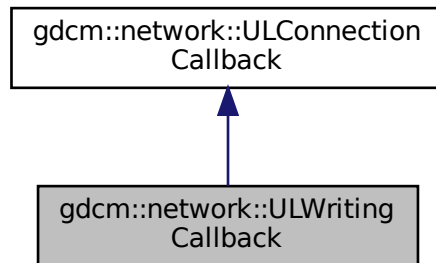
The documentation for this class was generated from the following file:

- [gdcmULTransitionTable.h](#)

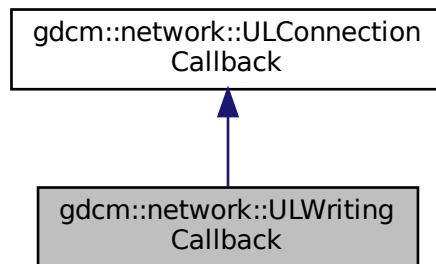
## 10.353 gdcm::network::ULWritingCallback Class Reference

```
#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.*

## Additional Inherited Members

### 10.353.1 Constructor & Destructor Documentation

#### 10.353.1.1 ULWritingCallback()

```
gdcm::network::ULWritingCallback::ULWritingCallback ( ) [default]
```

#### 10.353.1.2 ~ULWritingCallback()

```
gdcm::network::ULWritingCallback::~~ULWritingCallback ( ) [override], [default]
```

### 10.353.2 Member Function Documentation

#### 10.353.2.1 HandleDataSet()

```
void gdcm::network::ULWritingCallback::HandleDataSet (
    const DataSet & inDataSet ) [override], [virtual]
```

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

#### 10.353.2.2 HandleResponse()

```
void gdcm::network::ULWritingCallback::HandleResponse (
    const DataSet & inDataSet ) [override], [virtual]
```

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

### 10.353.2.3 SetDirectory()

```
void gdcmm::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:

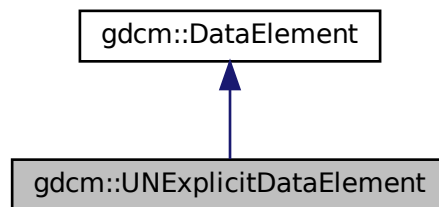
- [gdcmmULWritingCallback.h](#)

## 10.354 gdcmm::UNExplicitDataElement Class Reference

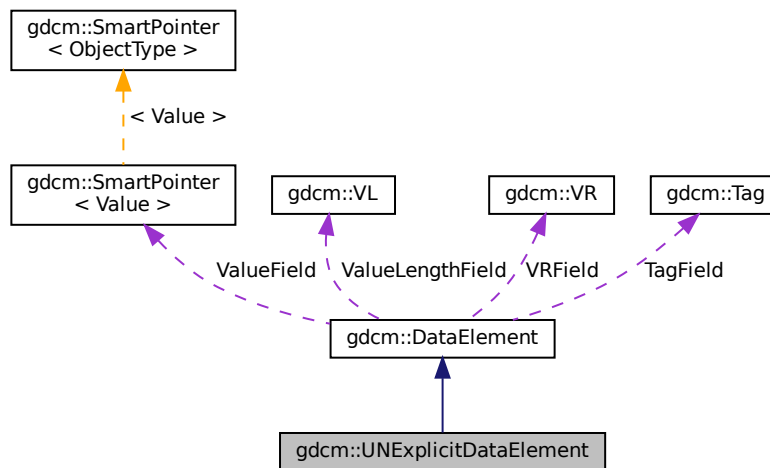
Class to read/write a [DataElement](#) as UNExplicit Data [Element](#).

```
#include <gdcmmUNExplicitDataElement.h>
```

Inheritance diagram for gdcmm::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)

## Additional Inherited Members

### 10.354.1 Detailed Description

Class to read/write a [DataElement](#) as UNExplicit Data [Element](#).

Note

bla

### 10.354.2 Member Function Documentation

### 10.354.2.1 GetLength()

```
VL gdcM::UNExplicitDataElement::GetLength ( ) const
```

### 10.354.2.2 Read()

```
template<typename TSwap >  
std::istream& gdcM::UNExplicitDataElement::Read (   
    std::istream & is )
```

### 10.354.2.3 ReadPreValue()

```
template<typename TSwap >  
std::istream& gdcM::UNExplicitDataElement::ReadPreValue (   
    std::istream & is )
```

### 10.354.2.4 ReadValue()

```
template<typename TSwap >  
std::istream& gdcM::UNExplicitDataElement::ReadValue (   
    std::istream & is,   
    bool readvalues = true )
```

### 10.354.2.5 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](#)

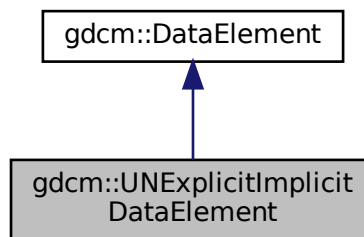


## 10.355 gdcm::UNExplicitImplicitDataElement Class Reference

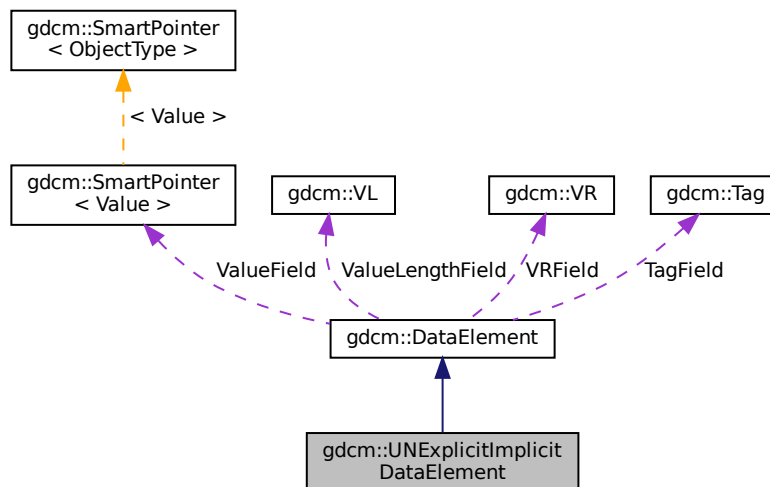
Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#).

```
#include <gdcmUNExplicitImplicitDataElement.h>
```

Inheritance diagram for gdcm::UNExplicitImplicitDataElement:



Collaboration diagram for gdcm::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)`

## Additional Inherited Members

### 10.355.1 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`

### 10.355.2 Member Function Documentation

#### 10.355.2.1 GetLength()

```
VL gdcM::UNExplicitImplicitDataElement::GetLength ( ) const
```

#### 10.355.2.2 Read()

```
template<typename TSwap >
std::istream& gdcM::UNExplicitImplicitDataElement::Read (
    std::istream & is )
```

#### 10.355.2.3 ReadPreValue()

```
template<typename TSwap >
std::istream& gdcM::UNExplicitImplicitDataElement::ReadPreValue (
    std::istream & is )
```

#### 10.355.2.4 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](#)

## 10.356 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)

#### 10.356.1 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](#)

#### 10.356.2 Member Function Documentation

### 10.356.2.1 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  $(n / 2) * 3$

### 10.356.2.2 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](#)

## 10.357 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)

## 10.357.1 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.

## 10.357.2 Member Enumeration Documentation

### 10.357.2.1 UsageType

```
enum gdcmm::Usage::UsageType
```

#### Enumerator

Mandatory	
Conditional	
UserOption	
Invalid	

### 10.357.3 Constructor & Destructor Documentation

#### 10.357.3.1 Usage()

```
gdcm::Usage::Usage (
    UsageType type = Invalid ) [inline]
```

### 10.357.4 Member Function Documentation

#### 10.357.4.1 GetUsageString()

```
static const char* gdcm::Usage::GetUsageString (
    UsageType type ) [static]
```

#### 10.357.4.2 GetUsageType()

```
static UsageType gdcm::Usage::GetUsageType (
    const char * type ) [static]
```

#### 10.357.4.3 operator UsageType()

```
gdcm::Usage::operator UsageType ( ) const [inline]
```

### 10.357.5 Friends And Related Function Documentation

#### 10.357.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const Usage & vr ) [friend]
```

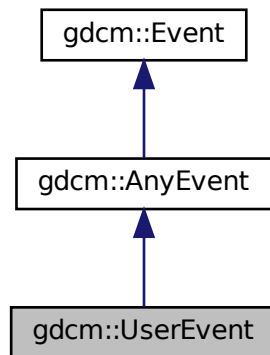
The documentation for this class was generated from the following file:

- [gdcmUsage.h](#)

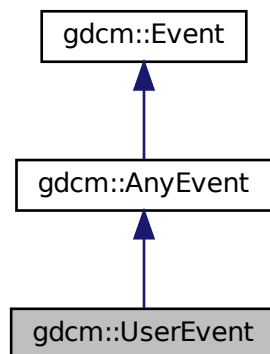
## 10.358 gdcm::UserEvent Class Reference

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

Inheritance diagram for gdcm::UserEvent:



Collaboration diagram for gdcm::UserEvent:



### Additional Inherited Members

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

- [gdcmEvent.h](#)

## 10.359 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

### 10.359.1 Detailed Description

[UserInformation.](#)

[Table 9-16](#) USER INFORMATION ITEM FIELDS

TODO what is the goal of :

[Table 9-20](#) USER INFORMATION ITEM FIELDS

### 10.359.2 Constructor & Destructor Documentation

#### 10.359.2.1 [UserInformation\(\)](#) [1/2]

```
gdcm::network::UserInformation::UserInformation ( )
```

#### 10.359.2.2 [~UserInformation\(\)](#)

```
gdcm::network::UserInformation::~~UserInformation ( )
```



### 10.359.2.3 UserInfo() [2/2]

```
gdcm::network::UserInfo::UserInfo (
    const UserInfo & ) [delete]
```

## 10.359.3 Member Function Documentation

### 10.359.3.1 AddRoleSelectionSub()

```
void gdcm::network::UserInfo::AddRoleSelectionSub (
    RoleSelectionSub const & r )
```

### 10.359.3.2 AddSOPClassExtendedNegociationSub()

```
void gdcm::network::UserInfo::AddSOPClassExtendedNegociationSub (
    SOPClassExtendedNegociationSub const & s )
```

### 10.359.3.3 GetMaximumLengthSub() [1/2]

```
MaximumLengthSub& gdcm::network::UserInfo::GetMaximumLengthSub ( ) [inline]
```

### 10.359.3.4 GetMaximumLengthSub() [2/2]

```
const MaximumLengthSub& gdcm::network::UserInfo::GetMaximumLengthSub ( ) const [inline]
```

### 10.359.3.5 operator=()

```
UserInfo& gdcm::network::UserInfo::operator= (
    const UserInfo & )
```

#### 10.359.3.6 Print()

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

#### 10.359.3.7 Read()

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

#### 10.359.3.8 Size()

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

#### 10.359.3.9 Write()

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

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

- [gdcmUserInformation.h](#)

## 10.360 gdcm::UUIDGenerator Class Reference

Class for generating unique UUID.

```
#include <gdcmUUIDGenerator.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.*

### 10.360.1 Detailed Description

Class for generating unique UUID.

generate DCE 1.1 uid

### 10.360.2 Member Function Documentation

#### 10.360.2.1 Generate()

```
const char* gdcm::UUIDGenerator::Generate ( )
```

Return the generated uuid NOT THREAD SAFE

#### 10.360.2.2 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:

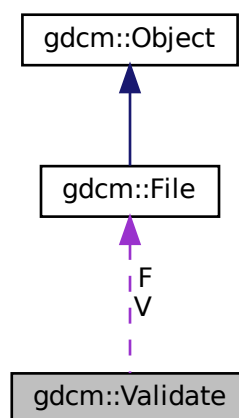
- [gdcmUUIDGenerator.h](#)

## 10.361 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](#)

### 10.361.1 Detailed Description

[Validate](#) class.

### 10.361.2 Constructor & Destructor Documentation

#### 10.361.2.1 [Validate](#)()

```
gdcmm::Validate::Validate ( )
```

#### 10.361.2.2 [~Validate](#)()

```
gdcmm::Validate::~~Validate ( )
```

### 10.361.3 Member Function Documentation

#### 10.361.3.1 [GetValidatedFile](#)()

```
const File& gdcmm::Validate::GetValidatedFile ( ) [inline]
```

### 10.361.3.2 SetFile()

```
void gdcm::Validate::SetFile (
    File const & f ) [inline]
```

### 10.361.3.3 Validation()

```
void gdcm::Validate::Validation ( )
```

## 10.361.4 Member Data Documentation

### 10.361.4.1 F

```
const File* gdcm::Validate::F [protected]
```

### 10.361.4.2 V

```
File gdcm::Validate::V [protected]
```

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

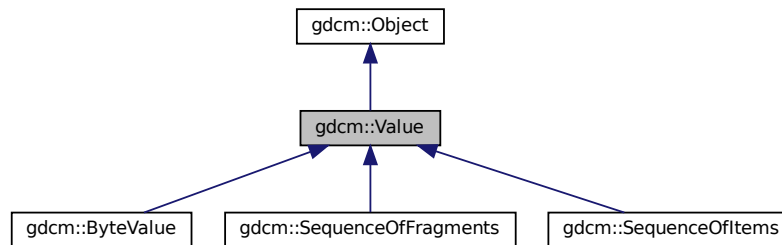
- [gdcmValidate.h](#)

## 10.362 gdcm::Value Class Reference

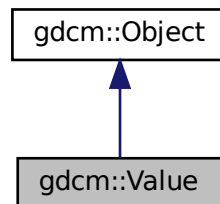
Class to represent the value of a Data [Element](#).

```
#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

### Protected Member Functions

- virtual void `SetLengthOnly` (`VL` l)

## Friends

- class [DataElement](#)

### 10.362.1 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.

### 10.362.2 Constructor & Destructor Documentation

#### 10.362.2.1 Value()

```
gdcm::Value::Value ( ) [default]
```

#### 10.362.2.2 ~Value()

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

### 10.362.3 Member Function Documentation

#### 10.362.3.1 Clear()

```
virtual void gdcm::Value::Clear ( ) [pure virtual]
```

Implemented in [gdcm::SequenceOfItems](#), [gdcm::SequenceOfFragments](#), and [gdcm::ByteValue](#).

### 10.362.3.2 GetLength()

```
virtual VL gdcM::Value::GetLength ( ) const [pure virtual]
```

Implemented in [gdcM::SequenceOfItems](#), [gdcM::SequenceOfFragments](#), and [gdcM::ByteValue](#).

Referenced by [gdcM::DataSet::InsertDataElement\(\)](#), and [gdcM::DataElement::SetValue\(\)](#).

### 10.362.3.3 operator==( )

```
virtual bool gdcM::Value::operator== (
    const Value & val ) const [pure virtual]
```

Implemented in [gdcM::SequenceOfItems](#), [gdcM::SequenceOfFragments](#), and [gdcM::ByteValue](#).

### 10.362.3.4 SetLength()

```
virtual void gdcM::Value::SetLength (
    VL l ) [pure virtual]
```

Implemented in [gdcM::ByteValue](#), [gdcM::SequenceOfItems](#), and [gdcM::SequenceOfFragments](#).

### 10.362.3.5 SetLengthOnly()

```
virtual void gdcM::Value::SetLengthOnly (
    VL l ) [protected], [virtual]
```

Reimplemented in [gdcM::ByteValue](#).

## 10.362.4 Friends And Related Function Documentation

### 10.362.4.1 DataElement

```
friend class DataElement [friend]
```

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

- [gdcMValue.h](#)



## 10.363 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)

### 10.363.1 Detailed Description

```
template<typename TDE, typename TSwap, typename TType = uint8_t>  
class gdcm::ValueIO< TDE, TSwap, TType >
```

Class to dispatch template calls.

### 10.363.2 Member Function Documentation

#### 10.363.2.1 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]
```

#### 10.363.2.2 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](#)

## 10.364 gdcM::MrProtocol::Vector3 Struct Reference

```
#include <gdcMMrProtocol.h>
```

### Public Attributes

- double [dCor](#)
- double [dSag](#)
- double [dTra](#)

### 10.364.1 Member Data Documentation

#### 10.364.1.1 dCor

```
double gdcM::MrProtocol::Vector3::dCor
```

#### 10.364.1.2 dSag

```
double gdcM::MrProtocol::Vector3::dSag
```

#### 10.364.1.3 dTra

```
double gdcM::MrProtocol::Vector3::dTra
```

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

- [gdcMMrProtocol.h](#)

## 10.365 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)

### 10.365.1 Detailed Description

major/minor and build version

### 10.365.2 Constructor & Destructor Documentation

#### 10.365.2.1 Version()

```
gdcm::Version::Version ( ) [default]
```

#### 10.365.2.2 ~Version()

```
gdcm::Version::~~Version ( ) [default]
```

### 10.365.3 Member Function Documentation

#### 10.365.3.1 GetBuildVersion()

```
static int gdcm::Version::GetBuildVersion ( ) [static]
```

#### 10.365.3.2 GetMajorVersion()

```
static int gdcm::Version::GetMajorVersion ( ) [static]
```

#### 10.365.3.3 GetMinorVersion()

```
static int gdcm::Version::GetMinorVersion ( ) [static]
```

#### 10.365.3.4 GetVersion()

```
static const char* gdcm::Version::GetVersion ( ) [static]
```

#### 10.365.3.5 Print()

```
void gdcm::Version::Print (
    std::ostream & os = std::cout ) const
```

### 10.365.4 Friends And Related Function Documentation

#### 10.365.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const Version & v ) [friend]
```

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

- [gdcmVersion.h](#)

## 10.366 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)

### 10.366.1 Detailed Description

[Value](#) Length.

#### Warning

this is a 4bytes value ! Do not try to use it for 2bytes value length

#### Examples

[ReadAndDumpDICOMDIR2.cxx](#), and [rle2img.cxx](#).

### 10.366.2 Member Typedef Documentation

#### 10.366.2.1 Type

```
typedef uint32_t gdcm::VL::Type
```

### 10.366.3 Constructor & Destructor Documentation

#### 10.366.3.1 VL()

```
gdcm::VL::VL (
    uint32_t vl = 0 ) [inline]
```

### 10.366.4 Member Function Documentation

#### 10.366.4.1 GetLength()

```
VL gdcm::VL::GetLength ( ) const [inline]
```

#### Examples

[ReadAndDumpDICOMDIR2.cxx](#).

Referenced by `gdcm::FileMetaInformation::GetFullLength()`, `gdcm::DataSet::GetLength()`, and `gdcm::Item::Write()`.

#### 10.366.4.2 GetVL16Max()

```
static uint16_t gdcm::VL::GetVL16Max ( ) [inline], [static]
```

#### 10.366.4.3 GetVL32Max()

```
static uint32_t gdcm::VL::GetVL32Max ( ) [inline], [static]
```

#### 10.366.4.4 IsOdd()

```
bool gdcm::VL::IsOdd ( ) const [inline]
```

Return whether or not the [VL](#) is odd or not.

#### 10.366.4.5 IsUndefined()

```
bool gdcm::VL::IsUndefined ( ) const [inline]
```

#### 10.366.4.6 operator uint32\_t()

```
gdcm::VL::operator uint32_t ( ) const [inline]
```

#### 10.366.4.7 operator++() [1/2]

```
VL& gdcm::VL::operator++ ( ) [inline]
```

#### 10.366.4.8 operator++() [2/2]

```
VL gdcm::VL::operator++ (
    int ) [inline]
```

#### 10.366.4.9 operator+=()

```
VL& gdcmm::VL::operator+= (
    VL const & vl ) [inline]
```

+= operator

#### 10.366.4.10 Read()

```
template<typename TSwap >
std::istream& gdcmm::VL::Read (
    std::istream & is ) [inline]
```

#### 10.366.4.11 Read16()

```
template<typename TSwap >
std::istream& gdcmm::VL::Read16 (
    std::istream & is ) [inline]
```

#### 10.366.4.12 SetToUndefined()

```
void gdcmm::VL::SetToUndefined ( ) [inline]
```

#### 10.366.4.13 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()`.

#### 10.366.4.14 Write16()

```
template<typename TSwap >
const std::ostream& gdcmm::VL::Write16 (
    std::ostream & os ) const [inline]
```



## 10.366.5 Friends And Related Function Documentation

### 10.366.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const VL & vl ) [friend]
```

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

- [gdcmVL.h](#)

## 10.367 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)

## 10.367.1 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

## 10.367.2 Member Enumeration Documentation

### 10.367.2.1 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	
VM2_n	
VM3_4	
VM3_3n	
VM3_n	
VM4_4n	
VM6_6n	
VM6_n	
VM7_7n	
VM30_30n	
VM47_47n	
VM_END	

### 10.367.3 Constructor & Destructor Documentation

#### 10.367.3.1 VM()

```
gdcmm::VM::VM (
    VMType type = VM0 ) [inline]
```

### 10.367.4 Member Function Documentation

#### 10.367.4.1 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.

#### 10.367.4.2 GetIndex()

```
static unsigned int gdcmm::VM::GetIndex (
    VMType vm ) [static], [protected]
```

#### 10.367.4.3 GetLength()

```
unsigned int gdcmm::VM::GetLength ( ) const
```

#### 10.367.4.4 GetNumberOfElementsFromArray()

```
static size_t gdcmm::VM::GetNumberOfElementsFromArray (
    const char * array,
    size_t length ) [static]
```

#### 10.367.4.5 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

#### 10.367.4.6 GetVMType()

```
static VMType gdcm::VM::GetVMType (
    const char * vm ) [static]
```

#### 10.367.4.7 GetVMTypeFromLength()

```
static VMType gdcm::VM::GetVMTypeFromLength (
    size_t length,
    unsigned int size ) [static]
```

#### 10.367.4.8 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

#### 10.367.4.9 operator VMType()

```
gdcm::VM::operator VMType ( ) const [inline]
```

### 10.367.5 Friends And Related Function Documentation

#### 10.367.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const VM & vm ) [friend]
```

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

- [gdcmVM.h](#)

## 10.368 gdcm::VMToLength< T > Struct Template Reference

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

- [gdcmVM.h](#)

## 10.369 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)

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

### 10.369.2 Member Enumeration Documentation

#### 10.369.2.1 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	



## Enumerator

SL	
SQ	
SS	
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

[NewSequence.cs](#).

## 10.369.3 Constructor & Destructor Documentation

### 10.369.3.1 VR()

```
gdcm::VR::VR (
    VRType vr = INVALID ) [inline]
```

## 10.369.4 Member Function Documentation

#### 10.369.4.1 CanDisplay()

```
static bool gdcm::VR::CanDisplay (
    VRType vr ) [static]
```

#### 10.369.4.2 Compatible()

```
bool gdcm::VR::Compatible (
    VR const & vr ) const
```

#### 10.369.4.3 GetLength() [1/2]

```
int gdcm::VR::GetLength ( ) const [inline]
```

#### 10.369.4.4 GetLength() [2/2]

```
static uint32_t gdcm::VR::GetLength (
    VRType vr ) [inline], [static]
```

#### 10.369.4.5 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.

#### 10.369.4.6 GetSizeof()

```
unsigned int gdcm::VR::GetSizeof ( ) const
```

#### 10.369.4.7 GetVRString()

```
static const char* gdcm::VR::GetVRString (
    VRType vr ) [static]
```

#### 10.369.4.8 GetVRStringFromFile()

```
static const char* gdcm::VR::GetVRStringFromFile (
    VRType vr ) [static]
```

#### 10.369.4.9 GetVRType()

```
static VRType gdcm::VR::GetVRType (
    const char * vr ) [static]
```

#### 10.369.4.10 GetVRTypeFromFile()

```
static VRType gdcm::VR::GetVRTypeFromFile (
    const char * vr ) [static]
```

#### 10.369.4.11 IsASCII()

```
static bool gdcm::VR::IsASCII (
    VRType vr ) [static]
```

#### 10.369.4.12 IsASCII2()

```
static bool gdcm::VR::IsASCII2 (
    VRType vr ) [static]
```

**10.369.4.13 IsBinary()**

```
static bool gdcm::VR::IsBinary (
    VRType vr ) [static]
```

**10.369.4.14 IsBinary2()**

```
static bool gdcm::VR::IsBinary2 (
    VRType vr ) [static]
```

**10.369.4.15 IsDual()**

```
bool gdcm::VR::IsDual ( ) const
```

**10.369.4.16 IsSwap()**

```
static bool gdcm::VR::IsSwap (
    const char * vr ) [static]
```

**10.369.4.17 IsValid() [1/2]**

```
static bool gdcm::VR::IsValid (
    const char * vr ) [static]
```

**10.369.4.18 IsValid() [2/2]**

```
static bool gdcm::VR::IsValid (
    const char * vr1,
    VRType vr2 ) [static]
```

#### 10.369.4.19 IsVRFile()

```
bool gdcm::VR::IsVRFile ( ) const
```

Referenced by `gdcm::DataElement::SetVR()`.

#### 10.369.4.20 operator VRType()

```
gdcm::VR::operator VRType ( ) const [inline]
```

#### 10.369.4.21 Read()

```
std::istream& gdcm::VR::Read (
    std::istream & is ) [inline]
```

References `gdcmDebugMacro`, `INVALID`, and `VR_END`.

#### 10.369.4.22 Write()

```
const std::ostream& gdcm::VR::Write (
    std::ostream & os ) const [inline]
```

References `gdcmAssertAlwaysMacro`, and `INVALID`.

### 10.369.5 Friends And Related Function Documentation

#### 10.369.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const VR & vr ) [friend]
```

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

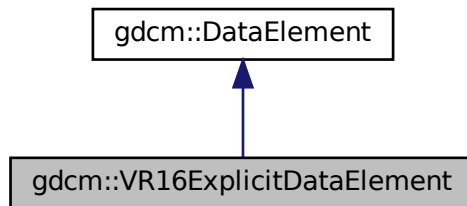
- [gdcmVR.h](#)

## 10.370 gdcm::VR16ExplicitDataElement Class Reference

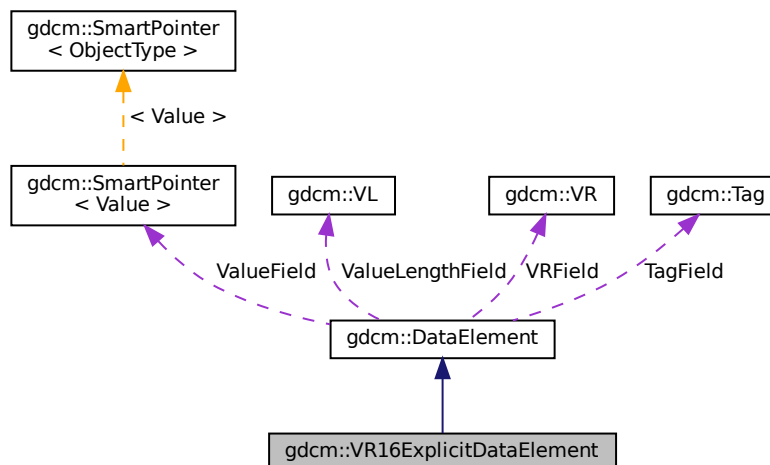
Class to read/write a [DataElement](#) as Explicit Data [Element](#).

```
#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)

## Additional Inherited Members

### 10.370.1 Detailed Description

Class to read/write a [DataElement](#) as Explicit Data [Element](#).

#### Note

This class support 16 bits when finding an unkown [VR](#): For instance: Siemens\_CT\_Sensation64\_has\_VR\_RT.dcm

### 10.370.2 Member Function Documentation

#### 10.370.2.1 GetLength()

```
VL gdcm::VR16ExplicitDataElement::GetLength ( ) const
```

#### 10.370.2.2 Read()

```
template<typename TSwap >  
std::istream& gdcm::VR16ExplicitDataElement::Read (  
    std::istream & is )
```

#### 10.370.2.3 ReadPreValue()

```
template<typename TSwap >  
std::istream& gdcm::VR16ExplicitDataElement::ReadPreValue (  
    std::istream & is )
```

#### 10.370.2.4 ReadValue()

```
template<typename TSwap >
std::istream& gdc::VR16ExplicitDataElement::ReadValue (
    std::istream & is,
    bool readvalues = true )
```

#### 10.370.2.5 ReadWithLength()

```
template<typename TSwap >
std::istream& gdc::VR16ExplicitDataElement::ReadWithLength (
    std::istream & is,
    VL & length )
```

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

- [gdcVR16ExplicitDataElement.h](#)

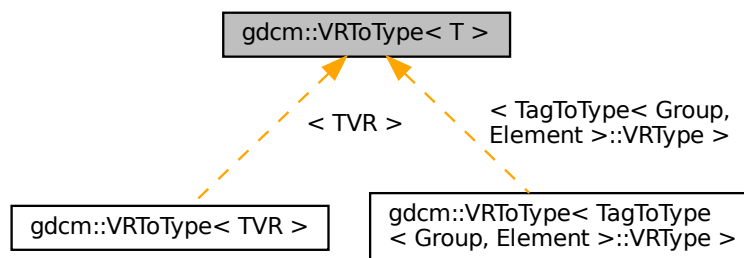
### 10.371 gdc::VRToEncoding< T > Struct Template Reference

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

- [gdcVR.h](#)

### 10.372 gdc::VRToType< T > Struct Template Reference

Inheritance diagram for gdc::VRToType< T >:



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

- [gdcVR.h](#)



## 10.373 gdcm::VRVLSIZE< T > Class Template Reference

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

- [gdcmAttribute.h](#)

## 10.374 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)

### 10.374.1 Member Function Documentation

#### 10.374.1.1 Read()

```
static uint16_t gdcm::VRVLSIZE< 0 >::Read (  
    std::istream & _is ) [inline], [static]
```

#### 10.374.1.2 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](#)

## 10.375 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)

### 10.375.1 Member Function Documentation

#### 10.375.1.1 Read()

```
static uint32_t gdcM::VRVLSize< 1 >::Read (  
    std::istream & _is )    [inline], [static]
```

#### 10.375.1.2 Write()

```
static void gdcM::VRVLSize< 1 >::Write (  
    std::ostream & os )    [inline], [static]
```

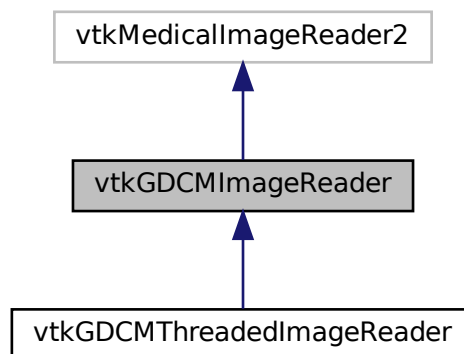
The documentation for this class was generated from the following file:

- [gdcMAttribute.h](#)

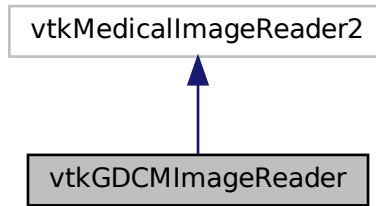
## 10.376 vtkGDCMImageReader Class Reference

```
#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 [gdcml::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](#)

## 10.376.1 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](#), [MetaImageMD5Activiz.cs](#), [ReadSeriesIntoVTK.java](#), [RefCounting.cs](#), [gdcmmorthoplanes.cxx](#), [gdcmmreslice.cxx](#), [gdcmmtexture.cxx](#), [gdcmmvolume.cxx](#), [offscreenimage.cxx](#), and [reslicesphere.cxx](#).

## 10.376.2 Constructor & Destructor Documentation

### 10.376.2.1 vtkGDCMImageReader()

```
vtkGDCMImageReader::vtkGDCMImageReader ( ) [protected]
```

### Examples

[HelloActiviz2.cs](#).

### 10.376.2.2 ~vtkGDCMImageReader()

```
vtkGDCMImageReader::~~vtkGDCMImageReader ( ) [protected]
```

## 10.376.3 Member Function Documentation

### 10.376.3.1 CanReadFile()

```
virtual int vtkGDCMImageReader::CanReadFile (
    const char * fname ) [virtual]
```

### Examples

[AWTMedical3.java](#), and [MetaImageMD5Activiz.cs](#).

### 10.376.3.2 ExecuteData()

```
void vtkGDCMImageReader::ExecuteData (
    vtkDataObject * out ) [protected]
```

### 10.376.3.3 ExecuteInformation()

```
void vtkGDCMImageReader::ExecuteInformation ( ) [protected]
```

### 10.376.3.4 FillMedicalImageInformation()

```
void vtkGDCMImageReader::FillMedicalImageInformation (
    const gdcm::ImageReader & reader ) [protected]
```

### 10.376.3.5 GetDescriptiveName()

```
virtual const char* vtkGDCMImageReader::GetDescriptiveName ( ) [inline], [virtual]
```

### 10.376.3.6 GetFileExtensions()

```
virtual const char* vtkGDCMImageReader::GetFileExtensions ( ) [inline], [virtual]
```

### 10.376.3.7 GetIconImage()

```
vtkImageData* vtkGDCMImageReader::GetIconImage ( )
```

### 10.376.3.8 GetOverlay()

```
vtkImageData* vtkGDCMImageReader::GetOverlay (
    int i )
```

### 10.376.3.9 LoadSingleFile()

```
int vtkGDCMImageReader::LoadSingleFile (
    const char * filename,
    char * pointer,
    unsigned long & outlen ) [protected]
```

### 10.376.3.10 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](#), [gdcmmorthoplanes.cxx](#), [gdcmreslice.cxx](#), [gdcmttexture.cxx](#), [gdcmvolume.cxx](#), [offscreenimage.cxx](#), and [reslicesphere.cxx](#).

### 10.376.3.11 PrintSelf()

```
virtual void vtkGDCMImageReader::PrintSelf (
    ostream & os,
    vtkIndent indent ) [virtual]
```

Reimplemented in [vtkGDCMThreadedImageReader](#).

### 10.376.3.12 RequestDataCompat()

```
int vtkGDCMImageReader::RequestDataCompat ( ) [protected]
```

### 10.376.3.13 RequestInformationCompat()

```
int vtkGDCMImageReader::RequestInformationCompat ( ) [protected]
```

**10.376.3.14 SetCurve()**

```
virtual void vtkGDCMImageReader::SetCurve (
    vtkPolyData * pd ) [virtual]
```

**10.376.3.15 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 [gdcmmorthoplanes.cxx](#).

**10.376.3.16 SetFilePattern()**

```
void vtkGDCMImageReader::SetFilePattern (
    const char * ) [inline], [protected]
```

**10.376.3.17 SetFilePrefix()**

```
void vtkGDCMImageReader::SetFilePrefix (
    const char * ) [inline], [protected]
```

**10.376.3.18 SetMedicalImageProperties()**

```
virtual void vtkGDCMImageReader::SetMedicalImageProperties (
    vtkMedicalImageProperties * pd ) [virtual]
```

**10.376.3.19 vtkBooleanMacro()** [1/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
    ApplyLookupTable ,
    int )
```



**10.376.3.20 vtkBooleanMacro()** [2/5]

```
int vtkGDCMImageReader::vtkBooleanMacro (
    ApplyYBRToRGB ,
    int )
```

**10.376.3.21 vtkBooleanMacro()** [3/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
    LoadIconImage ,
    int )
```

**10.376.3.22 vtkBooleanMacro()** [4/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
    LoadOverlays ,
    int )
```

**10.376.3.23 vtkBooleanMacro()** [5/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
    LossyFlag ,
    int )
```

**10.376.3.24 vtkGetMacro()** [1/11]

```
vtkGDCMImageReader::vtkGetMacro (
    ApplyLookupTable ,
    int )
```

**10.376.3.25 vtkGetMacro()** [2/11]

```
vtkGDCMImageReader::vtkGetMacro (
    ApplyYBRToRGB ,
    int )
```

**10.376.3.26 vtkGetMacro()** [3/11]

```
vtkGDCMImageReader::vtkGetMacro (
    ImageFormat ,
    int )
```

**10.376.3.27 vtkGetMacro()** [4/11]

```
vtkGDCMImageReader::vtkGetMacro (
    LoadIconImage ,
    int )
```

**10.376.3.28 vtkGetMacro()** [5/11]

```
vtkGDCMImageReader::vtkGetMacro (
    LoadOverlays ,
    int )
```

**10.376.3.29 vtkGetMacro()** [6/11]

```
vtkGDCMImageReader::vtkGetMacro (
    LossyFlag ,
    int )
```

**10.376.3.30 vtkGetMacro()** [7/11]

```
vtkGDCMImageReader::vtkGetMacro (
    NumberOfIconImages ,
    int )
```

**10.376.3.31 vtkGetMacro()** [8/11]

```
vtkGDCMImageReader::vtkGetMacro (
    NumberOfOverlays ,
    int )
```

**10.376.3.32 vtkGetMacro()** [9/11]

```
vtkGDCMImageReader::vtkGetMacro (
    PlanarConfiguration ,
    int )
```

**10.376.3.33 vtkGetMacro()** [10/11]

```
vtkGDCMImageReader::vtkGetMacro (
    Scale ,
    double )
```

**10.376.3.34 vtkGetMacro()** [11/11]

```
vtkGDCMImageReader::vtkGetMacro (
    Shift ,
    double )
```

**10.376.3.35 vtkGetObjectMacro()** [1/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
    Curve ,
    vtkPolyData )
```

**10.376.3.36 vtkGetObjectMacro()** [2/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
    DirectionCosines ,
    vtkMatrix4x4 )
```

**10.376.3.37 vtkGetObjectMacro()** [3/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
    FileNames ,
    vtkStringArray )
```

**10.376.3.38 vtkGetObjectMacro() [4/4]**

```
vtkGDCMImageReader::vtkGetObjectMacro (
    MedicalImageProperties ,
    vtkMedicalImageProperties )
```

**10.376.3.39 vtkGetStringMacro() [1/2]**

```
vtkGDCMImageReader::vtkGetStringMacro (
    FilePattern ) [protected]
```

**10.376.3.40 vtkGetStringMacro() [2/2]**

```
vtkGDCMImageReader::vtkGetStringMacro (
    FilePrefix ) [protected]
```

**10.376.3.41 vtkGetVector3Macro()**

```
vtkGDCMImageReader::vtkGetVector3Macro (
    ImagePositionPatient ,
    double )
```

**10.376.3.42 vtkGetVector6Macro()**

```
vtkGDCMImageReader::vtkGetVector6Macro (
    ImageOrientationPatient ,
    double )
```

**10.376.3.43 vtkSetMacro() [1/4]**

```
vtkGDCMImageReader::vtkSetMacro (
    ApplyLookupTable ,
    int )
```

**10.376.3.44 vtkSetMacro() [2/4]**

```
vtkGDCMImageReader::vtkSetMacro (
    LoadIconImage ,
    int )
```

**10.376.3.45 vtkSetMacro() [3/4]**

```
vtkGDCMImageReader::vtkSetMacro (
    LoadOverlays ,
    int )
```

**10.376.3.46 vtkSetMacro() [4/4]**

```
vtkGDCMImageReader::vtkSetMacro (
    LossyFlag ,
    int )
```

**10.376.3.47 vtkSetVector6Macro()**

```
vtkGDCMImageReader::vtkSetVector6Macro (
    ImageOrientationPatient ,
    double ) [protected]
```

**10.376.3.48 vtkTypeMacro()**

```
vtkGDCMImageReader::vtkTypeMacro (
    vtkGDCMImageReader ,
    vtkMedicalImageReader2 )
```

**10.376.4 Member Data Documentation**

#### 10.376.4.1 ApplyInverseVideo

```
int vtkGDCMImageReader::ApplyInverseVideo [protected]
```

#### 10.376.4.2 ApplyLookupTable

```
int vtkGDCMImageReader::ApplyLookupTable [protected]
```

#### 10.376.4.3 ApplyPlanarConfiguration

```
int vtkGDCMImageReader::ApplyPlanarConfiguration [protected]
```

#### 10.376.4.4 ApplyShiftScale

```
int vtkGDCMImageReader::ApplyShiftScale [protected]
```

#### 10.376.4.5 ApplyYBRToRGB

```
int vtkGDCMImageReader::ApplyYBRToRGB [protected]
```

#### 10.376.4.6 Curve

```
vtkPolyData* vtkGDCMImageReader::Curve [protected]
```

#### 10.376.4.7 DirectionCosines

```
vtkMatrix4x4* vtkGDCMImageReader::DirectionCosines [protected]
```

#### 10.376.4.8 FileNames

```
vtkStringArray* vtkGDCMImageReader::FileNames [protected]
```

#### 10.376.4.9 ForceRescale

```
int vtkGDCMImageReader::ForceRescale [protected]
```

#### 10.376.4.10 IconDataScalarType

```
int vtkGDCMImageReader::IconDataScalarType [protected]
```

#### 10.376.4.11 IconImageDataExtent

```
int vtkGDCMImageReader::IconImageDataExtent[6] [protected]
```

#### 10.376.4.12 IconNumberOfScalarComponents

```
int vtkGDCMImageReader::IconNumberOfScalarComponents [protected]
```

#### 10.376.4.13 ImageFormat

```
int vtkGDCMImageReader::ImageFormat [protected]
```

#### 10.376.4.14 ImageOrientationPatient

```
double vtkGDCMImageReader::ImageOrientationPatient[6] [protected]
```

**10.376.4.15 ImagePositionPatient**

```
double vtkGDCMImageReader::ImagePositionPatient[3] [protected]
```

**10.376.4.16 LoadIconImage**

```
int vtkGDCMImageReader::LoadIconImage [protected]
```

**10.376.4.17 LoadOverlays**

```
int vtkGDCMImageReader::LoadOverlays [protected]
```

**10.376.4.18 LossyFlag**

```
int vtkGDCMImageReader::LossyFlag [protected]
```

**10.376.4.19 MedicalImageProperties**

```
vtkMedicalImageProperties* vtkGDCMImageReader::MedicalImageProperties [protected]
```

**10.376.4.20 NumberOfIconImages**

```
int vtkGDCMImageReader::NumberOfIconImages [protected]
```

**10.376.4.21 NumberOfOverlays**

```
int vtkGDCMImageReader::NumberOfOverlays [protected]
```



#### 10.376.4.22 PlanarConfiguration

```
int vtkGDCMImageReader::PlanarConfiguration [protected]
```

#### 10.376.4.23 Scale

```
double vtkGDCMImageReader::Scale [protected]
```

#### 10.376.4.24 Shift

```
double vtkGDCMImageReader::Shift [protected]
```

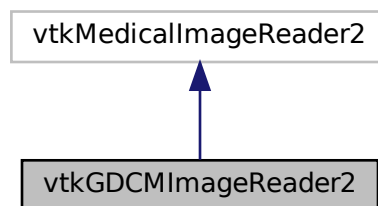
The documentation for this class was generated from the following file:

- [vtkGDCMImageReader.h](#)

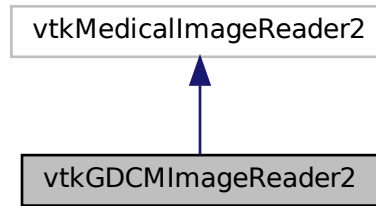
### 10.377 vtkGDCMImageReader2 Class Reference

```
#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 [ApplyYBRTToRGB](#)
- 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](#)

### 10.377.1 Detailed Description

#### Examples

[Compute3DSpacing.cxx](#).

### 10.377.2 Constructor & Destructor Documentation

#### 10.377.2.1 vtkGDCMImageReader2()

```
vtkGDCMImageReader2::vtkGDCMImageReader2 ( ) [protected]
```

#### 10.377.2.2 ~vtkGDCMImageReader2()

```
vtkGDCMImageReader2::~~vtkGDCMImageReader2 ( ) [protected]
```

### 10.377.3 Member Function Documentation

#### 10.377.3.1 CanReadFile()

```
virtual int vtkGDCMImageReader2::CanReadFile (
    const char * fname ) [virtual]
```

#### 10.377.3.2 FillMedicalImageInformation()

```
void vtkGDCMImageReader2::FillMedicalImageInformation (
    const gdcm::ImageReader & reader ) [protected]
```

#### 10.377.3.3 GetDescriptiveName()

```
virtual const char* vtkGDCMImageReader2::GetDescriptiveName ( ) [inline], [virtual]
```

#### 10.377.3.4 GetFileExtensions()

```
virtual const char* vtkGDCMImageReader2::GetFileExtensions ( ) [inline], [virtual]
```

#### 10.377.3.5 GetIconImage()

```
vtkImageData* vtkGDCMImageReader2::GetIconImage ( )
```

#### 10.377.3.6 GetIconImagePort()

```
vtkAlgorithmOutput* vtkGDCMImageReader2::GetIconImagePort ( )
```

#### 10.377.3.7 GetOverlay()

```
vtkImageData* vtkGDCMImageReader2::GetOverlay (
    int i )
```

#### 10.377.3.8 GetOverlayPort()

```
vtkAlgorithmOutput* vtkGDCMImageReader2::GetOverlayPort (
    int index )
```

#### 10.377.3.9 LoadSingleFile()

```
int vtkGDCMImageReader2::LoadSingleFile (
    const char * filename,
    char * pointer,
    unsigned long & outlen ) [protected]
```

#### 10.377.3.10 New()

```
static vtkGDCMImageReader2\* vtkGDCMImageReader2::New ( ) [static]
```

#### Examples

[Compute3DSpacing.cxx](#).

#### 10.377.3.11 PrintSelf()

```
virtual void vtkGDCMImageReader2::PrintSelf (
    ostream & os,
    vtkIndent indent ) [virtual]
```

#### 10.377.3.12 ProcessRequest()

```
int vtkGDCMImageReader2::ProcessRequest (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector ) [protected]
```

#### 10.377.3.13 RequestData()

```
int vtkGDCMImageReader2::RequestData (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector ) [protected]
```

#### 10.377.3.14 RequestDataCompat()

```
int vtkGDCMImageReader2::RequestDataCompat ( ) [protected]
```

**10.377.3.15 RequestInformation()**

```
int vtkGDCMImageReader2::RequestInformation (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector ) [protected]
```

**10.377.3.16 RequestInformationCompat()**

```
int vtkGDCMImageReader2::RequestInformationCompat ( ) [protected]
```

**10.377.3.17 SetCurve()**

```
virtual void vtkGDCMImageReader2::SetCurve (
    vtkPolyData * pd ) [virtual]
```

**10.377.3.18 SetFilePattern()**

```
void vtkGDCMImageReader2::SetFilePattern (
    const char * ) [inline], [protected]
```

**10.377.3.19 SetFilePrefix()**

```
void vtkGDCMImageReader2::SetFilePrefix (
    const char * ) [inline], [protected]
```

**10.377.3.20 SetMedicalImageProperties()**

```
virtual void vtkGDCMImageReader2::SetMedicalImageProperties (
    vtkMedicalImageProperties * pd ) [virtual]
```

**10.377.3.21 vtkBooleanMacro()** [1/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (
    ApplyLookupTable ,
    int )
```

**10.377.3.22 vtkBooleanMacro()** [2/5]

```
int vtkGDCMImageReader2::vtkBooleanMacro (
    ApplyYBRToRGB ,
    int )
```

**10.377.3.23 vtkBooleanMacro()** [3/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (
    LoadIconImage ,
    int )
```

**10.377.3.24 vtkBooleanMacro()** [4/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (
    LoadOverlays ,
    int )
```

**10.377.3.25 vtkBooleanMacro()** [5/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (
    LossyFlag ,
    int )
```

**10.377.3.26 vtkGetMacro()** [1/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    ApplyLookupTable ,
    int )
```



**10.377.3.27 vtkGetMacro()** [2/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    ApplyYBRToRGB ,
    int )
```

**10.377.3.28 vtkGetMacro()** [3/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    ImageFormat ,
    int )
```

**10.377.3.29 vtkGetMacro()** [4/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    LoadIconImage ,
    int )
```

**10.377.3.30 vtkGetMacro()** [5/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    LoadOverlays ,
    int )
```

**10.377.3.31 vtkGetMacro()** [6/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    LossyFlag ,
    int )
```

**10.377.3.32 vtkGetMacro()** [7/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    NumberOfIconImages ,
    int )
```

**10.377.3.33 vtkGetMacro()** [8/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    NumberOfOverlays ,
    int )
```

**10.377.3.34 vtkGetMacro()** [9/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    PlanarConfiguration ,
    int )
```

**10.377.3.35 vtkGetMacro()** [10/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    Scale ,
    double )
```

**10.377.3.36 vtkGetMacro()** [11/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    Shift ,
    double )
```

**10.377.3.37 vtkGetObjectMacro()** [1/2]

```
vtkGDCMImageReader2::vtkGetObjectMacro (
    Curve ,
    vtkPolyData )
```

**10.377.3.38 vtkGetObjectMacro()** [2/2]

```
vtkGDCMImageReader2::vtkGetObjectMacro (
    DirectionCosines ,
    vtkMatrix4x4 )
```

**10.377.3.39 vtkGetStringMacro() [1/2]**

```
vtkGDCMImageReader2::vtkGetStringMacro (
    FilePattern ) [protected]
```

**10.377.3.40 vtkGetStringMacro() [2/2]**

```
vtkGDCMImageReader2::vtkGetStringMacro (
    FilePrefix ) [protected]
```

**10.377.3.41 vtkGetVector3Macro()**

```
vtkGDCMImageReader2::vtkGetVector3Macro (
    ImagePositionPatient ,
    double )
```

**10.377.3.42 vtkGetVector6Macro()**

```
vtkGDCMImageReader2::vtkGetVector6Macro (
    ImageOrientationPatient ,
    double )
```

**10.377.3.43 vtkSetMacro() [1/4]**

```
vtkGDCMImageReader2::vtkSetMacro (
    ApplyLookupTable ,
    int )
```

**10.377.3.44 vtkSetMacro() [2/4]**

```
vtkGDCMImageReader2::vtkSetMacro (
    LoadIconImage ,
    int )
```

**10.377.3.45 vtkSetMacro() [3/4]**

```
vtkGDCMImageReader2::vtkSetMacro (
    LoadOverlays ,
    int )
```

**10.377.3.46 vtkSetMacro() [4/4]**

```
vtkGDCMImageReader2::vtkSetMacro (
    LossyFlag ,
    int )
```

**10.377.3.47 vtkSetVector6Macro()**

```
vtkGDCMImageReader2::vtkSetVector6Macro (
    ImageOrientationPatient ,
    double ) [protected]
```

**10.377.3.48 vtkTypeMacro()**

```
vtkGDCMImageReader2::vtkTypeMacro (
    vtkGDCMImageReader2 ,
    vtkMedicalImageReader2 )
```

**10.377.4 Member Data Documentation****10.377.4.1 ApplyInverseVideo**

```
int vtkGDCMImageReader2::ApplyInverseVideo [protected]
```

**10.377.4.2 ApplyLookupTable**

```
int vtkGDCMImageReader2::ApplyLookupTable [protected]
```

#### 10.377.4.3 ApplyPlanarConfiguration

```
int vtkGDCMImageReader2::ApplyPlanarConfiguration [protected]
```

#### 10.377.4.4 ApplyShiftScale

```
int vtkGDCMImageReader2::ApplyShiftScale [protected]
```

#### 10.377.4.5 ApplyYBRToRGB

```
int vtkGDCMImageReader2::ApplyYBRToRGB [protected]
```

#### 10.377.4.6 Curve

```
vtkPolyData* vtkGDCMImageReader2::Curve [protected]
```

#### 10.377.4.7 DirectionCosines

```
vtkMatrix4x4* vtkGDCMImageReader2::DirectionCosines [protected]
```

#### 10.377.4.8 ForceRescale

```
int vtkGDCMImageReader2::ForceRescale [protected]
```

#### 10.377.4.9 IconDataScalarType

```
int vtkGDCMImageReader2::IconDataScalarType [protected]
```

**10.377.4.10 IconImageDataExtent**

```
int vtkGDCMImageReader2::IconImageDataExtent[6] [protected]
```

**10.377.4.11 IconNumberOfScalarComponents**

```
int vtkGDCMImageReader2::IconNumberOfScalarComponents [protected]
```

**10.377.4.12 ImageFormat**

```
int vtkGDCMImageReader2::ImageFormat [protected]
```

**10.377.4.13 ImageOrientationPatient**

```
double vtkGDCMImageReader2::ImageOrientationPatient[6] [protected]
```

**10.377.4.14 ImagePositionPatient**

```
double vtkGDCMImageReader2::ImagePositionPatient[3] [protected]
```

**10.377.4.15 LoadIconImage**

```
int vtkGDCMImageReader2::LoadIconImage [protected]
```

**10.377.4.16 LoadOverlays**

```
int vtkGDCMImageReader2::LoadOverlays [protected]
```

#### 10.377.4.17 LossyFlag

```
int vtkGDCMImageReader2::LossyFlag [protected]
```

#### 10.377.4.18 NumberOfIconImages

```
int vtkGDCMImageReader2::NumberOfIconImages [protected]
```

#### 10.377.4.19 NumberOfOverlays

```
int vtkGDCMImageReader2::NumberOfOverlays [protected]
```

#### 10.377.4.20 PlanarConfiguration

```
int vtkGDCMImageReader2::PlanarConfiguration [protected]
```

#### 10.377.4.21 Scale

```
double vtkGDCMImageReader2::Scale [protected]
```

#### 10.377.4.22 Shift

```
double vtkGDCMImageReader2::Shift [protected]
```

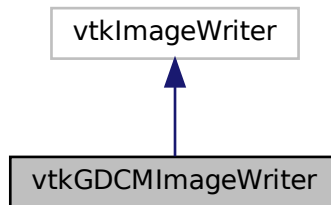
The documentation for this class was generated from the following file:

- [vtkGDCMImageReader2.h](#)

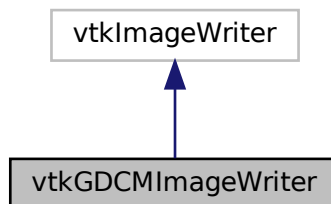
## 10.378 vtkGDCMImageWriter Class Reference

```
#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)

### 10.378.1 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 [gdcmorthoplanes.cxx](#).

### 10.378.2 Member Enumeration Documentation

#### 10.378.2.1 CompressionTypes

```
enum vtkGDCMImageWriter::CompressionTypes
```

##### Enumerator

NO_COMPRESSION	
JPEG_COMPRESSION	
JPEG2000_COMPRESSION	
JPEGLS_COMPRESSION	
RLE_COMPRESSION	

### 10.378.3 Constructor & Destructor Documentation

#### 10.378.3.1 vtkGDCMImageWriter()

```
vtkGDCMImageWriter::vtkGDCMImageWriter ( ) [protected]
```

#### 10.378.3.2 ~vtkGDCMImageWriter()

```
vtkGDCMImageWriter::~~vtkGDCMImageWriter ( ) [protected]
```

### 10.378.4 Member Function Documentation

#### 10.378.4.1 GetDescriptiveName()

```
virtual const char* vtkGDCMImageWriter::GetDescriptiveName ( ) [inline], [virtual]
```

#### 10.378.4.2 GetFileExtensions()

```
virtual const char* vtkGDCMImageWriter::GetFileExtensions ( ) [inline], [virtual]
```

#### 10.378.4.3 GetFileName()

```
virtual char* vtkGDCMImageWriter::GetFileName ( ) [protected], [virtual]
```

#### 10.378.4.4 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 [gdcmorthoplanes.cxx](#).

#### 10.378.4.5 PrintSelf()

```
virtual void vtkGDCMImageWriter::PrintSelf (
    ostream & os,
    vtkIndent indent ) [virtual]
```

#### 10.378.4.6 SetDirectionCosines()

```
virtual void vtkGDCMImageWriter::SetDirectionCosines (
    vtkMatrix4x4 * matrix ) [virtual]
```

##### Examples

[Convert16BitsTo8Bits.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [HelloActiviz2.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld.java](#), [MagnifyFile.cxx](#), and [gdcmorthoplanes.cxx](#).

#### 10.378.4.7 SetDirectionCosinesFromImageOrientationPatient()

```
virtual void vtkGDCMImageWriter::SetDirectionCosinesFromImageOrientationPatient (
    const double dircos[6] ) [virtual]
```

#### 10.378.4.8 SetFileNames()

```
virtual void vtkGDCMImageWriter::SetFileNames (
    vtkStringArray * ) [virtual]
```

##### Examples

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

#### 10.378.4.9 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 [gdcmmorthoplanes.cxx](#).

#### 10.378.4.10 vtkBooleanMacro() [1/2]

```
vtkGDCMImageWriter::vtkBooleanMacro (
    FileLowerLeft ,
    int )
```

#### 10.378.4.11 vtkBooleanMacro() [2/2]

```
vtkGDCMImageWriter::vtkBooleanMacro (
    LossyFlag ,
    int )
```

**10.378.4.12 vtkGetMacro()** [1/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    CompressionType ,
    int )
```

**10.378.4.13 vtkGetMacro()** [2/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    FileLowerLeft ,
    int )
```

**10.378.4.14 vtkGetMacro()** [3/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    ImageFormat ,
    int )
```

**10.378.4.15 vtkGetMacro()** [4/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    LossyFlag ,
    int )
```

**10.378.4.16 vtkGetMacro()** [5/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    PlanarConfiguration ,
    int )
```

**10.378.4.17 vtkGetMacro()** [6/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    Scale ,
    double )
```

**10.378.4.18 vtkGetMacro() [7/7]**

```
vtkGDCMImageWriter::vtkGetMacro (
    Shift ,
    double )
```

**10.378.4.19 vtkGetObjectMacro() [1/3]**

```
vtkGDCMImageWriter::vtkGetObjectMacro (
    DirectionCosines ,
    vtkMatrix4x4 )
```

**10.378.4.20 vtkGetObjectMacro() [2/3]**

```
vtkGDCMImageWriter::vtkGetObjectMacro (
    FileNames ,
    vtkStringArray )
```

**10.378.4.21 vtkGetObjectMacro() [3/3]**

```
vtkGDCMImageWriter::vtkGetObjectMacro (
    MedicalImageProperties ,
    vtkMedicalImageProperties )
```

**10.378.4.22 vtkGetStringMacro() [1/2]**

```
vtkGDCMImageWriter::vtkGetStringMacro (
    SeriesUID )
```

**10.378.4.23 vtkGetStringMacro() [2/2]**

```
vtkGDCMImageWriter::vtkGetStringMacro (
    StudyUID )
```

**10.378.4.24 vtkSetMacro()** [1/7]

```
vtkGDCMImageWriter::vtkSetMacro (
    CompressionType ,
    int )
```

**10.378.4.25 vtkSetMacro()** [2/7]

```
vtkGDCMImageWriter::vtkSetMacro (
    FileLowerLeft ,
    int )
```

**10.378.4.26 vtkSetMacro()** [3/7]

```
vtkGDCMImageWriter::vtkSetMacro (
    ImageFormat ,
    int )
```

**10.378.4.27 vtkSetMacro()** [4/7]

```
vtkGDCMImageWriter::vtkSetMacro (
    LossyFlag ,
    int )
```

**10.378.4.28 vtkSetMacro()** [5/7]

```
vtkGDCMImageWriter::vtkSetMacro (
    PlanarConfiguration ,
    int )
```

**10.378.4.29 vtkSetMacro()** [6/7]

```
vtkGDCMImageWriter::vtkSetMacro (
    Scale ,
    double )
```

**10.378.4.30 vtkSetMacro() [7/7]**

```
vtkGDCMImageWriter::vtkSetMacro (
    Shift ,
    double )
```

**10.378.4.31 vtkSetStringMacro() [1/2]**

```
vtkGDCMImageWriter::vtkSetStringMacro (
    SeriesUID )
```

**10.378.4.32 vtkSetStringMacro() [2/2]**

```
vtkGDCMImageWriter::vtkSetStringMacro (
    StudyUID )
```

**10.378.4.33 vtkTypeMacro()**

```
vtkGDCMImageWriter::vtkTypeMacro (
    vtkGDCMImageWriter ,
    vtkImageWriter )
```

**10.378.4.34 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 [gdcmmorthoplanes.cxx](#).



#### 10.378.4.35 WriteGDCMData()

```
int vtkGDCMImageWriter::WriteGDCMData (
    vtkImageData * data,
    int timeStep ) [protected]
```

#### 10.378.4.36 WriteSlice()

```
void vtkGDCMImageWriter::WriteSlice (
    vtkImageData * data ) [protected]
```

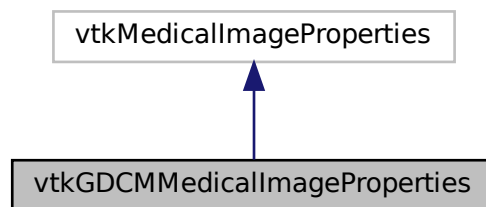
The documentation for this class was generated from the following file:

- [vtkGDCMImageWriter.h](#)

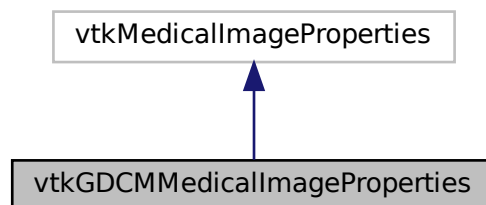
### 10.379 vtkGDCMMedicalImageProperties Class Reference

```
#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](#)

## 10.379.1 Constructor & Destructor Documentation

### 10.379.1.1 [vtkGDCMMedicalImageProperties\(\)](#)

```
vtkGDCMMedicalImageProperties::vtkGDCMMedicalImageProperties ( ) [protected]
```

### 10.379.1.2 [~vtkGDCMMedicalImageProperties\(\)](#)

```
vtkGDCMMedicalImageProperties::~~vtkGDCMMedicalImageProperties ( ) [protected]
```

## 10.379.2 Member Function Documentation

### 10.379.2.1 Clear()

```
virtual void vtkGDCMMedicalImageProperties::Clear ( ) [virtual]
```

### 10.379.2.2 GetFile()

```
gdcM::File const& vtkGDCMMedicalImageProperties::GetFile (
    unsigned int t ) [protected]
```

### 10.379.2.3 New()

```
static vtkGDCMMedicalImageProperties* vtkGDCMMedicalImageProperties::New ( ) [static]
```

### 10.379.2.4 PrintSelf()

```
void vtkGDCMMedicalImageProperties::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

### 10.379.2.5 PushBackFile()

```
void vtkGDCMMedicalImageProperties::PushBackFile (
    gdcM::File const & f ) [protected]
```

### 10.379.2.6 vtkTypeMacro()

```
vtkGDCMMedicalImageProperties::vtkTypeMacro (
    vtkGDCMMedicalImageProperties ,
    vtkMedicalImageProperties )
```

## 10.379.3 Friends And Related Function Documentation

### 10.379.3.1 vtkGDCMImageReader

```
friend class vtkGDCMImageReader [friend]
```

### 10.379.3.2 vtkGDCMImageReader2

```
friend class vtkGDCMImageReader2 [friend]
```

### 10.379.3.3 vtkGDCMImageWriter

```
friend class vtkGDCMImageWriter [friend]
```

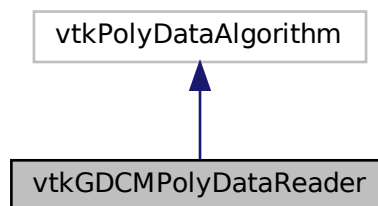
The documentation for this class was generated from the following file:

- [vtkGDCMMedicalImageProperties.h](#)

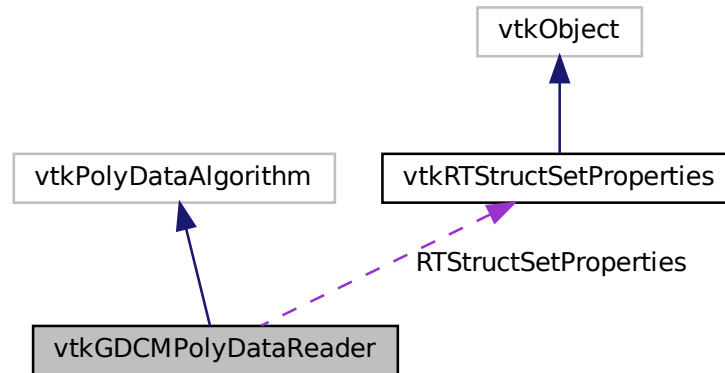
## 10.380 vtkGDCMPolyDataReader Class Reference

```
#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 [gdcmm::Reader](#) &reader)
- int [RequestData](#) (vtkInformation \*, vtkInformationVector \*\*, vtkInformationVector \*)
- int [RequestData\\_HemodynamicWaveformStorage](#) ([gdcmm::Reader](#) const &reader, vtkInformationVector \*outputVector)
- int [RequestData\\_RTStructureSetStorage](#) ([gdcmm::Reader](#) const &reader, vtkInformationVector \*outputVector)
- int [RequestInformation](#) (vtkInformation \*vtkNotUsed(request), vtkInformationVector \*\*vtkNotUsed(inputVector), vtkInformationVector \*outputVector)
- int [RequestInformation\\_HemodynamicWaveformStorage](#) ([gdcmm::Reader](#) const &reader)
- int [RequestInformation\\_RTStructureSetStorage](#) ([gdcmm::Reader](#) const &reader)

## Protected Attributes

- char \* [FileName](#)
- vtkMedicalImageProperties \* [MedicalImageProperties](#)
- vtkRTStructSetProperties \* [RTStructSetProperties](#)

### 10.380.1 Detailed Description

#### Examples

[GenerateRTSTRUCT.cxx](#), [gdcmscene.cxx](#), and [rtstructapp.cxx](#).

### 10.380.2 Constructor & Destructor Documentation

#### 10.380.2.1 vtkGDCMPolyDataReader()

```
vtkGDCMPolyDataReader::vtkGDCMPolyDataReader ( ) [protected]
```

#### 10.380.2.2 ~vtkGDCMPolyDataReader()

```
vtkGDCMPolyDataReader::~~vtkGDCMPolyDataReader ( ) [protected]
```

### 10.380.3 Member Function Documentation

#### 10.380.3.1 FillMedicalImageInformation()

```
void vtkGDCMPolyDataReader::FillMedicalImageInformation (
    const gdcm::Reader & reader ) [protected]
```

#### 10.380.3.2 New()

```
static vtkGDCMPolyDataReader\* vtkGDCMPolyDataReader::New ( ) [static]
```

#### Examples

[GenerateRTSTRUCT.cxx](#), [gdcmscene.cxx](#), and [rtstructapp.cxx](#).

### 10.380.3.3 PrintSelf()

```
virtual void vtkGDCMPolyDataReader::PrintSelf (
    ostream & os,
    vtkIndent indent ) [virtual]
```

### 10.380.3.4 RequestData()

```
int vtkGDCMPolyDataReader::RequestData (
    vtkInformation * ,
    vtkInformationVector ** ,
    vtkInformationVector * ) [protected]
```

### 10.380.3.5 RequestData\_HemodynamicWaveformStorage()

```
int vtkGDCMPolyDataReader::RequestData_HemodynamicWaveformStorage (
    gdcM::Reader const & reader,
    vtkInformationVector * outputVector ) [protected]
```

### 10.380.3.6 RequestData\_RTStructureSetStorage()

```
int vtkGDCMPolyDataReader::RequestData_RTStructureSetStorage (
    gdcM::Reader const & reader,
    vtkInformationVector * outputVector ) [protected]
```

### 10.380.3.7 RequestInformation()

```
int vtkGDCMPolyDataReader::RequestInformation (
    vtkInformation * vtkNotUsed(request),
    vtkInformationVector ** vtkNotUsed(inputVector),
    vtkInformationVector * outputVector ) [protected]
```

### 10.380.3.8 RequestInformation\_HemodynamicWaveformStorage()

```
int vtkGDCMPolyDataReader::RequestInformation_HemodynamicWaveformStorage (
    gdcM::Reader const & reader ) [protected]
```

**10.380.3.9 RequestInformation\_RTStructureSetStorage()**

```
int vtkGDCMPolyDataReader::RequestInformation_RTStructureSetStorage (
    gdcM::Reader const & reader ) [protected]
```

**10.380.3.10 vtkGetObjectMacro() [1/2]**

```
vtkGDCMPolyDataReader::vtkGetObjectMacro (
    MedicalImageProperties ,
    vtkMedicalImageProperties )
```

**10.380.3.11 vtkGetObjectMacro() [2/2]**

```
vtkGDCMPolyDataReader::vtkGetObjectMacro (
    RTStructSetProperties ,
    vtkRTStructSetProperties )
```

**10.380.3.12 vtkGetStringMacro()**

```
vtkGDCMPolyDataReader::vtkGetStringMacro (
    FileName )
```

**10.380.3.13 vtkSetStringMacro()**

```
vtkGDCMPolyDataReader::vtkSetStringMacro (
    FileName )
```

**10.380.3.14 vtkTypeMacro()**

```
vtkGDCMPolyDataReader::vtkTypeMacro (
    vtkGDCMPolyDataReader ,
    vtkPolyDataAlgorithm )
```



## 10.380.4 Member Data Documentation

### 10.380.4.1 FileName

```
char* vtkGDCMPolyDataReader::FileName [protected]
```

### 10.380.4.2 MedicalImageProperties

```
vtkMedicalImageProperties* vtkGDCMPolyDataReader::MedicalImageProperties [protected]
```

### 10.380.4.3 RTStructSetProperties

```
vtkRTStructSetProperties* vtkGDCMPolyDataReader::RTStructSetProperties [protected]
```

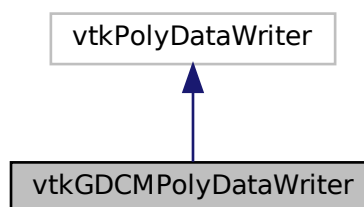
The documentation for this class was generated from the following file:

- [vtkGDCMPolyDataReader.h](#)

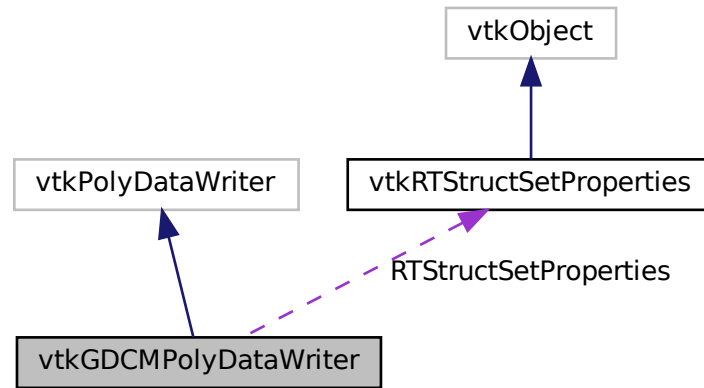
## 10.381 vtkGDCMPolyDataWriter Class Reference

```
#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](#) (gdcm::File &file, int num)
- void [WriteRTSTRUCTInfo](#) (gdcm::File &file)

## Protected Attributes

- vtkMedicalImageProperties \* [MedicalImageProperties](#)
- [vtkRTStructSetProperties](#) \* [RTStructSetProperties](#)

## 10.381.1 Detailed Description

### Examples

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

## 10.381.2 Constructor & Destructor Documentation

### 10.381.2.1 vtkGDCMPolyDataWriter()

```
vtkGDCMPolyDataWriter::vtkGDCMPolyDataWriter ( ) [protected]
```

### 10.381.2.2 ~vtkGDCMPolyDataWriter()

```
vtkGDCMPolyDataWriter::~~vtkGDCMPolyDataWriter ( ) [protected]
```

## 10.381.3 Member Function Documentation

### 10.381.3.1 InitializeRTStructSet()

```
void vtkGDCMPolyDataWriter::InitializeRTStructSet (
    vtkStdString inDirectory,
    vtkStdString inStructLabel,
    vtkStdString inStructName,
    vtkStringArray * inROINames,
    vtkStringArray * inROIAlgorithmName,
    vtkStringArray * inROIType )
```

### Examples

[GenerateRTSTRUCT.cxx](#).

### 10.381.3.2 New()

```
static vtkGDCMPolyDataWriter* vtkGDCMPolyDataWriter::New ( ) [static]
```

#### Examples

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

### 10.381.3.3 PrintSelf()

```
virtual void vtkGDCMPolyDataWriter::PrintSelf (
    ostream & os,
    vtkIndent indent ) [virtual]
```

### 10.381.3.4 SetMedicalImageProperties()

```
virtual void vtkGDCMPolyDataWriter::SetMedicalImageProperties (
    vtkMedicalImageProperties * pd ) [virtual]
```

#### Examples

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

### 10.381.3.5 SetNumberOfInputPorts()

```
void vtkGDCMPolyDataWriter::SetNumberOfInputPorts (
    int n )
```

#### Examples

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

### 10.381.3.6 SetRTStructSetProperties()

```
virtual void vtkGDCMPolyDataWriter::SetRTStructSetProperties (
    vtkRTStructSetProperties * pd ) [virtual]
```

#### Examples

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

### 10.381.3.7 vtkTypeMacro()

```
vtkGDCMPolyDataWriter::vtkTypeMacro (
    vtkGDCMPolyDataWriter ,
    vtkPolyDataWriter )
```

### 10.381.3.8 WriteData()

```
void vtkGDCMPolyDataWriter::WriteData ( ) [protected]
```

### 10.381.3.9 WriteRTSTRUCTData()

```
void vtkGDCMPolyDataWriter::WriteRTSTRUCTData (
    gdcM::File & file,
    int num ) [protected]
```

### 10.381.3.10 WriteRTSTRUCTInfo()

```
void vtkGDCMPolyDataWriter::WriteRTSTRUCTInfo (
    gdcM::File & file ) [protected]
```

## 10.381.4 Member Data Documentation

#### 10.381.4.1 MedicalImageProperties

```
vtkMedicalImageProperties* vtkGDCMPolyDataWriter::MedicalImageProperties [protected]
```

#### 10.381.4.2 RTStructSetProperties

```
vtkRTStructSetProperties* vtkGDCMPolyDataWriter::RTStructSetProperties [protected]
```

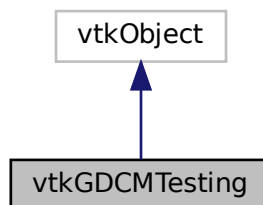
The documentation for this class was generated from the following file:

- [vtkGDCMPolyDataWriter.h](#)

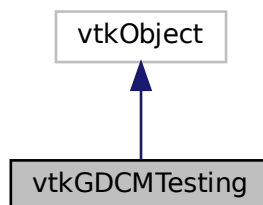
### 10.382 vtkGDCMTesting Class Reference

```
#include <vtkGDCMTesting.h>
```

Inheritance diagram for vtkGDCMTesting:



Collaboration diagram for vtkGDCMTesting:



## Public Types

- typedef const char \*const (\* [MD5MetalmagesType](#))[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 \* [GetMD5Metalmage](#) (unsigned int file)
- static const char \* [GetMHDMD5FromFile](#) (const char \*filepath)
- static unsigned int [GetNumberOfMD5Metalmages](#) ()
- static const char \* [GetRAWMD5FromFile](#) (const char \*filepath)
- static const char \* [GetVTKDataRoot](#) ()
- static [vtkGDCMTesting](#) \* [New](#) ()

## Protected Member Functions

- [vtkGDCMTesting](#) ()
- [~vtkGDCMTesting](#) ()

### 10.382.1 Detailed Description

#### Examples

[HelloActiviz5.cs](#), [HelloVTKWorld2.cs](#), [MetalmageMD5Activiz.cs](#), [ReadSeriesIntoVTK.java](#), and [RefCounting.cs](#).

### 10.382.2 Member Typedef Documentation

#### 10.382.2.1 MD5MetalmagesType

```
typedef const char* const (* vtkGDCMTesting::MD5MetaImagesType) [3]
```

### 10.382.3 Constructor & Destructor Documentation

### 10.382.3.1 vtkGDCMTesting()

```
vtkGDCMTesting::vtkGDCMTesting ( ) [protected]
```

### 10.382.3.2 ~vtkGDCMTesting()

```
vtkGDCMTesting::~~vtkGDCMTesting ( ) [protected]
```

## 10.382.4 Member Function Documentation

### 10.382.4.1 GetGDCMDataRoot()

```
static const char* vtkGDCMTesting::GetGDCMDataRoot ( ) [static]
```

#### Examples

[HelloActiviz5.cs](#), and [ReadSeriesIntoVTK.java](#).

### 10.382.4.2 GetMD5MetaImage()

```
static const char* const* vtkGDCMTesting::GetMD5MetaImage (
    unsigned int file ) [static]
```

### 10.382.4.3 GetMHDMD5FromFile()

```
static const char* vtkGDCMTesting::GetMHDMD5FromFile (
    const char * filepath ) [static]
```

#### Examples

[MetaImageMD5Activiz.cs](#).



#### 10.382.4.4 GetNumberOfMD5MetaImages()

```
static unsigned int vtkGDCMTesting::GetNumberOfMD5MetaImages ( ) [static]
```

#### 10.382.4.5 GetRAWMD5FromFile()

```
static const char* vtkGDCMTesting::GetRAWMD5FromFile (
    const char * filepath ) [static]
```

##### Examples

[MetaImageMD5Activiz.cs](#).

#### 10.382.4.6 GetVTKDataRoot()

```
static const char* vtkGDCMTesting::GetVTKDataRoot ( ) [static]
```

##### Examples

[HelloActiviz5.cs](#), and [HelloVTKWorld2.cs](#).

#### 10.382.4.7 New()

```
static vtkGDCMTesting\* vtkGDCMTesting::New ( ) [static]
```

##### Examples

[RefCounting.cs](#).

#### 10.382.4.8 PrintSelf()

```
void vtkGDCMTesting::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

#### 10.382.4.9 vtkTypeMacro()

```
vtkGDCMTesting::vtkTypeMacro (
    vtkGDCMTesting ,
    vtkObject )
```

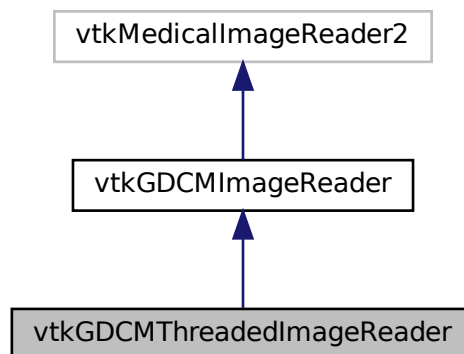
The documentation for this class was generated from the following file:

- [vtkGDCMTesting.h](#)

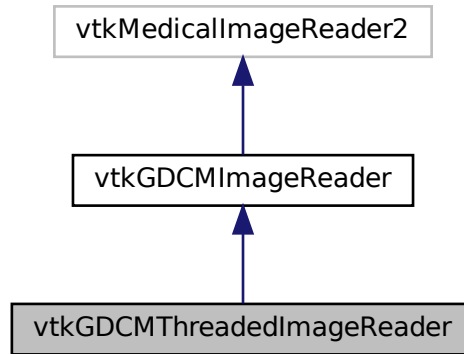
### 10.383 vtkGDCMThreadedImageReader Class Reference

```
#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)

### Static Public Member Functions

- static [vtkGDCMThreadedImageReader](#) \* [New](#) ()

### Protected Member Functions

- [vtkGDCMThreadedImageReader](#) ()
- [~vtkGDCMThreadedImageReader](#) ()
- void [ExecuteData](#) (vtkDataObject \*out)
- void [ExecuteInformation](#) ()
- void [ReadFiles](#) (unsigned int nfiles, const char \*filenames[])
- void [RequestDataCompat](#) ()

### Additional Inherited Members

#### 10.383.1 Constructor & Destructor Documentation

### 10.383.1.1 `vtkGDCMThreadedImageReader()`

```
vtkGDCMThreadedImageReader::vtkGDCMThreadedImageReader ( ) [protected]
```

### 10.383.1.2 `~vtkGDCMThreadedImageReader()`

```
vtkGDCMThreadedImageReader::~~vtkGDCMThreadedImageReader ( ) [protected]
```

## 10.383.2 Member Function Documentation

### 10.383.2.1 `ExecuteData()`

```
void vtkGDCMThreadedImageReader::ExecuteData (
    vtkDataObject * out ) [protected]
```

### 10.383.2.2 `ExecuteInformation()`

```
void vtkGDCMThreadedImageReader::ExecuteInformation ( ) [protected]
```

### 10.383.2.3 `New()`

```
static vtkGDCMThreadedImageReader\* vtkGDCMThreadedImageReader::New ( ) [static]
```

### 10.383.2.4 `PrintSelf()`

```
virtual void vtkGDCMThreadedImageReader::PrintSelf (
    ostream & os,
    vtkIndent indent ) [virtual]
```

Reimplemented from [vtkGDCMImageReader](#).

### 10.383.2.5 ReadFiles()

```
void vtkGDCMThreadedImageReader::ReadFiles (
    unsigned int nfiles,
    const char * filenames[] ) [protected]
```

### 10.383.2.6 RequestDataCompat()

```
void vtkGDCMThreadedImageReader::RequestDataCompat ( ) [protected]
```

### 10.383.2.7 vtkBooleanMacro()

```
vtkGDCMThreadedImageReader::vtkBooleanMacro (
    UseShiftScale ,
    int )
```

### 10.383.2.8 vtkGetMacro()

```
vtkGDCMThreadedImageReader::vtkGetMacro (
    UseShiftScale ,
    int )
```

### 10.383.2.9 vtkSetMacro() [1/3]

```
vtkGDCMThreadedImageReader::vtkSetMacro (
    Scale ,
    double )
```

### 10.383.2.10 vtkSetMacro() [2/3]

```
vtkGDCMThreadedImageReader::vtkSetMacro (
    Shift ,
    double )
```

### 10.383.2.11 vtkSetMacro() [3/3]

```
vtkGDCMThreadedImageReader::vtkSetMacro (
    UseShiftScale ,
    int )
```

### 10.383.2.12 vtkTypeMacro()

```
vtkGDCMThreadedImageReader::vtkTypeMacro (
    vtkGDCMThreadedImageReader ,
    vtkGDCMImageReader )
```

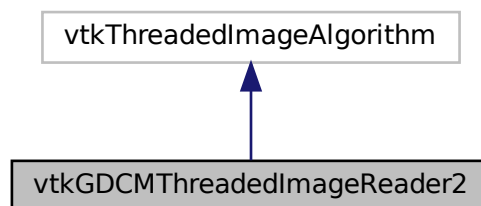
The documentation for this class was generated from the following file:

- [vtkGDCMThreadedImageReader.h](#)

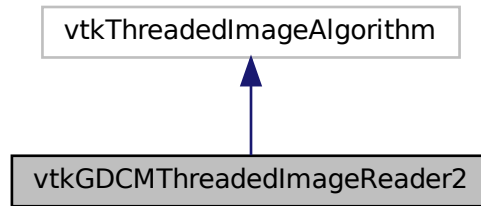
## 10.384 vtkGDCMThreadedImageReader2 Class Reference

```
#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)

## 10.384.1 Constructor & Destructor Documentation

### 10.384.1.1 [vtkGDCMThreadedImageReader2\(\)](#)

```
vtkGDCMThreadedImageReader2::vtkGDCMThreadedImageReader2 ( ) [protected]
```

### 10.384.1.2 [~vtkGDCMThreadedImageReader2\(\)](#)

```
vtkGDCMThreadedImageReader2::~~vtkGDCMThreadedImageReader2 ( ) [protected]
```

## 10.384.2 Member Function Documentation

### 10.384.2.1 [GetFileName\(\)](#)

```
virtual const char* vtkGDCMThreadedImageReader2::GetFileName (
    int i = 0 ) [virtual]
```

### 10.384.2.2 [New\(\)](#)

```
static vtkGDCMThreadedImageReader2\* vtkGDCMThreadedImageReader2::New ( ) [static]
```



### 10.384.2.3 PrintSelf()

```
virtual void vtkGDCMThreadedImageReader2::PrintSelf (
    ostream & os,
    vtkIndent indent ) [virtual]
```

### 10.384.2.4 RequestInformation()

```
int vtkGDCMThreadedImageReader2::RequestInformation (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector ) [protected]
```

### 10.384.2.5 SetFileName()

```
virtual void vtkGDCMThreadedImageReader2::SetFileName (
    const char * filename ) [virtual]
```

### 10.384.2.6 SetFileNames()

```
virtual void vtkGDCMThreadedImageReader2::SetFileNames (
    vtkStringArray * ) [virtual]
```

### 10.384.2.7 SplitExtent()

```
int vtkGDCMThreadedImageReader2::SplitExtent (
    int splitExt[6],
    int startExt[6],
    int num,
    int total )
```

**10.384.2.8 ThreadedRequestData()**

```
void vtkGDCMThreadedImageReader2::ThreadedRequestData (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector,
    vtkImageData *** inData,
    vtkImageData ** outData,
    int outExt[6],
    int id ) [protected]
```

**10.384.2.9 vtkBooleanMacro()** [1/3]

```
vtkGDCMThreadedImageReader2::vtkBooleanMacro (
    FileLowerLeft ,
    int )
```

**10.384.2.10 vtkBooleanMacro()** [2/3]

```
vtkGDCMThreadedImageReader2::vtkBooleanMacro (
    LoadOverlays ,
    int )
```

**10.384.2.11 vtkBooleanMacro()** [3/3]

```
vtkGDCMThreadedImageReader2::vtkBooleanMacro (
    UseShiftScale ,
    int )
```

**10.384.2.12 vtkGetMacro()** [1/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    DataScalarType ,
    int )
```

**10.384.2.13 vtkGetMacro()** [2/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    FileLowerLeft ,
    int )
```

**10.384.2.14 vtkGetMacro()** [3/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    LoadOverlays ,
    int )
```

**10.384.2.15 vtkGetMacro()** [4/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    NumberOfOverlays ,
    int )
```

**10.384.2.16 vtkGetMacro()** [5/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    NumberOfScalarComponents ,
    int )
```

**10.384.2.17 vtkGetMacro()** [6/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    Scale ,
    double )
```

**10.384.2.18 vtkGetMacro()** [7/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    Shift ,
    double )
```

**10.384.2.19 vtkGetMacro() [8/8]**

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    UseShiftScale ,
    int )
```

**10.384.2.20 vtkGetObjectMacro()**

```
vtkGDCMThreadedImageReader2::vtkGetObjectMacro (
    FileNames ,
    vtkStringArray )
```

**10.384.2.21 vtkGetVector3Macro() [1/2]**

```
vtkGDCMThreadedImageReader2::vtkGetVector3Macro (
    DataOrigin ,
    double )
```

**10.384.2.22 vtkGetVector3Macro() [2/2]**

```
vtkGDCMThreadedImageReader2::vtkGetVector3Macro (
    DataSpacing ,
    double )
```

**10.384.2.23 vtkGetVector6Macro()**

```
vtkGDCMThreadedImageReader2::vtkGetVector6Macro (
    DataExtent ,
    int )
```

**10.384.2.24 vtkSetMacro() [1/7]**

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    DataScalarType ,
    int )
```

**10.384.2.25 vtkSetMacro()** [2/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    FileLowerLeft ,
    int )
```

**10.384.2.26 vtkSetMacro()** [3/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    LoadOverlays ,
    int )
```

**10.384.2.27 vtkSetMacro()** [4/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    NumberOfScalarComponents ,
    int )
```

**10.384.2.28 vtkSetMacro()** [5/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    Scale ,
    double )
```

**10.384.2.29 vtkSetMacro()** [6/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    Shift ,
    double )
```

**10.384.2.30 vtkSetMacro()** [7/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    UseShiftScale ,
    int )
```

**10.384.2.31 vtkSetVector3Macro()** [1/2]

```
vtkGDCMThreadedImageReader2::vtkSetVector3Macro (
    DataOrigin ,
    double )
```

**10.384.2.32 vtkSetVector3Macro()** [2/2]

```
vtkGDCMThreadedImageReader2::vtkSetVector3Macro (
    DataSpacing ,
    double )
```

**10.384.2.33 vtkSetVector6Macro()**

```
vtkGDCMThreadedImageReader2::vtkSetVector6Macro (
    DataExtent ,
    int )
```

**10.384.2.34 vtkTypeMacro()**

```
vtkGDCMThreadedImageReader2::vtkTypeMacro (
    vtkGDCMThreadedImageReader2 ,
    vtkThreadedImageAlgorithm )
```

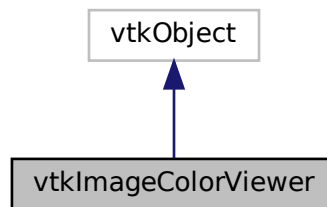
The documentation for this class was generated from the following file:

- [vtkGDCMThreadedImageReader2.h](#)

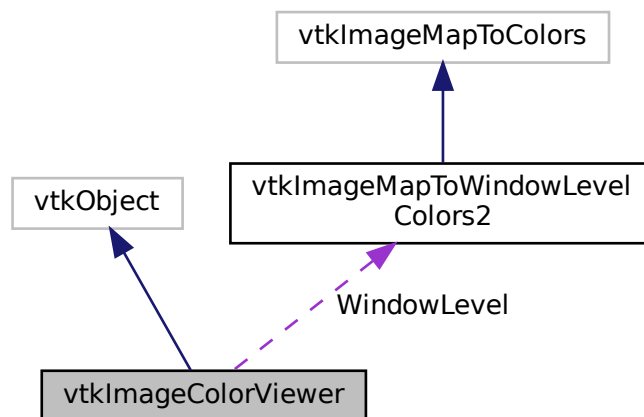
## 10.385 vtkImageColorViewer Class Reference

```
#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](#) ()
- [VTK\\_LEGACY](#) (int GetWholeZMax())
- [VTK\\_LEGACY](#) (int GetWholeZMin())
- [VTK\\_LEGACY](#) (int GetZSlice())
- [VTK\\_LEGACY](#) (void SetZSlice(int))
- [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](#)

### 10.385.1 Detailed Description

#### Examples

[gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

### 10.385.2 Member Enumeration Documentation

#### 10.385.2.1 anonymous enum

anonymous enum

## Enumerator

SLICE_ORIENTATION_YZ	
SLICE_ORIENTATION_XZ	
SLICE_ORIENTATION_XY	

### 10.385.3 Constructor & Destructor Documentation

#### 10.385.3.1 vtkImageColorViewer()

```
vtkImageColorViewer::vtkImageColorViewer ( ) [protected]
```

#### 10.385.3.2 ~vtkImageColorViewer()

```
vtkImageColorViewer::~~vtkImageColorViewer ( ) [protected]
```

### 10.385.4 Member Function Documentation

#### 10.385.4.1 AddInput()

```
virtual void vtkImageColorViewer::AddInput (
    vtkImageData * input ) [virtual]
```

#### 10.385.4.2 AddInputConnection()

```
virtual void vtkImageColorViewer::AddInputConnection (
    vtkAlgorithmOutput * input ) [virtual]
```

#### 10.385.4.3 GetColorLevel()

```
virtual double vtkImageColorViewer::GetColorLevel ( ) [virtual]
```

#### 10.385.4.4 GetColorWindow()

```
virtual double vtkImageColorViewer::GetColorWindow ( ) [virtual]
```

#### 10.385.4.5 GetInput()

```
virtual vtkImageData* vtkImageColorViewer::GetInput ( ) [virtual]
```

#### 10.385.4.6 GetOffScreenRendering()

```
virtual int vtkImageColorViewer::GetOffScreenRendering ( ) [virtual]
```

#### 10.385.4.7 GetOverlayVisibility()

```
double vtkImageColorViewer::GetOverlayVisibility ( )
```

#### 10.385.4.8 GetPosition()

```
virtual int* vtkImageColorViewer::GetPosition ( ) [virtual]
```

#### 10.385.4.9 GetSize()

```
virtual int* vtkImageColorViewer::GetSize ( ) [virtual]
```

**10.385.4.10 GetSliceMax()**

```
virtual int vtkImageColorViewer::GetSliceMax ( ) [virtual]
```

**10.385.4.11 GetSliceMin()**

```
virtual int vtkImageColorViewer::GetSliceMin ( ) [virtual]
```

**10.385.4.12 GetSliceRange() [1/3]**

```
virtual int* vtkImageColorViewer::GetSliceRange ( ) [virtual]
```

**10.385.4.13 GetSliceRange() [2/3]**

```
virtual void vtkImageColorViewer::GetSliceRange (
    int & min,
    int & max ) [virtual]
```

**10.385.4.14 GetSliceRange() [3/3]**

```
virtual void vtkImageColorViewer::GetSliceRange (
    int range[2] ) [inline], [virtual]
```

**10.385.4.15 GetWindowName()**

```
virtual const char* vtkImageColorViewer::GetWindowName ( ) [virtual]
```

**10.385.4.16 InstallPipeline()**

```
virtual void vtkImageColorViewer::InstallPipeline ( ) [protected], [virtual]
```

#### 10.385.4.17 New()

```
static vtkImageColorViewer* vtkImageColorViewer::New ( ) [static]
```

##### Examples

[gdcmrptionplan.cxx](#), and [gdcmrtpplan.cxx](#).

#### 10.385.4.18 PrintSelf()

```
void vtkImageColorViewer::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

#### 10.385.4.19 Render()

```
virtual void vtkImageColorViewer::Render (
    void ) [virtual]
```

##### Examples

[gdcmrptionplan.cxx](#), and [gdcmrtpplan.cxx](#).

#### 10.385.4.20 SetColorLevel()

```
virtual void vtkImageColorViewer::SetColorLevel (
    double s ) [virtual]
```

#### 10.385.4.21 SetColorWindow()

```
virtual void vtkImageColorViewer::SetColorWindow (
    double s ) [virtual]
```

**10.385.4.22 SetDisplayId()**

```
virtual void vtkImageColorViewer::SetDisplayId (
    void * a ) [virtual]
```

**10.385.4.23 SetInput()**

```
virtual void vtkImageColorViewer::SetInput (
    vtkImageData * in ) [virtual]
```

**Examples**

[gdcmrptionplan.cxx](#), and [gdcmrtpplan.cxx](#).

**10.385.4.24 SetInputConnection()**

```
virtual void vtkImageColorViewer::SetInputConnection (
    vtkAlgorithmOutput * input ) [virtual]
```

**10.385.4.25 SetOffScreenRendering()**

```
virtual void vtkImageColorViewer::SetOffScreenRendering (
    int ) [virtual]
```

**10.385.4.26 SetOverlayVisibility()**

```
void vtkImageColorViewer::SetOverlayVisibility (
    double vis )
```

**10.385.4.27 SetParentId()**

```
virtual void vtkImageColorViewer::SetParentId (
    void * a ) [virtual]
```

**10.385.4.28 SetPosition() [1/2]**

```
virtual void vtkImageColorViewer::SetPosition (
    int a,
    int b ) [virtual]
```

**10.385.4.29 SetPosition() [2/2]**

```
virtual void vtkImageColorViewer::SetPosition (
    int a[2] ) [inline], [virtual]
```

References SetPosition().

Referenced by SetPosition().

**10.385.4.30 SetRenderer()**

```
virtual void vtkImageColorViewer::SetRenderer (
    vtkRenderer * arg ) [virtual]
```

**10.385.4.31 SetRenderWindow()**

```
virtual void vtkImageColorViewer::SetRenderWindow (
    vtkRenderWindow * arg ) [virtual]
```

**10.385.4.32 SetSize() [1/2]**

```
virtual void vtkImageColorViewer::SetSize (
    int a,
    int b ) [virtual]
```

**Examples**

[gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

**10.385.4.33 SetSize() [2/2]**

```
virtual void vtkImageColorViewer::SetSize (
    int a[2] ) [inline], [virtual]
```

References SetSize().

Referenced by SetSize().

**10.385.4.34 SetSlice()**

```
virtual void vtkImageColorViewer::SetSlice (
    int s ) [virtual]
```

**10.385.4.35 SetSliceOrientation()**

```
virtual void vtkImageColorViewer::SetSliceOrientation (
    int orientation ) [virtual]
```

**10.385.4.36 SetSliceOrientationToXY()**

```
virtual void vtkImageColorViewer::SetSliceOrientationToXY ( ) [inline], [virtual]
```

References SLICE\_ORIENTATION\_XY.

**10.385.4.37 SetSliceOrientationToXZ()**

```
virtual void vtkImageColorViewer::SetSliceOrientationToXZ ( ) [inline], [virtual]
```

References SLICE\_ORIENTATION\_XZ.

**10.385.4.38 SetSliceOrientationToYZ()**

```
virtual void vtkImageColorViewer::SetSliceOrientationToYZ ( ) [inline], [virtual]
```

References SLICE\_ORIENTATION\_YZ.



#### 10.385.4.39 SetupInteractor()

```
virtual void vtkImageColorViewer::SetupInteractor (
    vtkRenderWindowInteractor * ) [virtual]
```

#### Examples

[gdcmrptionplan.cxx](#), and [gdcmrtpian.cxx](#).

#### 10.385.4.40 SetWindowId()

```
virtual void vtkImageColorViewer::SetWindowId (
    void * a ) [virtual]
```

#### 10.385.4.41 UnInstallPipeline()

```
virtual void vtkImageColorViewer::UnInstallPipeline ( ) [protected], [virtual]
```

#### 10.385.4.42 UpdateDisplayExtent()

```
virtual void vtkImageColorViewer::UpdateDisplayExtent ( ) [virtual]
```

#### 10.385.4.43 UpdateOrientation()

```
virtual void vtkImageColorViewer::UpdateOrientation ( ) [protected], [virtual]
```

#### 10.385.4.44 VTK\_LEGACY() [1/4]

```
vtkImageColorViewer::VTK_LEGACY (
    int GetWholeZMax() )
```

**10.385.4.45 VTK\_LEGACY() [2/4]**

```
vtkImageColorViewer::VTK_LEGACY (
    int  GetWholeZMin() )
```

**10.385.4.46 VTK\_LEGACY() [3/4]**

```
vtkImageColorViewer::VTK_LEGACY (
    int  GetZSlice() )
```

**10.385.4.47 VTK\_LEGACY() [4/4]**

```
vtkImageColorViewer::VTK_LEGACY (
    void  SetZSlice(int) )
```

**10.385.4.48 vtkBooleanMacro()**

```
vtkImageColorViewer::vtkBooleanMacro (
    OffScreenRendering ,
    int )
```

**10.385.4.49 vtkGetMacro() [1/2]**

```
vtkImageColorViewer::vtkGetMacro (
    Slice ,
    int )
```

**10.385.4.50 vtkGetMacro() [2/2]**

```
vtkImageColorViewer::vtkGetMacro (
    SliceOrientation ,
    int )
```

**10.385.4.51 vtkGetObjectMacro() [1/5]**

```
vtkImageColorViewer::vtkGetObjectMacro (
    ImageActor ,
    vtkImageActor )
```

**10.385.4.52 vtkGetObjectMacro() [2/5]**

```
vtkImageColorViewer::vtkGetObjectMacro (
    InteractorStyle ,
    vtkInteractorStyleImage )
```

**10.385.4.53 vtkGetObjectMacro() [3/5]**

```
vtkImageColorViewer::vtkGetObjectMacro (
    Renderer ,
    vtkRenderer )
```

**10.385.4.54 vtkGetObjectMacro() [4/5]**

```
vtkImageColorViewer::vtkGetObjectMacro (
    RenderWindow ,
    vtkRenderWindow )
```

**10.385.4.55 vtkGetObjectMacro() [5/5]**

```
vtkImageColorViewer::vtkGetObjectMacro (
    WindowLevel ,
    vtkImageMapToWindowLevelColors2 )
```

**10.385.4.56 vtkTypeMacro()**

```
vtkImageColorViewer::vtkTypeMacro (
    vtkImageColorViewer ,
    vtkObject )
```

## 10.385.5 Friends And Related Function Documentation

### 10.385.5.1 vtkImageColorViewerCallback

```
friend class vtkImageColorViewerCallback [friend]
```

## 10.385.6 Member Data Documentation

### 10.385.6.1 FirstRender

```
int vtkImageColorViewer::FirstRender [protected]
```

### 10.385.6.2 ImageActor

```
vtkImageActor* vtkImageColorViewer::ImageActor [protected]
```

### 10.385.6.3 Interactor

```
vtkRenderWindowInteractor* vtkImageColorViewer::Interactor [protected]
```

### 10.385.6.4 InteractorStyle

```
vtkInteractorStyleImage* vtkImageColorViewer::InteractorStyle [protected]
```

### 10.385.6.5 OverlayImageActor

```
vtkImageActor* vtkImageColorViewer::OverlayImageActor [protected]
```

### 10.385.6.6 Renderer

```
vtkRenderer* vtkImageColorViewer::Renderer [protected]
```

### 10.385.6.7 RenderWindow

```
vtkRenderWindow* vtkImageColorViewer::RenderWindow [protected]
```

### 10.385.6.8 Slice

```
int vtkImageColorViewer::Slice [protected]
```

### 10.385.6.9 SliceOrientation

```
int vtkImageColorViewer::SliceOrientation [protected]
```

### 10.385.6.10 WindowLevel

```
vtkImageMapToWindowLevelColors2* vtkImageColorViewer::WindowLevel [protected]
```

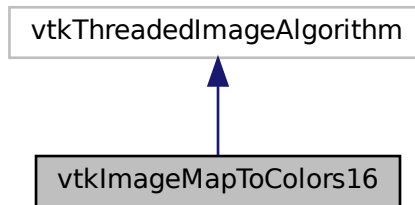
The documentation for this class was generated from the following file:

- [vtkImageColorViewer.h](#)

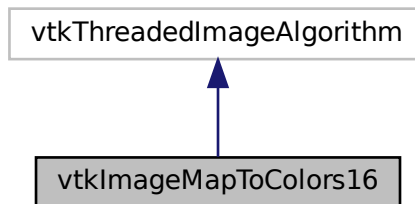
## 10.386 vtkImageMapToColors16 Class Reference

```
#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](#)

## 10.386.1 Constructor & Destructor Documentation

### 10.386.1.1 vtkImageMapToColors16()

```
vtkImageMapToColors16::vtkImageMapToColors16 ( ) [protected]
```

### 10.386.1.2 ~vtkImageMapToColors16()

```
vtkImageMapToColors16::~~vtkImageMapToColors16 ( ) [protected]
```

## 10.386.2 Member Function Documentation

### 10.386.2.1 GetMTime()

```
virtual unsigned long vtkImageMapToColors16::GetMTime ( ) [virtual]
```

### 10.386.2.2 New()

```
static vtkImageMapToColors16\* vtkImageMapToColors16::New ( ) [static]
```

### 10.386.2.3 PrintSelf()

```
void vtkImageMapToColors16::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

### 10.386.2.4 RequestData()

```
virtual int vtkImageMapToColors16::RequestData (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector ) [protected], [virtual]
```

### 10.386.2.5 RequestInformation()

```
virtual int vtkImageMapToColors16::RequestInformation (
    vtkInformation * ,
    vtkInformationVector ** ,
    vtkInformationVector * ) [protected], [virtual]
```



### 10.386.2.6 SetLookupTable()

```
virtual void vtkImageMapToColors16::SetLookupTable (
    vtkScalarsToColors * ) [virtual]
```

### 10.386.2.7 SetOutputFormatToLuminance()

```
void vtkImageMapToColors16::SetOutputFormatToLuminance ( ) [inline]
```

### 10.386.2.8 SetOutputFormatToLuminanceAlpha()

```
void vtkImageMapToColors16::SetOutputFormatToLuminanceAlpha ( ) [inline]
```

### 10.386.2.9 SetOutputFormatToRGB()

```
void vtkImageMapToColors16::SetOutputFormatToRGB ( ) [inline]
```

### 10.386.2.10 SetOutputFormatToRGBA()

```
void vtkImageMapToColors16::SetOutputFormatToRGBA ( ) [inline]
```

### 10.386.2.11 ThreadedRequestData()

```
void vtkImageMapToColors16::ThreadedRequestData (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector,
    vtkImageData *** inData,
    vtkImageData ** outData,
    int extent[6],
    int id ) [protected]
```

**10.386.2.12 vtkBooleanMacro()**

```
vtkImageMapToColors16::vtkBooleanMacro (
    PassAlphaToOutput ,
    int )
```

**10.386.2.13 vtkGetMacro() [1/3]**

```
vtkImageMapToColors16::vtkGetMacro (
    ActiveComponent ,
    int )
```

**10.386.2.14 vtkGetMacro() [2/3]**

```
vtkImageMapToColors16::vtkGetMacro (
    OutputFormat ,
    int )
```

**10.386.2.15 vtkGetMacro() [3/3]**

```
vtkImageMapToColors16::vtkGetMacro (
    PassAlphaToOutput ,
    int )
```

**10.386.2.16 vtkGetObjectMacro()**

```
vtkImageMapToColors16::vtkGetObjectMacro (
    LookupTable ,
    vtkScalarsToColors )
```

**10.386.2.17 vtkSetMacro() [1/3]**

```
vtkImageMapToColors16::vtkSetMacro (
    ActiveComponent ,
    int )
```

**10.386.2.18 vtkSetMacro() [2/3]**

```
vtkImageMapToColors16::vtkSetMacro (
    OutputFormat ,
    int )
```

**10.386.2.19 vtkSetMacro() [3/3]**

```
vtkImageMapToColors16::vtkSetMacro (
    PassAlphaToOutput ,
    int )
```

**10.386.2.20 vtkTypeMacro()**

```
vtkImageMapToColors16::vtkTypeMacro (
    vtkImageMapToColors16 ,
    vtkThreadedImageAlgorithm )
```

**10.386.3 Member Data Documentation****10.386.3.1 ActiveComponent**

```
int vtkImageMapToColors16::ActiveComponent [protected]
```

**10.386.3.2 DataWasPassed**

```
int vtkImageMapToColors16::DataWasPassed [protected]
```

**10.386.3.3 LookupTable**

```
vtkScalarsToColors* vtkImageMapToColors16::LookupTable [protected]
```

### 10.386.3.4 OutputFormat

```
int vtkImageMapToColors16::OutputFormat [protected]
```

### 10.386.3.5 PassAlphaToOutput

```
int vtkImageMapToColors16::PassAlphaToOutput [protected]
```

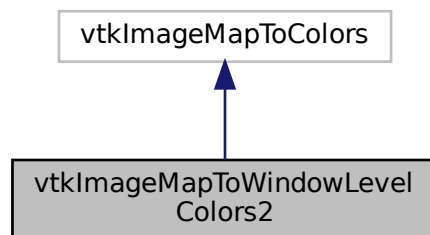
The documentation for this class was generated from the following file:

- [vtkImageMapToColors16.h](#)

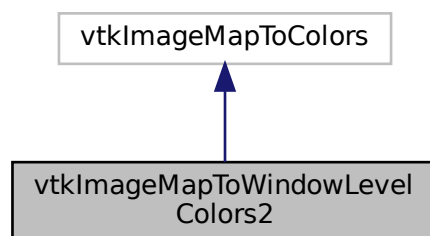
## 10.387 vtkImageMapToWindowLevelColors2 Class Reference

```
#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](#)

## 10.387.1 Constructor & Destructor Documentation

### 10.387.1.1 [vtkImageMapToWindowLevelColors2\(\)](#)

```
vtkImageMapToWindowLevelColors2::vtkImageMapToWindowLevelColors2 ( ) [protected]
```

### 10.387.1.2 [~vtkImageMapToWindowLevelColors2\(\)](#)

```
vtkImageMapToWindowLevelColors2::~~vtkImageMapToWindowLevelColors2 ( ) [protected]
```

## 10.387.2 Member Function Documentation

### 10.387.2.1 New()

```
static vtkImageMapToWindowLevelColors2* vtkImageMapToWindowLevelColors2::New ( ) [static]
```

### 10.387.2.2 PrintSelf()

```
void vtkImageMapToWindowLevelColors2::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

### 10.387.2.3 RequestData()

```
virtual int vtkImageMapToWindowLevelColors2::RequestData (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector ) [protected], [virtual]
```

### 10.387.2.4 RequestInformation()

```
virtual int vtkImageMapToWindowLevelColors2::RequestInformation (
    vtkInformation * ,
    vtkInformationVector ** ,
    vtkInformationVector * ) [protected], [virtual]
```

### 10.387.2.5 ThreadedRequestData()

```
void vtkImageMapToWindowLevelColors2::ThreadedRequestData (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector,
    vtkImageData *** inData,
    vtkImageData ** outData,
    int extent[6],
    int id ) [protected]
```

**10.387.2.6 vtkGetMacro() [1/2]**

```
vtkImageMapToWindowLevelColors2::vtkGetMacro (
    Level ,
    double )
```

**10.387.2.7 vtkGetMacro() [2/2]**

```
vtkImageMapToWindowLevelColors2::vtkGetMacro (
    Window ,
    double )
```

**10.387.2.8 vtkSetMacro() [1/2]**

```
vtkImageMapToWindowLevelColors2::vtkSetMacro (
    Level ,
    double )
```

**10.387.2.9 vtkSetMacro() [2/2]**

```
vtkImageMapToWindowLevelColors2::vtkSetMacro (
    Window ,
    double )
```

**10.387.2.10 vtkTypeMacro()**

```
vtkImageMapToWindowLevelColors2::vtkTypeMacro (
    vtkImageMapToWindowLevelColors2 ,
    vtkImageMapToColors )
```

**10.387.3 Member Data Documentation**

### 10.387.3.1 Level

```
double vtkImageMapToWindowLevelColors2::Level [protected]
```

### 10.387.3.2 Window

```
double vtkImageMapToWindowLevelColors2::Window [protected]
```

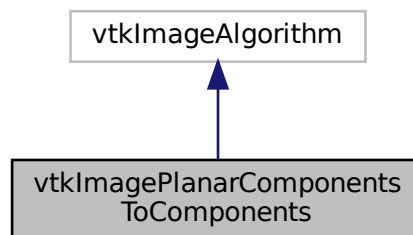
The documentation for this class was generated from the following file:

- [vtkImageMapToWindowLevelColors2.h](#)

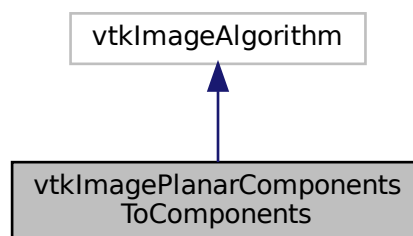
## 10.388 vtkImagePlanarComponentsToComponents Class Reference

```
#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 \*)

## 10.388.1 Constructor & Destructor Documentation

### 10.388.1.1 vtkImagePlanarComponentsToComponents()

```
vtkImagePlanarComponentsToComponents::vtkImagePlanarComponentsToComponents ( ) [protected]
```

### 10.388.1.2 ~vtkImagePlanarComponentsToComponents()

```
vtkImagePlanarComponentsToComponents::~~vtkImagePlanarComponentsToComponents ( ) [inline], [protected]
```

## 10.388.2 Member Function Documentation

### 10.388.2.1 New()

```
static vtkImagePlanarComponentsToComponents* vtkImagePlanarComponentsToComponents::New ( ) [static]
```

### 10.388.2.2 PrintSelf()

```
void vtkImagePlanarComponentsToComponents::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

### 10.388.2.3 RequestData()

```
virtual int vtkImagePlanarComponentsToComponents::RequestData (
    vtkInformation * ,
    vtkInformationVector ** ,
    vtkInformationVector * ) [protected], [virtual]
```

### 10.388.2.4 vtkTypeMacro()

```
vtkImagePlanarComponentsToComponents::vtkTypeMacro (
    vtkImagePlanarComponentsToComponents ,
    vtkImageAlgorithm )
```

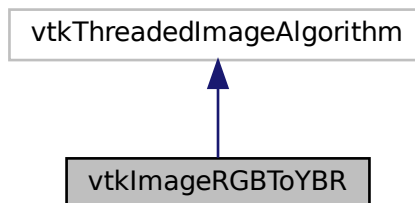
The documentation for this class was generated from the following file:

- [vtkImagePlanarComponentsToComponents.h](#)

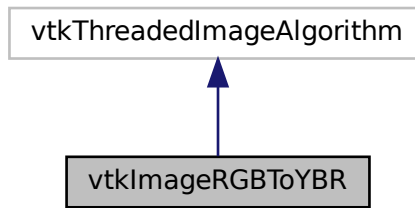
## 10.389 vtkImageRGBToYBR Class Reference

```
#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)

## 10.389.1 Constructor & Destructor Documentation

### 10.389.1.1 vtkImageRGBToYBR()

```
vtkImageRGBToYBR::vtkImageRGBToYBR ( ) [protected]
```

### 10.389.1.2 ~vtkImageRGBToYBR()

```
vtkImageRGBToYBR::~~vtkImageRGBToYBR ( ) [inline], [protected]
```

## 10.389.2 Member Function Documentation

### 10.389.2.1 New()

```
static vtkImageRGBToYBR* vtkImageRGBToYBR::New ( ) [static]
```

### 10.389.2.2 PrintSelf()

```
void vtkImageRGBToYBR::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

### 10.389.2.3 ThreadedExecute()

```
void vtkImageRGBToYBR::ThreadedExecute (
    vtkImageData * inData,
    vtkImageData * outData,
    int ext[6],
    int id ) [protected]
```

### 10.389.2.4 vtkTypeMacro()

```
vtkImageRGBToYBR::vtkTypeMacro (
    vtkImageRGBToYBR ,
    vtkThreadedImageAlgorithm )
```

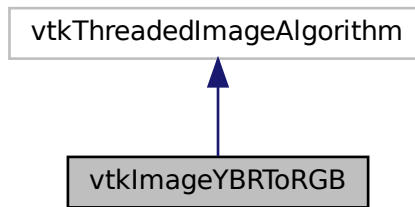
The documentation for this class was generated from the following file:

- [vtkImageRGBToYBR.h](#)

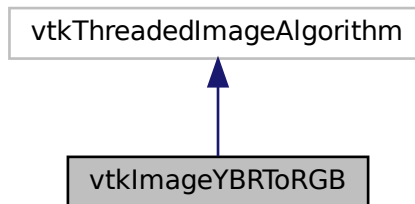
## 10.390 vtkImageYBRToRGB Class Reference

```
#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)

## 10.390.1 Constructor & Destructor Documentation

### 10.390.1.1 vtkImageYBRToRGB()

```
vtkImageYBRToRGB::vtkImageYBRToRGB ( ) [protected]
```

### 10.390.1.2 ~vtkImageYBRToRGB()

```
vtkImageYBRToRGB::~~vtkImageYBRToRGB ( ) [inline], [protected]
```

## 10.390.2 Member Function Documentation

### 10.390.2.1 New()

```
static vtkImageYBRToRGB\* vtkImageYBRToRGB::New ( ) [static]
```

### 10.390.2.2 PrintSelf()

```
void vtkImageYBRToRGB::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

### 10.390.2.3 ThreadedExecute()

```
void vtkImageYBRToRGB::ThreadedExecute (
    vtkImageData * inData,
    vtkImageData * outData,
    int ext[6],
    int id ) [protected]
```

### 10.390.2.4 vtkTypeMacro()

```
vtkImageYBRToRGB::vtkTypeMacro (
    vtkImageYBRToRGB ,
    vtkThreadedImageAlgorithm )
```

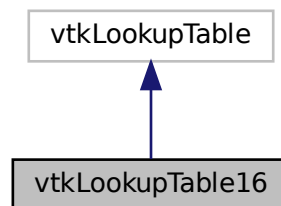
The documentation for this class was generated from the following file:

- [vtkImageYBRToRGB.h](#)

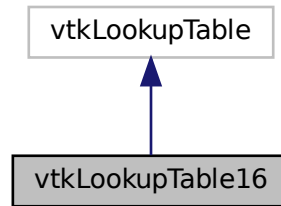
## 10.391 vtkLookupTable16 Class Reference

```
#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](#)

## 10.391.1 Constructor & Destructor Documentation



### 10.391.1.1 vtkLookupTable16()

```
vtkLookupTable16::vtkLookupTable16 (
    int size = 256,
    int ext = 256 ) [protected]
```

### 10.391.1.2 ~vtkLookupTable16()

```
vtkLookupTable16::~~vtkLookupTable16 ( ) [protected]
```

## 10.391.2 Member Function Documentation

### 10.391.2.1 Build()

```
void vtkLookupTable16::Build ( )
```

### 10.391.2.2 GetPointer()

```
unsigned short* vtkLookupTable16::GetPointer (
    const vtkIdType id ) [inline]
```

### 10.391.2.3 MapScalarsThroughTable2()

```
void vtkLookupTable16::MapScalarsThroughTable2 (
    void * input,
    unsigned char * output,
    int inputDataType,
    int numberOfValues,
    int inputIncrement,
    int outputFormat ) [protected]
```

#### 10.391.2.4 New()

```
static vtkLookupTable16* vtkLookupTable16::New ( ) [static]
```

#### 10.391.2.5 PrintSelf()

```
void vtkLookupTable16::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

#### 10.391.2.6 SetNumberOfTableValues()

```
void vtkLookupTable16::SetNumberOfTableValues (
    vtkIdType number )
```

#### 10.391.2.7 vtkTypeMacro()

```
vtkLookupTable16::vtkTypeMacro (
    vtkLookupTable16 ,
    vtkLookupTable )
```

#### 10.391.2.8 WritePointer()

```
unsigned char * vtkLookupTable16::WritePointer (
    const vtkIdType id,
    const int number ) [inline]
```

References Table16.

### 10.391.3 Member Data Documentation

### 10.391.3.1 Table16

```
vtkUnsignedShortArray* vtkLookupTable16::Table16 [protected]
```

Referenced by WritePointer().

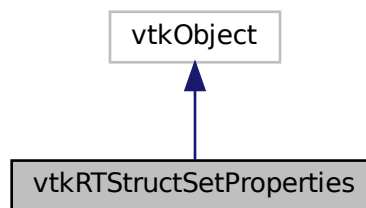
The documentation for this class was generated from the following file:

- [vtkLookupTable16.h](#)

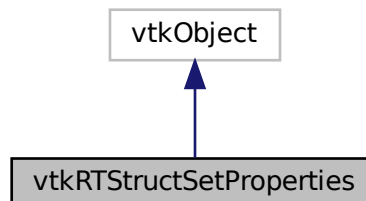
## 10.392 vtkRTStructSetProperties Class Reference

```
#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](#)

### 10.392.1 Detailed Description

Examples

[GenerateRTSTRUCT.cxx](#).

### 10.392.2 Constructor & Destructor Documentation

#### 10.392.2.1 vtkRTStructSetProperties()

```
vtkRTStructSetProperties::vtkRTStructSetProperties ( ) [protected]
```

#### 10.392.2.2 ~vtkRTStructSetProperties()

```
vtkRTStructSetProperties::~~vtkRTStructSetProperties ( ) [protected]
```

### 10.392.3 Member Function Documentation

### 10.392.3.1 AddContourReferencedFrameOfReference()

```
void vtkRTStructSetProperties::AddContourReferencedFrameOfReference (
    vtkIdType pdnum,
    const char * classuid,
    const char * instanceuid )
```

### 10.392.3.2 AddReferencedFrameOfReference()

```
void vtkRTStructSetProperties::AddReferencedFrameOfReference (
    const char * classuid,
    const char * instanceuid )
```

### 10.392.3.3 AddStructureSetROI()

```
void vtkRTStructSetProperties::AddStructureSetROI (
    int roinumber,
    const char * refframerefuid,
    const char * roiname,
    const char * ROIGenerationAlgorithm,
    const char * ROIDescription = 0 )
```

### 10.392.3.4 AddStructureSetROIObservation()

```
void vtkRTStructSetProperties::AddStructureSetROIObservation (
    int refnumber,
    int observationnumber,
    const char * rtroiinterpretedtype,
    const char * roiinterpreter,
    const char * roiobservationlabel = 0 )
```

### 10.392.3.5 Clear()

```
virtual void vtkRTStructSetProperties::Clear ( ) [virtual]
```

### 10.392.3.6 DeepCopy()

```
virtual void vtkRTStructSetProperties::DeepCopy (
    vtkRTStructSetProperties * p ) [virtual]
```

### 10.392.3.7 GetContourReferencedFrameOfReferenceClassUID()

```
const char* vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceClassUID (
    vtkIdType pdnum,
    vtkIdType id )
```

### 10.392.3.8 GetContourReferencedFrameOfReferenceInstanceUID()

```
const char* vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceInstanceUID (
    vtkIdType pdnum,
    vtkIdType id )
```

### 10.392.3.9 GetNumberOfContourReferencedFrameOfReferences() [1/2]

```
vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences ( )
```

### 10.392.3.10 GetNumberOfContourReferencedFrameOfReferences() [2/2]

```
vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences (
    vtkIdType pdnum )
```

### 10.392.3.11 GetNumberOfReferencedFrameOfReferences()

```
vtkIdType vtkRTStructSetProperties::GetNumberOfReferencedFrameOfReferences ( )
```

**10.392.3.12 GetNumberOfStructureSetROIs()**

```
vtkIdType vtkRTStructSetProperties::GetNumberOfStructureSetROIs ( )
```

**10.392.3.13 GetReferencedFrameOfReferenceClassUID()**

```
const char* vtkRTStructSetProperties::GetReferencedFrameOfReferenceClassUID (
    vtkIdType id )
```

**10.392.3.14 GetReferencedFrameOfReferenceInstanceUID()**

```
const char* vtkRTStructSetProperties::GetReferencedFrameOfReferenceInstanceUID (
    vtkIdType id )
```

**10.392.3.15 GetStructureSetObservationNumber()**

```
int vtkRTStructSetProperties::GetStructureSetObservationNumber (
    vtkIdType id )
```

**10.392.3.16 GetStructureSetROIDescription()**

```
const char* vtkRTStructSetProperties::GetStructureSetROIDescription (
    vtkIdType id )
```

**10.392.3.17 GetStructureSetROIGenerationAlgorithm()**

```
const char* vtkRTStructSetProperties::GetStructureSetROIGenerationAlgorithm (
    vtkIdType id )
```



**10.392.3.18 GetStructureSetROIName()**

```
const char* vtkRTStructSetProperties::GetStructureSetROIName (
    vtkIdType )
```

**10.392.3.19 GetStructureSetROINumber()**

```
int vtkRTStructSetProperties::GetStructureSetROINumber (
    vtkIdType id )
```

**10.392.3.20 GetStructureSetROIObservationLabel()**

```
const char* vtkRTStructSetProperties::GetStructureSetROIObservationLabel (
    vtkIdType id )
```

**10.392.3.21 GetStructureSetROIRefFrameRefUID()**

```
const char* vtkRTStructSetProperties::GetStructureSetROIRefFrameRefUID (
    vtkIdType )
```

**10.392.3.22 GetStructureSetRTROIInterpretedType()**

```
const char* vtkRTStructSetProperties::GetStructureSetRTROIInterpretedType (
    vtkIdType id )
```

**10.392.3.23 New()**

```
static vtkRTStructSetProperties\* vtkRTStructSetProperties::New ( ) [static]
```

**Examples**

[GenerateRTSTRUCT.cxx](#).

**10.392.3.24 PrintSelf()**

```
void vtkRTStructSetProperties::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

**10.392.3.25 vtkGetStringMacro() [1/9]**

```
vtkRTStructSetProperties::vtkGetStringMacro (
    ReferenceFrameOfReferenceUID )
```

**10.392.3.26 vtkGetStringMacro() [2/9]**

```
vtkRTStructSetProperties::vtkGetStringMacro (
    ReferenceSeriesInstanceUID )
```

**10.392.3.27 vtkGetStringMacro() [3/9]**

```
vtkRTStructSetProperties::vtkGetStringMacro (
    SeriesInstanceUID )
```

**10.392.3.28 vtkGetStringMacro() [4/9]**

```
vtkRTStructSetProperties::vtkGetStringMacro (
    SOPInstanceUID )
```

**10.392.3.29 vtkGetStringMacro() [5/9]**

```
vtkRTStructSetProperties::vtkGetStringMacro (
    StructureSetDate )
```

**10.392.3.30 vtkGetStringMacro()** [6/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
    StructureSetLabel )
```

**10.392.3.31 vtkGetStringMacro()** [7/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
    StructureSetName )
```

**10.392.3.32 vtkGetStringMacro()** [8/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
    StructureSetTime )
```

**10.392.3.33 vtkGetStringMacro()** [9/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
    StudyInstanceUID )
```

**10.392.3.34 vtkSetStringMacro()** [1/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    ReferenceFrameOfReferenceUID )
```

**10.392.3.35 vtkSetStringMacro()** [2/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    ReferenceSeriesInstanceUID )
```

**10.392.3.36 vtkSetStringMacro()** [3/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    SeriesInstanceUID )
```

**10.392.3.37 vtkSetStringMacro()** [4/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    SOPInstanceUID )
```

**10.392.3.38 vtkSetStringMacro()** [5/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    StructureSetDate )
```

**10.392.3.39 vtkSetStringMacro()** [6/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    StructureSetLabel )
```

**10.392.3.40 vtkSetStringMacro()** [7/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    StructureSetName )
```

**10.392.3.41 vtkSetStringMacro()** [8/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    StructureSetTime )
```

### 10.392.3.42 vtkSetStringMacro() [9/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    StudyInstanceUID )
```

### 10.392.3.43 vtkTypeMacro()

```
vtkRTStructSetProperties::vtkTypeMacro (
    vtkRTStructSetProperties ,
    vtkObject )
```

## 10.392.4 Member Data Documentation

### 10.392.4.1 Internals

```
vtkRTStructSetPropertiesInternals* vtkRTStructSetProperties::Internals [protected]
```

### 10.392.4.2 ReferenceFrameOfReferenceUID

```
char* vtkRTStructSetProperties::ReferenceFrameOfReferenceUID [protected]
```

### 10.392.4.3 ReferenceSeriesInstanceUID

```
char* vtkRTStructSetProperties::ReferenceSeriesInstanceUID [protected]
```

### 10.392.4.4 SeriesInstanceUID

```
char* vtkRTStructSetProperties::SeriesInstanceUID [protected]
```

#### 10.392.4.5 SOPInstanceUID

```
char* vtkRTStructSetProperties::SOPInstanceUID [protected]
```

#### 10.392.4.6 StructureSetDate

```
char* vtkRTStructSetProperties::StructureSetDate [protected]
```

#### 10.392.4.7 StructureSetLabel

```
char* vtkRTStructSetProperties::StructureSetLabel [protected]
```

#### 10.392.4.8 StructureSetName

```
char* vtkRTStructSetProperties::StructureSetName [protected]
```

#### 10.392.4.9 StructureSetTime

```
char* vtkRTStructSetProperties::StructureSetTime [protected]
```

#### 10.392.4.10 StudyInstanceUID

```
char* vtkRTStructSetProperties::StudyInstanceUID [protected]
```

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

- [vtkRTStructSetProperties.h](#)

## 10.393 gdcm::Waveform Class Reference

[Waveform](#) class.

```
#include <gdcmWaveform.h>
```

## Public Member Functions

- [Waveform](#) ()=default

### 10.393.1 Detailed Description

[Waveform](#) class.

### 10.393.2 Constructor & Destructor Documentation

#### 10.393.2.1 Waveform()

```
gdcm::Waveform::Waveform ( ) [default]
```

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

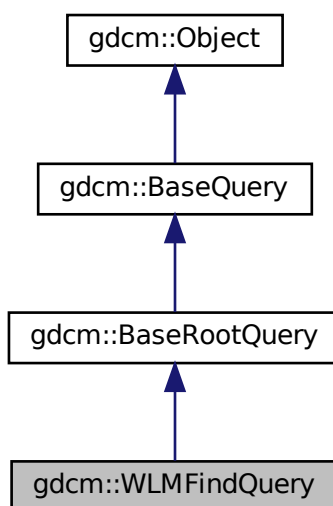
- [gdcmWaveform.h](#)

## 10.394 gdcm::WLMFindQuery Class Reference

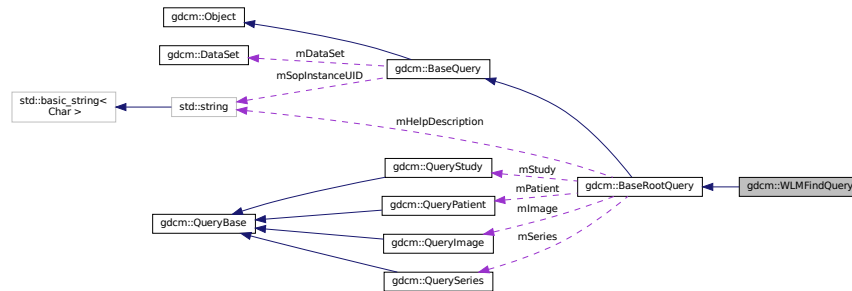
PatientRootQuery.

```
#include <gdcmWLMFindQuery.h>
```

Inheritance diagram for gdcm::WLMFindQuery:



Collaboration diagram for `gdcm::WLMFindQuery`:



## Public Member Functions

- [WLMFindQuery](#) ()
- [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

## Protected Member Functions

- [DataSet GetValidDataSet](#) () const

## Friends

- class [QueryFactory](#)

## Additional Inherited Members

### 10.394.1 Detailed Description

PatientRootQuery.

contains: the class which will produce a dataset for c-find with patient root

### 10.394.2 Constructor & Destructor Documentation



### 10.394.2.1 WLMFindQuery()

```
gdcm::WLMFindQuery::WLMFindQuery ( )
```

## 10.394.3 Member Function Documentation

### 10.394.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::WLMFindQuery::GetAbstractSyntaxUID ( ) const [override], [virtual]
```

Implements [gdcm::BaseQuery](#).

### 10.394.3.2 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](#).

### 10.394.3.3 GetValidDataSet()

```
DataSet gdcm::WLMFindQuery::GetValidDataSet ( ) const [protected]
```

### 10.394.3.4 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 dcmtk

Implements [gdcm::BaseRootQuery](#).

### 10.394.3.5 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](#).

## 10.394.4 Friends And Related Function Documentation

### 10.394.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

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

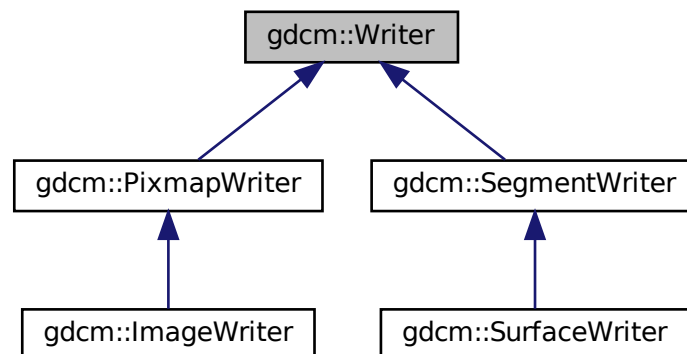
- [gdcmWLMFindQuery.h](#)

## 10.395 gdcm::Writer Class Reference

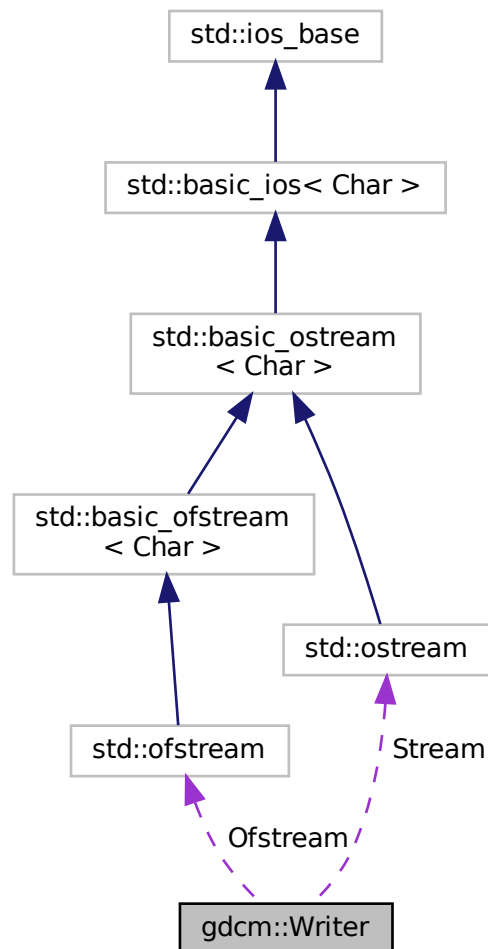
[Writer](#) ala DOM (Document [Object](#) Model)

```
#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](#)

## 10.395.1 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 (garanteed 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

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [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](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [StreamImageReaderTest.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

## 10.395.2 Constructor & Destructor Documentation

### 10.395.2.1 Writer()

```
gdcm::Writer::Writer ( )
```

### 10.395.2.2 ~Writer()

```
virtual gdcm::Writer::~~Writer ( ) [virtual]
```

## 10.395.3 Member Function Documentation

### 10.395.3.1 CheckFileMetaInformationOff()

```
void gdcm::Writer::CheckFileMetaInformationOff ( ) [inline]
```

#### Examples

[CreateFakeRTDOSE.cxx](#), [FixBrokenJ2K.cxx](#), and [HelloWorld.cxx](#).

### 10.395.3.2 CheckFileMetaInformationOn()

```
void gdcm::Writer::CheckFileMetaInformationOn ( ) [inline]
```

### 10.395.3.3 GetCheckFileMetaInformation()

```
bool gdcm::Writer::GetCheckFileMetaInformation ( ) const [inline], [protected]
```

#### 10.395.3.4 GetFile()

```
File& gdcm::Writer::GetFile ( ) [inline]
```

##### Examples

[CreateJPIPDataSet.cxx](#), [EncapsulateFileInRawData.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GetSubSequenceData.cxx](#), [QIDO-RS.cxx](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

#### 10.395.3.5 GetStreamPtr()

```
std::ostream* gdcm::Writer::GetStreamPtr ( ) const [inline], [protected]
```

#### 10.395.3.6 SetCheckFileMetaInformation()

```
void gdcm::Writer::SetCheckFileMetaInformation (
    bool b ) [inline]
```

Undocumented function, do not use (= leave default)

##### Examples

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), and [PatchFile.cxx](#).

#### 10.395.3.7 SetFile()

```
void gdcm::Writer::SetFile (
    const File & f ) [inline]
```

Set/Get the DICOM file ([DataSet](#) + Header)

##### Examples

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CompressImage.cxx](#), [CreateFakeRTDOSE.cxx](#), [DeriveSeries.cxx](#), [DuplicatePCDE.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [MergeTwoFiles.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

### 10.395.3.8 SetFileName()

```
void gdcm::Writer::SetFileName (
    const char * filename_native )
```

Set the filename of DICOM file to write:

#### Examples

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CompressImage.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [DeriveSeries.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [MergeTwoFiles.cxx](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [TemplateEmptyImage.cxx](#), [csa2img.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

### 10.395.3.9 SetStream()

```
void gdcm::Writer::SetStream (
    std::ostream & output_stream ) [inline]
```

Set user ostream buffer.

### 10.395.3.10 SetWriteDataSetOnly()

```
void gdcm::Writer::SetWriteDataSetOnly (
    bool b ) [inline], [protected]
```

### 10.395.3.11 Write()

```
virtual bool gdcm::Writer::Write ( ) [virtual]
```

Main function to tell the writer to write.

Reimplemented in [gdcm::SurfaceWriter](#), [gdcm::SegmentWriter](#), [gdcm::PixmapWriter](#), and [gdcm::ImageWriter](#).

#### Examples

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [DeriveSeries.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

## 10.395.4 Friends And Related Function Documentation

### 10.395.4.1 StreamImageWriter

```
friend class StreamImageWriter [friend]
```

## 10.395.5 Member Data Documentation

### 10.395.5.1 Ofstream

```
std::ofstream* gdcM::Writer::Ofstream [protected]
```

### 10.395.5.2 Stream

```
std::ostream* gdcM::Writer::Stream [protected]
```

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

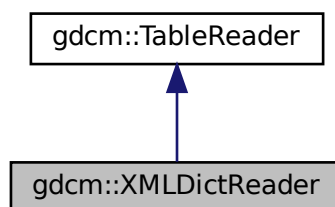
- [gdcMWriter.h](#)

## 10.396 gdcM::XMLDictReader Class Reference

Class for representing a [XMLDictReader](#).

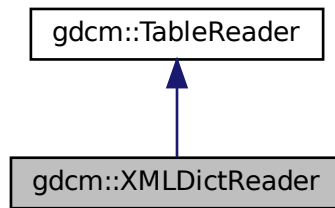
```
#include <gdcXMLDictReader.h>
```

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)

## Protected Member Functions

- void [HandleDescription](#) (const char \*\*atts)
- void [HandleEntry](#) (const char \*\*atts)

### 10.396.1 Detailed Description

Class for representing a [XMLDictReader](#).

Note

bla Will read the DICOMV3.xml file

### 10.396.2 Constructor & Destructor Documentation

#### 10.396.2.1 XMLDictReader()

```
gdcm::XMLDictReader::XMLDictReader ( )
```

### 10.396.2.2 ~XMLDictReader()

```
gdcm::XMLDictReader::~~XMLDictReader ( ) [inline]
```

## 10.396.3 Member Function Documentation

### 10.396.3.1 CharacterDataHandler()

```
void gdcm::XMLDictReader::CharacterDataHandler (
    const char * data,
    int length ) [virtual]
```

Reimplemented from [gdcm::TableReader](#).

### 10.396.3.2 EndElement()

```
void gdcm::XMLDictReader::EndElement (
    const char * name ) [virtual]
```

Reimplemented from [gdcm::TableReader](#).

### 10.396.3.3 GetDict()

```
const Dict& gdcm::XMLDictReader::GetDict ( ) [inline]
```

### 10.396.3.4 HandleDescription()

```
void gdcm::XMLDictReader::HandleDescription (
    const char ** atts ) [protected]
```

### 10.396.3.5 HandleEntry()

```
void gdcm::XMLDictReader::HandleEntry (
    const char ** atts ) [protected]
```

### 10.396.3.6 StartElement()

```
void gdcm::XMLDictReader::StartElement (
    const char * name,
    const char ** atts ) [virtual]
```

Reimplemented from [gdcm::TableReader](#).

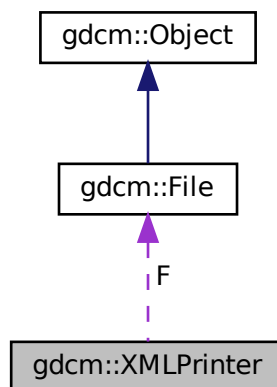
The documentation for this class was generated from the following file:

- [gdcmXMLDictReader.h](#)

## 10.397 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](#)

### 10.397.1 Member Enumeration Documentation

#### 10.397.1.1 [PrintStyles](#)

enum [gdcm::XMLPrinter::PrintStyles](#)

Enumerator

<a href="#">OnlyUUID</a>	
<a href="#">LOADBULKDATA</a>	

## 10.397.2 Constructor & Destructor Documentation

### 10.397.2.1 XMLPrinter()

```
gdcm::XMLPrinter::XMLPrinter ( )
```

### 10.397.2.2 ~XMLPrinter()

```
virtual gdcm::XMLPrinter::~~XMLPrinter ( ) [virtual]
```

## 10.397.3 Member Function Documentation

### 10.397.3.1 GetPrintStyle()

```
PrintStyles gdcm::XMLPrinter::GetPrintStyle ( ) const [inline]
```

### 10.397.3.2 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

### 10.397.3.3 Print()

```
void gdcm::XMLPrinter::Print (
    std::ostream & os )
```

#### 10.397.3.4 PrintDataElement()

```
VR gdcM::XMLPrinter::PrintDataElement (
    std::ostream & os,
    const Dicts & dicts,
    const DataSet & ds,
    const DataElement & de,
    const TransferSyntax & ts ) [protected]
```

#### 10.397.3.5 PrintDataSet()

```
void gdcM::XMLPrinter::PrintDataSet (
    const DataSet & ds,
    const TransferSyntax & ts,
    std::ostream & os )
```

#### 10.397.3.6 PrintSQ()

```
void gdcM::XMLPrinter::PrintSQ (
    const SequenceOfItems * sqi,
    const TransferSyntax & ts,
    std::ostream & os ) [protected]
```

#### 10.397.3.7 SetFile()

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

#### 10.397.3.8 SetStyle()

```
void gdcM::XMLPrinter::SetStyle (
    PrintStyles ps ) [inline]
```

### 10.397.4 Member Data Documentation

### 10.397.4.1 F

```
const File* gdcm::XMLPrinter::F [protected]
```

### 10.397.4.2 PrintStyle

```
PrintStyles gdcm::XMLPrinter::PrintStyle [protected]
```

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

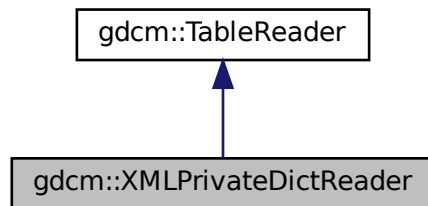
- [gdcmXMLPrinter.h](#)

## 10.398 gdcm::XMLPrivateDictReader Class Reference

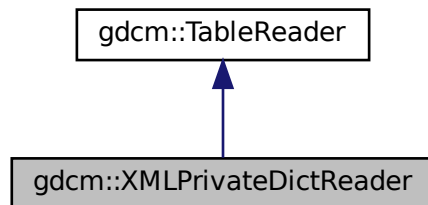
Class for representing a [XMLPrivateDictReader](#).

```
#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)

## Protected Member Functions

- void [HandleDescription](#) (const char \*\*atts)
- void [HandleEntry](#) (const char \*\*atts)

### 10.398.1 Detailed Description

Class for representing a [XMLPrivateDictReader](#).

#### Note

bla Will read the Private.xml file

### 10.398.2 Constructor & Destructor Documentation

#### 10.398.2.1 XMLPrivateDictReader()

```
gdcm::XMLPrivateDictReader::XMLPrivateDictReader ( )
```

#### 10.398.2.2 ~XMLPrivateDictReader()

```
gdcm::XMLPrivateDictReader::~XMLPrivateDictReader ( ) [inline]
```

### 10.398.3 Member Function Documentation



### 10.398.3.1 CharacterDataHandler()

```
void gdcm::XMLPrivateDictReader::CharacterDataHandler (
    const char * data,
    int length ) [virtual]
```

Reimplemented from [gdcm::TableReader](#).

### 10.398.3.2 EndElement()

```
void gdcm::XMLPrivateDictReader::EndElement (
    const char * name ) [virtual]
```

Reimplemented from [gdcm::TableReader](#).

### 10.398.3.3 GetPrivateDict()

```
const PrivateDict& gdcm::XMLPrivateDictReader::GetPrivateDict ( ) [inline]
```

### 10.398.3.4 HandleDescription()

```
void gdcm::XMLPrivateDictReader::HandleDescription (
    const char ** atts ) [protected]
```

### 10.398.3.5 HandleEntry()

```
void gdcm::XMLPrivateDictReader::HandleEntry (
    const char ** atts ) [protected]
```

### 10.398.3.6 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](#)

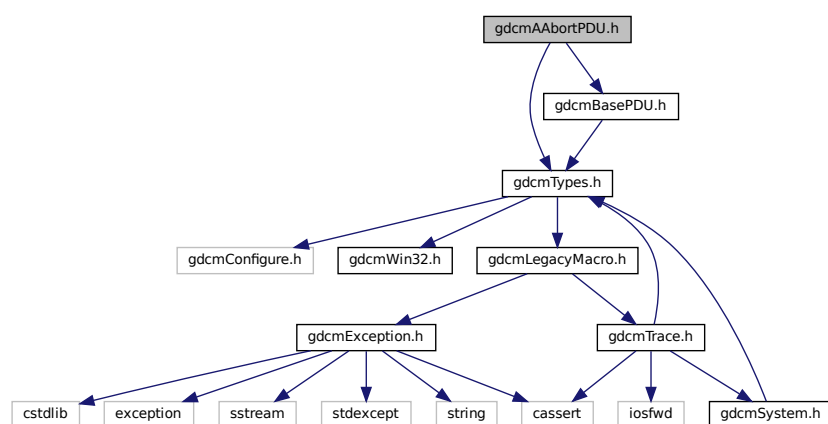


## Chapter 11

# File Documentation

### 11.1 gdcmAAbortPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
Include dependency graph for gdcmAAbortPDU.h:
```



### Classes

- class `gdcm::network::AAabortPDU`  
*AAabortPDU.*

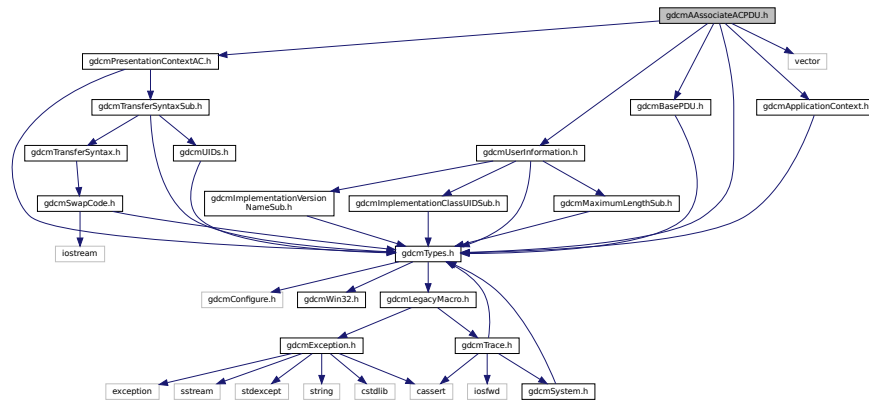
### Namespaces

- `gdcm`
- `gdcm::network`

## 11.2 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.*

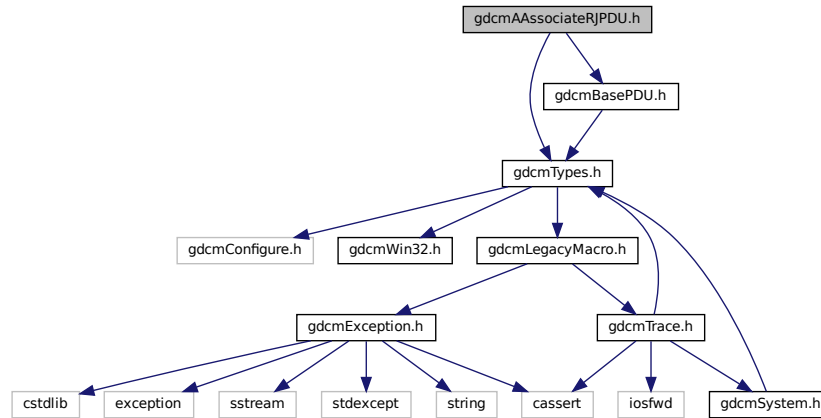
### Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.3 gdcmAAssociateRJPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAAssociateRJPDU.h:



## Classes

- class [gdcm::network::AAssociateRJPDU](#)  
*AAssociateRJPDU.*

## Namespaces

- [gdcm](#)
- [gdcm::network](#)

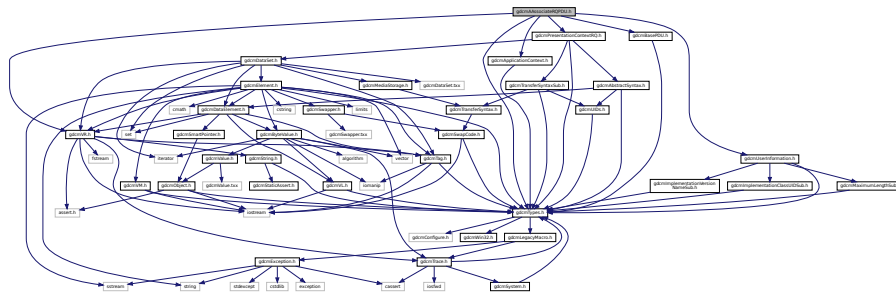
## 11.4 gdcmAAssociateRQPDU.h File Reference

```

#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 [gdcm::network::AAssociateRQPDU](#)  
*AAssociateRQPDU.*

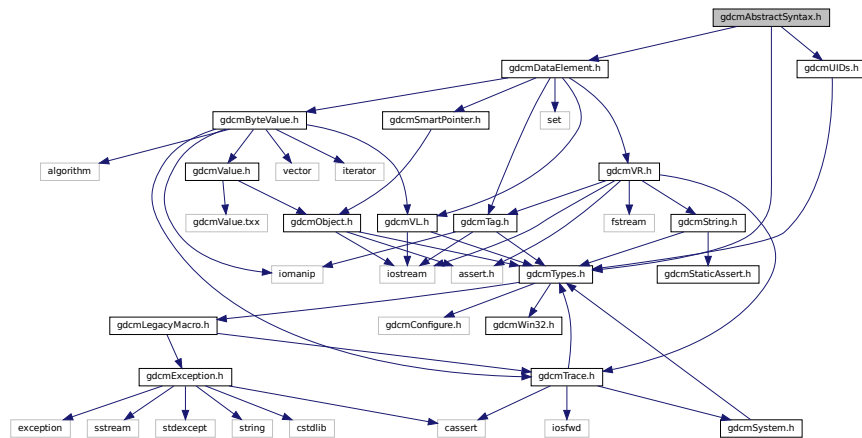
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

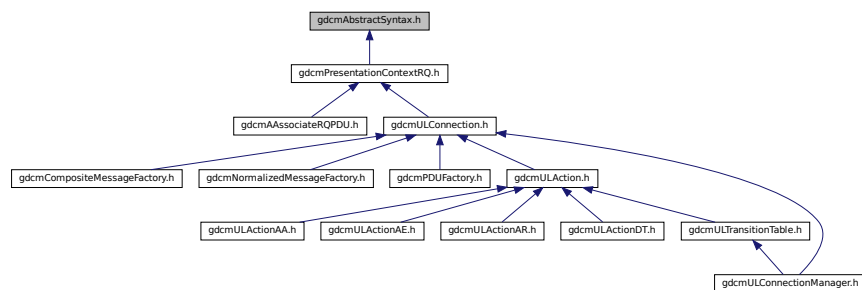
## 11.5 gdcmAbstractSyntax.h File Reference

```
#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 [gdcm::network::AbstractSyntax](#)  
*AbstractSyntax.*

## Namespaces

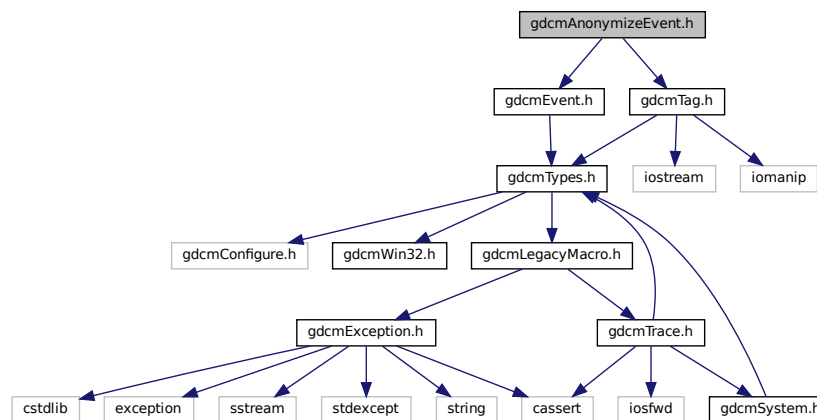
- [gdcm](#)
- [gdcm::network](#)

## 11.6 gdcmAnonymizeEvent.h File Reference

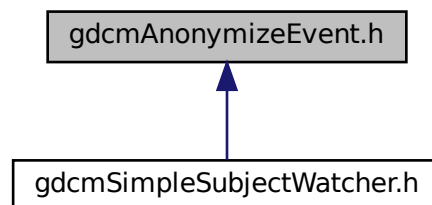
```
#include "gdcmEvent.h"
```

```
#include "gdcmTag.h"
```

Include dependency graph for gdcmAnonymizeEvent.h:



This graph shows which files directly or indirectly include this file:



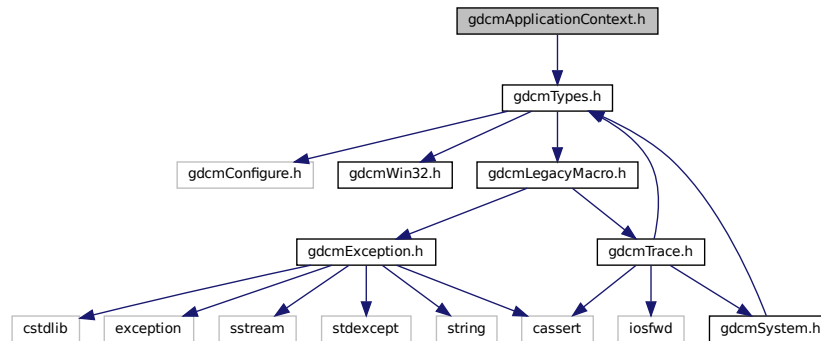




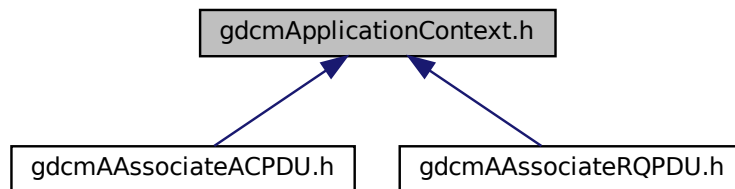
## 11.8 gdcmApplicationContext.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmApplicationContext.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class `gdcm::network::ApplicationContext`  
*ApplicationContext.*

### Namespaces

- `gdcm`
- `gdcm::network`

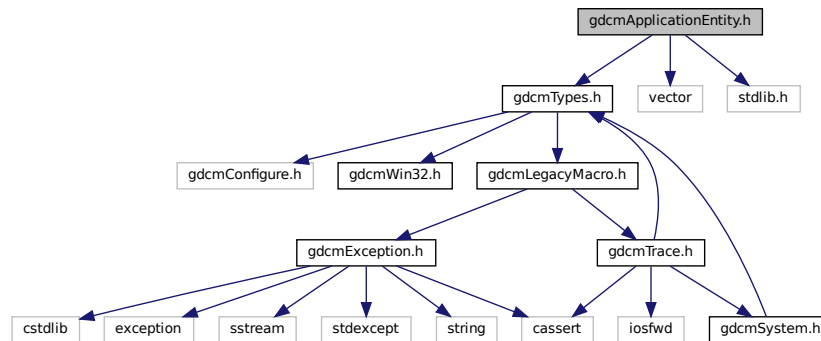
## 11.9 gdcmApplicationEntity.h File Reference

```
#include "gdcmTypes.h"
```

```
#include <vector>
```

```
#include <stdlib.h>
```

Include dependency graph for gdcmApplicationEntity.h:



### Classes

- class `gdcm::ApplicationEntity`  
*ApplicationEntity.*

### Namespaces

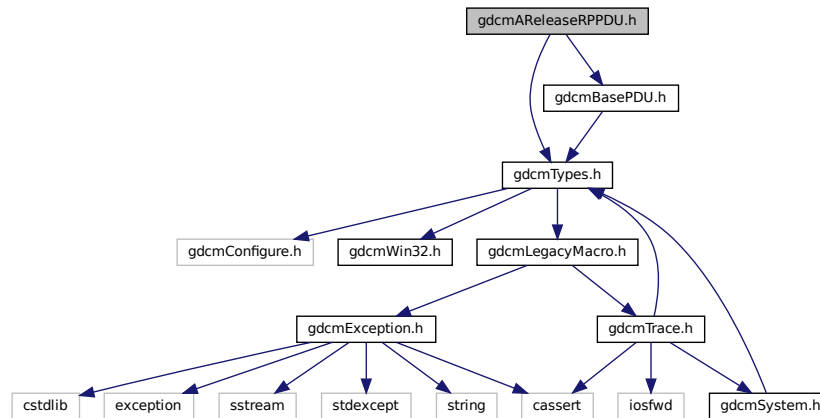
- `gdcm`

## 11.10 gdcmAResponseRPPDU.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAReleaseRPPDU.h:



## Classes

- class `gdcm::network::AReleaseRPPDU`  
*AReleaseRPPDU*.

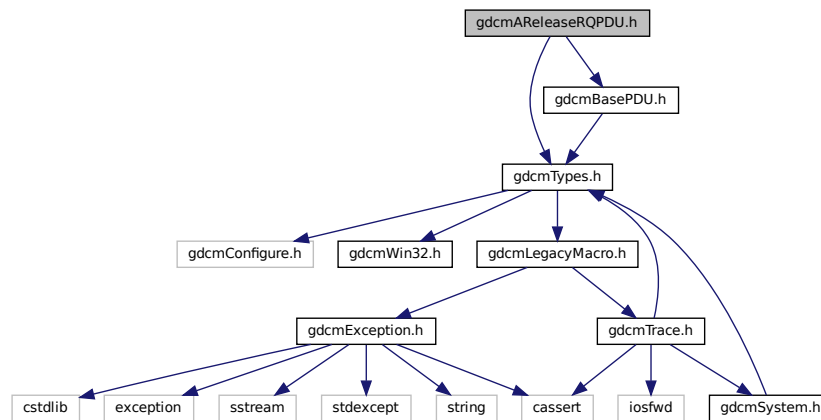
## Namespaces

- `gdcm`
- `gdcm::network`

## 11.11 gdcmAReleaseRPPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for `gdcmAReleaseRQPDU.h`:



## Classes

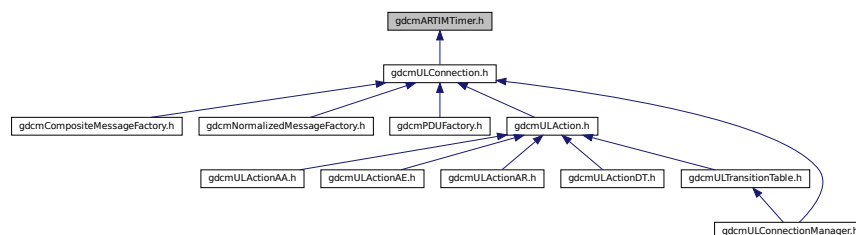
- class [gdcm::network::AReleaseRQPDU](#)  
*AReleaseRQPDU*.

## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.12 gdcmARTIMTimer.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::network::ARTIMTimer](#)  
*ARTIMTimer.*

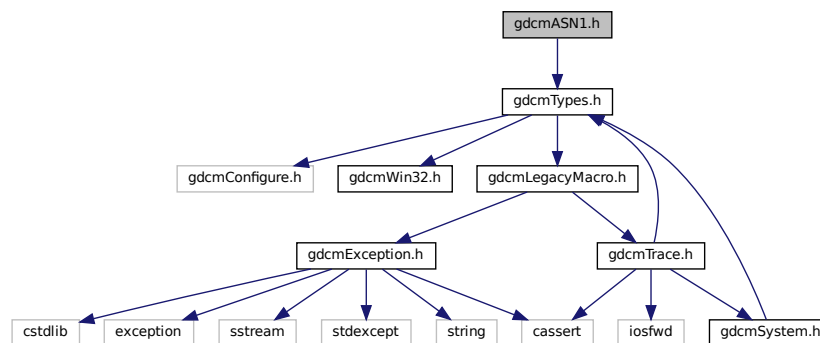
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.13 gdcmASN1.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmASN1.h:



## Classes

- class [gdcm::ASN1](#)  
*Class for ASN1.*

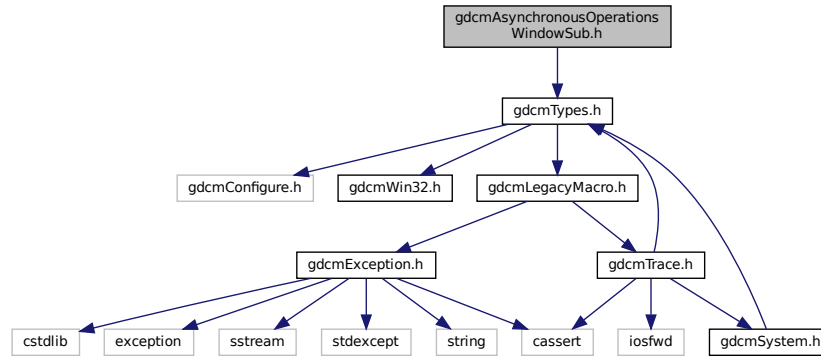
## Namespaces

- [gdcm](#)

## 11.14 gdcmAsynchronousOperationsWindowSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmAsynchronousOperationsWindowSub.h:



### Classes

- class [gdcm::network::AsynchronousOperationsWindowSub](#)  
*AsynchronousOperationsWindowSub.*

### Namespaces

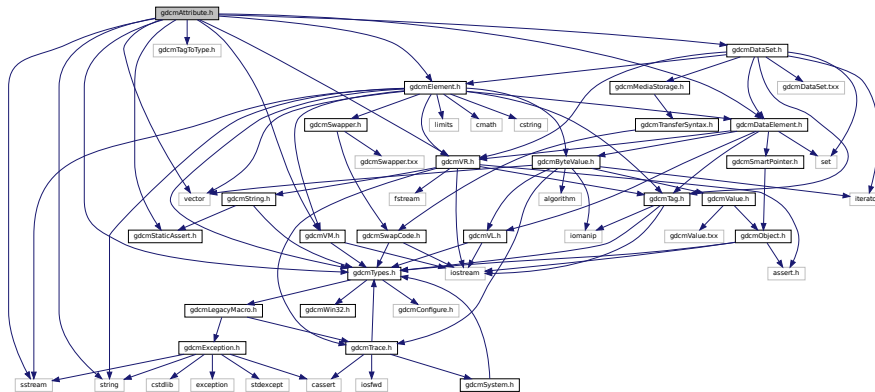
- [gdcm](#)
- [gdcm::network](#)

## 11.15 gdcmAttribute.h File Reference

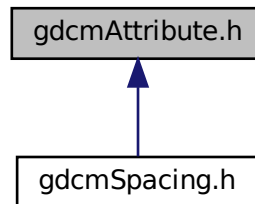
```
#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmTagToType.h"
#include "gdcmVM.h"
#include "gdcmElement.h"
#include "gdcmDataElement.h"
#include "gdcmDataSet.h"
#include "gdcmStaticAssert.h"
#include <string>
#include <vector>
```

```
#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 instanciation 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 >](#)

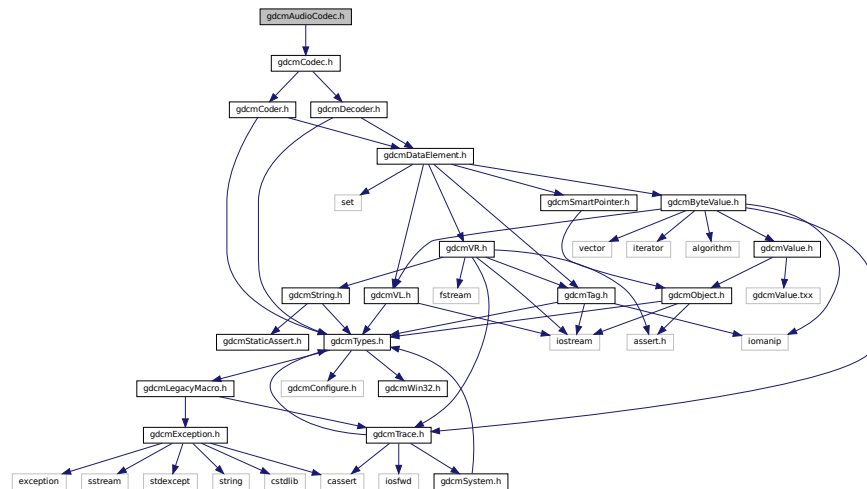
## Namespaces

- [gdcm](#)

## 11.16 gdcmAudioCodec.h File Reference

```
#include "gdcmCodec.h"
```

Include dependency graph for gdcmAudioCodec.h:



## Classes

- class [gdcm::AudioCodec](#)  
*AudioCodec.*

## Namespaces

- [gdcm](#)







## Classes

- class [gdcmbasePDU.h](#)  
[BaseNormalizedMessage](#).

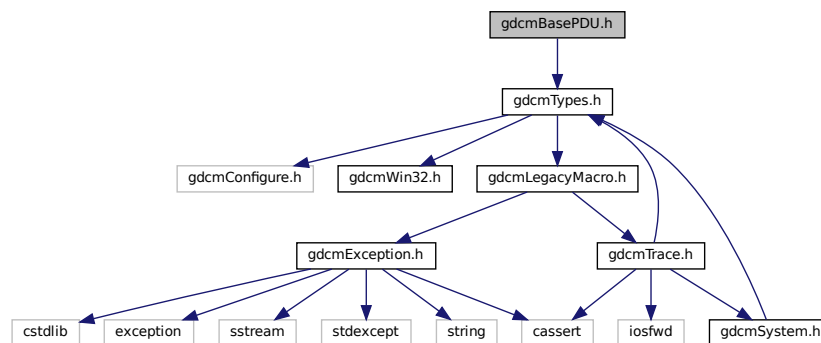
## Namespaces

- [gdcmbasePDU.h](#)
- [gdcmbasePDU.h](#)

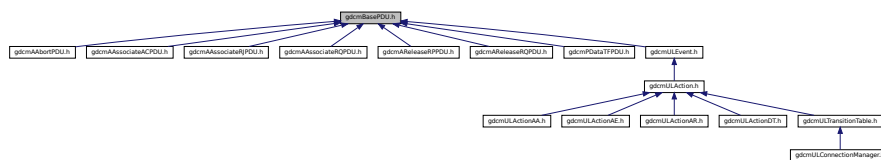
## 11.20 gdcmbasePDU.h File Reference

```
#include "gdcmbasePDU.h"
```

Include dependency graph for gdcmbasePDU.h:



This graph shows which files directly or indirectly include this file:



## Classes

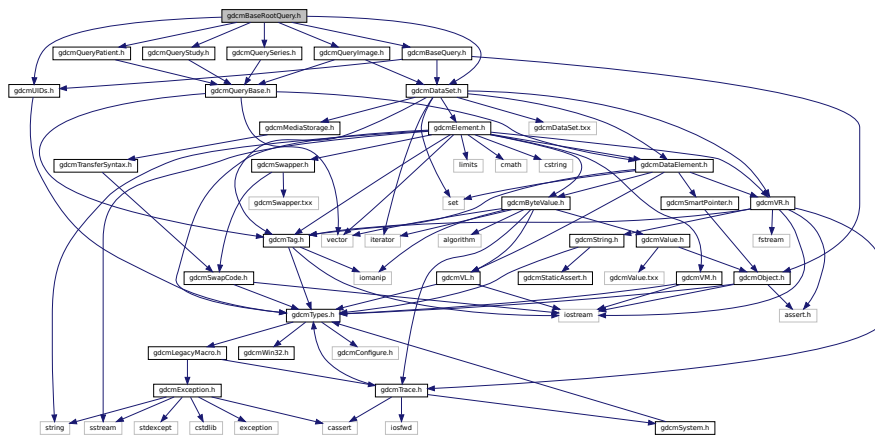
- class [gdcmbasePDU.h](#)  
[BasePDU](#).



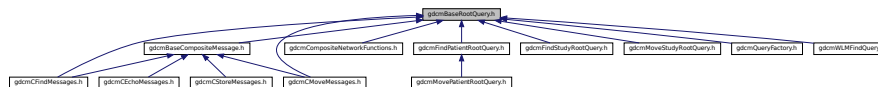
## 11.22 gdcmBaseRootQuery.h File Reference

```
#include "gdcmDataSet.h"
#include "gdcmUIDs.h"
#include "gdcmBaseQuery.h"
#include "gdcmQueryPatient.h"
#include "gdcmQueryStudy.h"
#include "gdcmQuerySeries.h"
#include "gdcmQueryImage.h"
```

Include dependency graph for gdcmbaseRootQuery.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::BaseRootQuery`  
*BaseRootQuery*.

## Namespaces

- **gdcm**

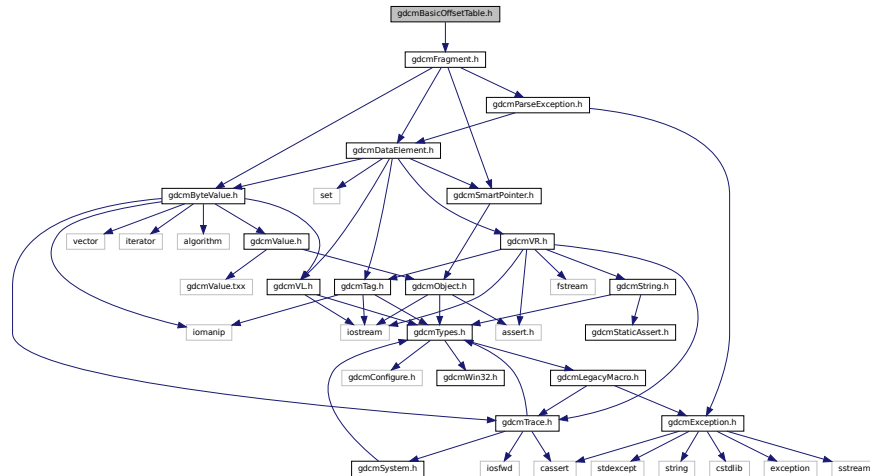
## Enumerations

- 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` }

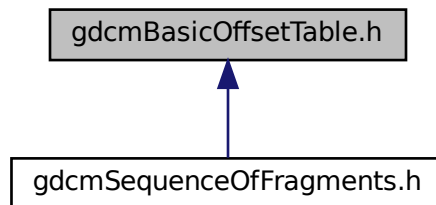
## 11.23 gdcmBasicOffsetTable.h File Reference

```
#include "gdcmFragment.h"
```

Include dependency graph for `gdcmBasicOffsetTable.h`:



This graph shows which files directly or indirectly include this file:



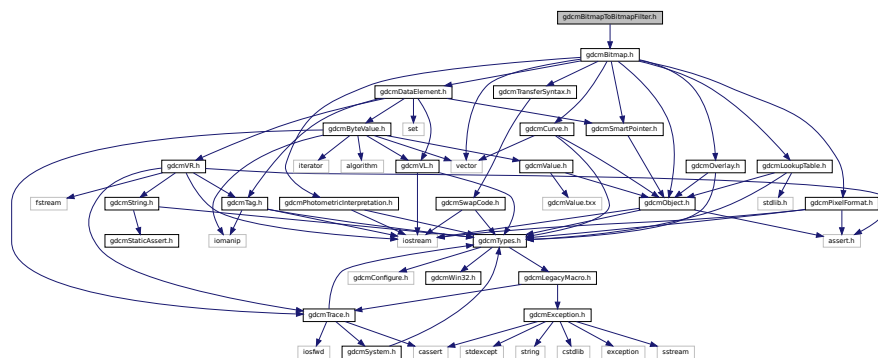


- class `gdcm::Bitmap`  
*Bitmap* class.

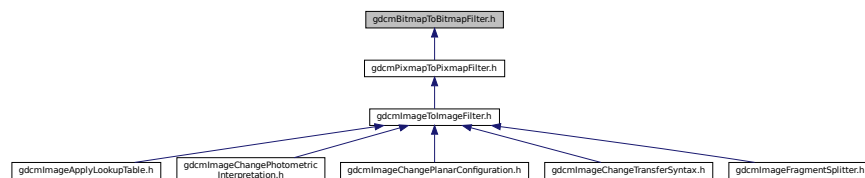
- **gdcm**

```
#include "gdcmBitmap.h"
```

Include dependency graph for `gdcmBitmapToBitmapFilter.h`:



This graph shows which files directly or indirectly include this file:



- class `gdcm::BitmapToBitmapFilter`  
*BitmapToBitmapFilter* class.



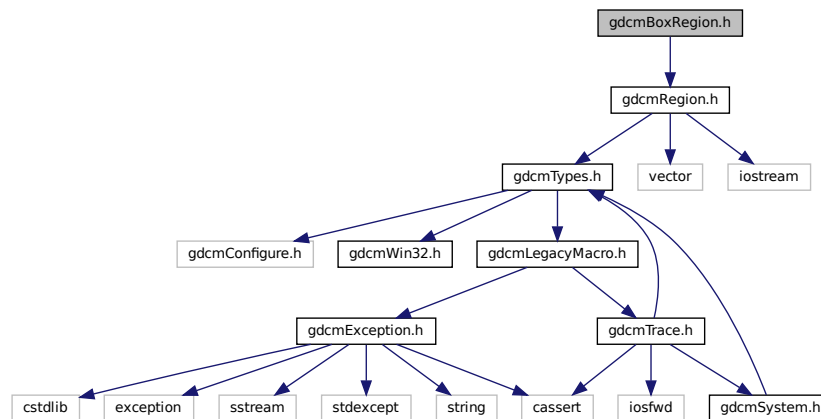
## Namespaces

- [gdcm](#)

## 11.26 gdcmBoxRegion.h File Reference

```
#include "gdcmRegion.h"
```

Include dependency graph for gdcmBoxRegion.h:



## Classes

- class [gdcm::BoxRegion](#)  
Class for manipulation box region.

## Namespaces

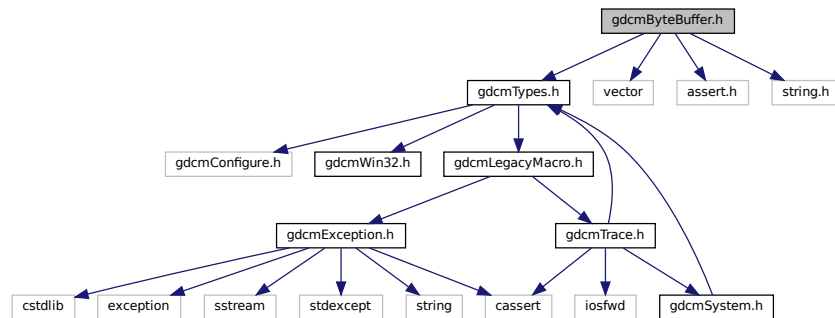
- [gdcm](#)

## 11.27 gdcmByteBuffer.h File Reference

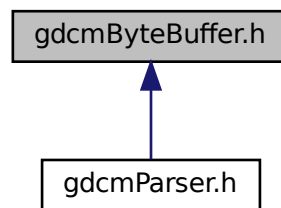
```
#include "gdcmTypes.h"
#include <vector>
#include <assert.h>
```

```
#include <string.h>
```

Include dependency graph for `gdcmByteBuffer.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::ByteBuffer`  
*ByteBuffer.*

## Namespaces

- `gdcm`



## Classes

- class [gdcm::ByteSwapFilter](#)  
[ByteSwapFilter](#).

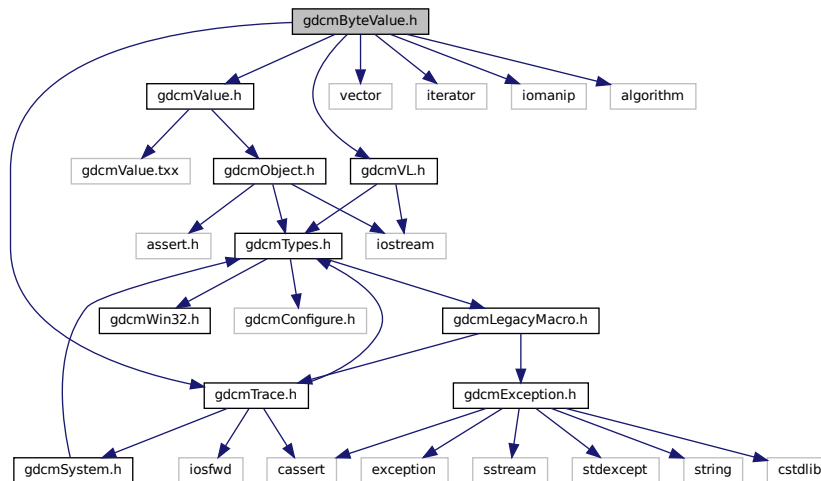
## Namespaces

- [gdcm](#)

## 11.30 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)*

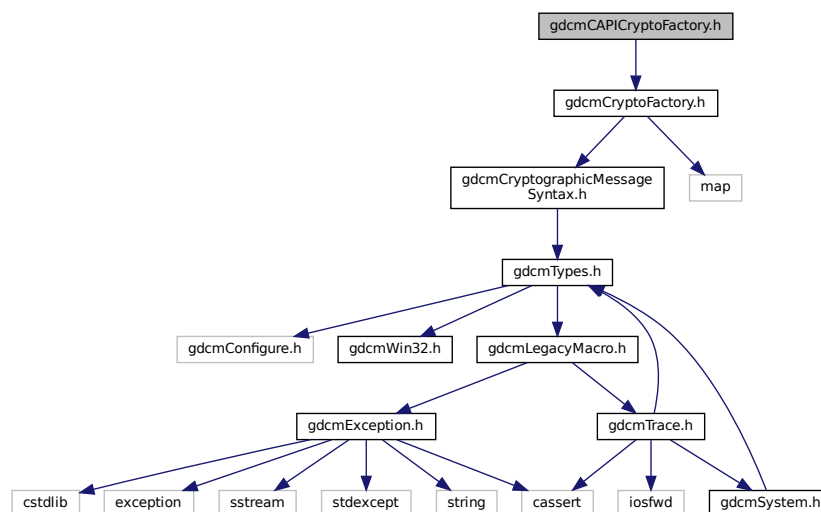
## Namespaces

- [gdcm](#)

## 11.31 gdcmCAPICryptoFactory.h File Reference

```
#include "gdcmCryptoFactory.h"
```

Include dependency graph for gdcmCAPICryptoFactory.h:



## Classes

- class [gdcm::CAPICryptoFactory](#)

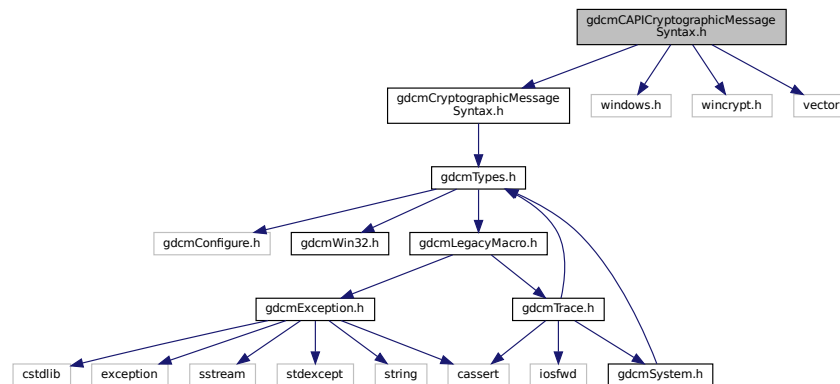
## Namespaces

- [gdcm](#)

## 11.32 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

- [gdcm](#)

## 11.33 gdcmCEchoMessages.h File Reference

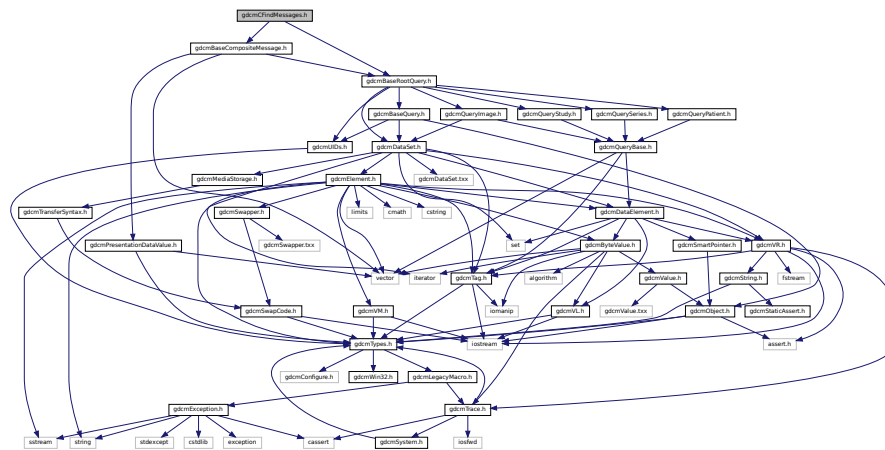
```
#include "gdcmBaseCompositeMessage.h"
```

[illegible]

- class `gdcm::network::CEchoRQ`  
`CEchoRQ`.
- class `gdcm::network::CEchoRSP`  
`CEchoRSP` *this file defines the messages for the cecho action.*

- gdc
- gdc::network

```
#include "gdcmBaseCompositeMessage.h"
#include "gdcmBaseRootQuery.h"
```



- 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.

- gdc
- gdc::network

```
#include "gdcmBaseCompositeMessage.h"
#include "gdcmBaseRootQuery.h"
```



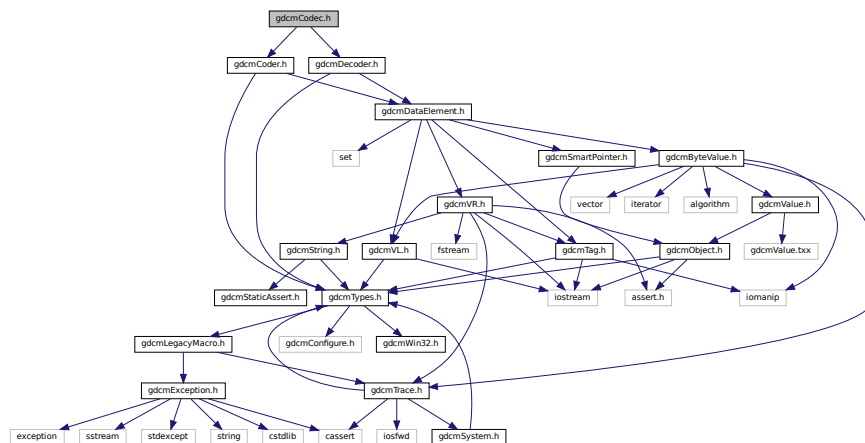
[illegible]

- class `gdcm::network::CMoveCancelRq`
- class `gdcm::network::CMoveRQ`  
`CMoveRQ`.
- class `gdcm::network::CMoveRSP`  
`CMoveRSP` this file defines the messages for the cmove action.

- `gdc`
- `gdc::network`

```
#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.*

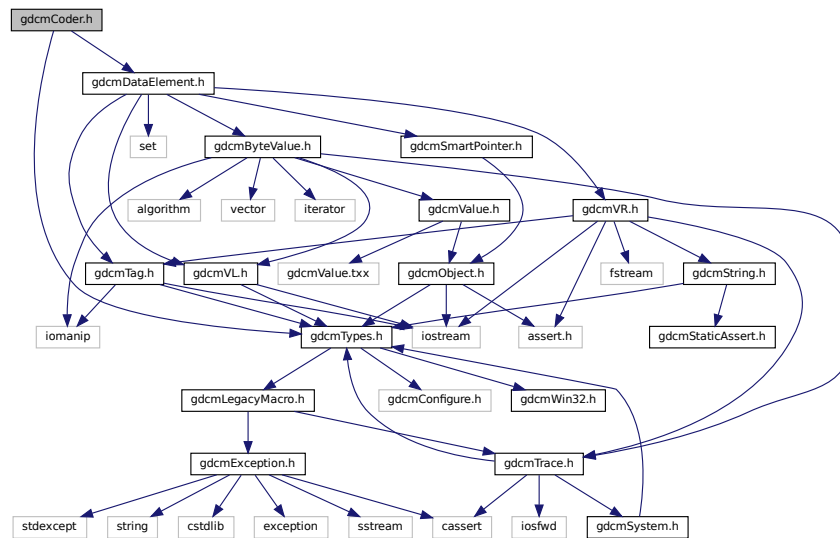
## Namespaces

- `gdcm`

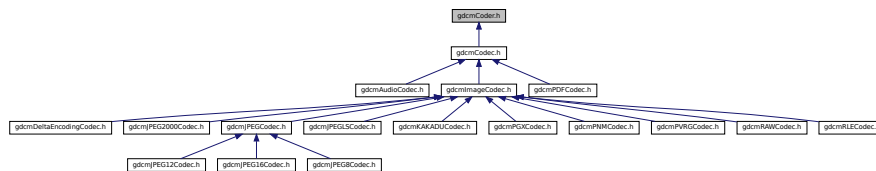
## 11.37 gdcmCoder.h File Reference

```
#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.*

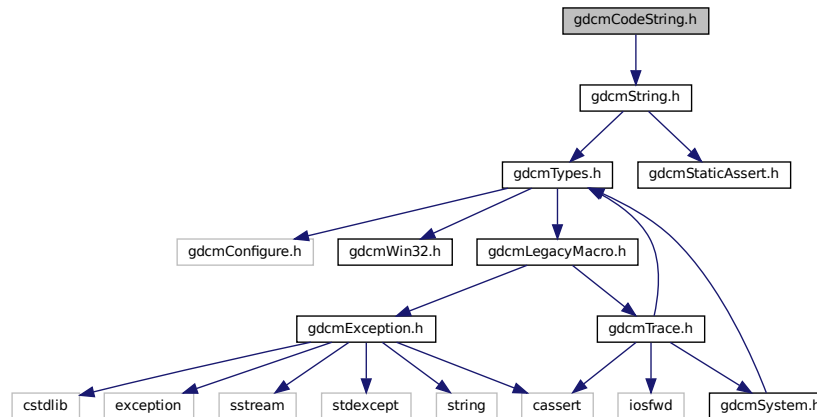
## Namespaces

- [gdcm](#)

## 11.38 gdcmCodeString.h File Reference

```
#include "gdcmString.h"
```

Include dependency graph for gdcmCodeString.h:



### Classes

- class `gdcm::CodeString`  
*CodeString.*

### Namespaces

- `gdcm`

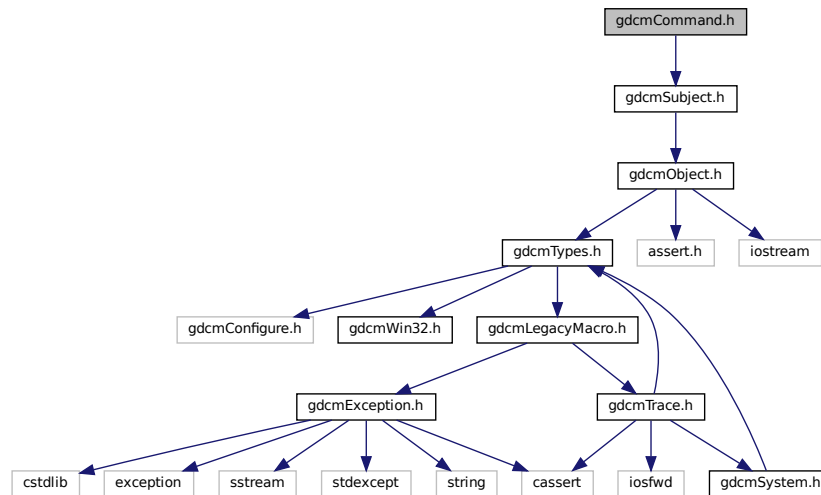
### Functions

- bool `gdcm::operator!=` (const CodeString &ref, const CodeString &cs)
- std::ostream & `gdcm::operator<<` (std::ostream &os, const CodeString &str)
- bool `gdcm::operator==` (const CodeString &ref, const CodeString &cs)

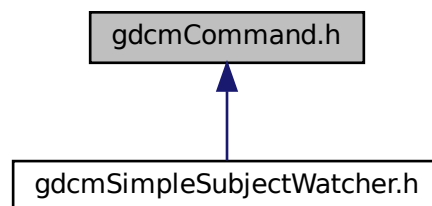
## 11.39 gdcmCommand.h File Reference

```
#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.*



[illegible]

## Classes

- class [gdcm::CompositeNetworkFunctions](#)  
*Composite Network Functions.*

## Namespaces

- [gdcm](#)

## 11.43 gdcmConstCharWrapper.h File Reference

### Classes

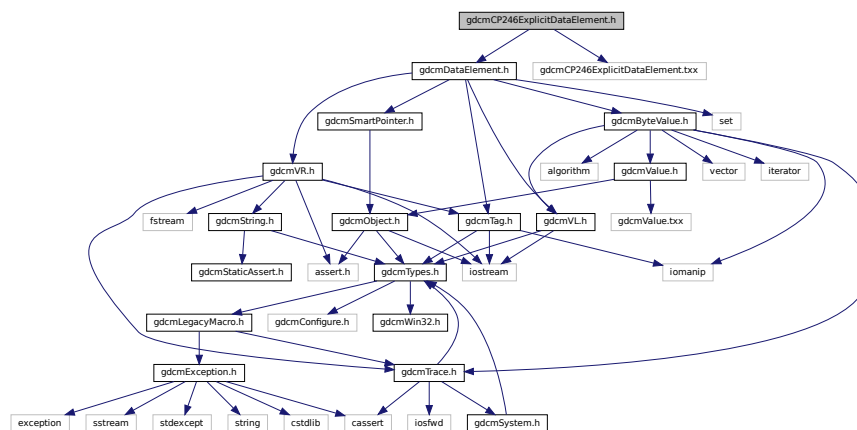
- class [gdcm::ConstCharWrapper](#)  
*Do not use me.*

### Namespaces

- [gdcm](#)

## 11.44 gdcmCP246ExplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmCP246ExplicitDataElement.txx"
Include dependency graph for gdcmCP246ExplicitDataElement.h:
```





## Classes

- class [gdcm::CP246ExplicitDataElement](#)  
Class to read/write a [DataElement](#) as CP246Explicit Data [Element](#).

## Namespaces

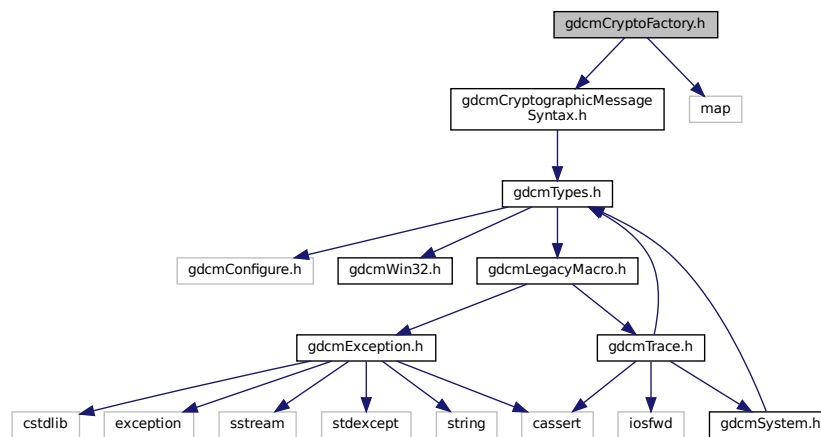
- [gdcm](#)

## 11.45 gdcmCryptoFactory.h File Reference

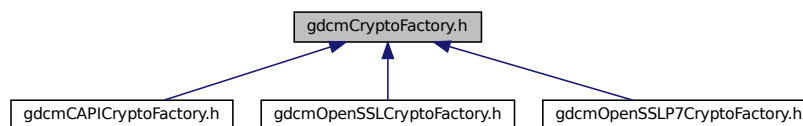
```
#include "gdcmCryptographicMessageSyntax.h"
```

```
#include <map>
```

Include dependency graph for gdcmCryptoFactory.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::CryptoFactory](#)  
Class to do handle the crypto factory.

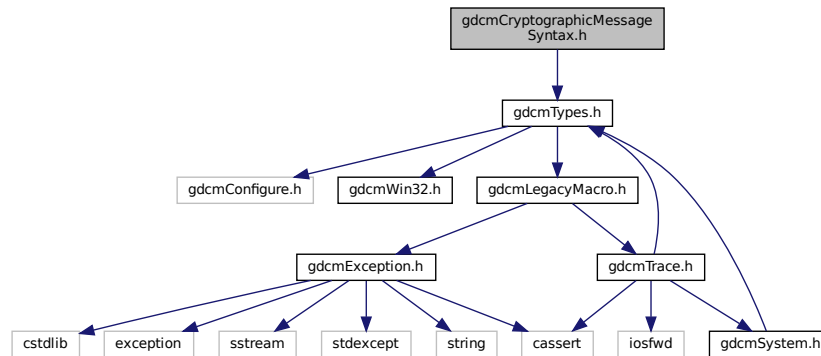
## Namespaces

- [gdcm](#)

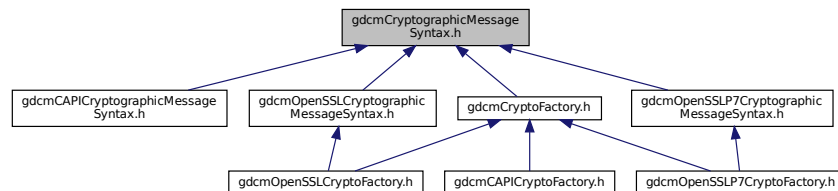
## 11.46 gdcmCryptographicMessageSyntax.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmCryptographicMessageSyntax.h:



This graph shows which files directly or indirectly include this file:



## Classes

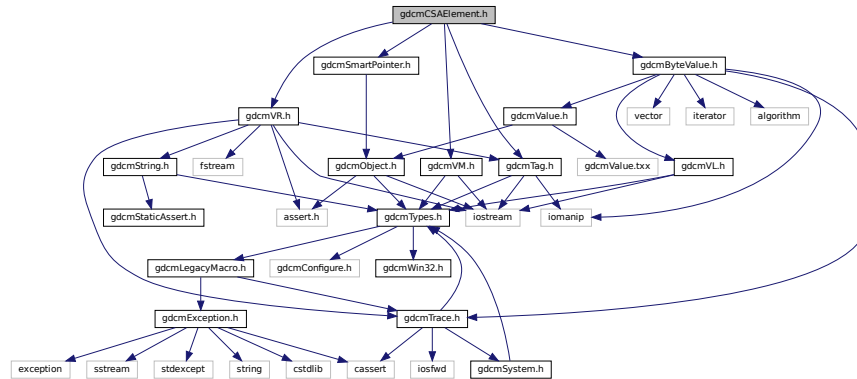
- class [gdcm::CryptographicMessageSyntax](#)

## Namespaces

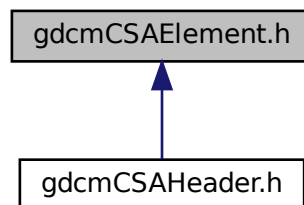
- [gdcm](#)

## 11.47 gdcmCSAElement.h File Reference

```
#include "gdcmTag.h"
#include "gdcmVM.h"
#include "gdcmVR.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"
Include dependency graph for gdcmCSAElement.h:
```



This graph shows which files directly or indirectly include this file:



### Classes

- class [gdcm::CSAElement](#)  
Class to represent a CSA *Element*.

### Namespaces

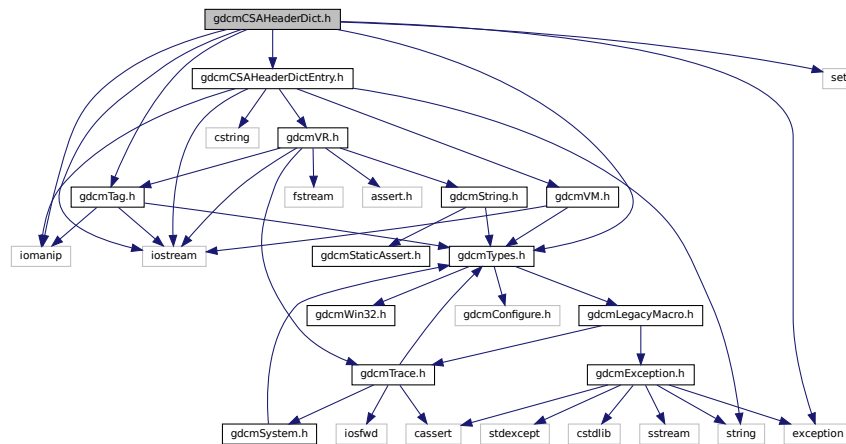
- [gdcm](#)



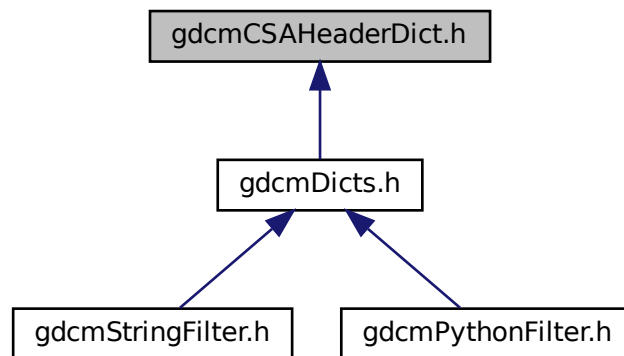
## 11.49 gdcmCSAHeaderDict.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmCSAHeaderDictEntry.h"
#include <iostream>
#include <iomanip>
#include <set>
#include <exception>
```

Include dependency graph for gdcmCSAHeaderDict.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::CSAHeaderDict](#)  
Class to represent a map of [CSAHeaderDictEntry](#).
- class [gdcm::CSAHeaderDictException](#)

## Namespaces

- [gdcm](#)

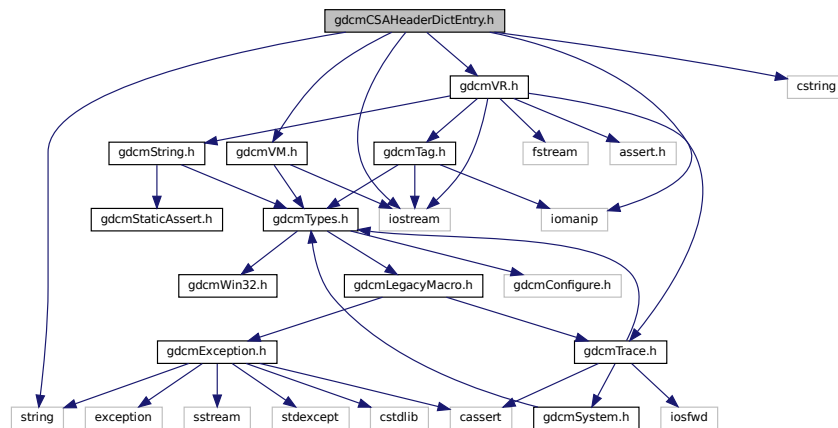
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeaderDict &val)`

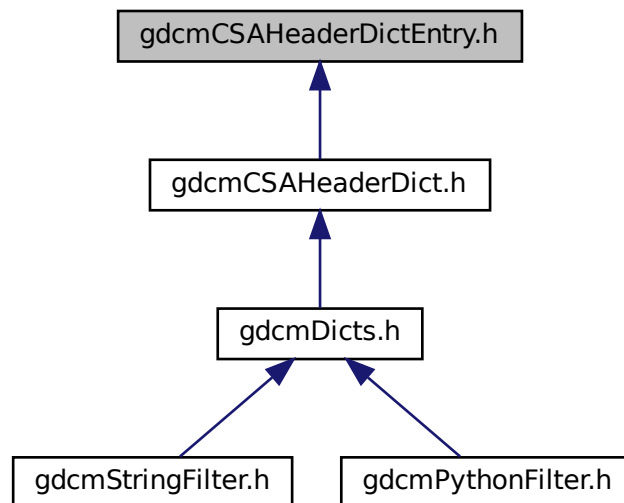
## 11.50 gdcmCSAHeaderDictEntry.h File Reference

```
#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 [gdcm::CSAHeaderDictEntry](#)  
*Class to represent an Entry in the [Dict](#).*

## Namespaces

- [gdcm](#)

## Functions

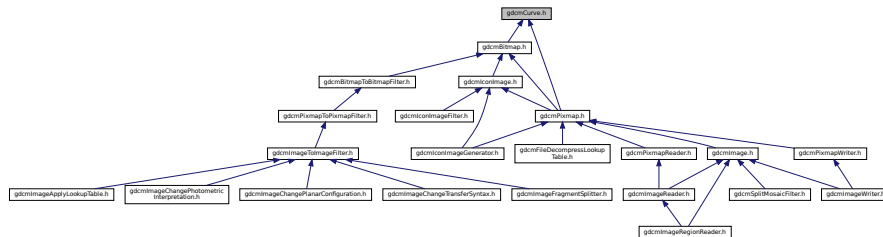
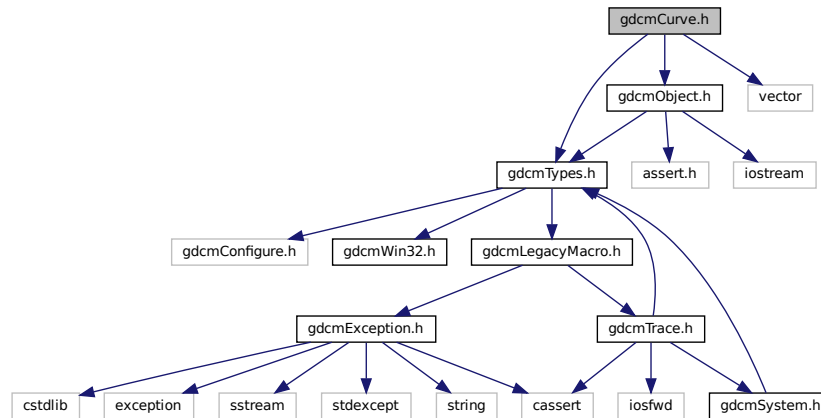
- `std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeaderDictEntry &val)`

## 11.51 gdcmCStoreMessages.h File Reference

```
#include "gdcmBaseCompositeMessage.h"
```

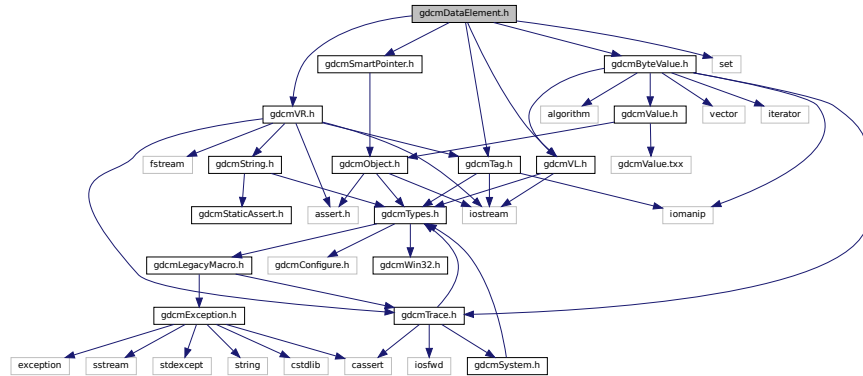




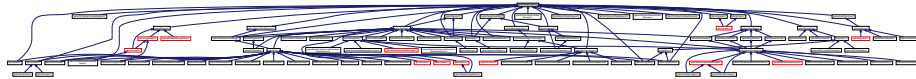


---

```
#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

- [gdcm](#)

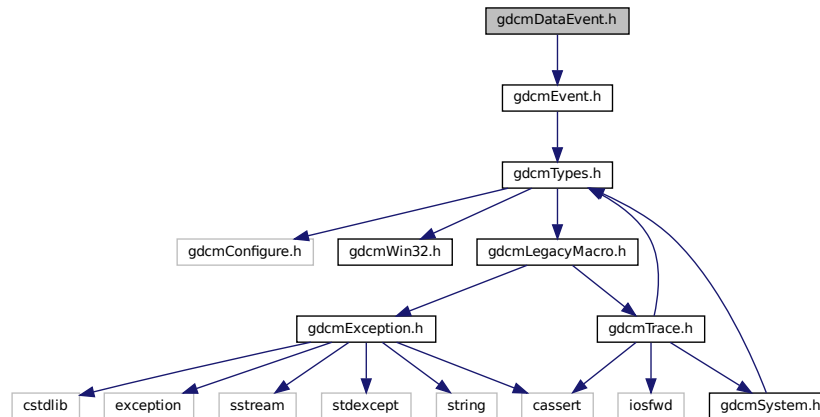
## Functions

- bool [gdcm::operator!=](#) (const DataElement &lhs, const DataElement &rhs)
- std::ostream & [gdcm::operator<<](#) (std::ostream &os, const DataElement &val)

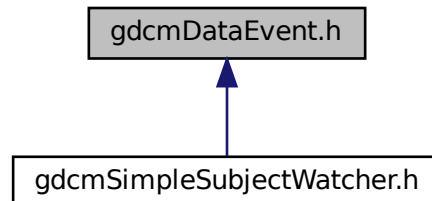
## 11.54 gdcmDataEvent.h File Reference

```
#include "gdcmEvent.h"
```

Include dependency graph for gdcmDataEvent.h:



This graph shows which files directly or indirectly include this file:



### Classes

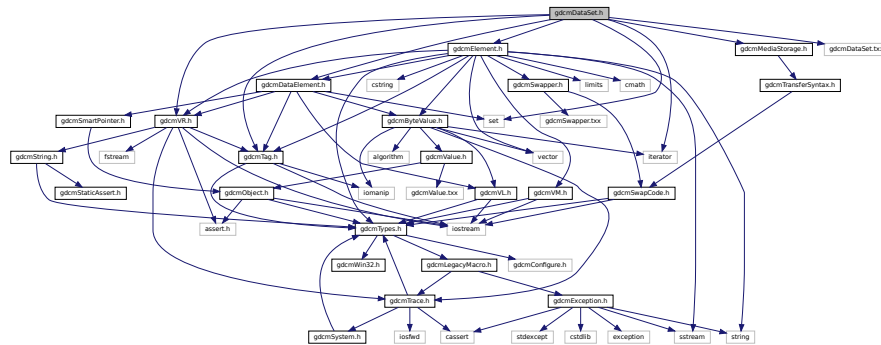
- class [gdcm::DataEvent](#)  
*DataEvent*.

### Namespaces

- [gdcm](#)

## 11.55 gdcmDataSet.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmTag.h"
#include "gdcmVR.h"
#include "gdcmElement.h"
#include "gdcmMediaStorage.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

- [gdcm](#)

### Functions

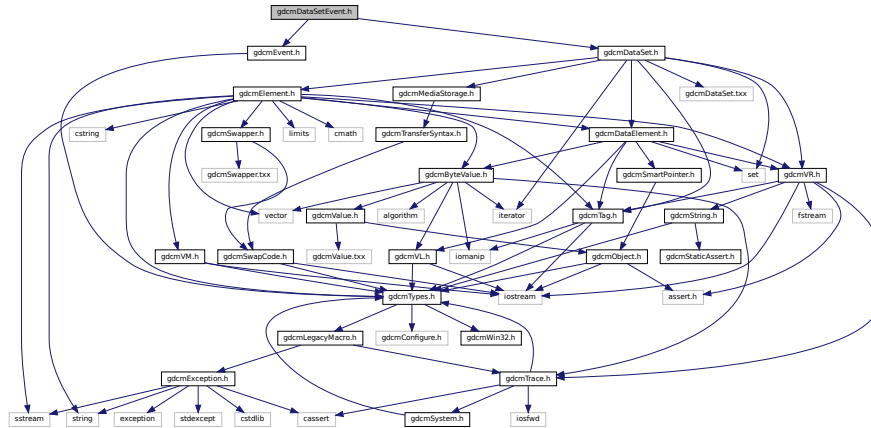
- `std::ostream & gdcm::operator<< (std::ostream &os, const DataSet &val)`

## 11.56 gdcmDataSetEvent.h File Reference

```
#include "gdcmEvent.h"
```

```
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmDataSetEvent.h:



### Classes

- class [gdcm::DataSetEvent](#)  
*DataSetEvent.*

### Namespaces

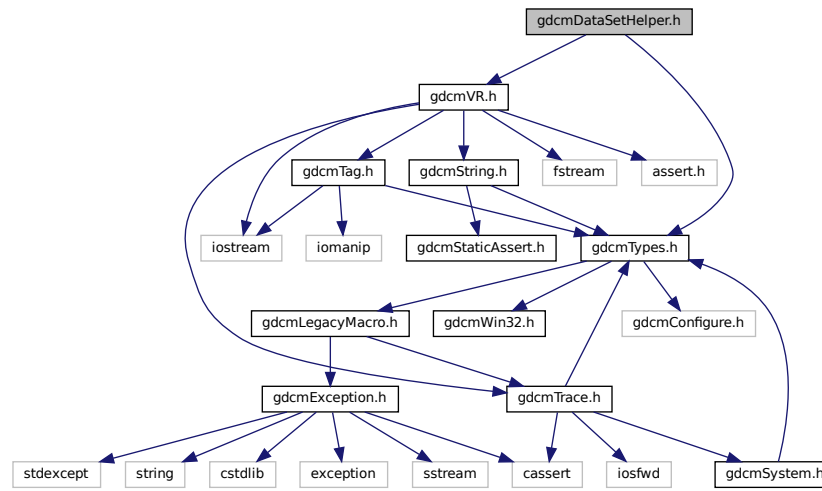
- [gdcm](#)

## 11.57 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

- [gdcm](#)

## 11.58 gdcmDecoder.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmDataElement.h"
```

[illegible][illegible]

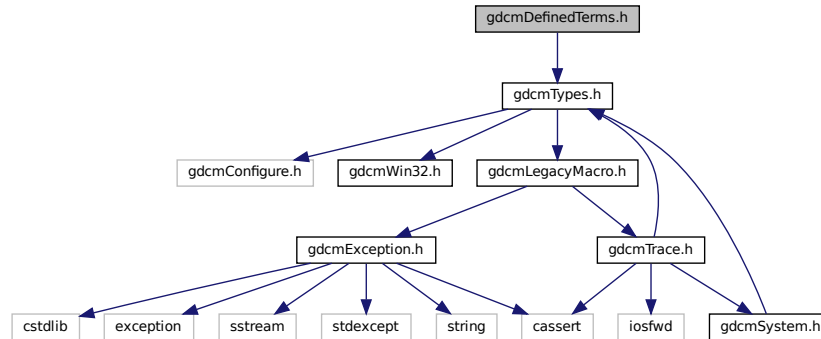
- class `gdc::Decoder`  
*Decoder*.

- **gdcm**

## 11.59 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

- [gdcm](#)

## 11.60 gdcmDeflateStream.h File Reference

```
#include "zipstreamimpl.h"
```

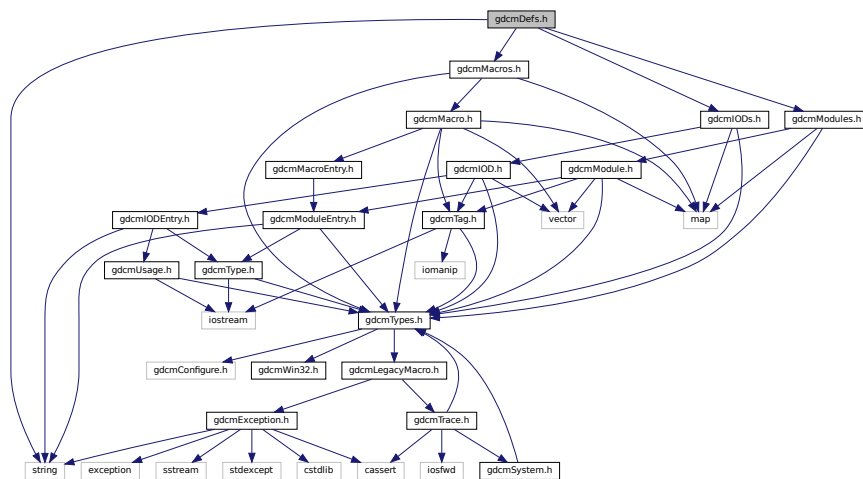


```
graph TD; A[gdcmDeflateStream.h] --> B[zipstreamimpl.h]
```

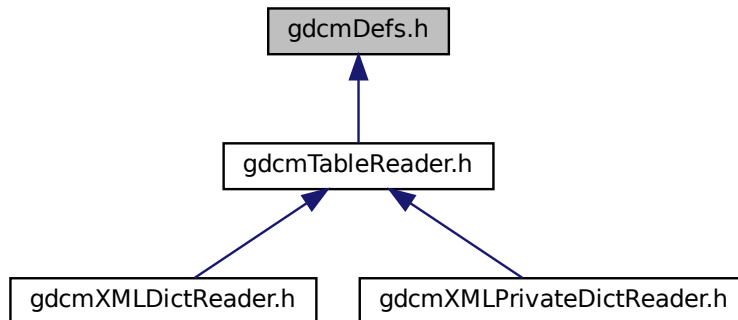
```
#include "gdcmModules.h"
#include "gdcmMacros.h"
#include "gdcmIODs.h"
#include <string>
```

Include dependency graph for gdcmDefs.h:

Include dependency graph for gdcMDefs.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcml::Defs](#)  
*FIXME I do not like the name 'Defs'.*

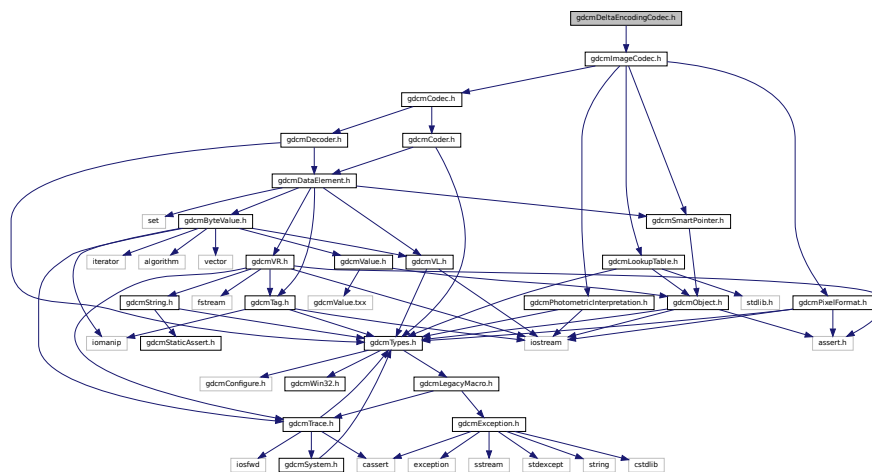
## Namespaces

- [gdcml](#)

## 11.62 gdcmlDeltaEncodingCodec.h File Reference

```
#include "gdcmlImageCodec.h"
```

Include dependency graph for `gdcmlDeltaEncodingCodec.h`:



## Classes

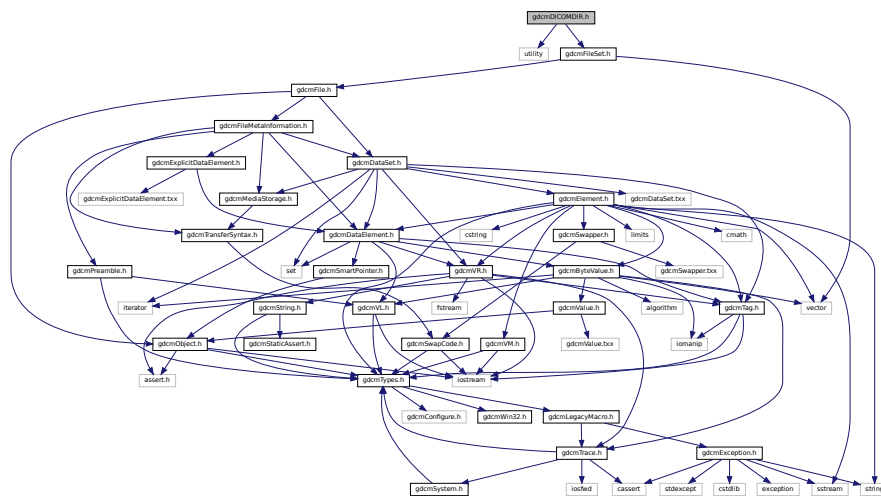
- class `gdcm::DeltaEncodingCodec`  
*DeltaEncodingCodec* compression used by some private vendor.

## Namespaces

- **gdcm**

## 11.63 gdcmdicomdir.h File Reference

```
#include <utility>
#include "gdcMFileSet.h"
Include dependency graph for gdcMDCOMDIR.h:
```



## Classes

- class `gdcm::DICOMDIR`  
*DICOMDIR* class.

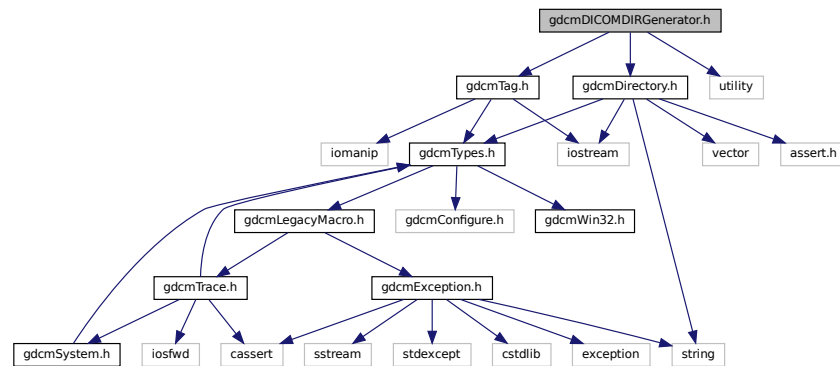
## Namespaces

- gdc

## 11.64 gdcmDICOMDIRGenerator.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmTag.h"
#include <utility>
```

Include dependency graph for gdcmDICOMDIRGenerator.h:



### Classes

- class [gdcm::DICOMDIRGenerator](#)  
*DICOMDIRGenerator* class.

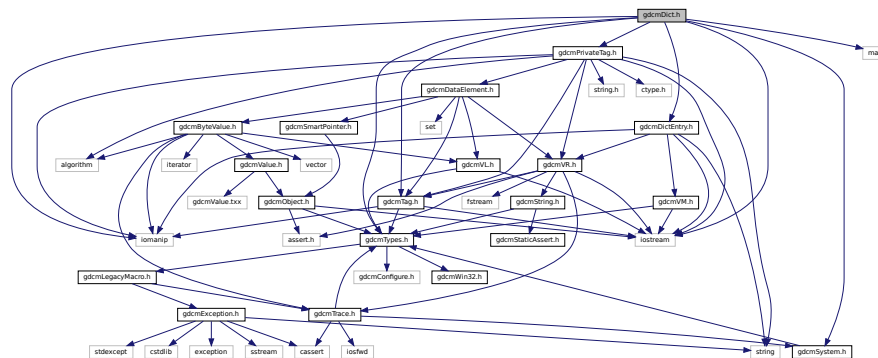
### Namespaces

- [gdcm](#)

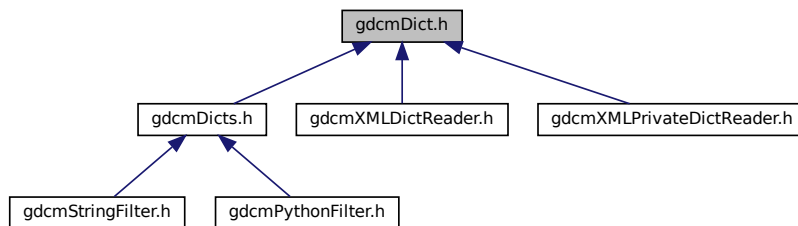
## 11.65 gdcmDict.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmDictEntry.h"
#include "gdcmSystem.h"
#include <iostream>
#include <iomanip>
```

Include dependency graph for gdcDict.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcmm::Dict`  
*Class to represent a map of `DictEntry`.*
- class `gdcmm::PrivateDict`  
*Private `Dict`.*

## Namespaces

- gdc

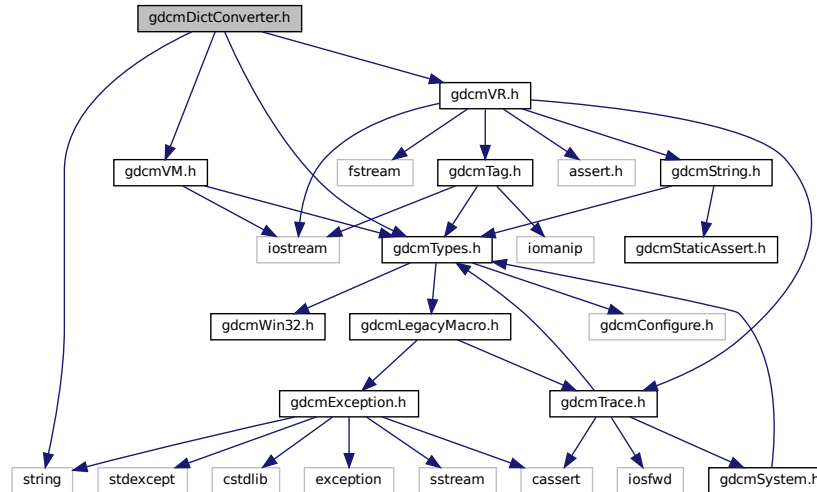
## Functions

- `std::ostream & gdcmm::operator<< (std::ostream &os, const Dict &val)`
- `std::ostream & gdcmm::operator<< (std::ostream &os, const PrivateDict &val)`

## 11.66 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

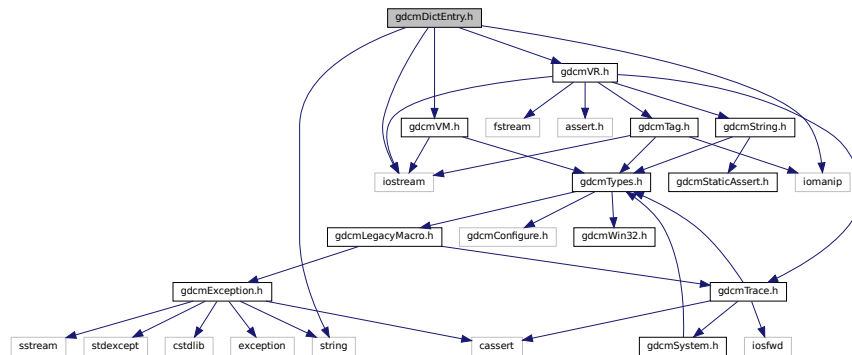
- [gdcm](#)

## 11.67 gdcmDictEntry.h File Reference

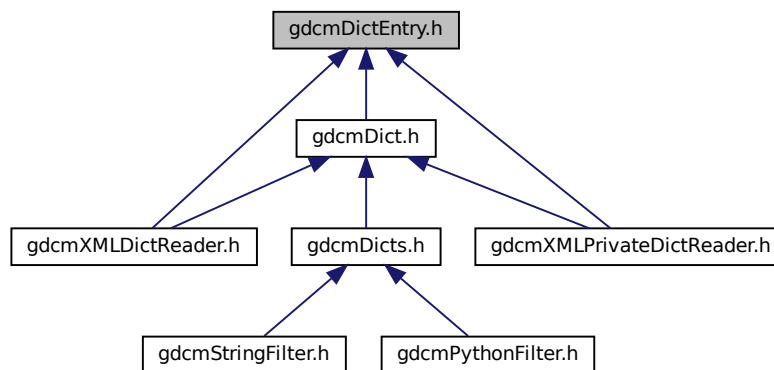
```
#include "gdcmVR.h"
#include "gdcmVM.h"
#include <string>
#include <iostream>
```

```
#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

- [gdcm](#)

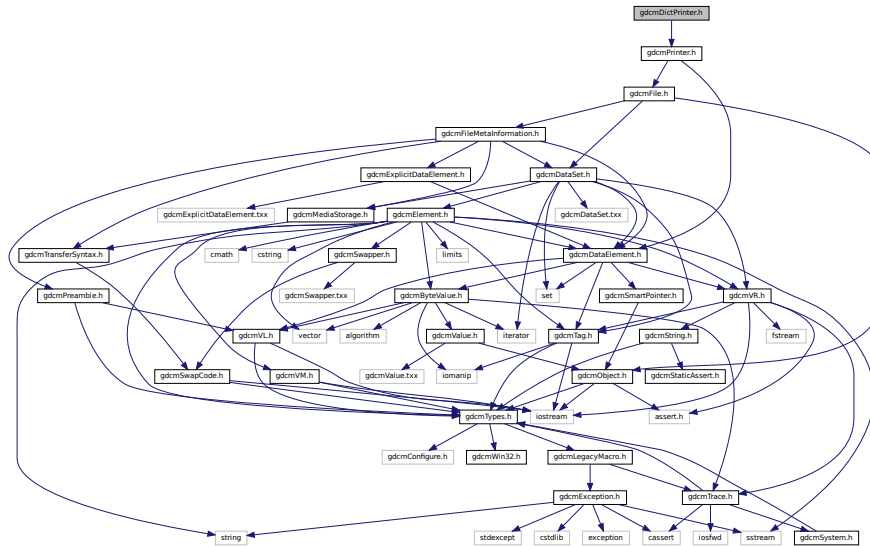
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const DictEntry &val)`

## 11.68 gdcmDictPrinter.h File Reference

```
#include "gdcmPrinter.h"
```

Include dependency graph for `gdcmDictPrinter.h`:



## Classes

- class `gdcm::DictPrinter`  
*DictPrinter* class.

## Namespaces

- **gdcm**

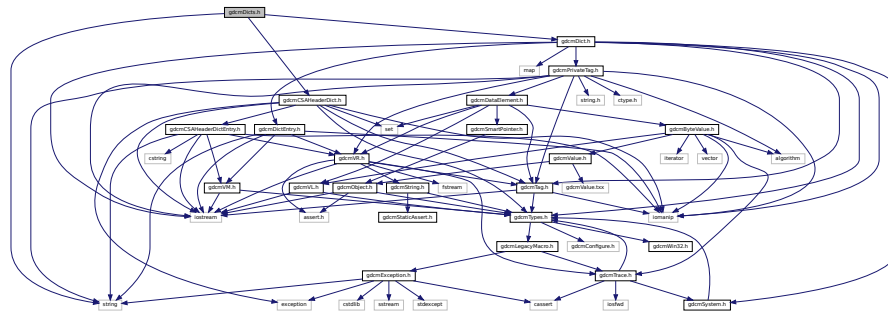
## 11.69 gdcmDicts.h File Reference

```
#include "gdcmDict.h"
```

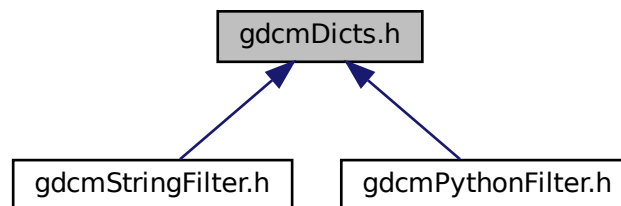
```
#include "gdcmCSAHeaderDict.h"
```



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

- **gdcm**

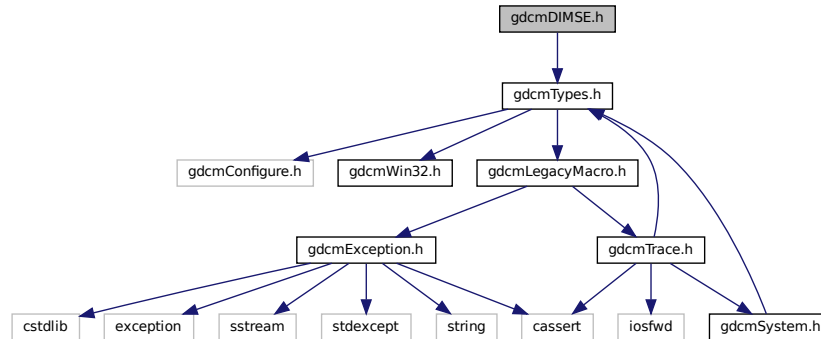
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Dicts &d)`

## 11.70 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.*

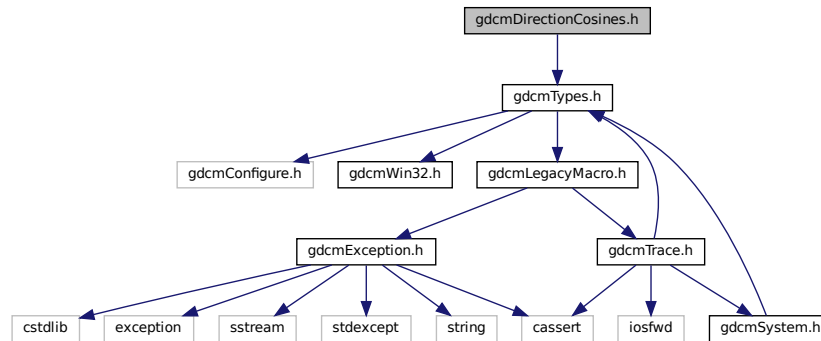
### Namespaces

- `gdcm`
- `gdcm::network`

## 11.71 gdcmDirectionCosines.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmDirectionCosines.h:



## Classes

- class [gdcm::DirectionCosines](#)  
class to handle *DirectionCosines*

## Namespaces

- [gdcm](#)

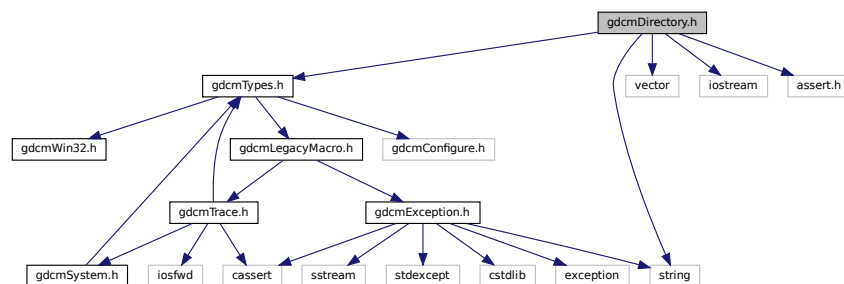
## 11.72 gdcmDirectory.h File Reference

```

#include "gdcmTypes.h"
#include <string>
#include <vector>
#include <iostream>
#include <assert.h>

```

Include dependency graph for gdcmDirectory.h:





## Classes

- class [gdcm::DirectoryHelper](#)  
*DirectoryHelper.*

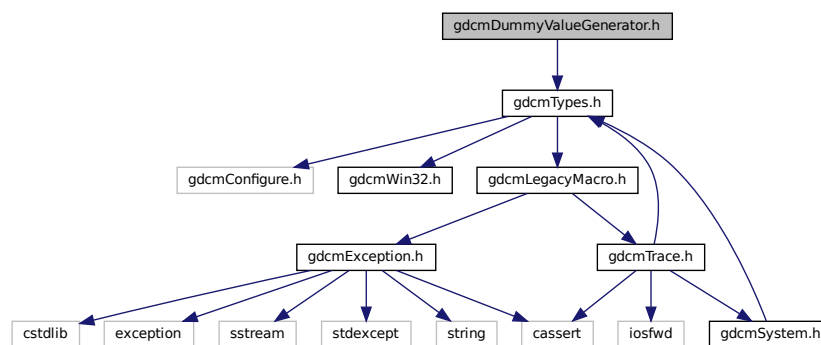
## Namespaces

- [gdcm](#)

## 11.74 gdcmDummyValueGenerator.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmDummyValueGenerator.h:



## Classes

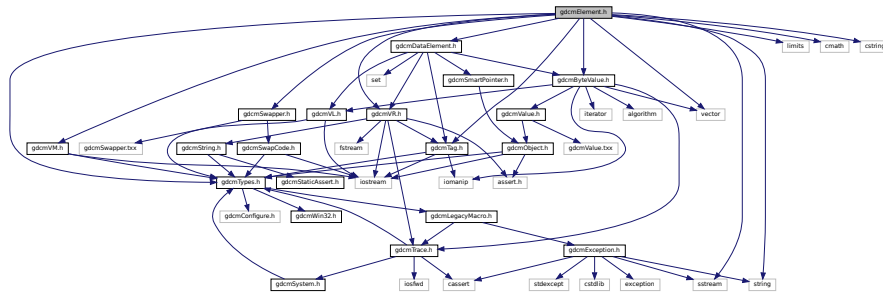
- class [gdcm::DummyValueGenerator](#)  
*Class for generating dummy value.*

## Namespaces

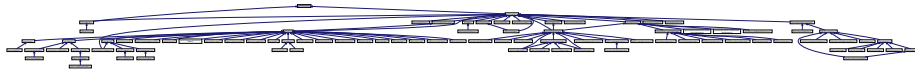
- [gdcm](#)



```
#include <cmath>
#include <cstring>
Include dependency graph for gdcmElement.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcmm::Element< TVR, TVM >`  
*Element class.*
- class `gdcmm::Element< TVR, VM::VM1_2 >`
- class `gdcmm::Element< TVR, VM::VM1_n >`
- class `gdcmm::Element< TVR, VM::VM2_2n >`
- class `gdcmm::Element< TVR, VM::VM2_n >`
- class `gdcmm::Element< TVR, VM::VM3_3n >`
- class `gdcmm::Element< TVR, VM::VM3_n >`
- class `gdcmm::Element< VR::AS, VM::VM5 >`
- class `gdcmm::Element< VR::OB, VM::VM1 >`
- class `gdcmm::Element< VR::OW, VM::VM1 >`
- class `gdcmm::ElementDisableCombinations< TVR, TVM >`  
*A class which is used to produce compile errors for an invalid combination of template parameters.*
- class `gdcmm::ElementDisableCombinations< VR::OB, VM::VM1_n >`
- class `gdcmm::ElementDisableCombinations< VR::OW, VM::VM1_n >`
- class `gdcmm::EncodingImplementation< VR::VRASCII >`
- class `gdcmm::EncodingImplementation< VR::VRBINARY >`
- struct `gdcmm::ignore_char`

## Namespaces

- **gdcm**

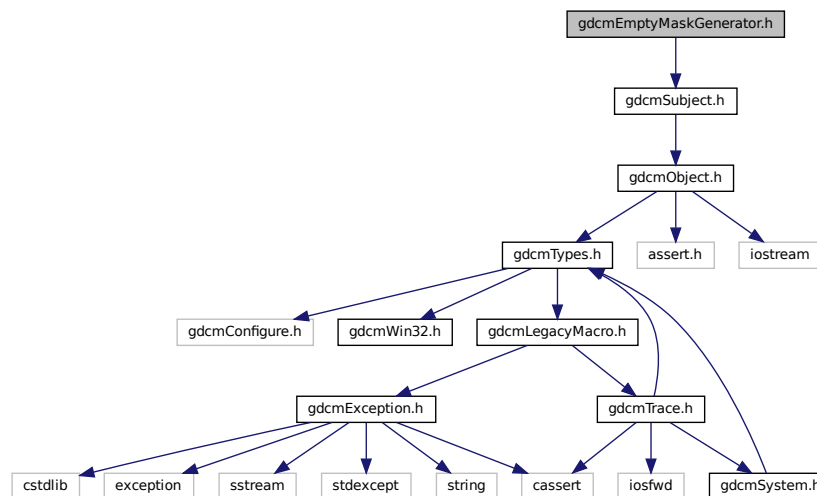
## Functions

- static int [gdcmm::add1](#) (char \*buf, int n)
- ignore\_char const [gdcmm::backslash](#) ('\')
- static void [gdcmm::clean](#) (char \*mant)
- static int [gdcmm::doround](#) (char \*buf, unsigned int n)
- std::istream & [gdcmm::operator>>](#) (std::istream &in, ignore\_char const &ic)
- static int [gdcmm::roundat](#) (char \*buf, unsigned int i, int iexp)
- template<typename Float >  
static void [gdcmm::x16printf](#) (char \*buf, int size, Float f)

## 11.77 gdcmmEmptyMaskGenerator.h File Reference

```
#include "gdcmmSubject.h"
```

Include dependency graph for gdcmmEmptyMaskGenerator.h:



## Classes

- class [gdcmm::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.*

## Namespaces

- [gdcmm](#)





## 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:

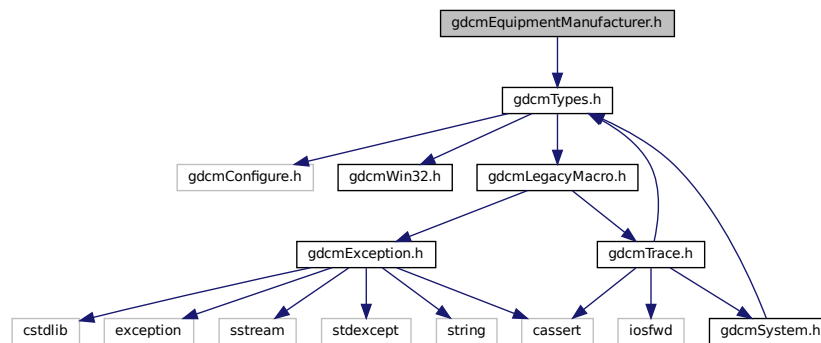
## Namespaces

- [gdcm](#)

## 11.80 gdcmEquipmentManufacturer.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmEquipmentManufacturer.h:



## Classes

- class [gdcm::EquipmentManufacturer](#)

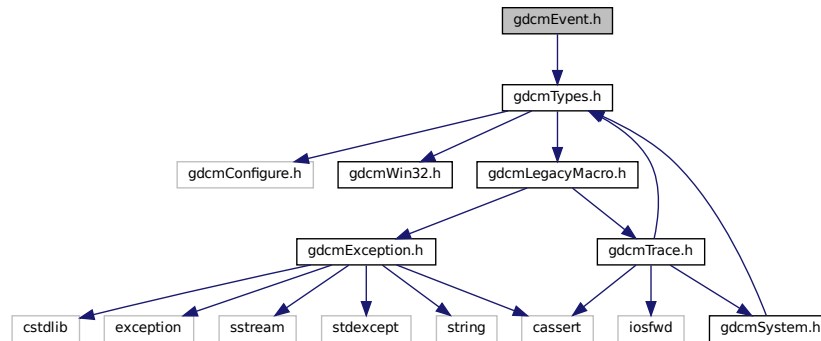
## Namespaces

- [gdcm](#)

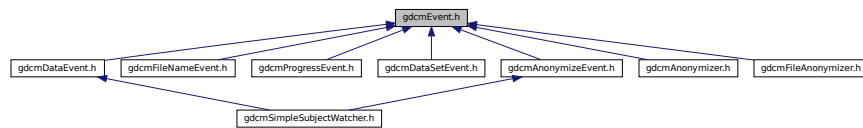
## 11.81 gdcmEvent.h File Reference

```
#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

- `gdcm`

## Macros

- `#define gdcmEventMacro(classname, super)`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, Event &e)`  
*Generic inserter operator for [Event](#) and its subclasses.*

### 11.81.1 Macro Definition Documentation

#### 11.81.1.1 [gdcmEventMacro](#)

```
#define gdcmEventMacro(
    classname,
    super )
```

#### Value:

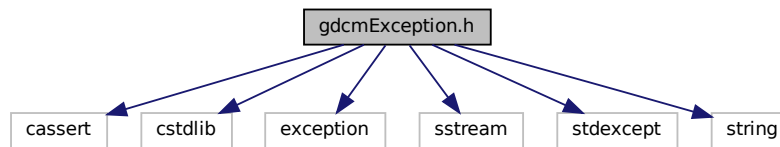
```
\
class classname : public super { \
public: \
    typedef classname Self; \
    typedef super Superclass; \
    classname() {} \
    virtual ~classname() override = default; \
    virtual const char * GetEventName() const { return #classname; } \
    virtual bool CheckEvent(const ::gdcm::Event* e) const \
    { return dynamic_cast<const Self*>(e) ? true : false; } \
    virtual ::gdcm::Event* MakeObject() const \
    { return new Self; } \
    classname(const Self&s) : super(s){}; \
private: \
    void operator=(const Self&); \
}
```

### 11.82 [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

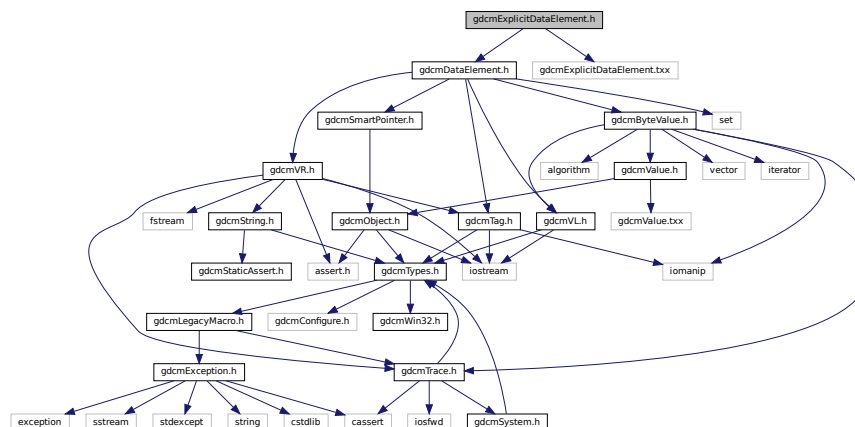
- [gdcm](#)

## 11.83 gdcmExplicitDataElement.h File Reference

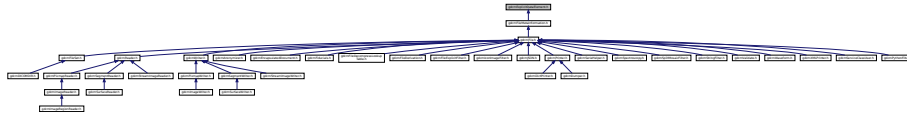
```
#include "gdcmDataElement.h"
```

```
#include "gdcmExplicitDataElement.txx"
```

Include dependency graph for gdcmExplicitDataElement.h:



This graph shows which files directly or indirectly include this file:



## Classes

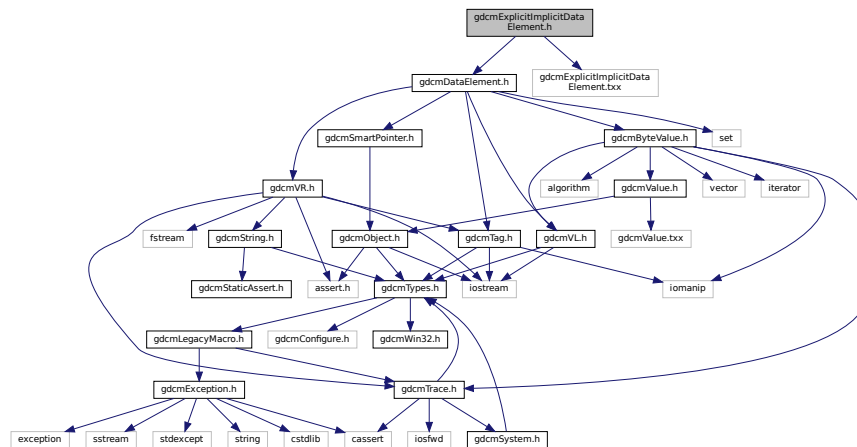
- class [gdcm::ExplicitDataElement](#)  
Class to read/write a [DataElement](#) as Explicit Data [Element](#).

## Namespaces

- [gdcm](#)

## 11.84 gdcmExplicitImplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmExplicitImplicitDataElement.txx"
Include dependency graph for gdcmExplicitImplicitDataElement.h:
```

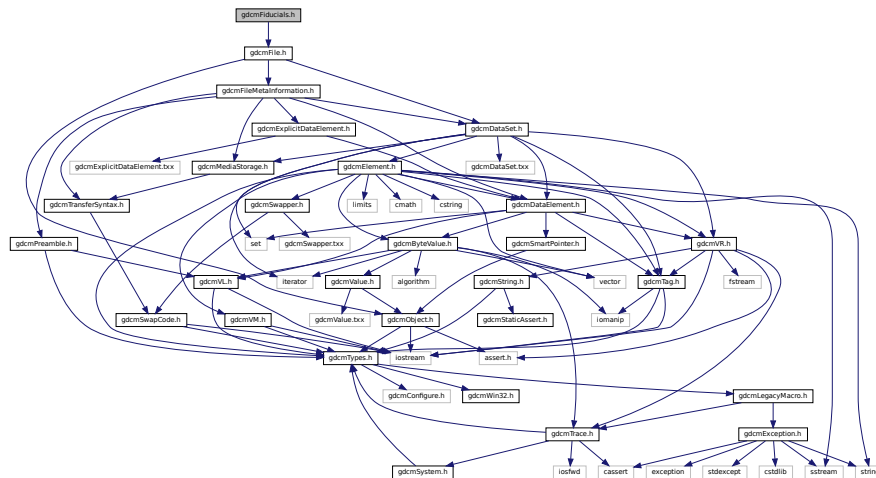


## Classes

- class [gdcm::ExplicitImplicitDataElement](#)  
Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#).

- **gdcm**

Include dependency graph for gdcmFiducials.h:

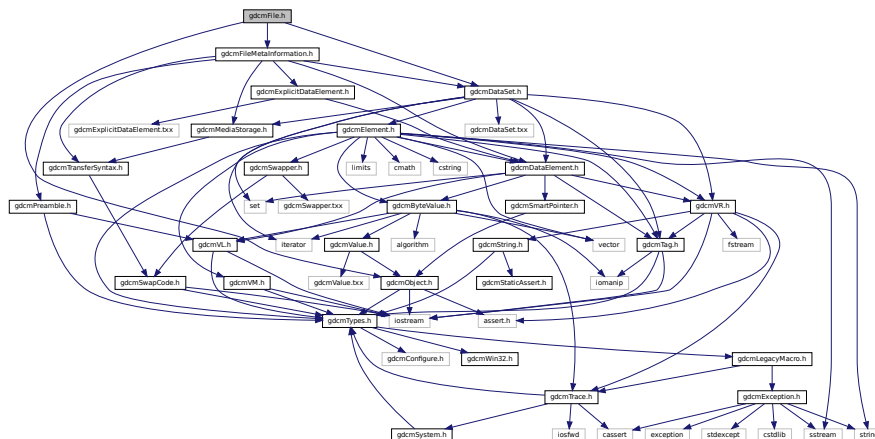


- class `gdcm::Fiducials`  
*Fiducials*.

- gdc

```
#include "gdcmDataSet.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

- **gdcm**

## Functions

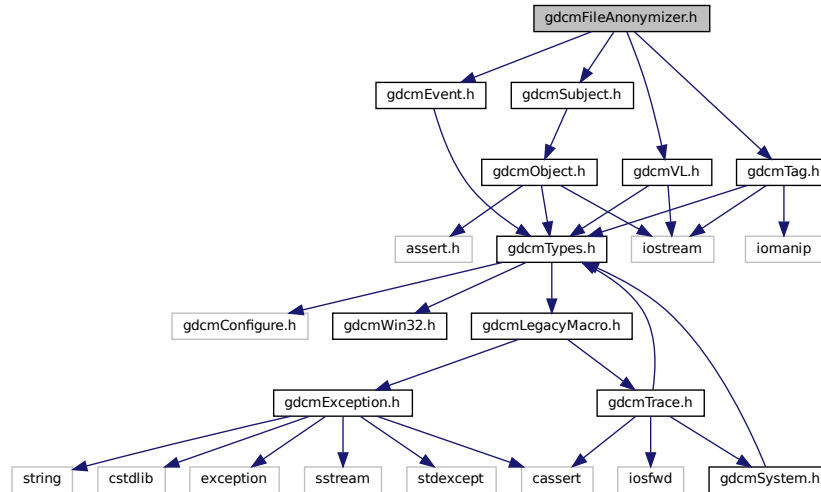
- `std::ostream & gdcmm::operator<< (std::ostream &os, const File &val)`



## 11.87 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.*

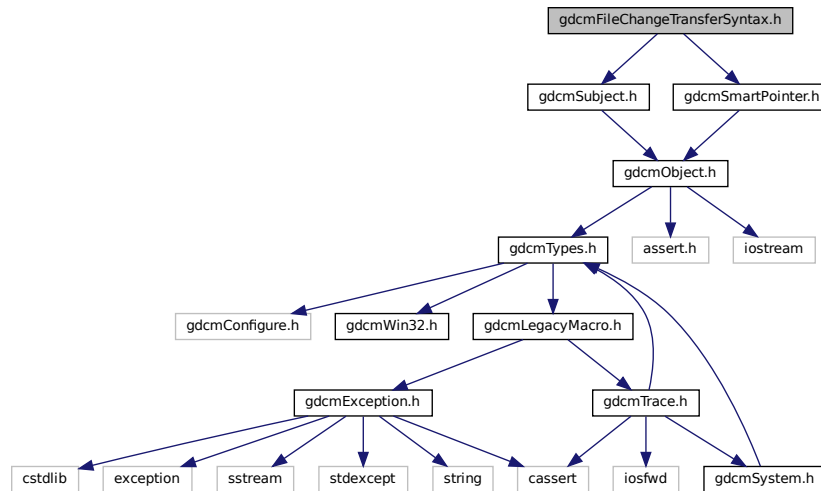
### Namespaces

- [gdcm](#)

## 11.88 gdcmFileChangeTransferSyntax.h File Reference

```
#include "gdcmSubject.h"
#include "gdcmSmartPointer.h"
```

Include dependency graph for `gdcmFileChangeTransferSyntax.h`:



## Classes

- class `gdcm::FileChangeTransferSyntax`  
*FileChangeTransferSyntax.*

## Namespaces

- `gdcm`

## 11.89 gdcmFileDecompressLookupTable.h File Reference

```

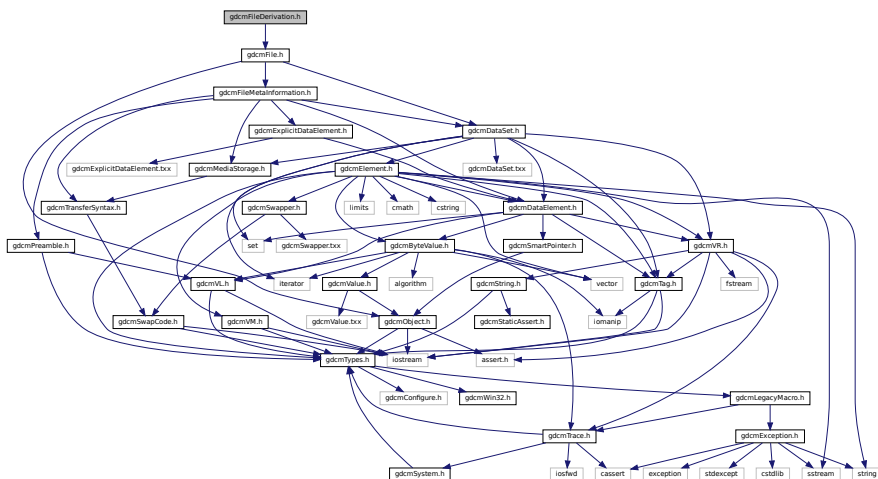
#include "gdcmSubject.h"
#include "gdcmFile.h"
#include "gdcmPixmap.h"

```

- class `gdc::FileDecompressLookupTable`  
*FileDecompressLookupTable* class.

- **gdcm**

```
#include "gdcmFile.h"
Include dependency graph for gdcmFileDerivation.h:
```





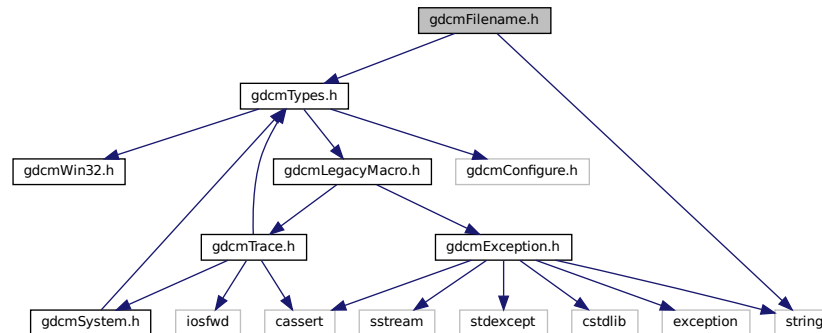


## 11.93 gdcmFilename.h File Reference

```
#include "gdcmTypes.h"
```

```
#include <string>
```

Include dependency graph for gdcmFilename.h:



### Classes

- class [gdcm::Filename](#)  
Class to manipulate file name's.

### Namespaces

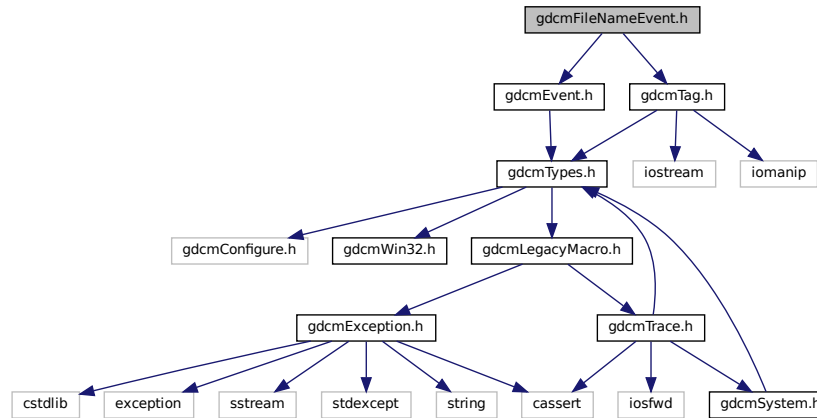
- [gdcm](#)

## 11.94 gdcmFileNameEvent.h File Reference

```
#include "gdcmEvent.h"
```

```
#include "gdcmTag.h"
```

Include dependency graph for gdcmFileNameEvent.h:



## Classes

- class [gdcm::FileNameEvent](#)  
*FileNameEvent.*

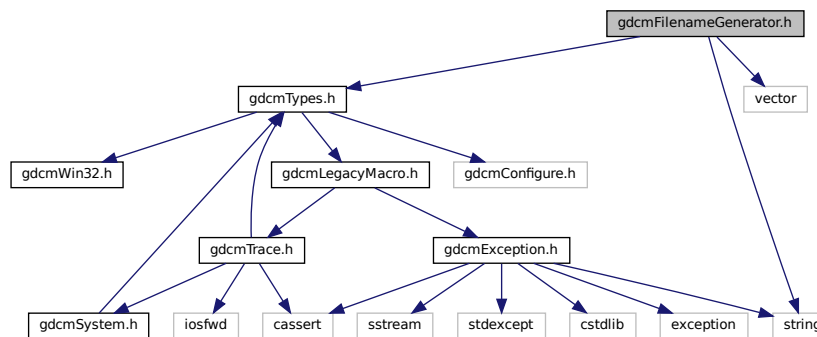
## Namespaces

- [gdcm](#)

## 11.95 gdcmFilenameGenerator.h File Reference

```
#include "gdcmTypes.h"
#include <string>
#include <vector>
```

Include dependency graph for gdcmFilenameGenerator.h:



## 11.96 gdcmFileSet.h File Reference



## Classes

- class [gdcm::FileSet](#)

## Namespaces

- [gdcm](#)

## Functions

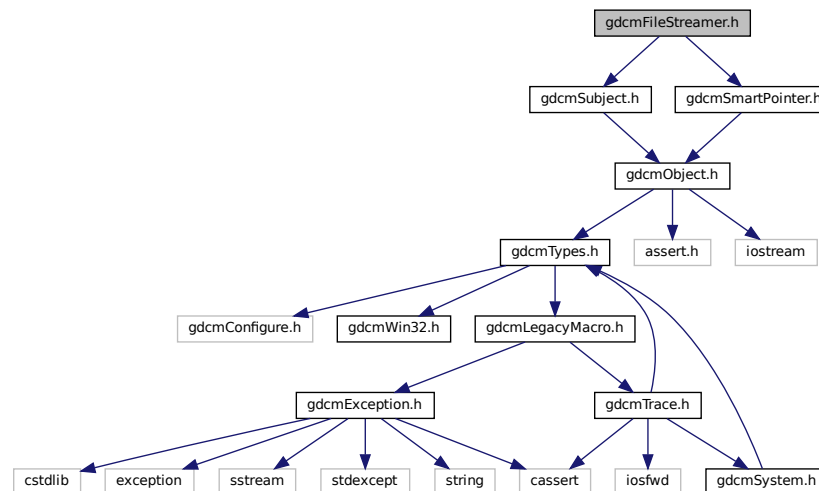
- `std::ostream & gdcm::operator<< (std::ostream &os, const FileSet &f)`

## 11.97 gdcmFileStreamer.h File Reference

```
#include "gdcmSubject.h"
```

```
#include "gdcmSmartPointer.h"
```

Include dependency graph for gdcmFileStreamer.h:



## Classes

- class [gdcm::FileStreamer](#)  
*FileStreamer.*

## Namespaces

- [gdcm](#)







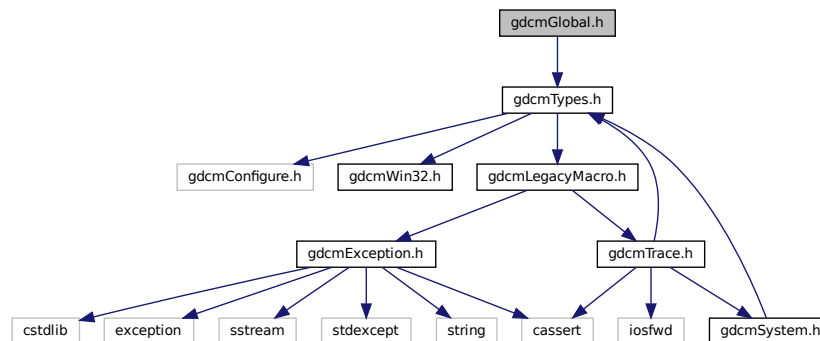
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Fragment &val)`

## 11.101 gdcmGlobal.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmGlobal.h:



## Classes

- class `gdcm::Global`  
*Global.*

## Namespaces

- `gdcm`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Global &g)`

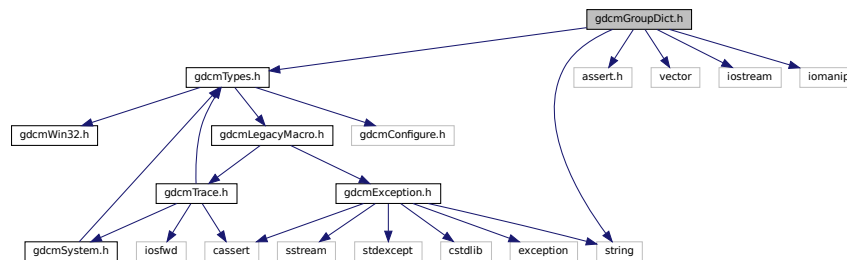
## Variables

- static Global `gdcm::GlobalInstance`

## 11.102 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

- [gdcm](#)

### Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const GroupDict &_val)`

## 11.103 gdcmIconImage.h File Reference

```
#include "gdcmBitmap.h"
```

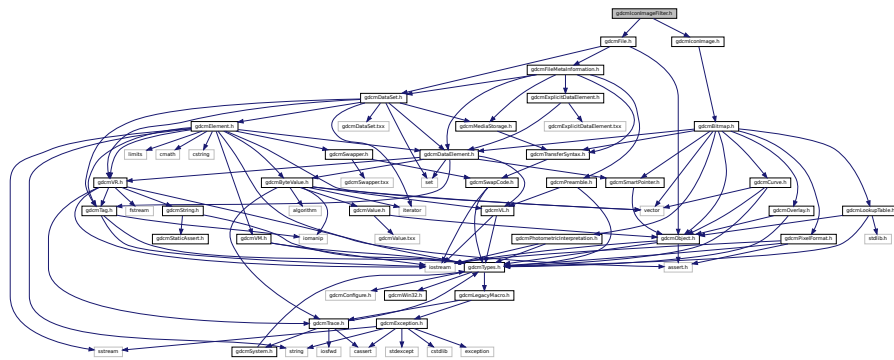
[illegible][illegible]

- **gdcm**

- typedef Bitmap **gdcm::IconImage**

```
#include "gdcmFile.h"
#include "gdcmIconImage.h"
```

Include dependency graph for `gdcmlconImageFilter.h`:



## Classes

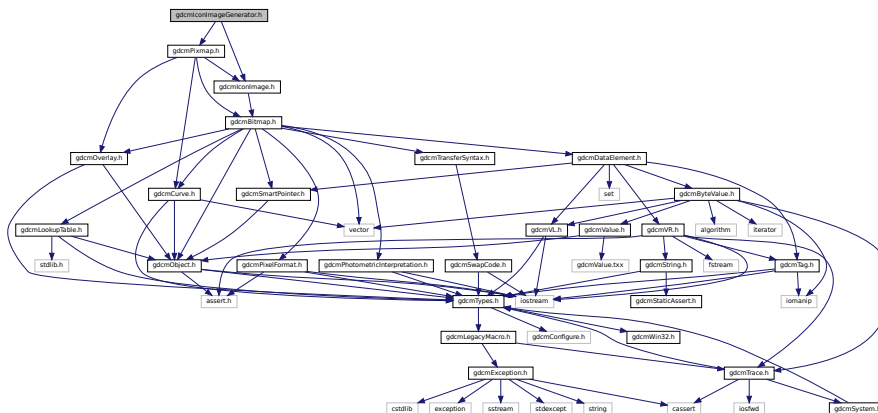
- class `gdcm::IconImageFilter`  
*IconImageFilter*.

## Namespaces

- **gdcm**

## 11.105 gdcmlconImageGenerator.h File Reference

```
#include "gdcmPixmap.h"
#include "gdcmIconImage.h"
Include dependency graph for gdcmIconImageGenerator.h:
```





## Classes

- class `gdcm::IconImageGenerator`  
*IconImageGenerator.*

## Namespaces

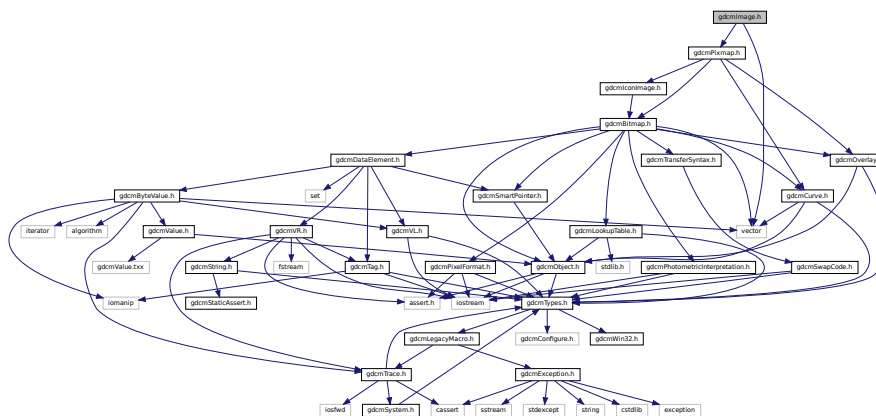
- **gdcm**

## 11.106 gdcmlImage.h File Reference

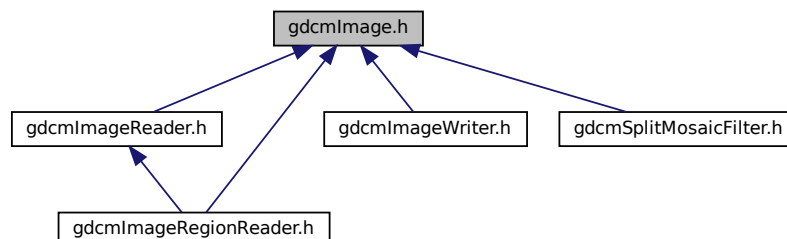
```
#include "gdcmPixmap.h"
```

```
#include <vector>
```

Include dependency graph for gdcmlImage.h:

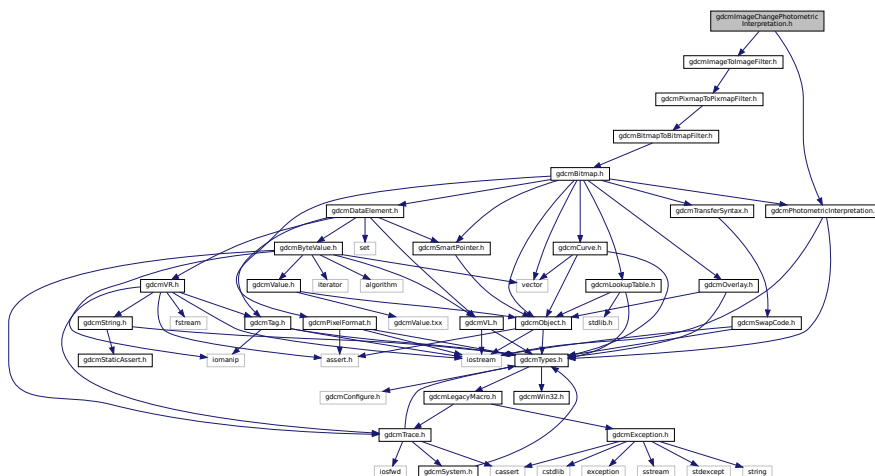


This graph shows which files directly or indirectly include this file:





```
#include "gdcmImageToImageFilter.h"
#include "gdcmPhotometricInterpretation.h"
Include dependency graph for gdcmImageChangePhotometricInterpretation.h:
```



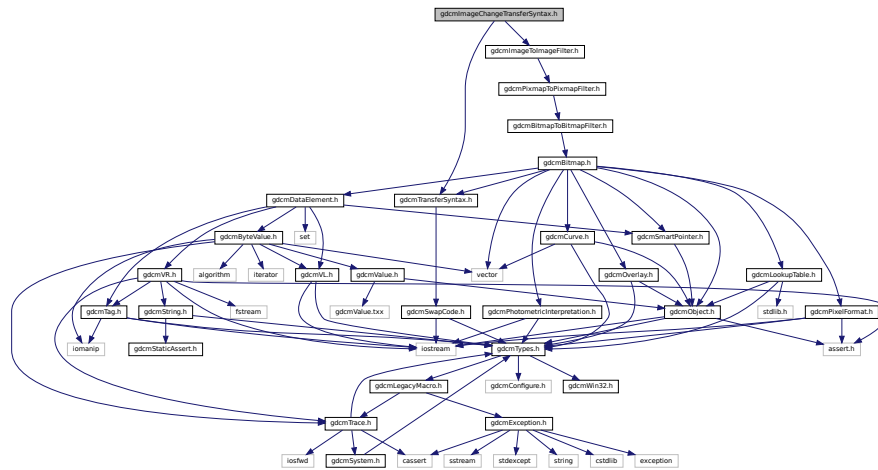
- class `gdcm::ImageChangePhotometricInterpretation`  
*ImageChangePhotometricInterpretation* class.

- **gdcm**

- `template<typename T >`  
`static T gdc::Clamp (int v)`
- `template<typename T >`  
`static int gdc::Round (T x)`



Include dependency graph for `gdcmImageChangeTransferSyntax.h`:



## Classes

- class `gdcm::ImageChangeTransferSyntax`  
*ImageChangeTransferSyntax* class.

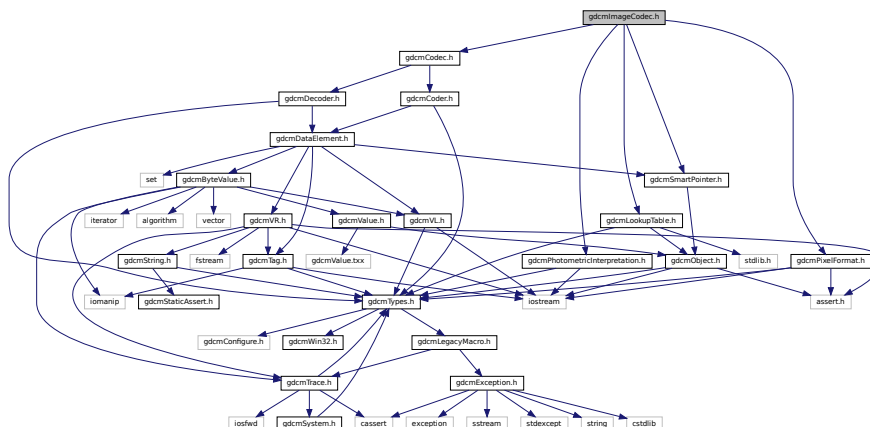
## Namespaces

- **gdcm**

## 11.111 gdcmlImageCodec.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

- `gdcm`

## 11.112 gdcmImageConverter.h File Reference

```
#include "gdcmTypes.h"
```

```

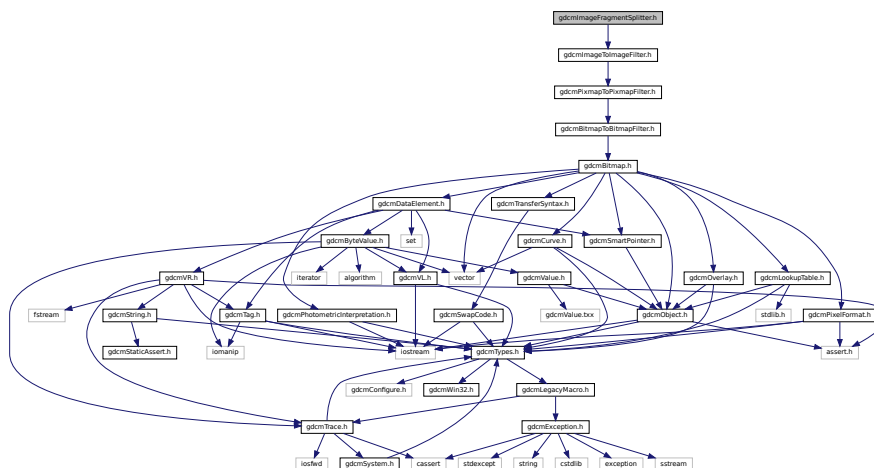
graph TD
    gdcmImageConverter.h[gdcmImageConverter.h] --> gdcmTypes.h[gdcmTypes.h]
    gdcmTypes.h --> gdcmConfigure.h[gdcmConfigure.h]
    gdcmTypes.h --> gdcmWin32.h[gdcmWin32.h]
    gdcmTypes.h --> gdcmLegacyMacro.h[gdcmLegacyMacro.h]
    gdcmLegacyMacro.h --> gdcmException.h[gdcmException.h]
    gdcmLegacyMacro.h --> gdcmTrace.h[gdcmTrace.h]
    gdcmException.h --> cstdlib[cstdlib]
    gdcmException.h --> exception[exception]
    gdcmException.h --> sstream[sstream]
    gdcmException.h --> stdexcept[stdexcept]
    gdcmException.h --> string[string]
    gdcmException.h --> cassert[cassert]
    gdcmException.h --> iosfwd[iosfwd]
    gdcmException.h --> gdcmSystem.h[gdcmSystem.h]
    gdcmTrace.h --> iosfwd
    gdcmTrace.h --> gdcmSystem.h

```

- class `gdcm::ImageConverter`  
*Image Converter.*

- **gdcm**

```
#include "gdcmImageToImageFilter.h"
Include dependency graph for gdcmImageFragmentSplitter.h:
```



## Classes

- class [gdcm::ImageFragmentSplitter](#)  
*ImageFragmentSplitter* class.

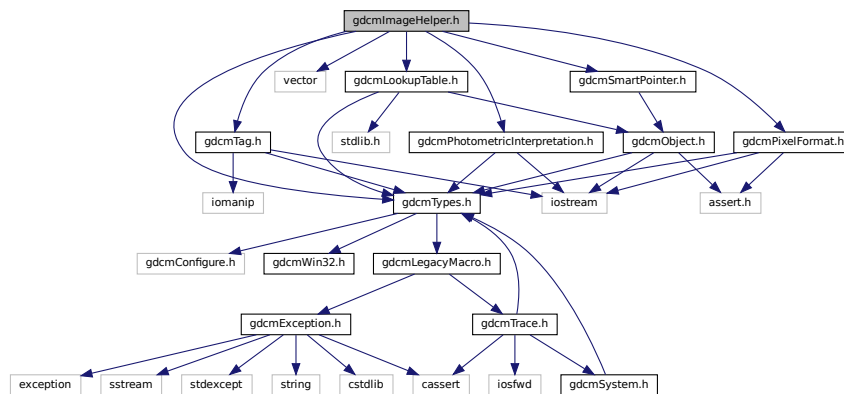
## Namespaces

- [gdcm](#)

### 11.114 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

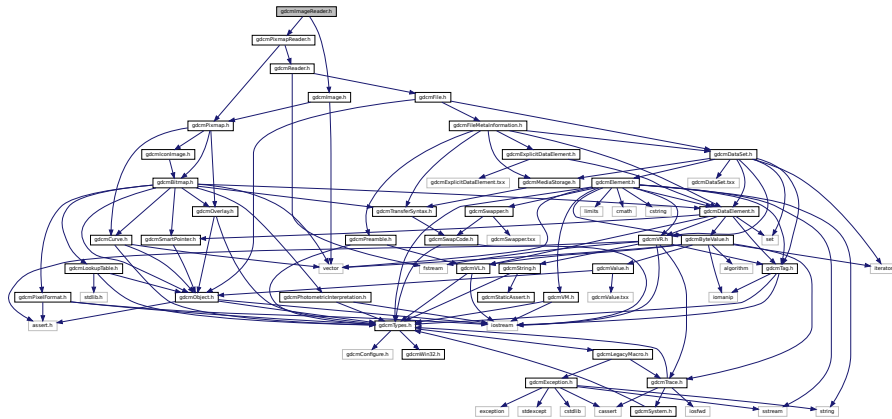
- class [gdcm::ImageHelper](#)  
*ImageHelper* (internal class, not intended for user level)
- struct [gdcm::RealWorldValueMappingContent](#)

## Namespaces

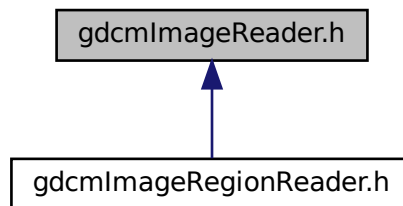
- [gdcm](#)



```
#include "gdcmPixmapReader.h"
#include "gdcmImage.h"
Include dependency graph for gdcmImageReader.h:
```



This graph shows which files directly or indirectly include this file:



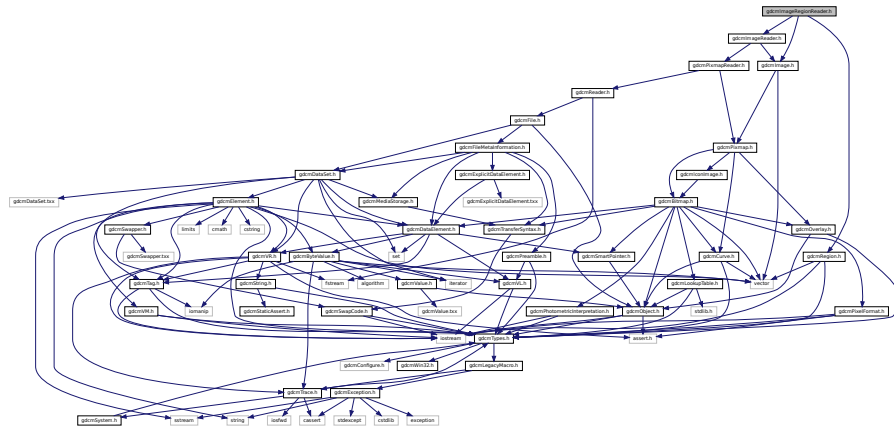
- class `gdcm::ImageReader`  
*ImageReader*.

- **gdcm**

## 11.116 gdcmlImageRegionReader.h File Reference

```
#include "gdcmImageReader.h"
#include "gdcmImage.h"
#include "gdcmRegion.h"
```

Include dependency graph for `gdcmlImageRegionReader.h`:



## Classes

- class `gdcm::ImageRegionReader`  
*ImageRegionReader.*

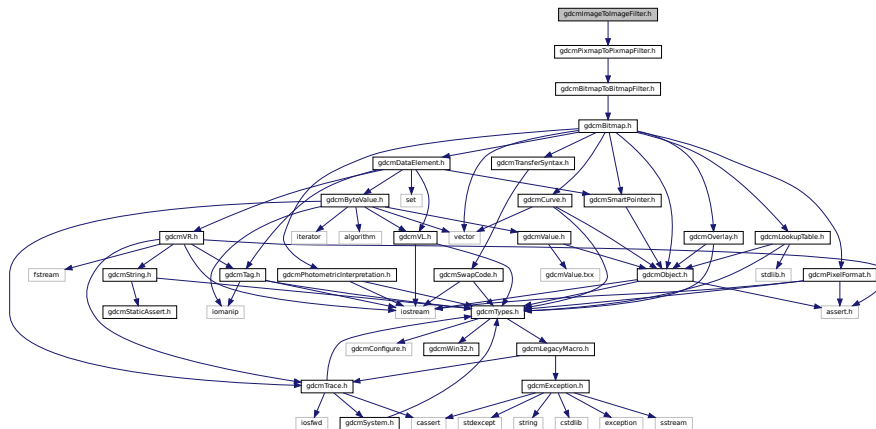
## Namespaces

- **gdcm**

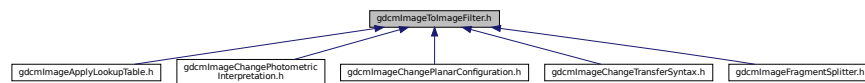
## 11.117 gdcmlImageToImageFilter.h File Reference

```
#include "gdcmPixmapToPixmapFilter.h"
```

Include dependency graph for gdcmImageToImageFilter.h:



This graph shows which files directly or indirectly include this file:



## Classes

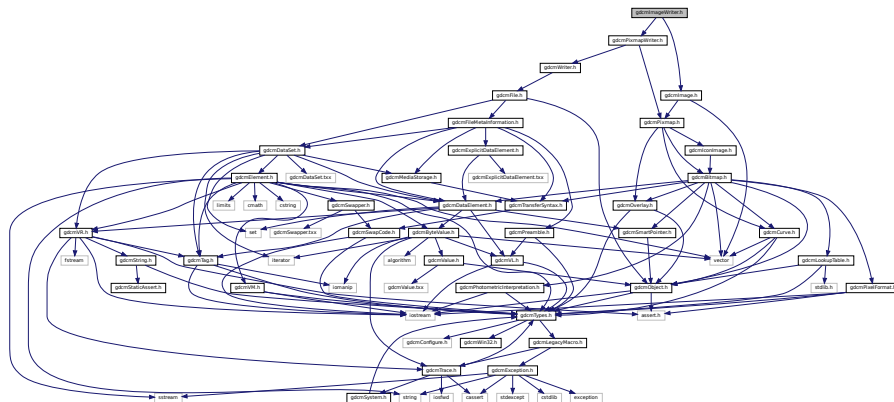
- class [gdcm::ImageToImageFilter](#)  
*ImageToImageFilter* class.

## Namespaces

- [gdcm](#)

## 11.118 gdcmImageWriter.h File Reference

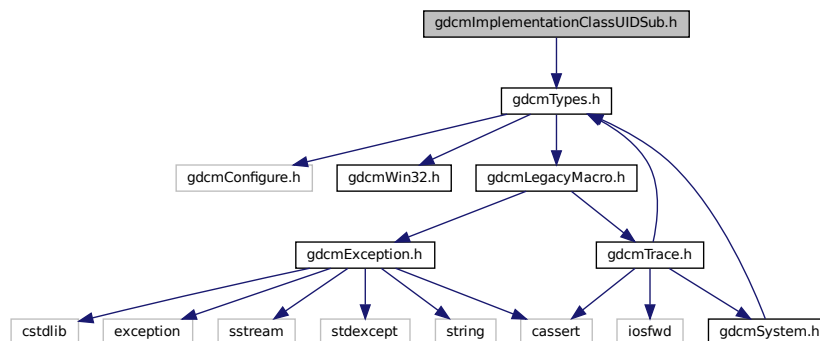
```
#include "gdcmPImageWriter.h"
#include "gdcmImage.h"
```



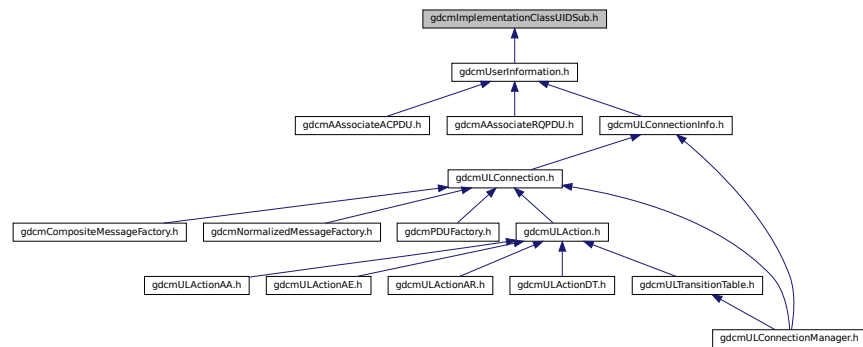
- class `gdcm::ImageWriter`  
*ImageWriter.*

- **gdcm**

```
#include "gdcmTypes.h"
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::network::ImplementationClassUIDSub](#)  
*ImplementationClassUIDSub.*

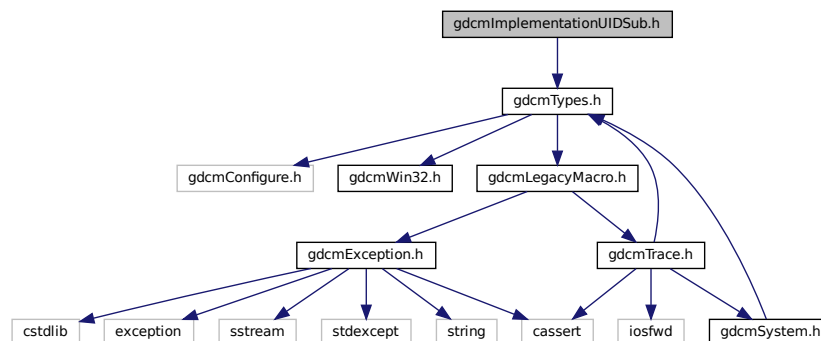
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.120 gdcmImplementationUIDSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmImplementationUIDSub.h:



## Classes

- class [gdcm::network::ImplementationUIDSub](#)  
*ImplementationUIDSub.*

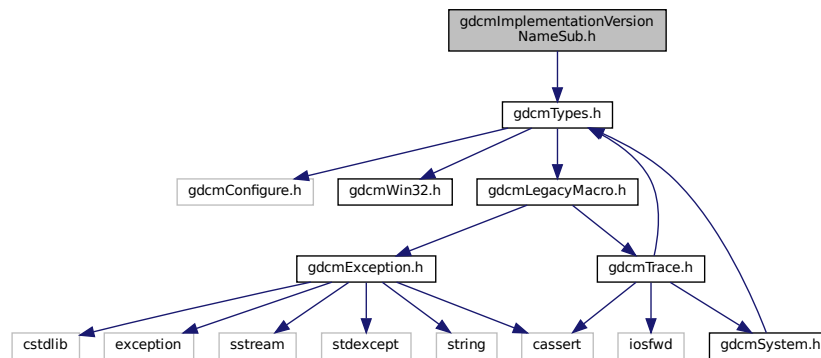
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

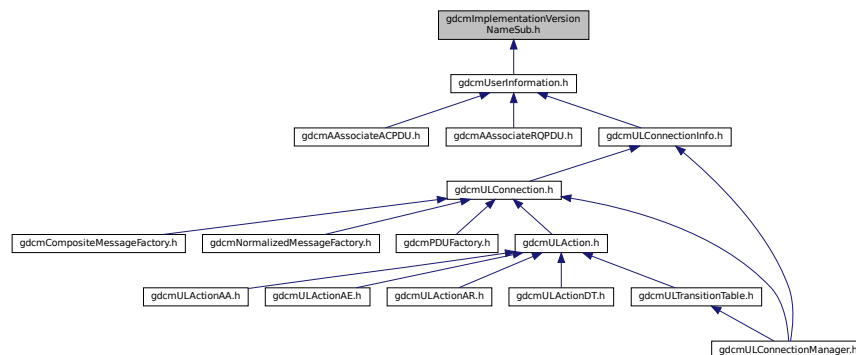
## 11.121 gdcmImplementationVersionNameSub.h File Reference

```
#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

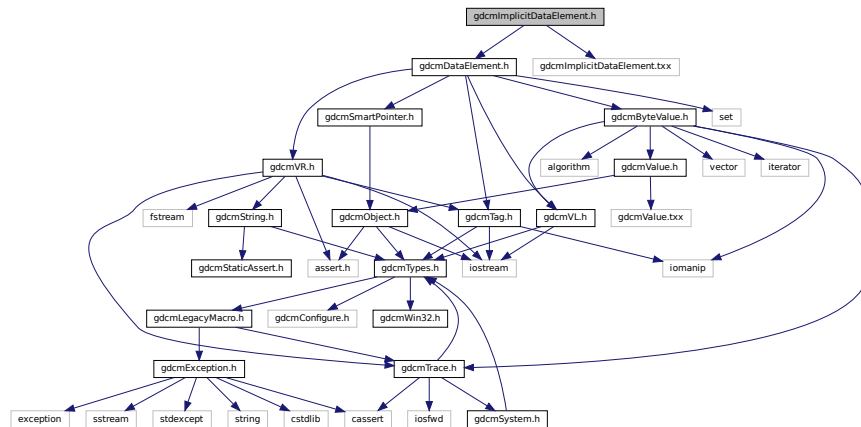
- [gdcm](#)
- [gdcm::network](#)

## 11.122 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.*

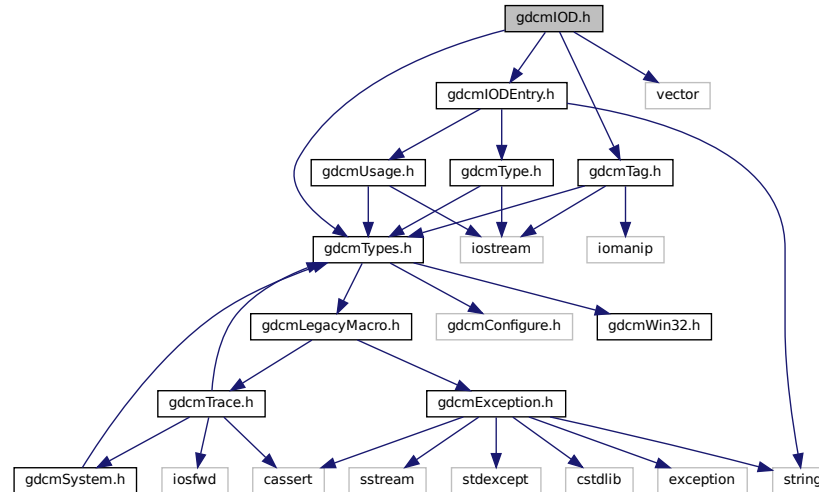
## Namespaces

- [gdcm](#)

## 11.123 gdcmIOD.h File Reference

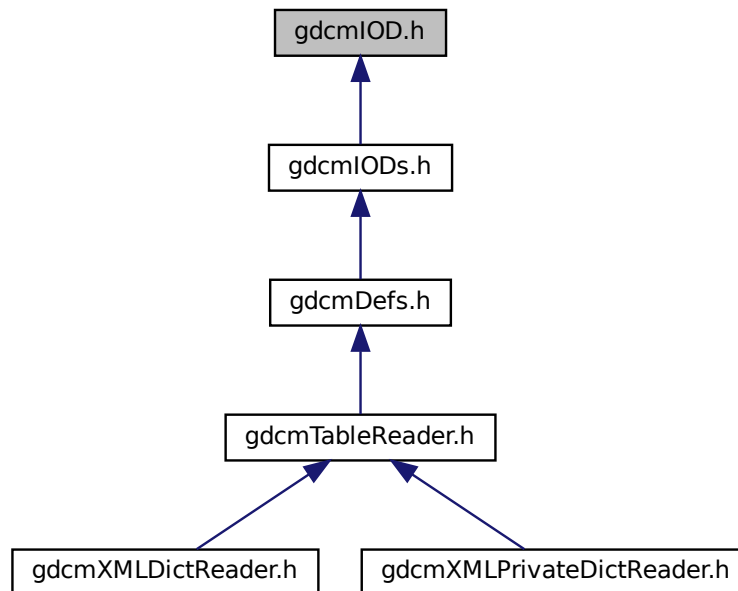
```
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmIOEntry.h"
#include <vector>
```

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

- [gdcm](#)

## Functions

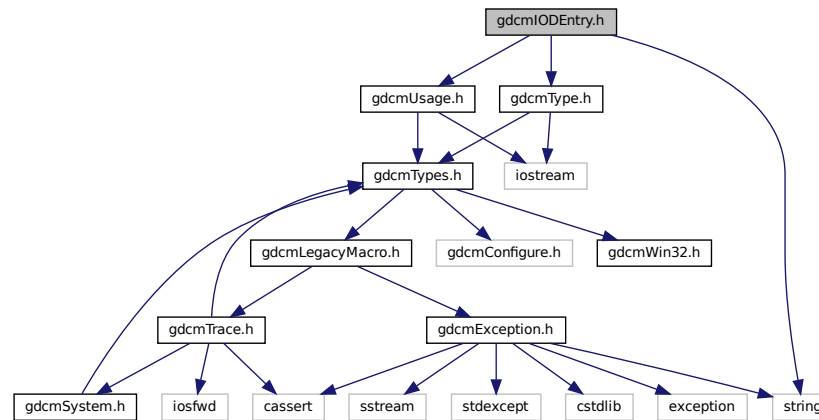
- `std::ostream & gdcm::operator<< (std::ostream &_os, const IOD &_val)`

## 11.124 gdcmIODEntry.h File Reference

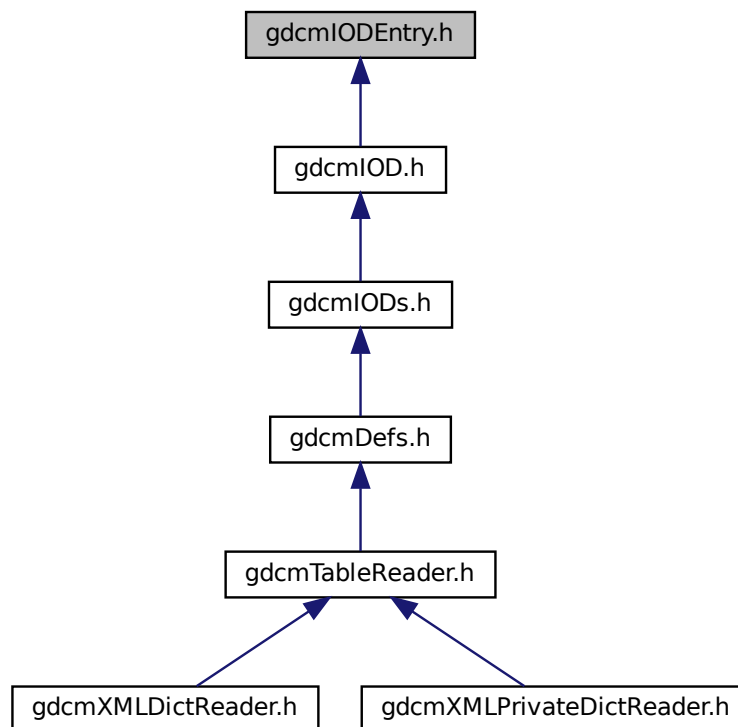
```
#include "gdcmUsage.h"  
#include "gdcmType.h"
```

```
#include <string>
```

Include dependency graph for gdcmlODEntry.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::IODEntry](#)  
Class for representing a *IODEntry*.

## Namespaces

- [gdcm](#)

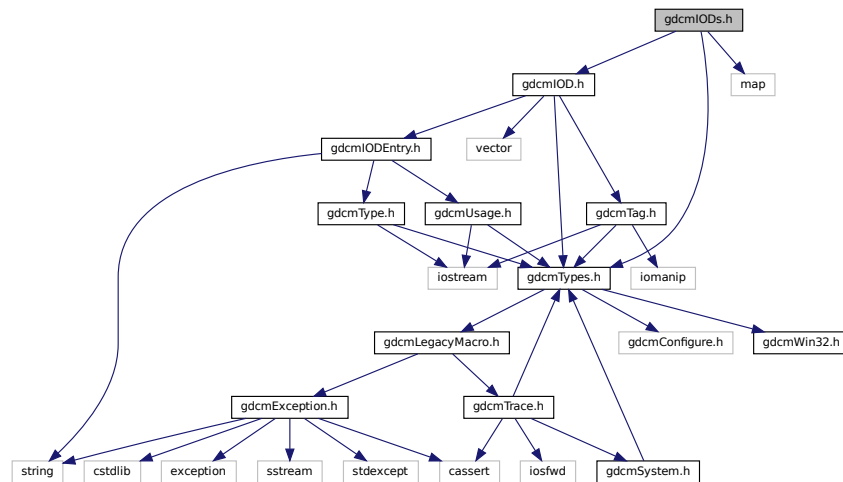
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const IODEntry &_val)`

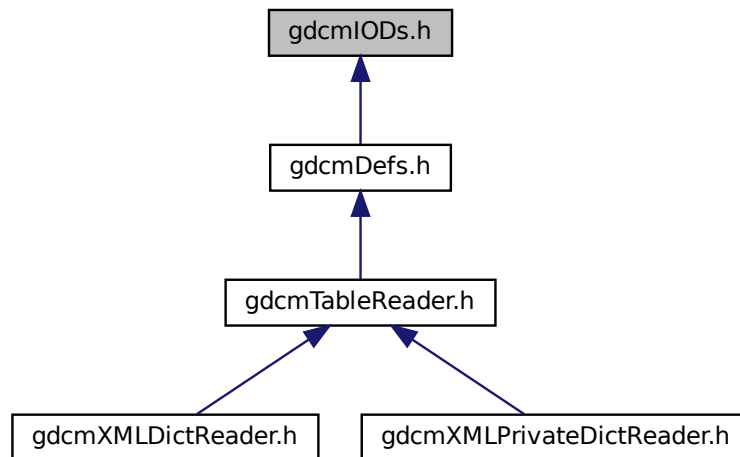
## 11.125 gdcmIODs.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmIOD.h"
#include <map>
```

Include dependency graph for gdcmIODs.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcml::IODs](#)  
*Class for representing a [IODs](#).*

## Namespaces

- [gdcml](#)

## Functions

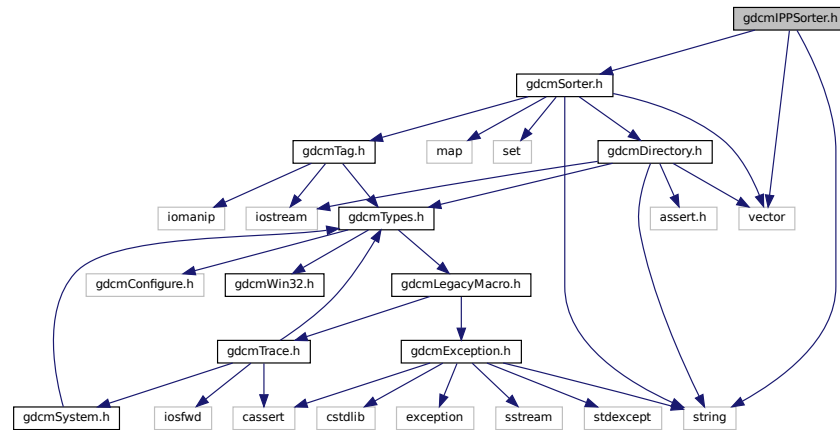
- `std::ostream & gdcml::operator<< (std::ostream &_os, const IODs &_val)`

## 11.126 gdcmlPPSorter.h File Reference

```
#include "gdcmlSorter.h"  
#include <vector>
```

```
#include <string>
```

Include dependency graph for gdcmIPPSorter.h:



## Classes

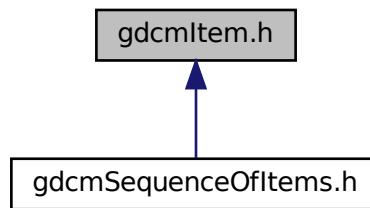
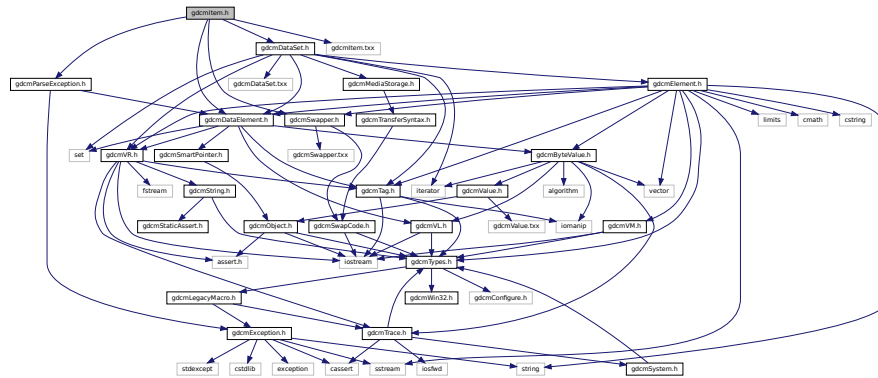
- class [gdcm::IPPSorter](#)  
*IPPSorter*.

## Namespaces

- [gdcm](#)

## 11.127 gdcmItem.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmDataSet.h"
#include "gdcmParseException.h"
#include "gdcmSwapper.h"
#include "gdcmItem.txx"
```



- class `gdcm::Item`  
*Class to represent an `Item`.*

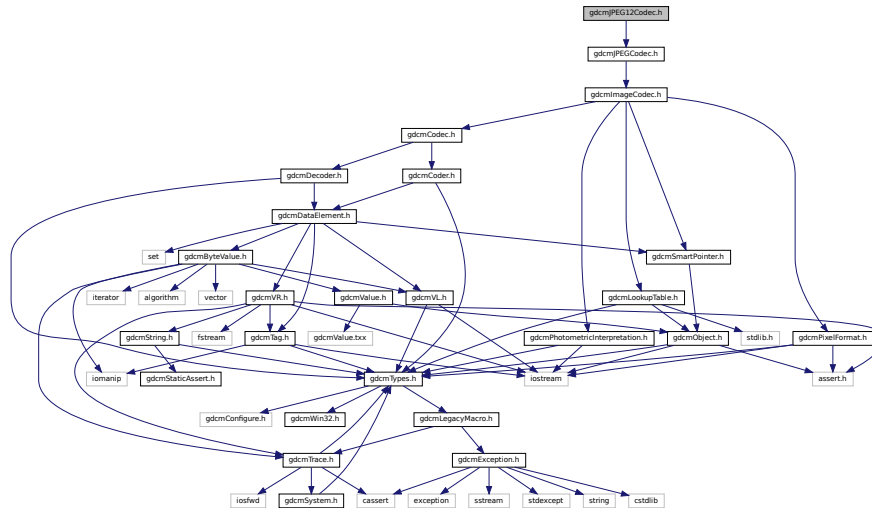
- **gdcm**

- `std::ostream & gdcm::operator<< (std::ostream &os, const Item &val)`

## 11.128 gdcmJPEG12Codec.h File Reference

```
#include "gdcmJPEGCodec.h"
```

Include dependency graph for gdcmJPEG12Codec.h:



### Classes

- class [gdcm::JPEG12Codec](#)  
*Class to do JPEG 12bits (lossy & lossless)*

### Namespaces

- [gdcm](#)

## 11.129 gdcmJPEG16Codec.h File Reference

```
#include "gdcmJPEGCodec.h"
```



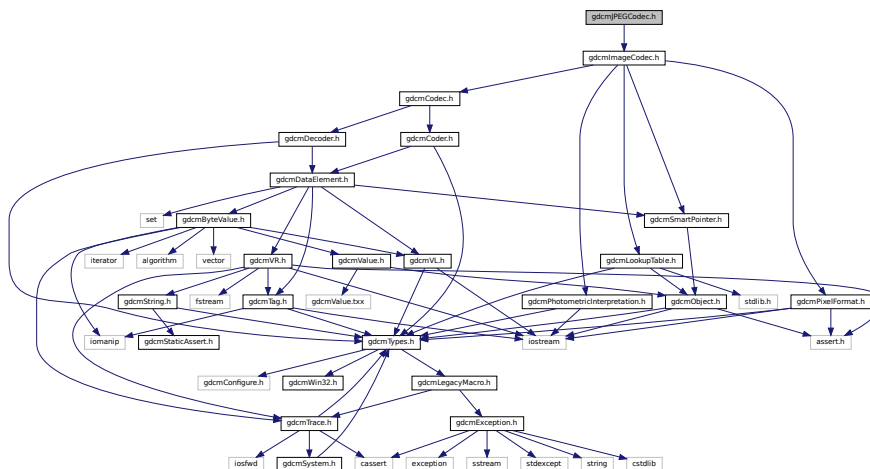




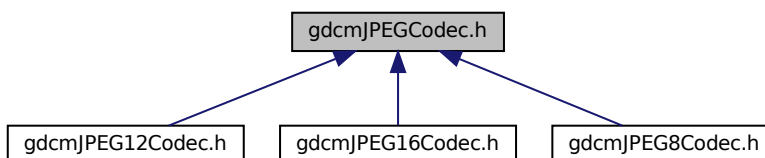
## 11.132 gdcmJPEGCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmJPEGCodec.h:



This graph shows which files directly or indirectly include this file:



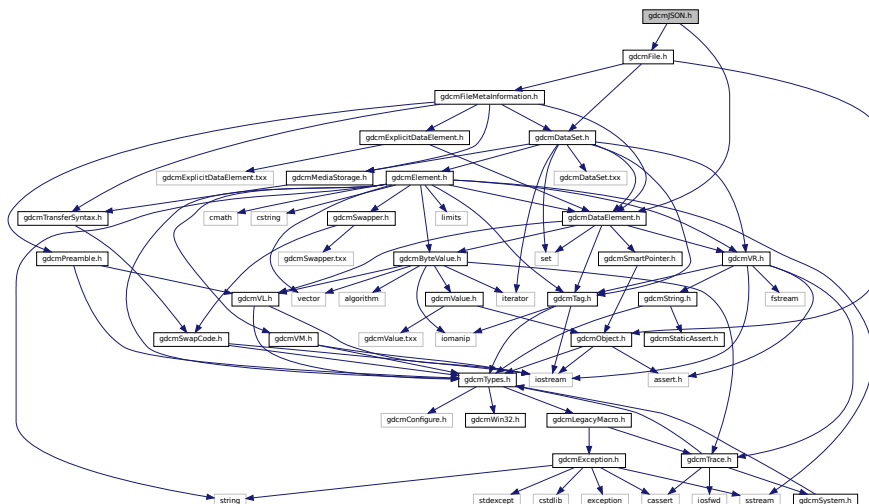
### Classes

- class [gdcm::JPEGCodec](#)  
*JPEG codec.*

### Namespaces

- [gdcm](#)





- class `gdcm::JSON`

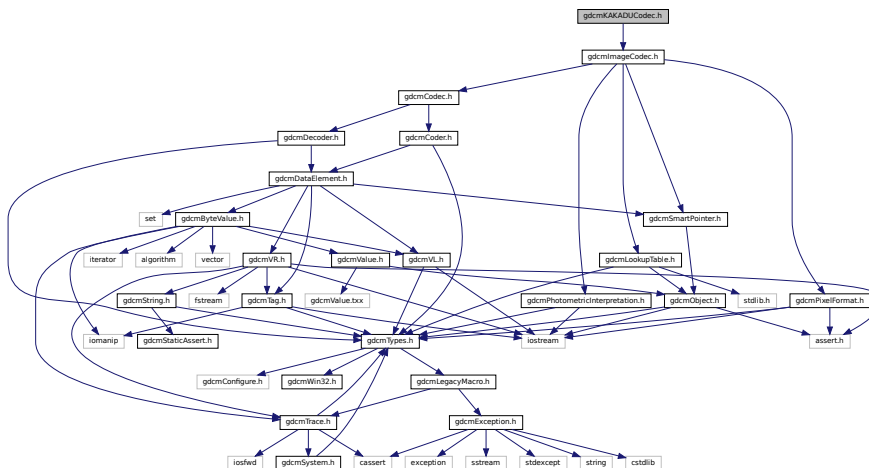
- class `gdcm::JSON`

- **gdcm**

- **gdcm**

Include dependency graph for gdcMkKADUCodec.h:

Include dependency graph for gdcMkKADUCodec.h:



## Classes

- class [gdcm::KAKADUCodec](#)  
*KAKADUCodec.*

## Namespaces

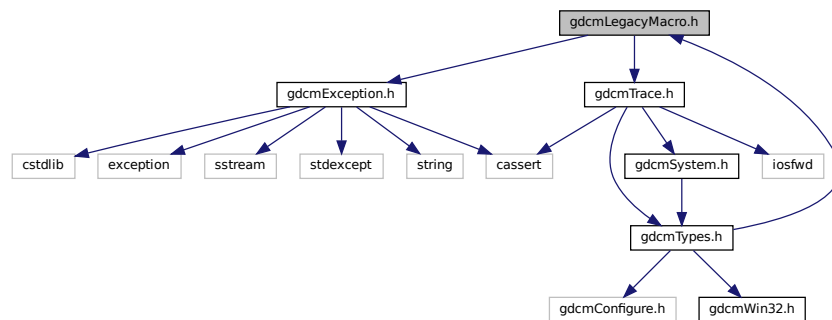
- [gdcm](#)

## 11.136 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.")`

### 11.136.1 Macro Definition Documentation

### 11.136.1.1 GDCM\_LEGACY

```
#define GDCM_LEGACY(  
    method ) method;
```

### 11.136.1.2 GDCM\_LEGACY\_BODY

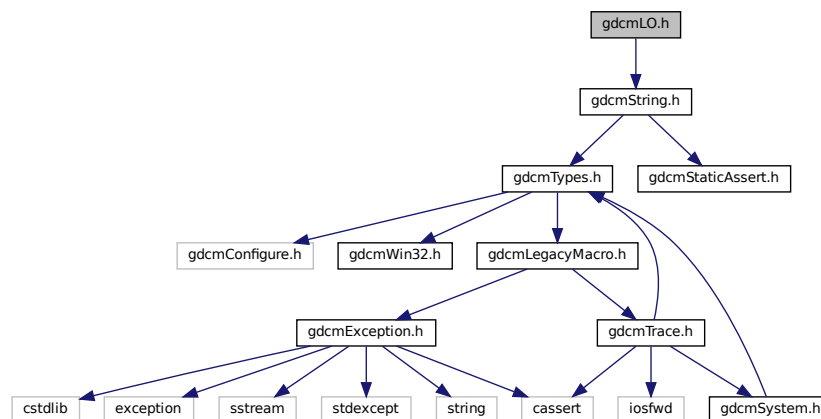
```
#define GDCM_LEGACY_BODY(  
    method,  
    version ) gdcmWarningMacro(#method " was deprecated for " version " and will be  
removed in a future version.")
```

### 11.136.1.3 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.)
```

## 11.137 gdcmlO.h File Reference

```
#include "gdcmlString.h"  
Include dependency graph for gdcmlO.h:
```



## Classes

- class [gdcm::LO](#)  
*LO.*

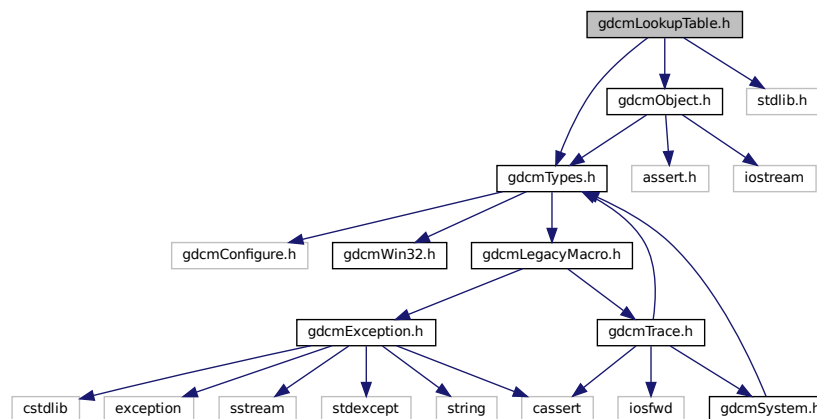
## Namespaces

- [gdcm](#)

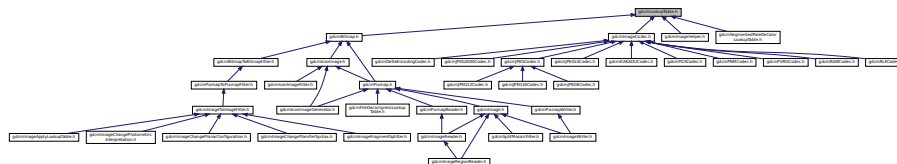
## 11.138 gdcmLookupTable.h File Reference

```
#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.

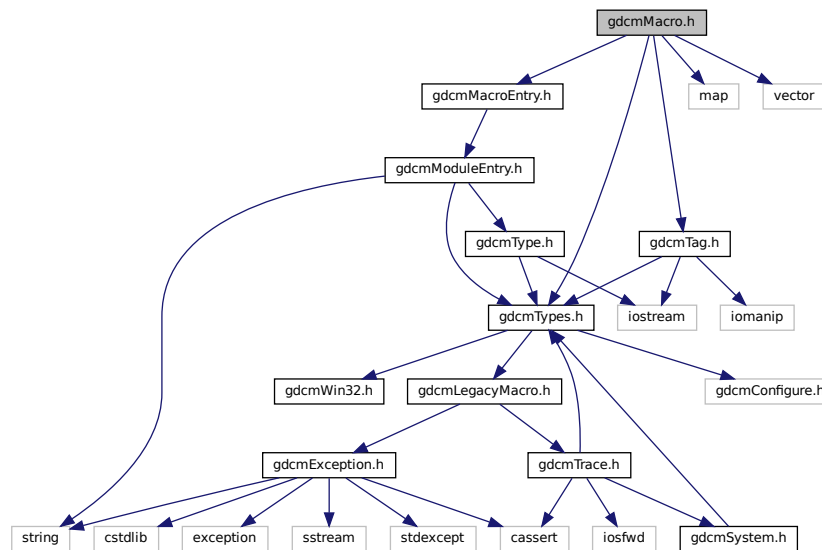
## Namespaces

- [gdcm](#)

## 11.139 gdcmMacro.h File Reference

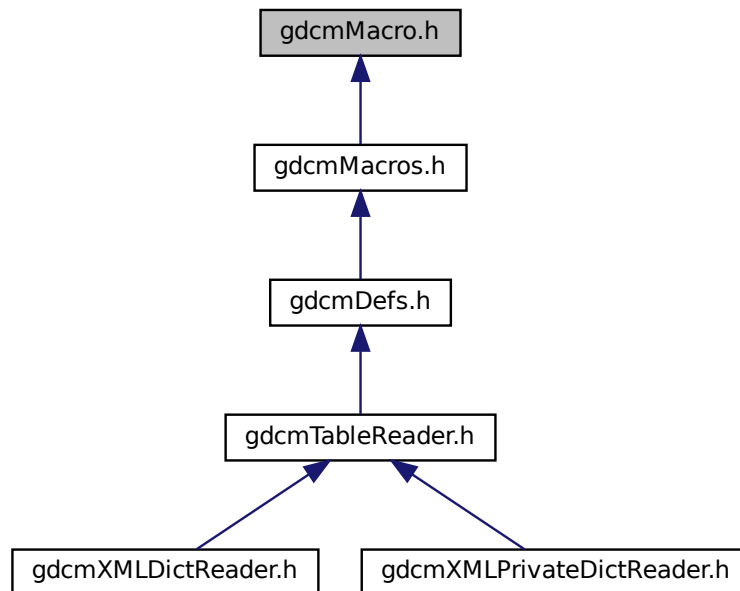
```
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmMacroEntry.h"
#include <map>
#include <vector>
```

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

- [gdcm](#)

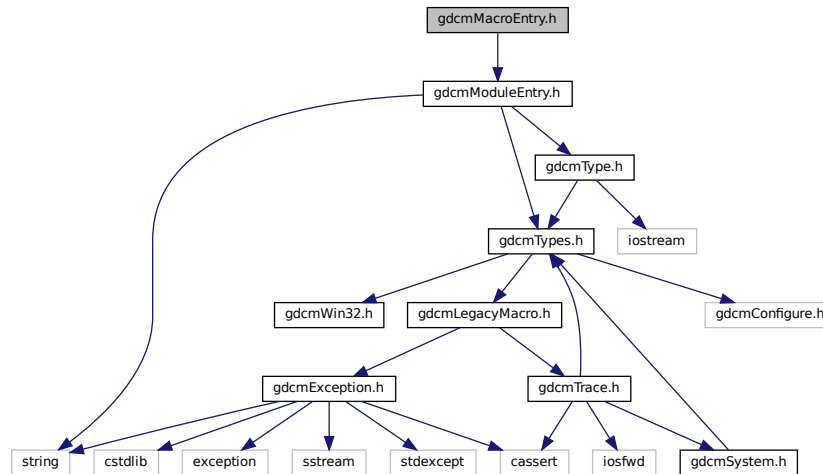
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Macro &_val)`

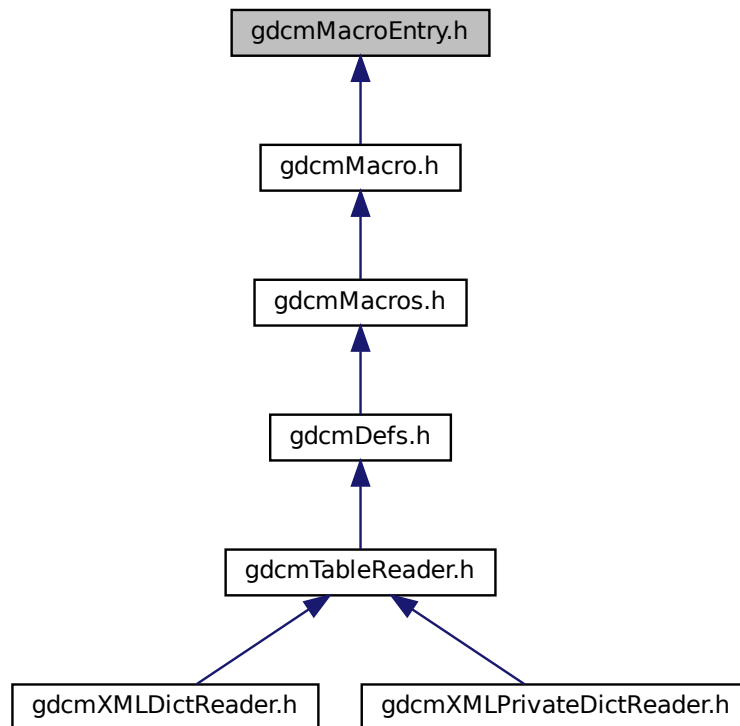
## 11.140 gdcmMacroEntry.h File Reference

```
#include "gdcmModuleEntry.h"
```

Include dependency graph for gdcmMacroEntry.h:



This graph shows which files directly or indirectly include this file:



## Macros

- #define [GDCMMACROENTRY\\_H](#)

### 11.140.1 Macro Definition Documentation

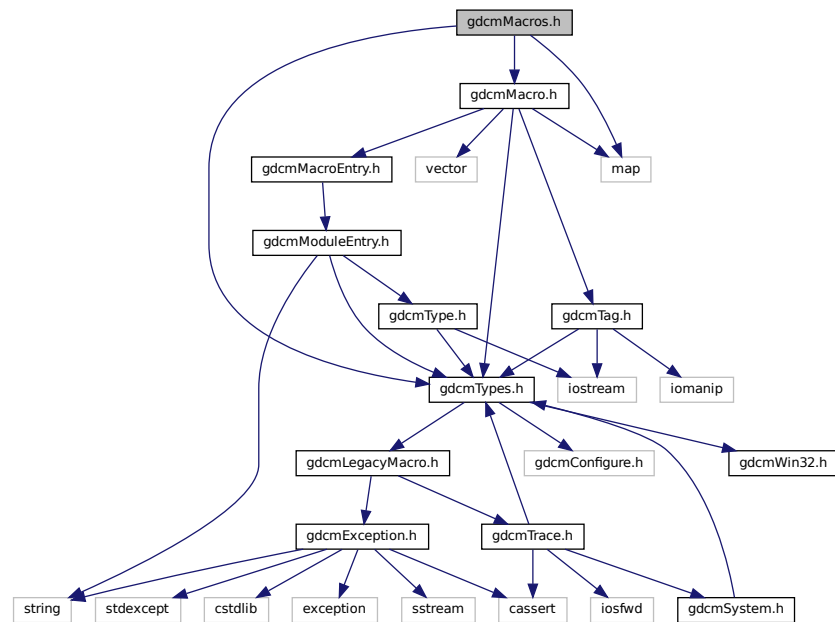
#### 11.140.1.1 GDCMMACROENTRY\_H

```
#define GDCMMACROENTRY_H
```

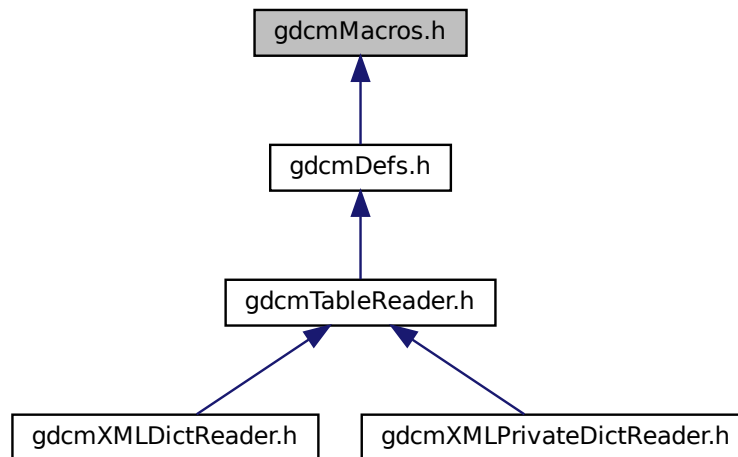
## 11.141 gdcmMacros.h File Reference

```
#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

- [gdcm](#)

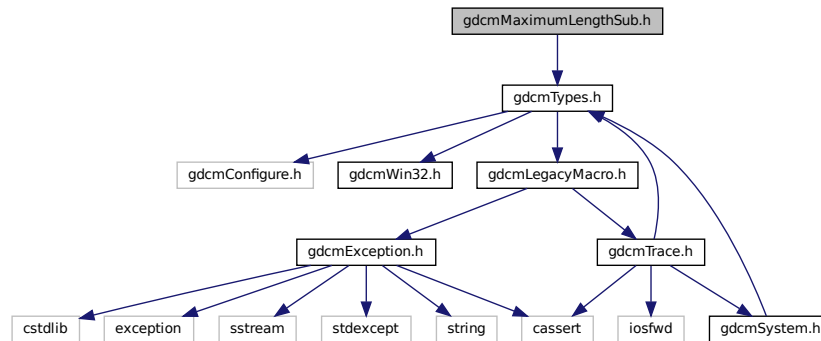
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Macros &_val)`

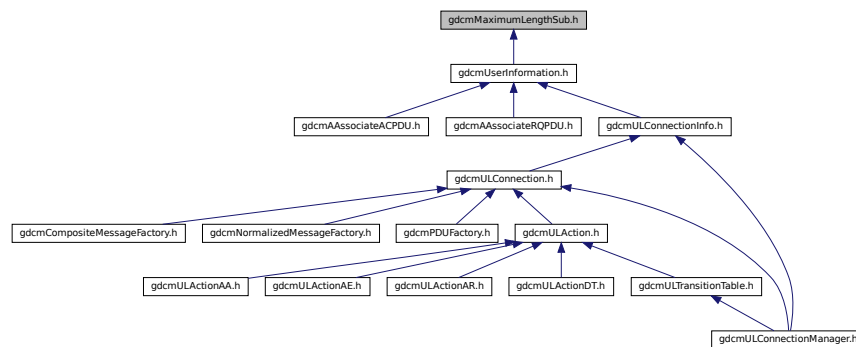
## 11.142 gdcmMaximumLengthSub.h File Reference

```
#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.*

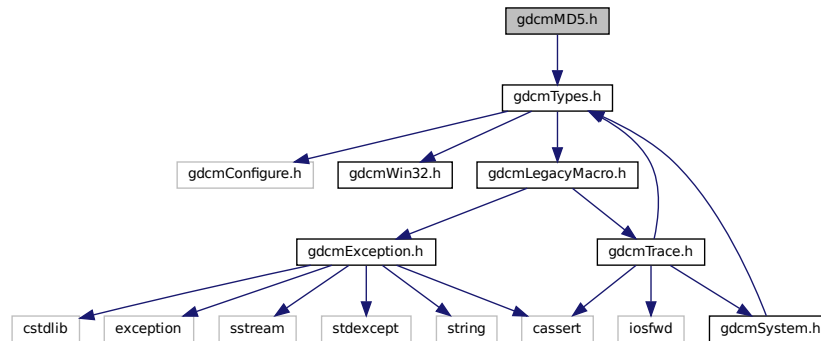
## Namespaces

- `gdcm`
- `gdcm::network`

## 11.143 gdcmMD5.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmMD5.h:



### Classes

- class [gdcm::MD5](#)  
Class for [MD5](#).

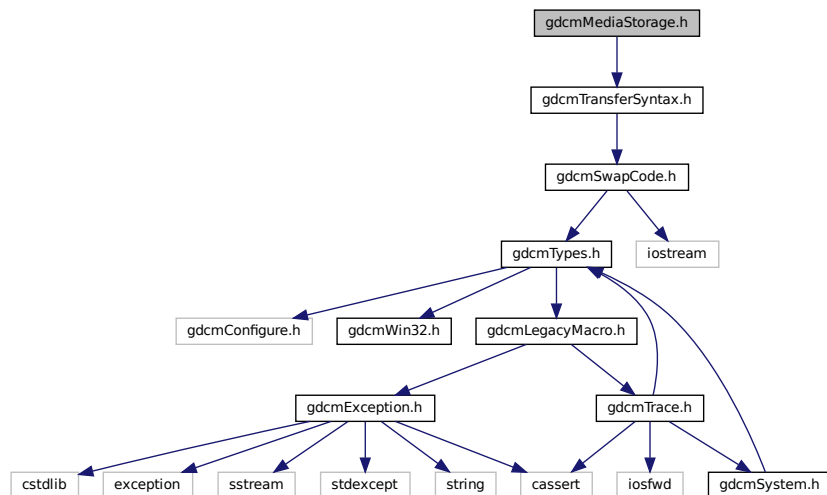
### Namespaces

- [gdcm](#)

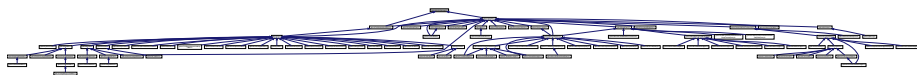
## 11.144 gdcmMediaStorage.h File Reference

```
#include "gdcmTransferSyntax.h"
```

Include dependency graph for `gdcmMediaStorage.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::MediaStorage`  
*MediaStorage.*

## Namespaces

- `gdcm`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const MediaStorage &ms)`

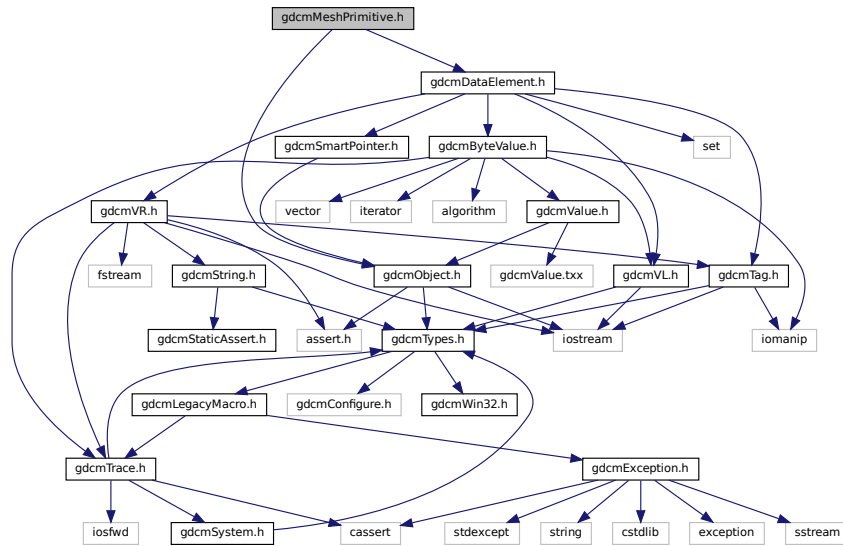


## 11.145 gdcmMeshPrimitive.h File Reference

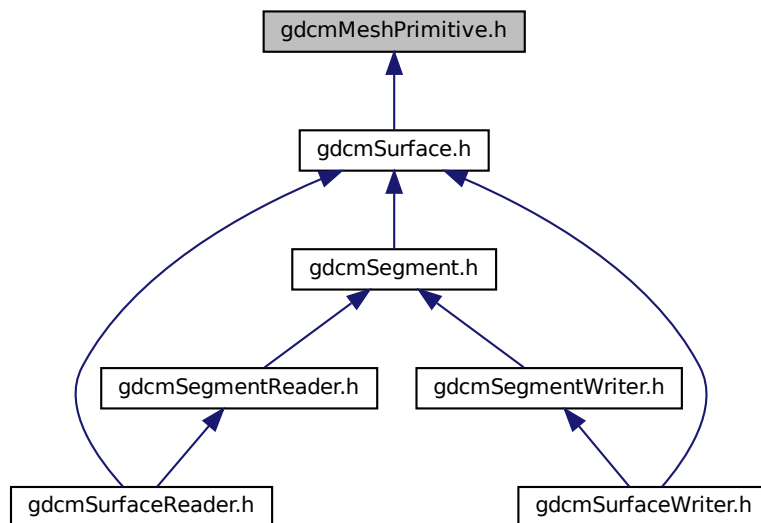
```
#include <gdcmObject.h>
```

```
#include <gdcmDataElement.h>
```

Include dependency graph for gdcmMeshPrimitive.h:



This graph shows which files directly or indirectly include this file:

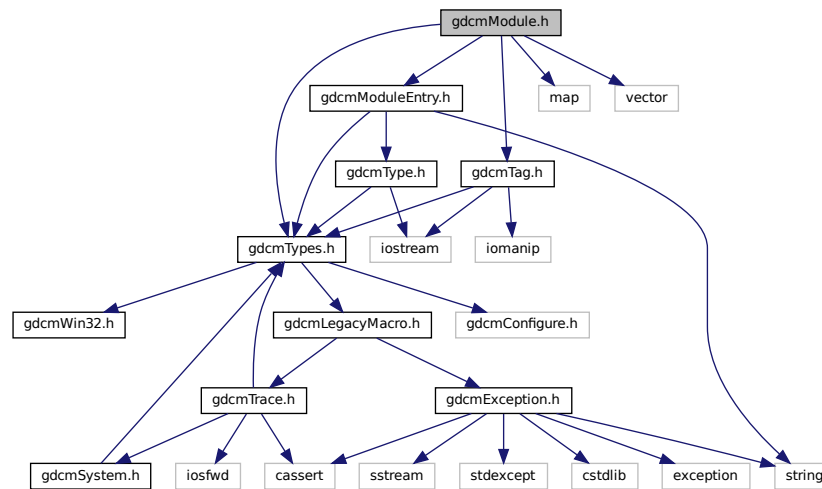




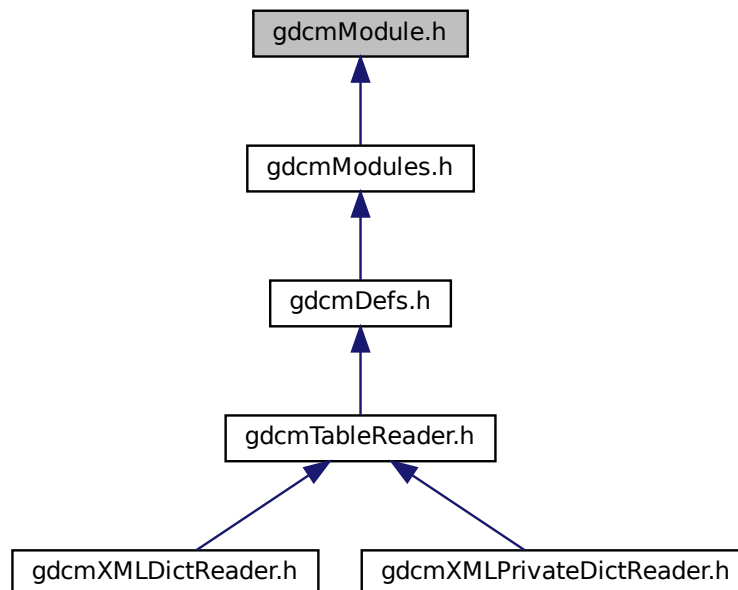


```
#include <vector>
```

Include dependency graph for gdcmModule.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::Module](#)  
*Class for representing a [Module](#).*

## Namespaces

- [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Module &_val)`

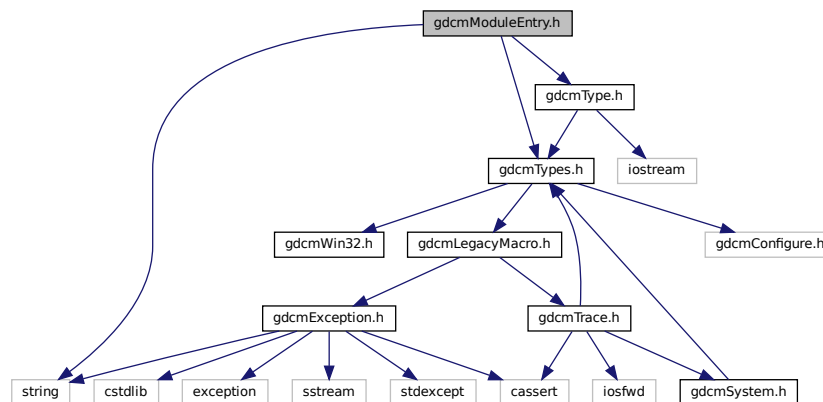
## 11.149 gdcmModuleEntry.h File Reference

```
#include "gdcmTypes.h"
```

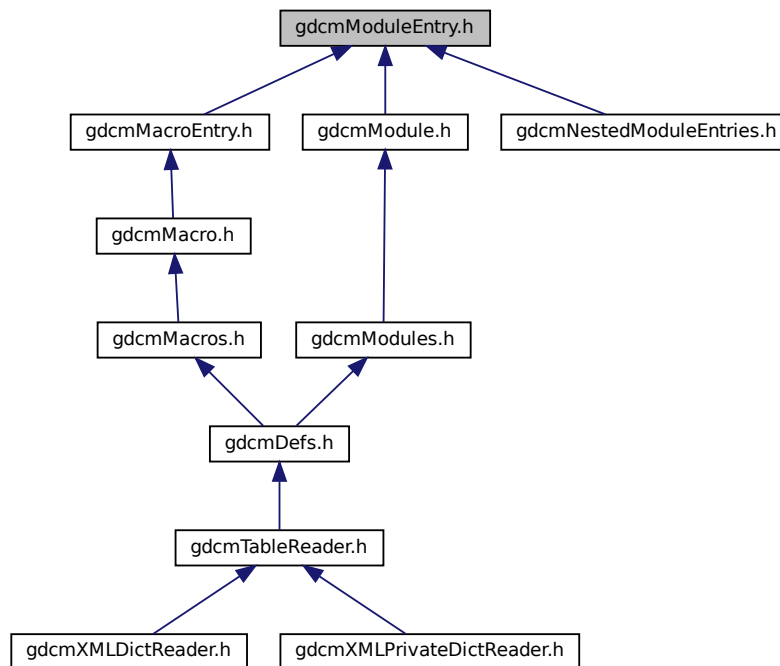
```
#include "gdcmType.h"
```

```
#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

- [gdc](#)

## Typedefs

- typedef ModuleEntry [gdcm::MacroEntry](#)

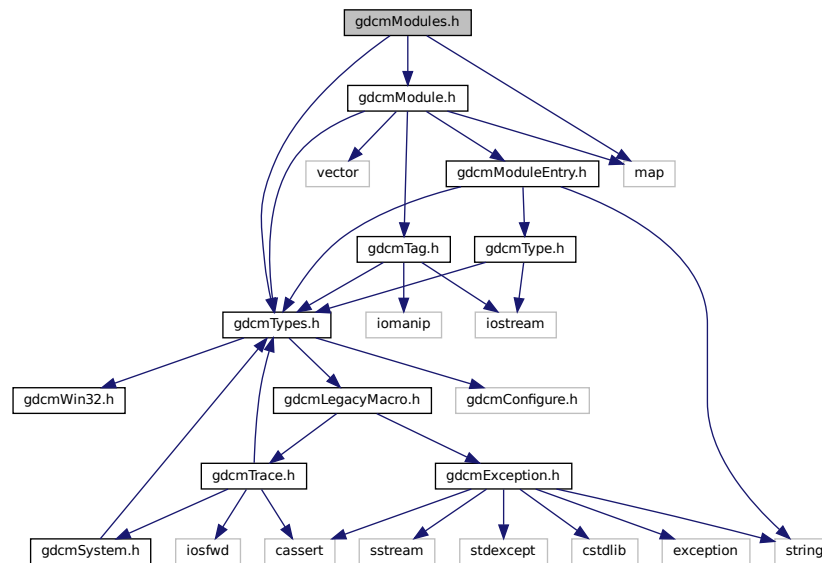
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const ModuleEntry &_val)`

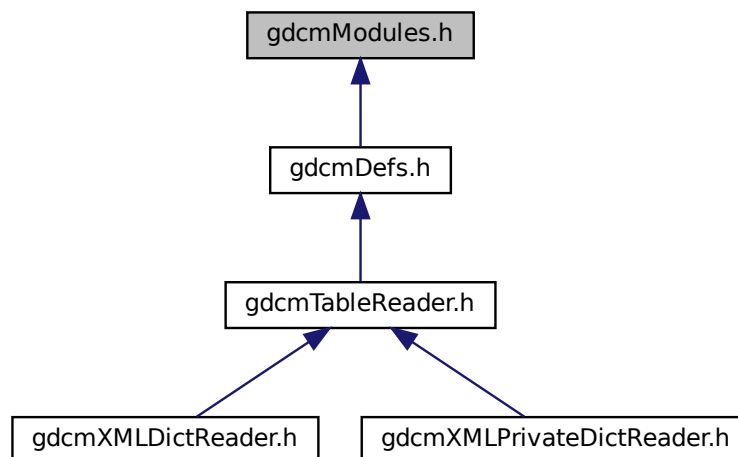
## 11.150 gdcmModules.h File Reference

```
#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:











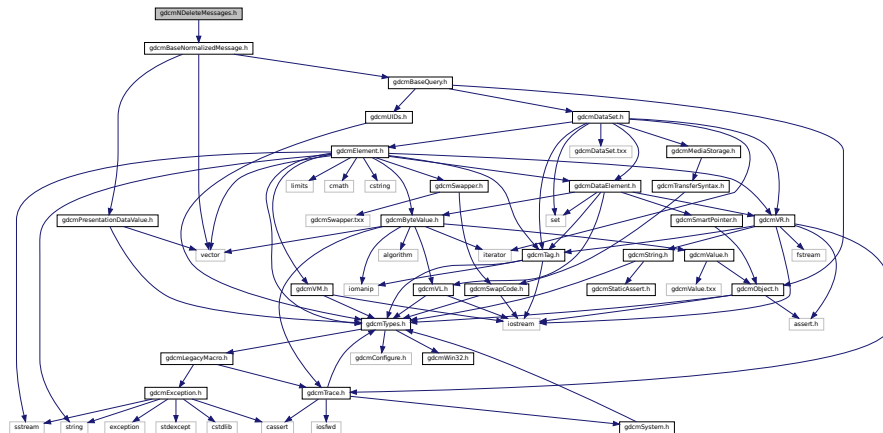


[illegible]

- class `gdcm::network::NCreateRQ`  
`NCreateRQ`.
- class `gdcm::network::NCreateRSP`  
`NCreateRSP` this file defines the messages for the ncreate action.

- `gdcm`
- `gdcm::network`

```
#include "gdcmBaseNormalizedMessage.h"
Include dependency graph for gdcmNDeleteMessages.h:
```



## Classes

- class [gdcm::network::NDeleteRQ](#)  
*NDeleteRQ.*
- class [gdcm::network::NDeleteRSP](#)  
*NDeleteRSP* this file defines the messages for the ndelete action.

## Namespaces

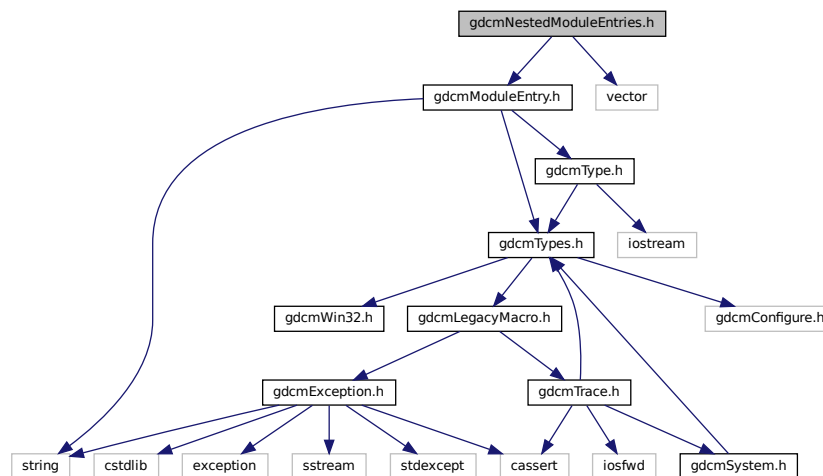
- [gdcm](#)
- [gdcm::network](#)

## 11.157 gdcmNestedModuleEntries.h File Reference

```
#include "gdcmModuleEntry.h"
```

```
#include <vector>
```

Include dependency graph for gdcmNestedModuleEntries.h:



## Classes

- class [gdcm::NestedModuleEntries](#)  
*Class for representing a [NestedModuleEntries](#).*

## Namespaces

- [gdcm](#)

## Typedefs

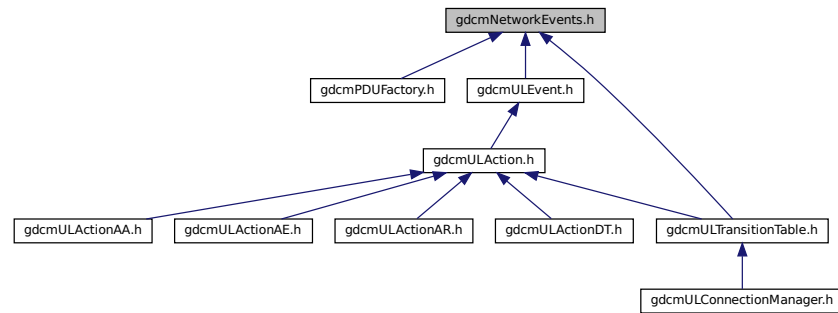
- typedef NestedModuleEntries [gdcm::NestedMacroEntries](#)

## Functions

- std::ostream & [gdcm::operator<<](#) (std::ostream &\_os, const NestedModuleEntries &\_val)

## 11.158 gdcmNetworkEvents.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- [gdcm](#)
- [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 }

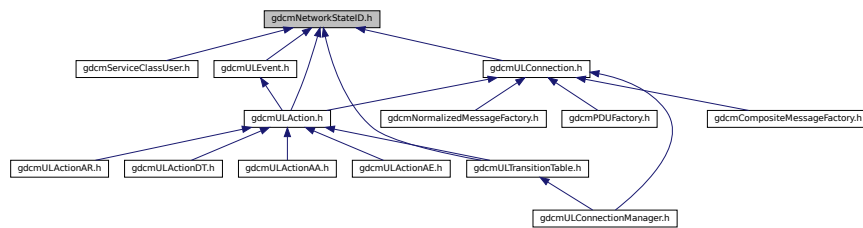
```

## Variables

- const int `gdcm::network::cMaxEventID` = eEventDoesNotExist

## 11.159 gdcmNetworkStateID.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- `gdcm`
- `gdcm::network`

## Enumerations

- enum `gdcm::network::EStateID` {  
`gdcm::network::eStaDoesNotExist` = 0 ,  
`gdcm::network::eSta1Idle` = 1 ,  
`gdcm::network::eSta2Open` = 2 ,  
`gdcm::network::eSta3WaitLocalAssoc` = 4 ,  
`gdcm::network::eSta4LocalAssocDone` = 8 ,  
`gdcm::network::eSta5WaitRemoteAssoc` = 16 ,  
`gdcm::network::eSta6TransferReady` = 32 ,  
`gdcm::network::eSta7WaitRelease` = 64 ,  
`gdcm::network::eSta8WaitLocalRelease` = 128 ,  
`gdcm::network::eSta9ReleaseCollisionRqLocal` = 256 ,  
`gdcm::network::eSta10ReleaseCollisionAc` = 512 ,  
`gdcm::network::eSta11ReleaseCollisionRq` = 1024 ,  
`gdcm::network::eSta12ReleaseCollisionAcLocal` = 2048 ,  
`gdcm::network::eSta13AwaitingClose` = 4096 }

## Functions

- int `gdcn::network::GetStateIndex` (EStateID inState)

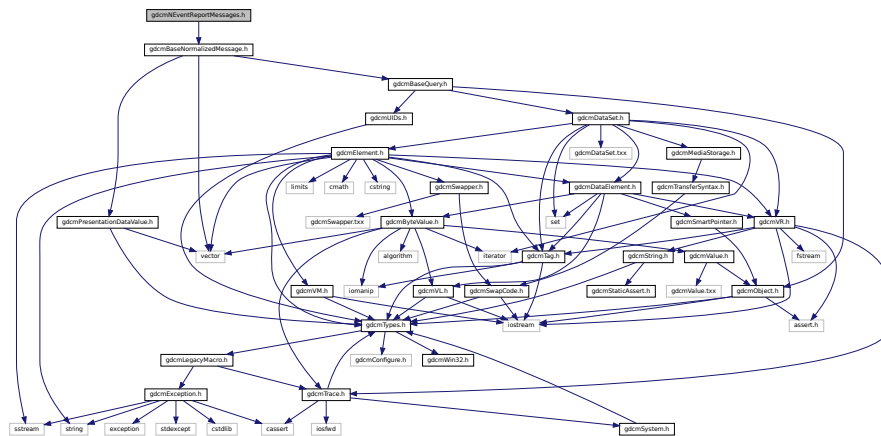
## Variables

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

## 11.160 gdcnNEventReportMessages.h File Reference

```
#include "gdcmbaseNormalizedMessage.h"
```

Include dependency graph for gdcmlEventReportMessages.h:



## Classes

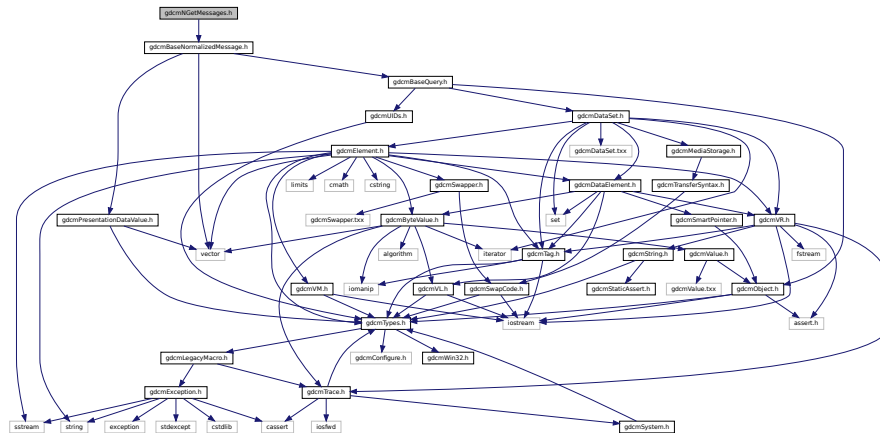
- class `gdcm::network::NEventReportRQ`  
`NEventReportRQ`.
- class `gdcm::network::NEventReportRSP`  
`NEventReportRSP` *this file defines the messages for the neventreport action.*

## Namespaces

- gdc
- gdc::network



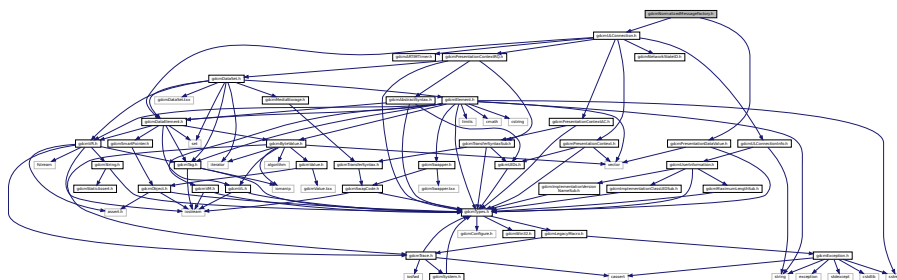
```
#include "gdcmBaseNormalizedMessage.h"
Include dependency graph for gdcmNGetMessages.h:
```



- class `gdcmm::network::NGetRQ`  
*NGetRQ.*
- class `gdcmm::network::NGetRSP`  
*NGetRSP* this file defines the messages for the *nget* action.

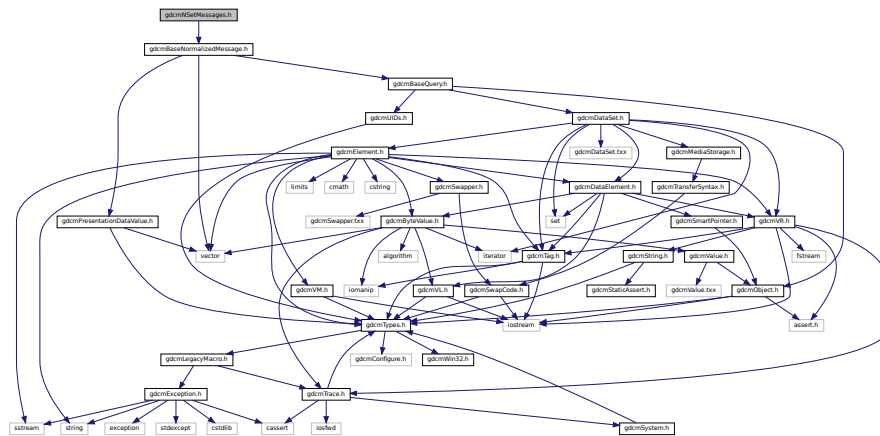
- `gdc`
- `gdc::network`

```
#include "gdcmPresentationDataValue.h"
#include "gdcmULConnection.h"
Include dependency graph for gdcmNormalizedMessageFactory.h:
```





```
#include "gdcmBaseNormalizedMessage.h"
Include dependency graph for gdcmNSetMessages.h:
```



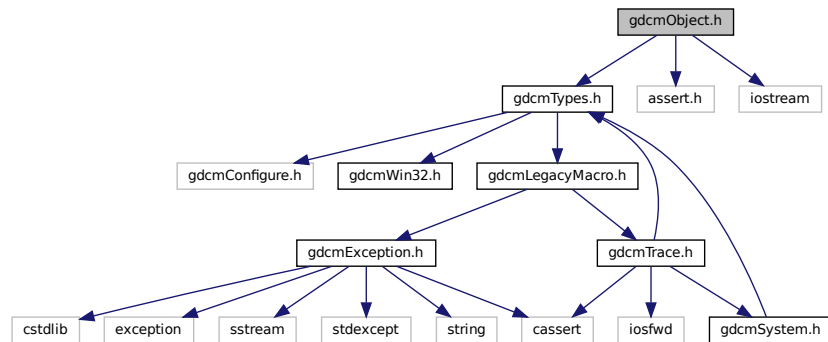
- class `gdcm::network::NSetRQ`  
`NSetRQ`.
- class `gdcm::network::NSetRSP`  
`NSetRSP` this file defines the messages for the nset action.

- gdc
- gdc::network

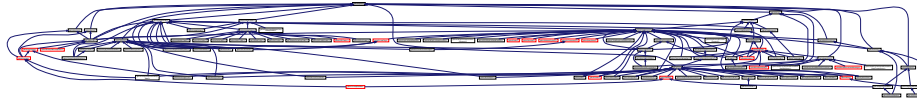
```
#include "gdcmTypes.h"
#include <assert.h>
```

```
#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

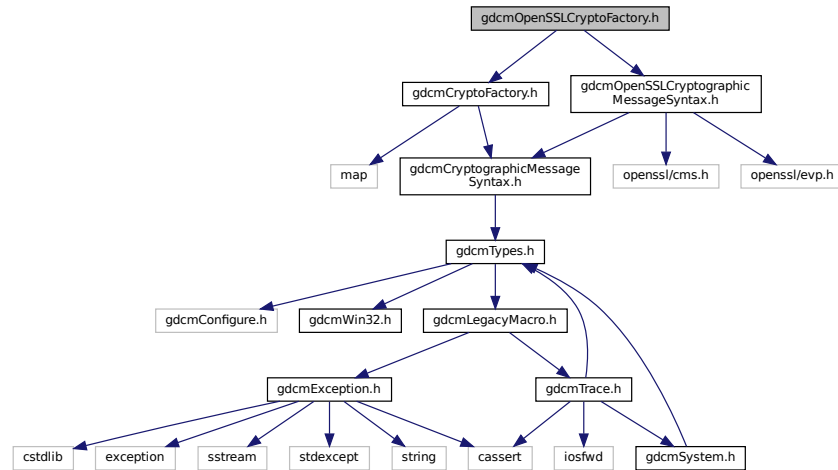
- `gdcm`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Object &obj)`

## 11.166 gdcmOpenSSLCryptoFactory.h File Reference

```
#include "gdcmCryptoFactory.h"
#include "gdcmOpenSSLCryptographicMessageSyntax.h"
Include dependency graph for gdcmOpenSSLCryptoFactory.h:
```



### Classes

- class [gdcm::OpenSSLCryptoFactory](#)

### Namespaces

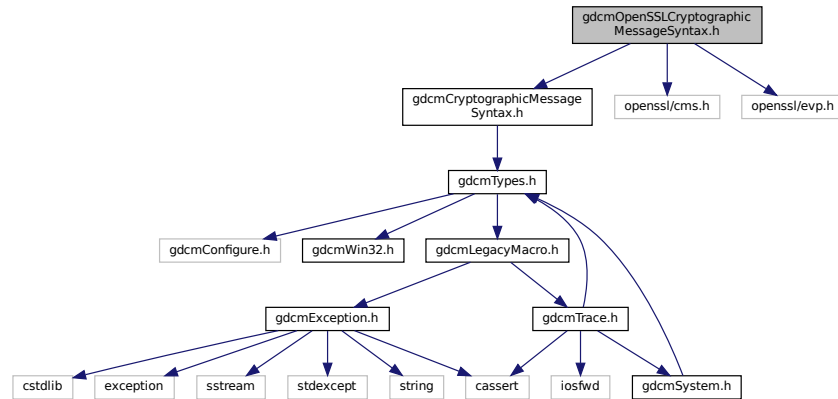
- [gdcm](#)

## 11.167 gdcmOpenSSLCryptographicMessageSyntax.h File Reference

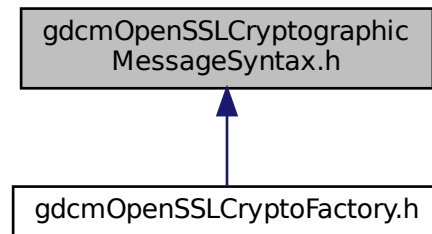
```
#include "gdcmCryptographicMessageSyntax.h"
#include <openssl/cms.h>
```

```
#include <openssl/evp.h>
```

Include dependency graph for `gdcmOpenSSLCryptographicMessageSyntax.h`:



This graph shows which files directly or indirectly include this file:



## Classes

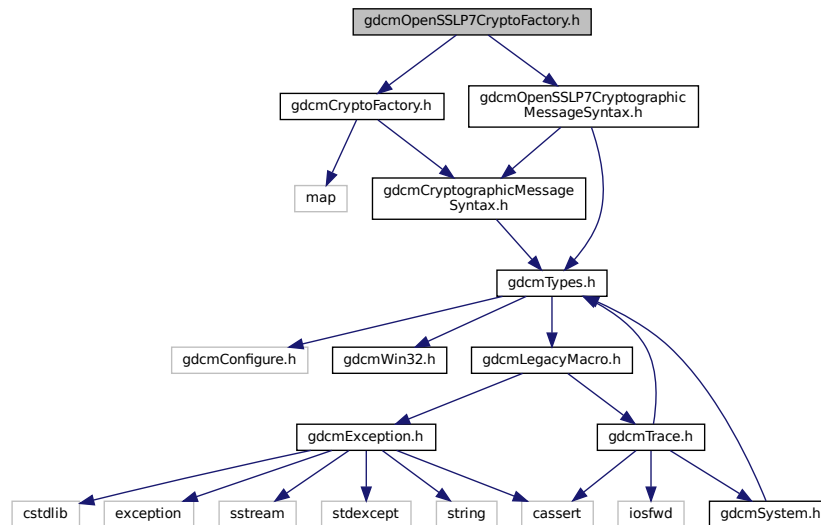
- class [gdcm::OpenSSLCryptographicMessageSyntax](#)

## Namespaces

- [gdcm](#)

## 11.168 gdcmOpenSSL7CryptoFactory.h File Reference

```
#include "gdcmCryptoFactory.h"
#include "gdcmOpenSSL7CryptographicMessageSyntax.h"
Include dependency graph for gdcmOpenSSL7CryptoFactory.h:
```



### Classes

- class [gdcm::OpenSSL7CryptoFactory](#)

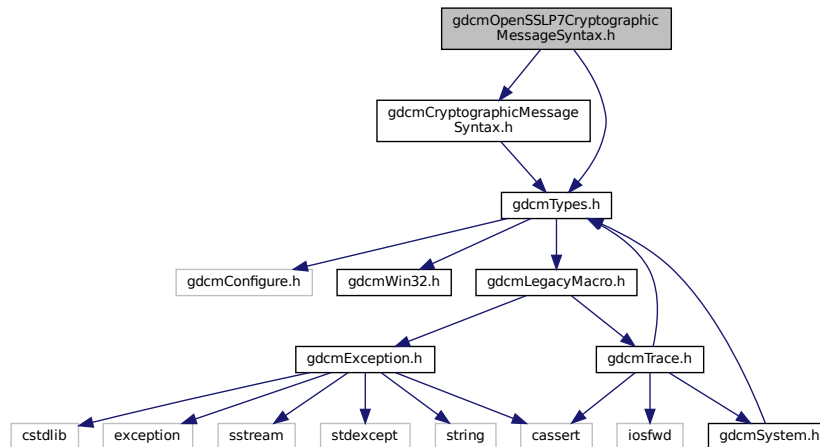
### Namespaces

- [gdcm](#)

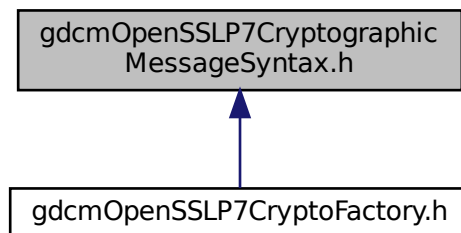
## 11.169 gdcmOpenSSL7CryptographicMessageSyntax.h File Reference

```
#include "gdcmCryptographicMessageSyntax.h"
#include "gdcmTypes.h"
```

Include dependency graph for `gdcOpenSSLP7CryptographicMessageSyntax.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdc::OpenSSLP7CryptographicMessageSyntax](#)

## Namespaces

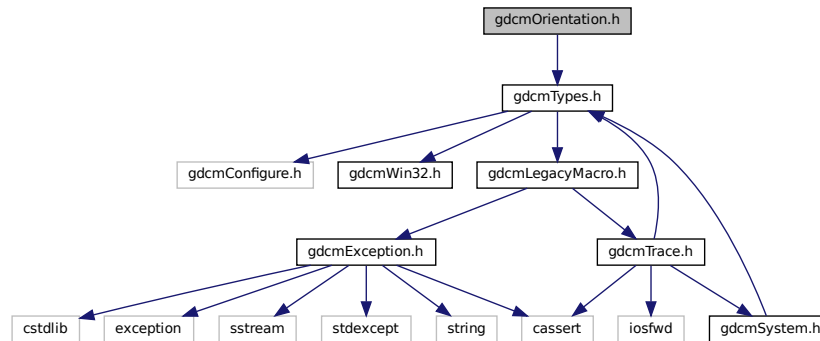
- [gdc](#)



## 11.170 gdcmOrientation.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmOrientation.h:



### Classes

- class [gdcm::Orientation](#)  
class to handle [Orientation](#)

### Namespaces

- [gdcm](#)

### Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Orientation &o)`

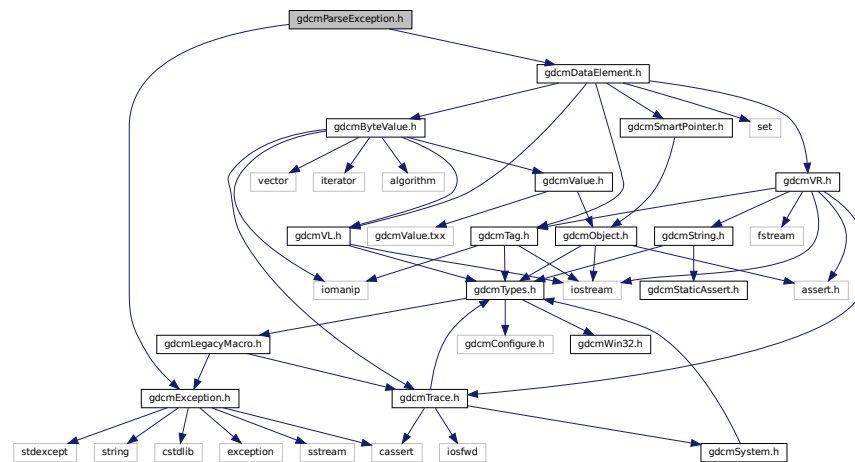
## 11.171 gdcmOverlay.h File Reference

```
#include "gdcmTypes.h"
```

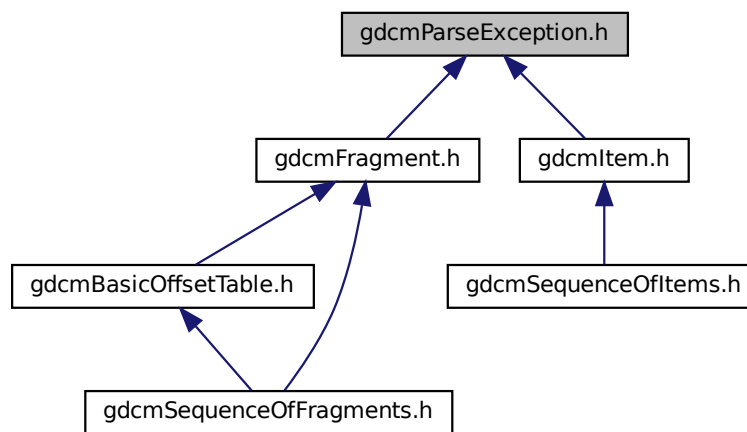
```
#include "gdcmObject.h"
```



Include dependency graph for gdcmParseException.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::ParseException](#)  
*ParseException* Standard exception handling object.

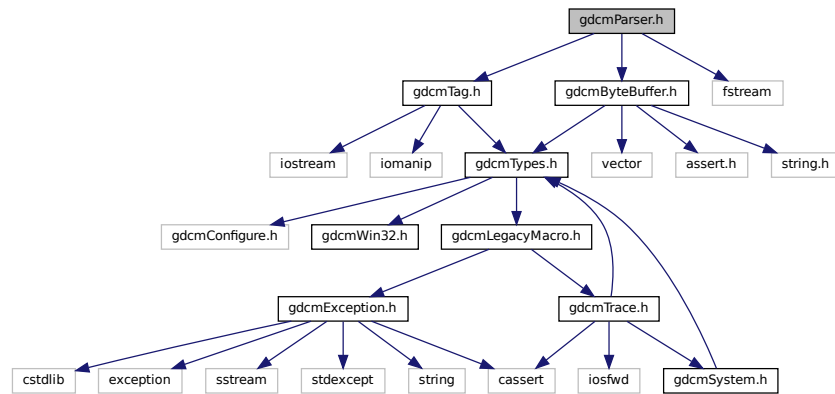
## Namespaces

- [gdcm](#)

## 11.173 gdcmParser.h File Reference

```
#include "gdcmTag.h"
#include "gdcmByteBuffer.h"
#include <fstream>
```

Include dependency graph for gdcmParser.h:



### Classes

- class [gdcm::Parser](#)  
*Parser ala XML\_Parser from expat (SAX)*

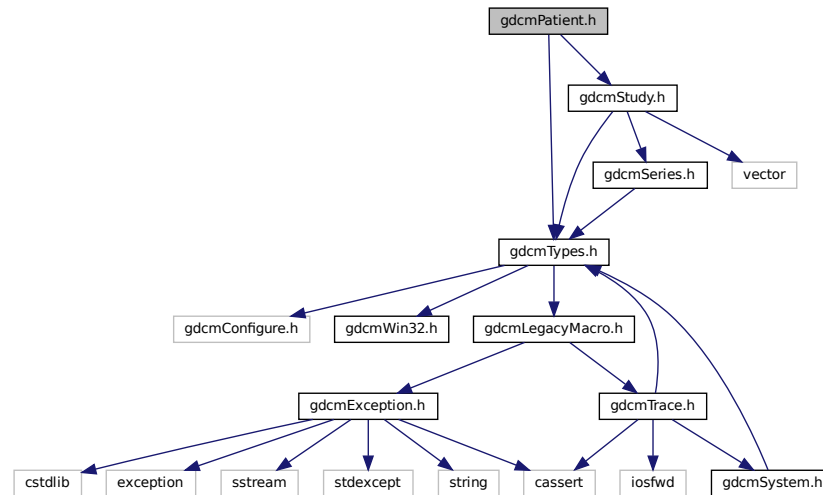
### Namespaces

- [gdcm](#)

## 11.174 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

- [gdcm](#)

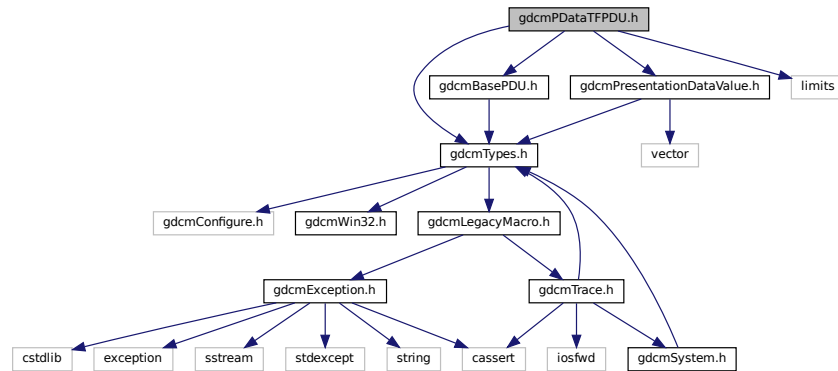
## 11.175 gdcmPDataTFPDU.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmPresentationDataValue.h"
#include "gdcmBasePDU.h"
#include <limits>

```

Include dependency graph for `gdcmPDataTFPDU.h`:



## Classes

- class `gdcm::network::PDataTFPDU`  
*PDataTFPDU*.

## Namespaces

- `gdcm`
- `gdcm::network`

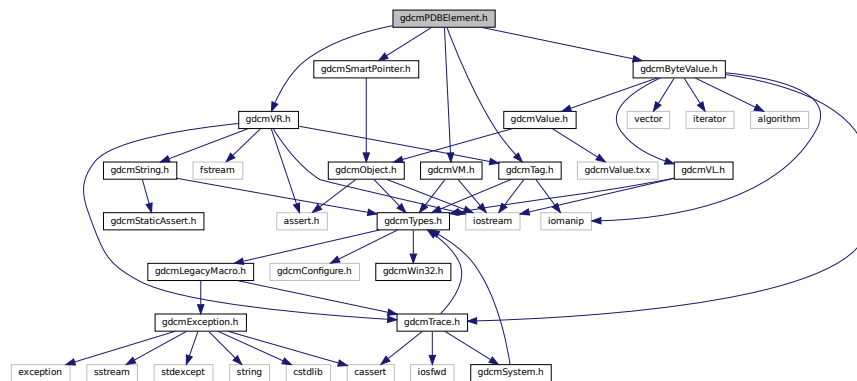
## 11.176 gdcmPDBelement.h File Reference

```

#include "gdcmTag.h"
#include "gdcmVM.h"
#include "gdcmVR.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"

```

Include dependency graph for `gdcmPDBelement.h`:



```
graph BT; gdcnPDBHeader.h --> gdcnPDBElement.h
```

- class `gdcm::PDBElement`  
*Class to represent a PDB Element.*

- **gdcm**

- `std::ostream & gdcmm::operator<< (std::ostream &os, const PDBelement &val)`

```
#include "gdcmTypes.h"
#include "gdcmDataSet.h"
#include "gdcmPDBelement.h"
```

[illegible]

## Classes

- class [gdcm::PDBHeader](#)  
*Class for PDBHeader.*

## Namespaces

- [gdcm](#)

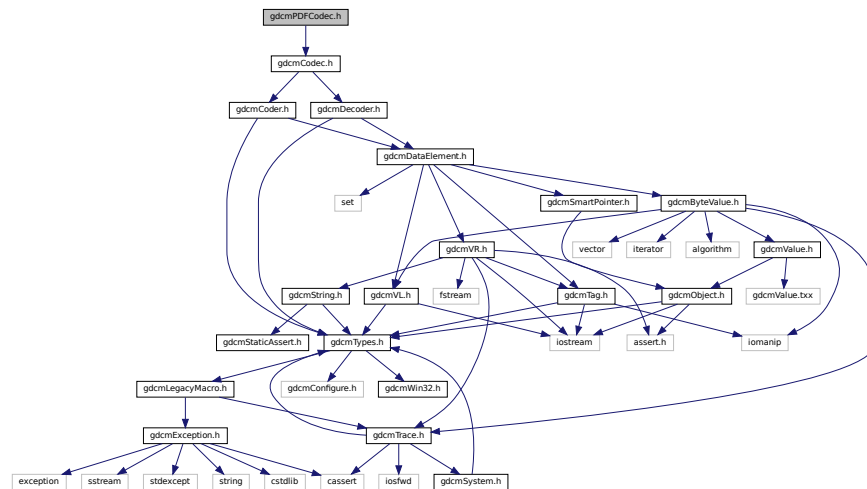
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const PDBHeader &d)`

## 11.178 gdcmPDFCodec.h File Reference

```
#include "gdcmCodec.h"
```

Include dependency graph for gdcmPDFCodec.h:



## Classes

- class [gdcm::PDFCodec](#)  
*PDFCodec class.*

## Namespaces

- [gdcm](#)



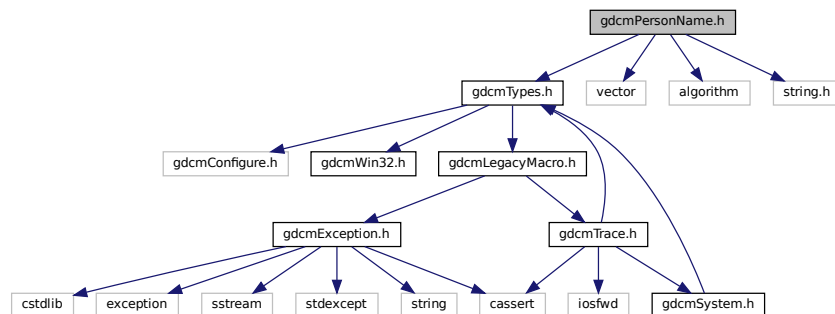
```
#include "gdcmTypes.h"
#include "gdcmNetworkEvents.h"
#include "gdcmULConnection.h"
#include "gdcmPresentationDataValue.h"
```

- class `gdcm::network::PDUFactory`  
*PDUFactory* basically, given an initial byte, construct the.

- `gdcm`
- `gdcm::network`

```
#include "gdcmTypes.h"
#include <vector>
#include <algorithm>
#include <string.h>
```

Include dependency graph for gdcmPersonName.h:



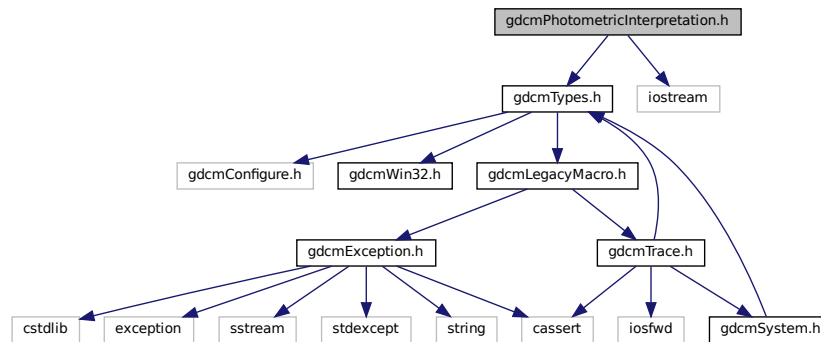


## 11.182 gdcmPhotometricInterpretation.h File Reference

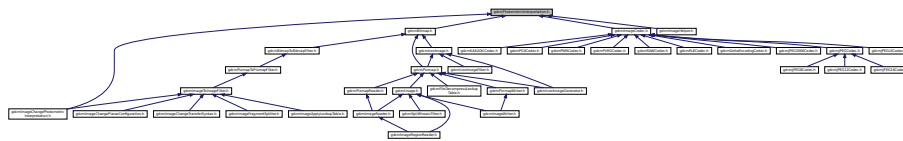
```
#include "gdcmTypes.h"
```

```
#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

- [gdcm](#)

### Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const PhotometricInterpretation &val)`

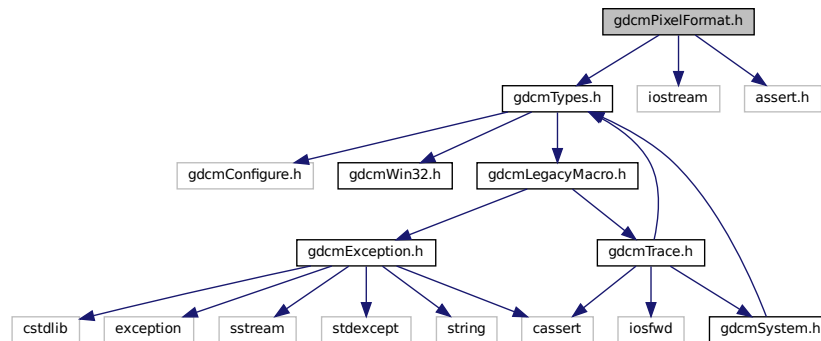
## 11.183 gdcmPixelFormat.h File Reference

```
#include "gdcmTypes.h"
```

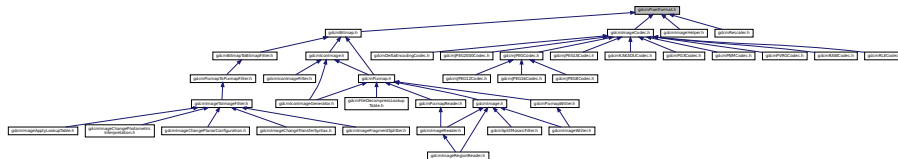
```
#include <iostream>
```

```
#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

- [gdcm](#)

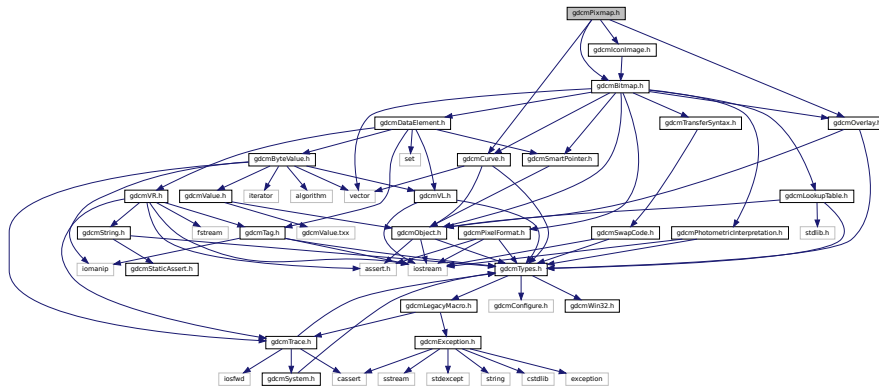
### Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const PixelFormat &pf)`

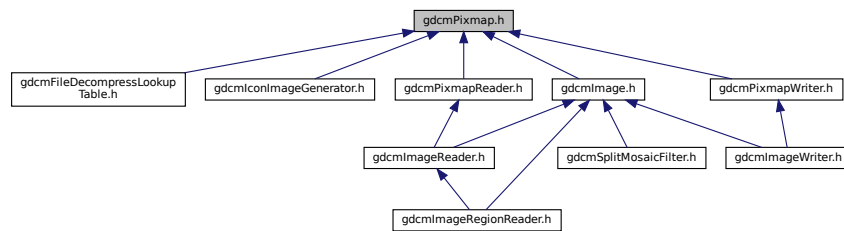
## 11.184 gdcmPixmap.h File Reference

```
#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

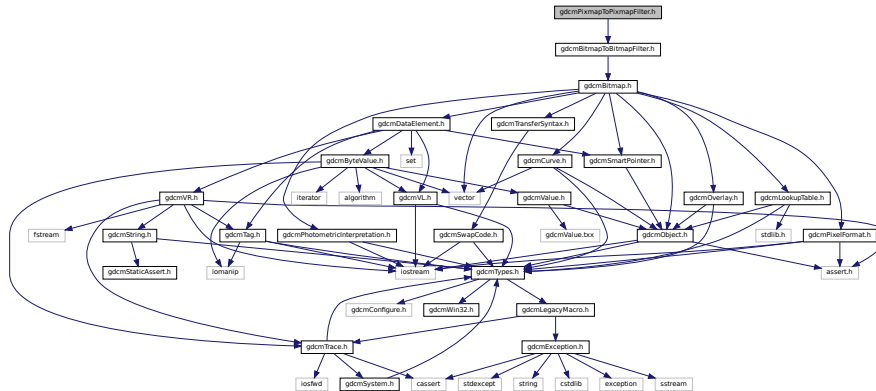
- [gdcm](#)



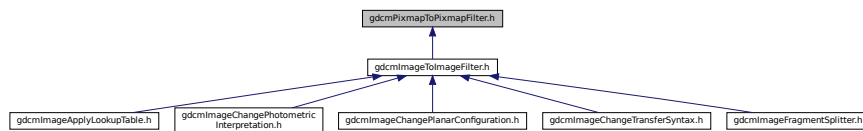
## 11.186 gdcmPixmapToPixmapFilter.h File Reference

```
#include "gdcmBitmapToBitmapFilter.h"
```

Include dependency graph for gdcmPixmapToPixmapFilter.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [gdcm::PixmapToPixmapFilter](#)  
*PixmapToPixmapFilter* class.

### Namespaces

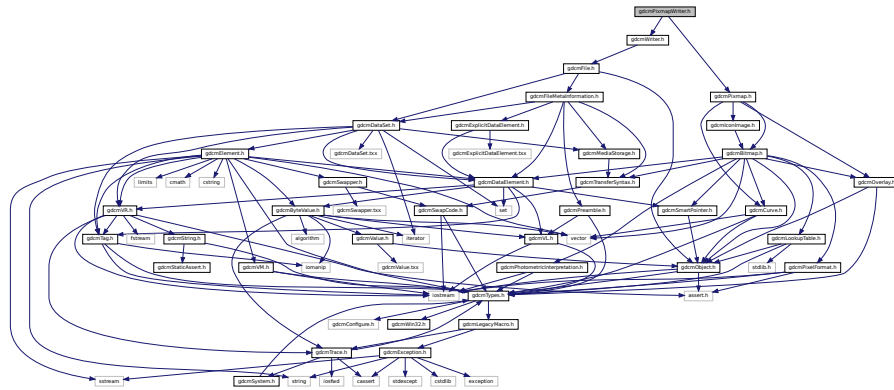
- [gdcm](#)

## 11.187 gdcmPixmapWriter.h File Reference

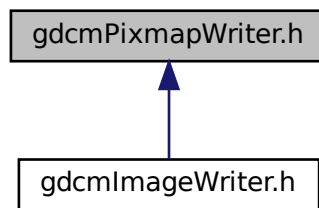
```
#include "gdcmWriter.h"
```

```
#include "gdcmPixmap.h"
```

Include dependency graph for gdcmPixmapWriter.h:



This graph shows which files directly or indirectly include this file:



## Classes

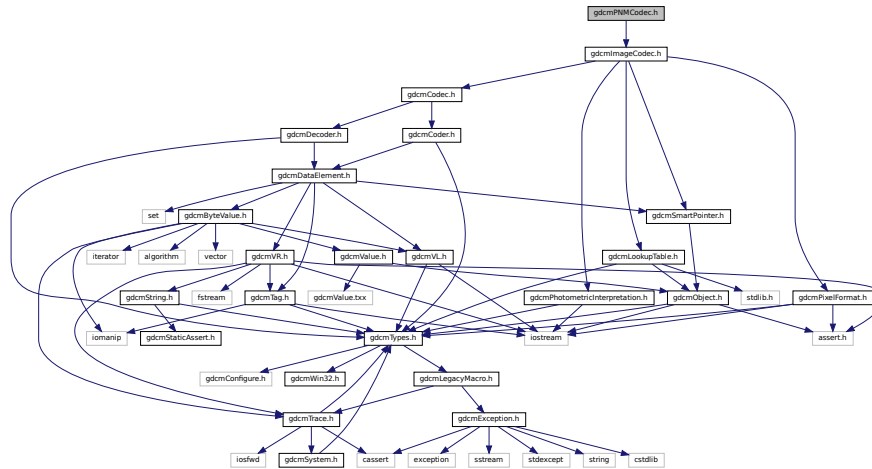
- class `gdcm::PixmapWriter`  
*PixmapWriter*.

## Namespaces

- gdc



```
#include "gdcImageCodec.h"
Include dependency graph for gdcPNMCodec.h:
```

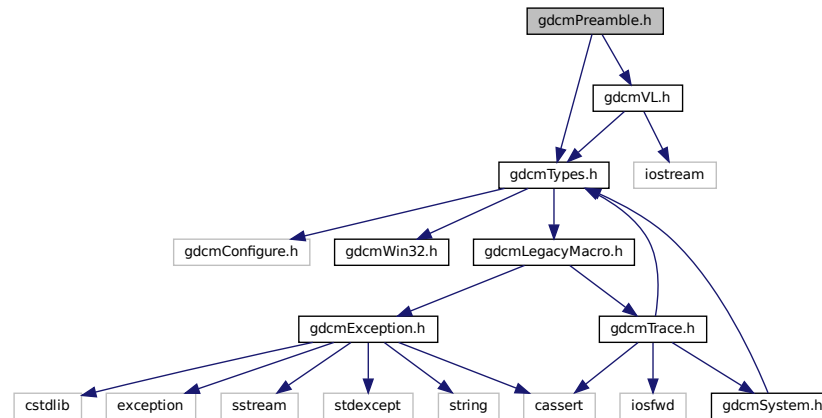


- class `gdcm::PNMCodec`  
*Class to do PNM.*

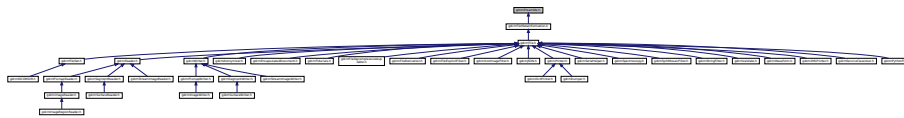
- **gdcm**

```
#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

- [gdcm](#)

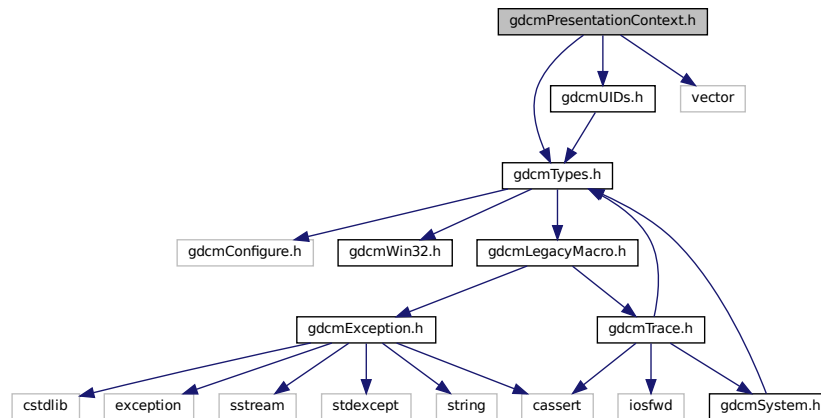
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Preamble &val)`

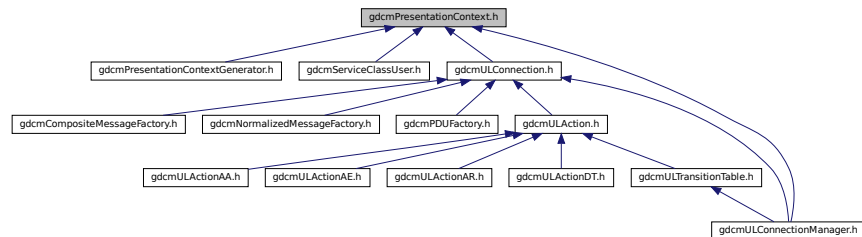
## 11.190 gdcmPresentationContext.h File Reference

```
#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

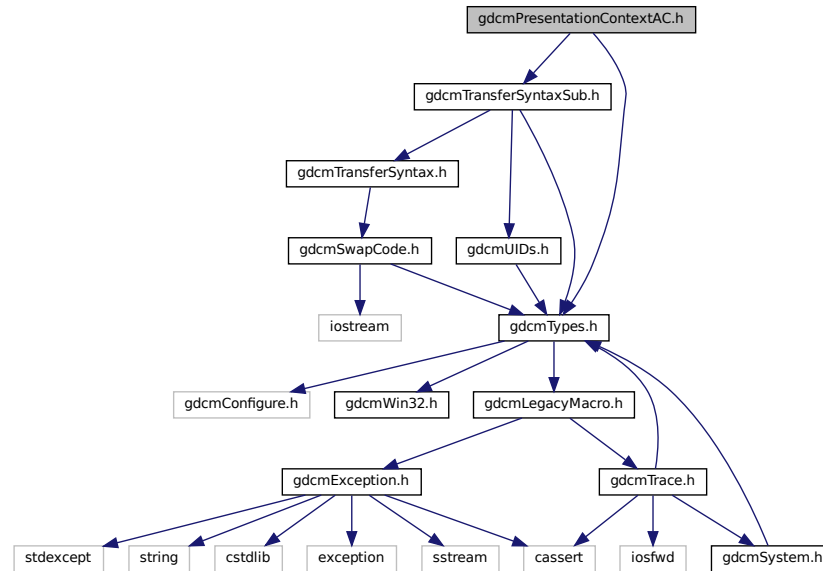
- class [gdcm::PresentationContext](#)  
*PresentationContext.*

### Namespaces

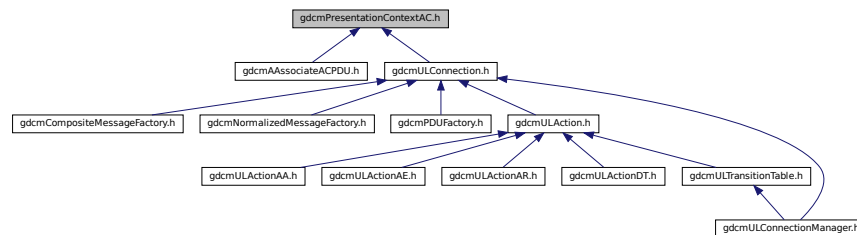
- [gdcm](#)

## 11.191 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

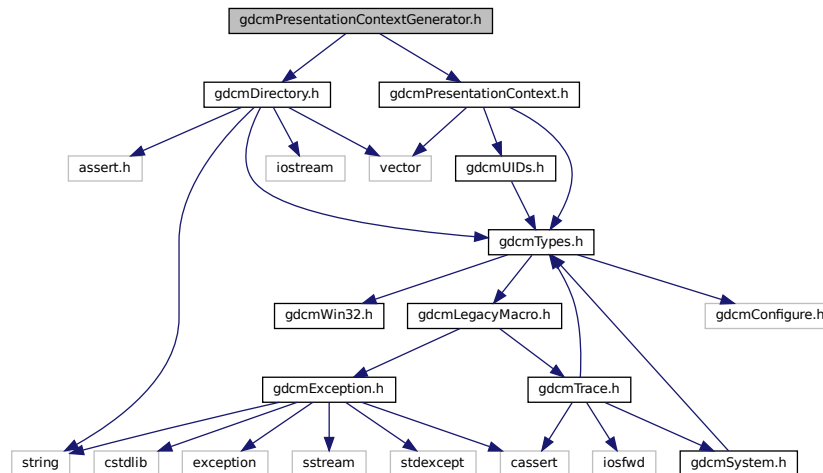
- [gdcm](#)
- [gdcm::network](#)

## 11.192 gdcmPresentationContextGenerator.h File Reference

```
#include "gdcmDirectory.h"
```

```
#include "gdcmPresentationContext.h"
```

Include dependency graph for gdcmPresentationContextGenerator.h:



### Classes

- class [gdcm::PresentationContextGenerator](#)  
*PresentationContextGenerator.*

### Namespaces

- [gdcm](#)

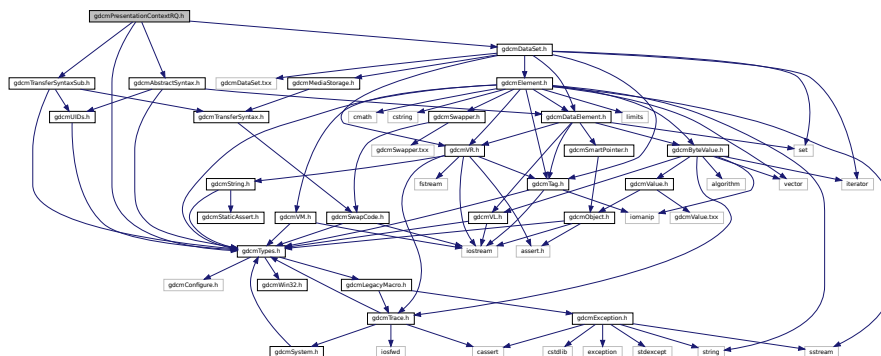
## 11.193 gdcmPresentationContextRQ.h File Reference

```
#include "gdcmTypes.h"
```

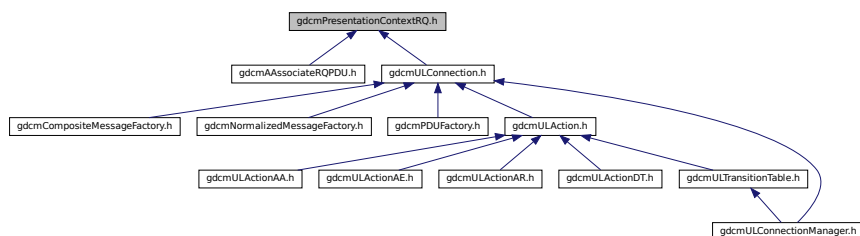
```
#include "gdcmAbstractSyntax.h"
```

```
#include "gdcmTransferSyntaxSub.h"
```

Include dependency graph for gdcmPresentationContextRQ.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcmm::network::PresentationContextRQ`  
*PresentationContextRQ.*

## Namespaces

- `gdcm`
- `gdcm::network`

## 11.194 gdcmpresentationdatavalue.h File Reference

```
#include "gdcmTypes.h"
#include <vector>
```



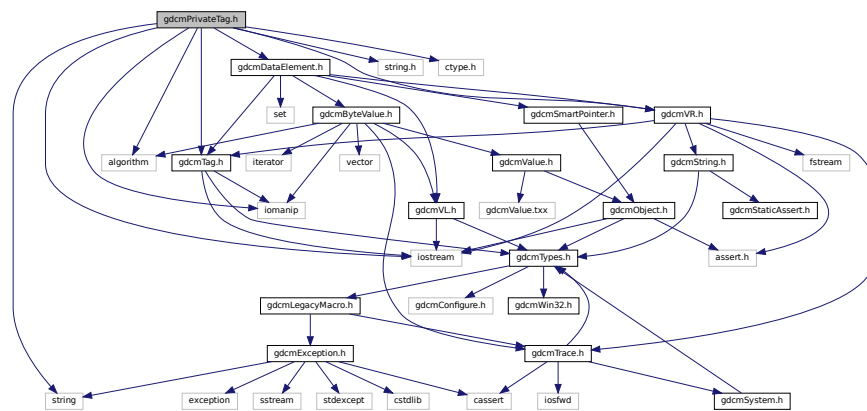




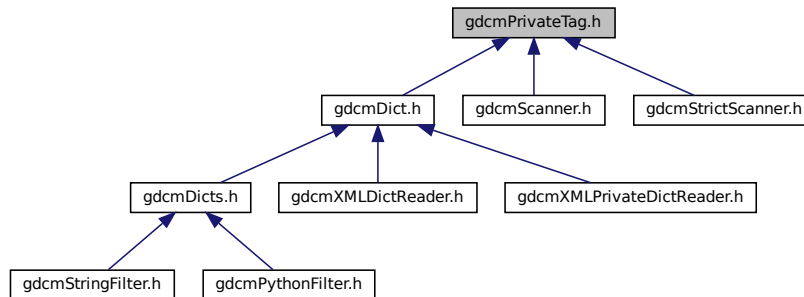
## 11.196 gdcmPrivateTag.h File Reference

```
#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

- [gdcm](#)

## Functions

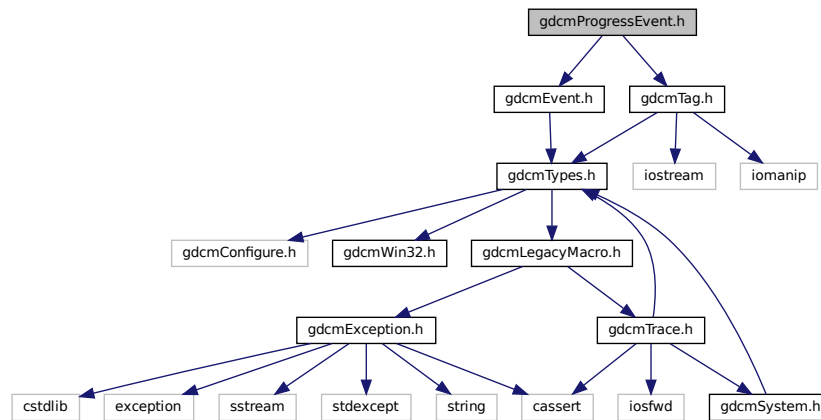
- `std::ostream & gdcm::operator<< (std::ostream &os, const PrivateTag &val)`

## 11.197 gdcmProgressEvent.h File Reference

```
#include "gdcmEvent.h"
```

```
#include "gdcmTag.h"
```

Include dependency graph for `gdcmProgressEvent.h`:



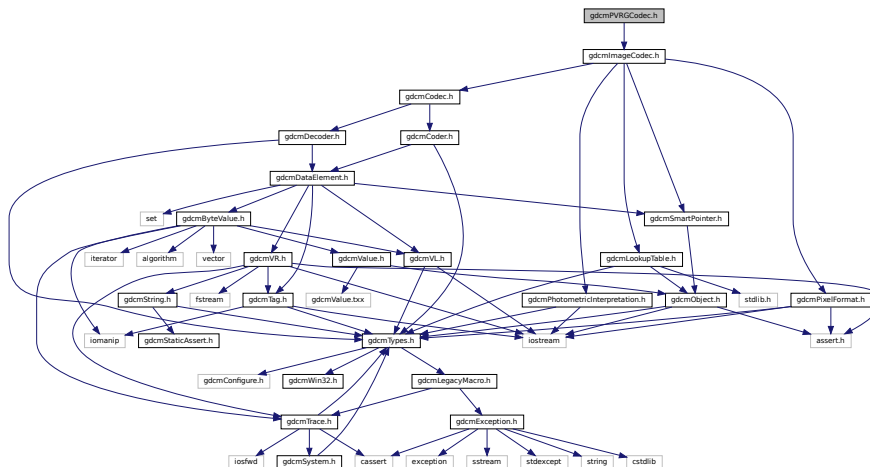
## Classes

- class [gdcm::ProgressEvent](#)  
*ProgressEvent.*

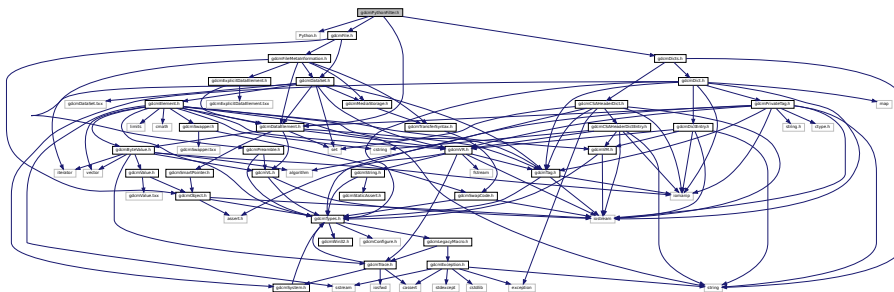
## Namespaces

- [gdcm](#)

Include dependency graph for gdcnPVRGCodec.h:



Include dependency graph for `gdcPythonFilter.h`:

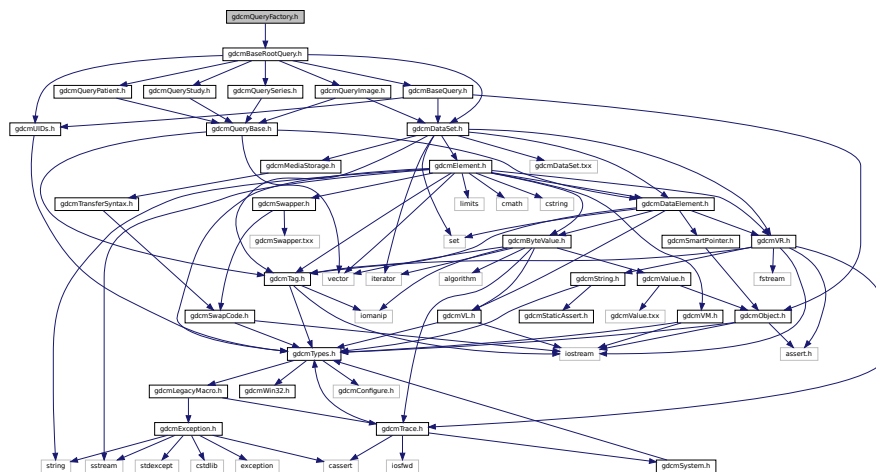




- **gdcm**

- enum `gdcm::ERootType` {  
`gdcm::ePatientRootType` ,  
`gdcm::eStudyRootType` }

```
#include "gdcmBaseRootQuery.h"
Include dependency graph for gdcmQueryFactory.h:
```



- class `gdcm::QueryFactory`  
*QueryFactory.h*.

- **gdcm**



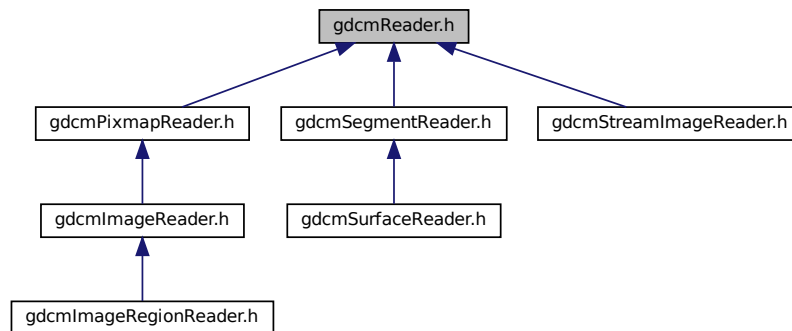
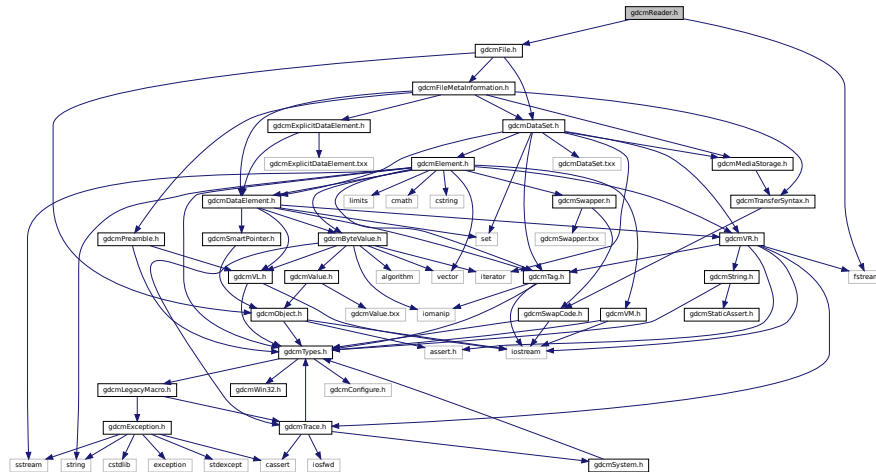












- class `gdc::Reader`  
*Reader* ala DOM (Document *Object* Model)

- **gdcm**

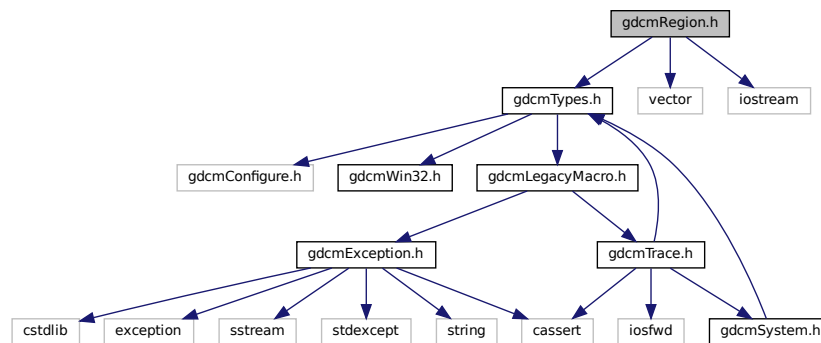
## 11.208 gdcmRegion.h File Reference

```
#include "gdcmTypes.h"
```

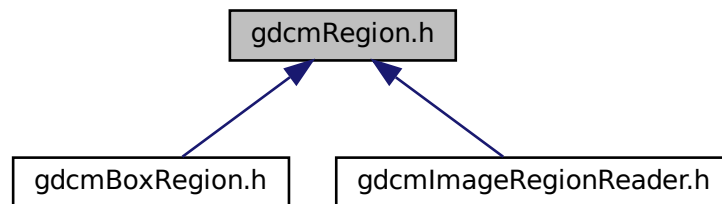
```
#include <vector>
```

```
#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

- `gdcm`

## Functions

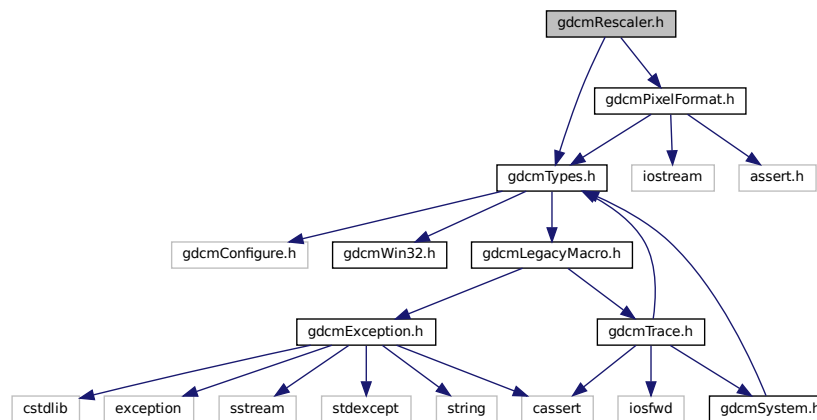
- `std::ostream & gdcm::operator<<` (`std::ostream &os`, `const Region &r`)

## 11.209 gdcmRescaler.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmPixelFormat.h"
```

Include dependency graph for `gdcmRescaler.h`:



## Classes

- class `gdcm::Rescaler`  
*Rescale class.*

## Namespaces

- `gdcm`



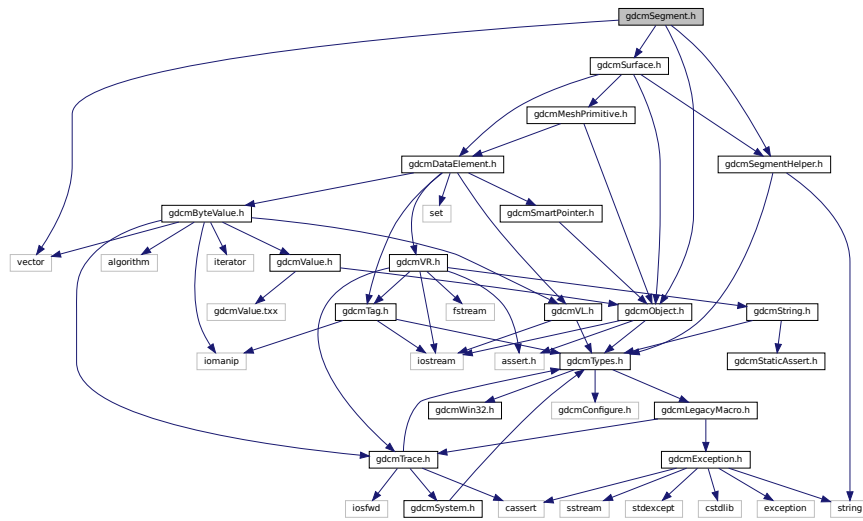


## Functions

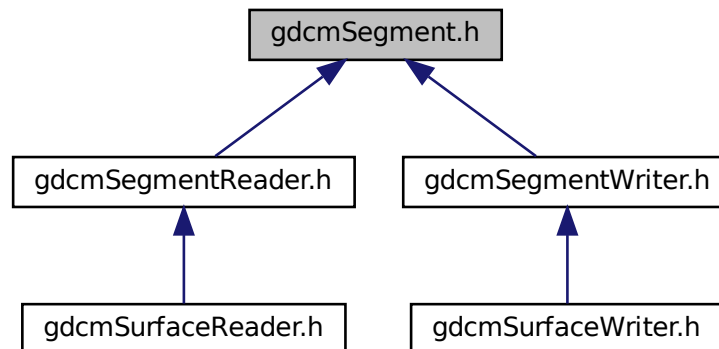
- `std::ostream & gdcmm::operator<< (std::ostream &os, const Scanner &s)`

## 11.213 gdcmmSegment.h File Reference

```
#include <vector>
#include <gdcmmObject.h>
#include <gdcmmSurface.h>
#include "gdcmmSegmentHelper.h"
Include dependency graph for gdcmmSegment.h:
```



This graph shows which files directly or indirectly include this file:





## Classes

- class [gdcm::Segment](#)  
*This class defines a segment.*

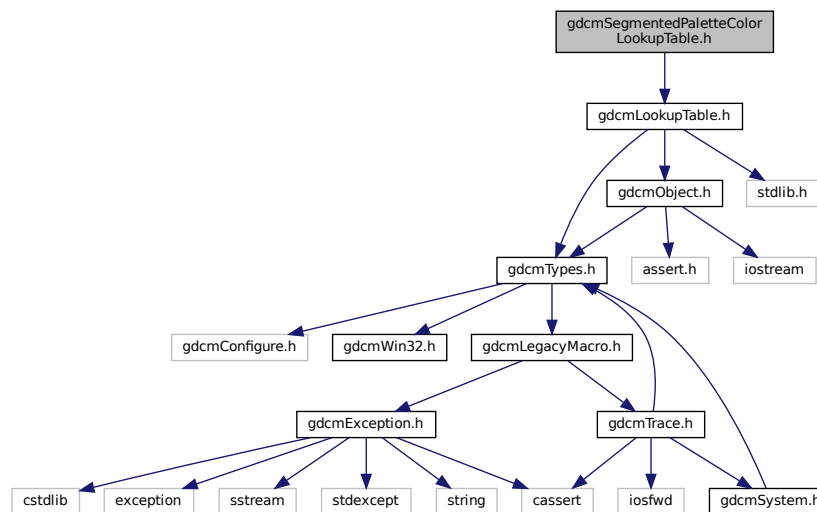
## Namespaces

- [gdcm](#)

## 11.214 gdcmSegmentedPaletteColorLookupTable.h File Reference

```
#include "gdcmLookupTable.h"
```

Include dependency graph for gdcmSegmentedPaletteColorLookupTable.h:



## Classes

- class [gdcm::SegmentedPaletteColorLookupTable](#)  
*SegmentedPaletteColorLookupTable class.*

## Namespaces

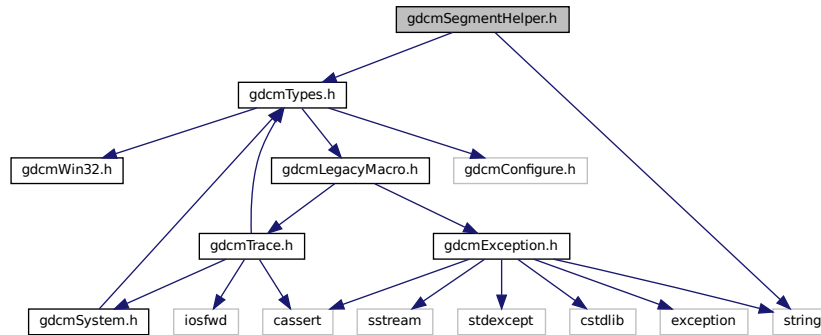
- [gdcm](#)

## 11.215 gdcmSegmentHelper.h File Reference

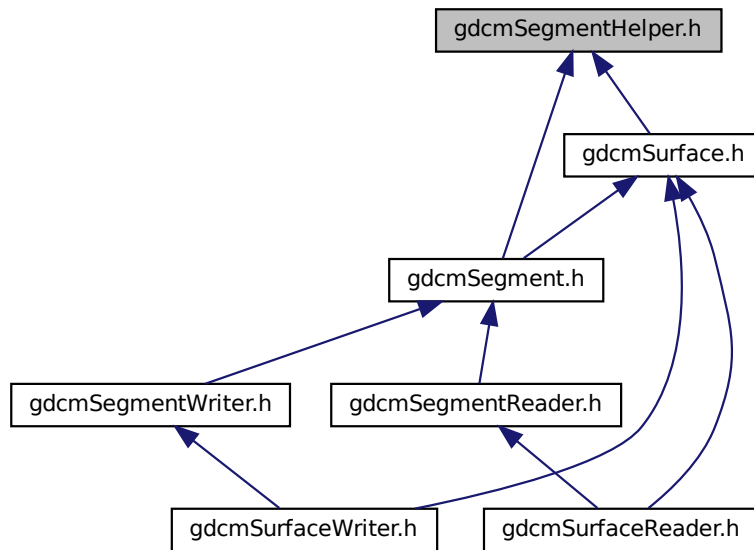
```
#include "gdcmTypes.h"
```

```
#include <string>
```

Include dependency graph for gdcmSegmentHelper.h:



This graph shows which files directly or indirectly include this file:



## Classes

- struct [gdcm::SegmentHelper::BasicCodedEntry](#)

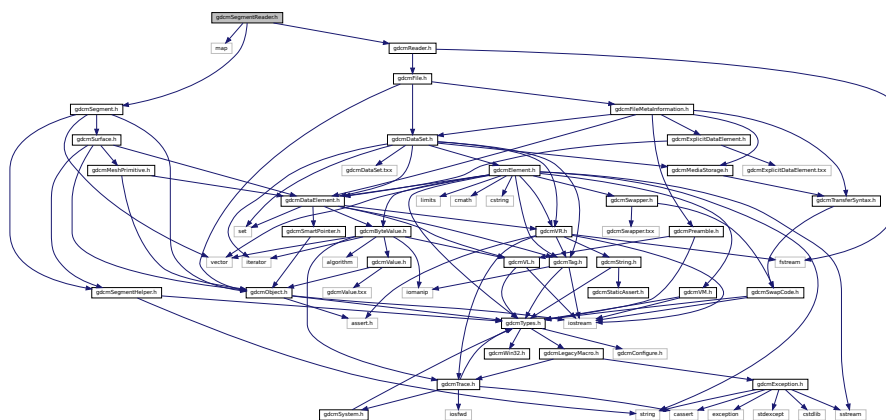
*This structure defines a basic coded entry with all of its attributes.*

## Namespaces

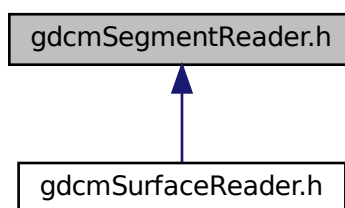
- [gdcm](#)
- [gdcm::SegmentHelper](#)

## 11.216 gdcmSegmentReader.h File Reference

```
#include <map>
#include <gdcmReader.h>
#include <gdcmSegment.h>
Include dependency graph for gdcmSegmentReader.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::SegmentReader](#)  
*This class defines a segment reader.*



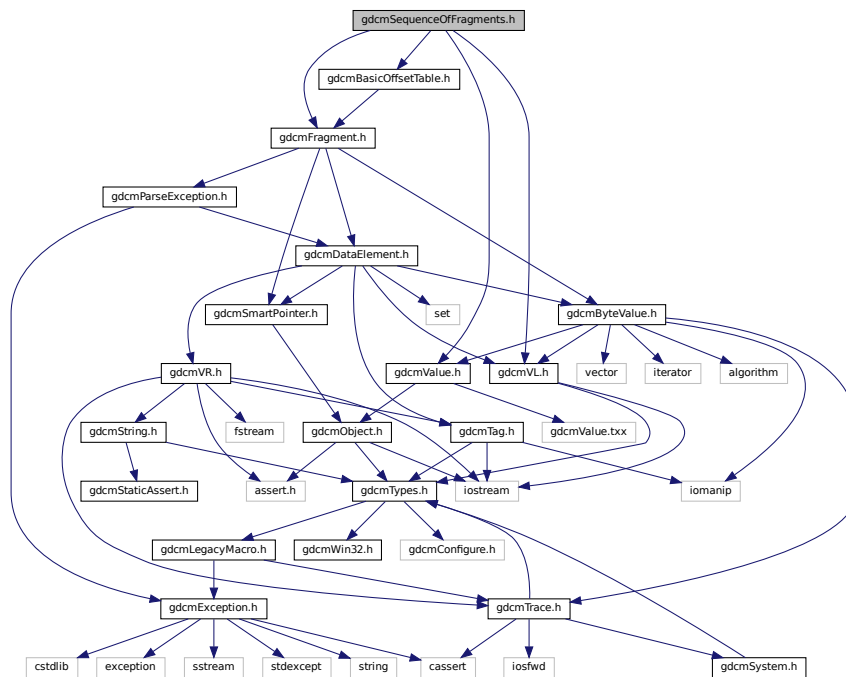
## Namespaces

- [gdcm](#)

## 11.218 gdcmSequenceOfFragments.h File Reference

```
#include "gdcmValue.h"
#include "gdcmVL.h"
#include "gdcmFragment.h"
#include "gdcmBasicOffsetTable.h"
```

Include dependency graph for gdcmSequenceOfFragments.h:



## Classes

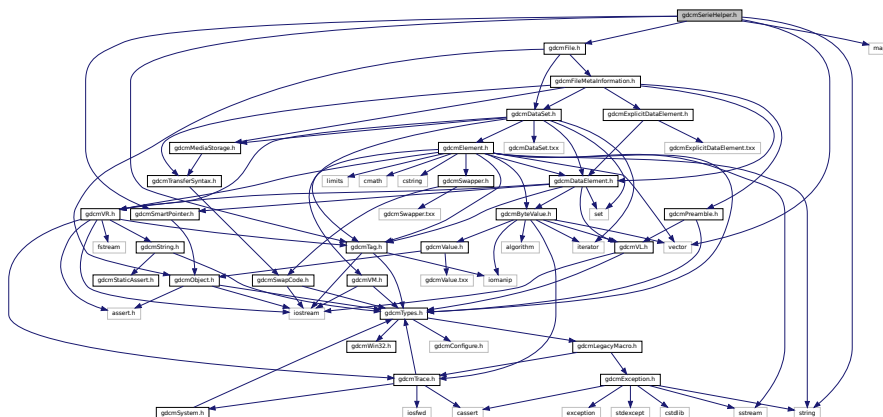
- class [gdcm::SequenceOfFragments](#)  
Class to represent a Sequence Of Fragments.

## Namespaces

- [gdcm](#)



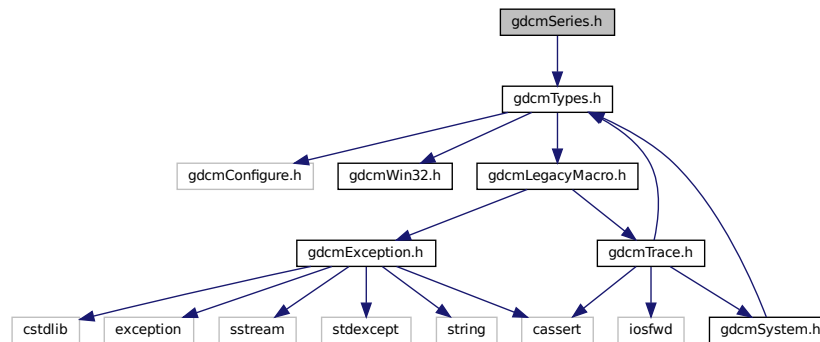
Include dependency graph for gdcSerieHelper.h:



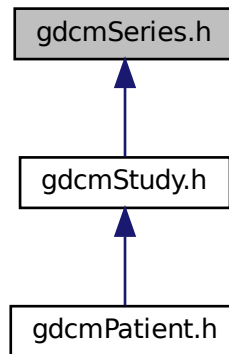
## 11.221 gdcmSeries.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmSeries.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class `gdcm::Series`  
*Series.*

### Namespaces

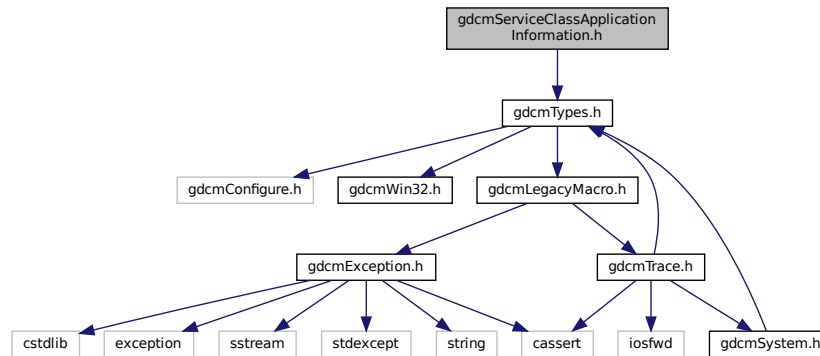
- `gdcm`



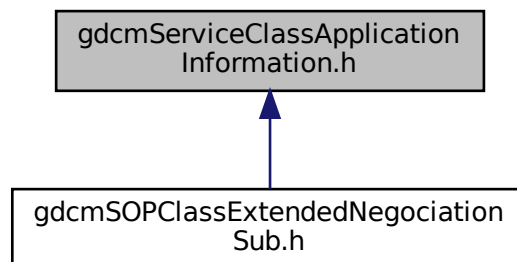
## 11.222 gdcmServiceClassApplicationInformation.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmServiceClassApplicationInformation.h:



This graph shows which files directly or indirectly include this file:



### Classes

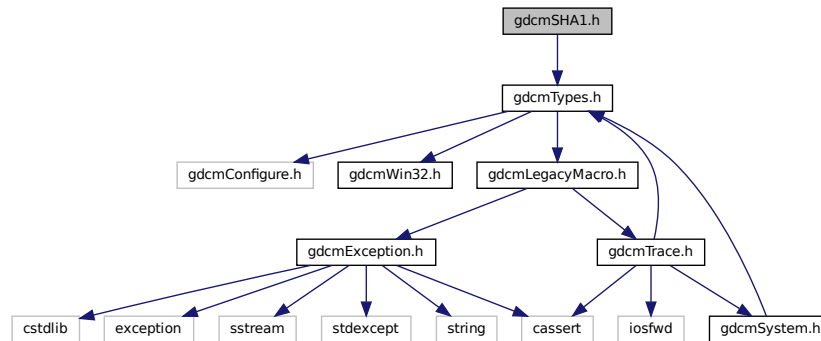
- class [gdcm::network::ServiceClassApplicationInformation](#)

### Namespaces

- [gdcm](#)
- [gdcm::network](#)



Include dependency graph for gdcmSHA1.h:



## Classes

- class [gdcm::SHA1](#)  
Class for [SHA1](#).

## Namespaces

- [gdcm](#)

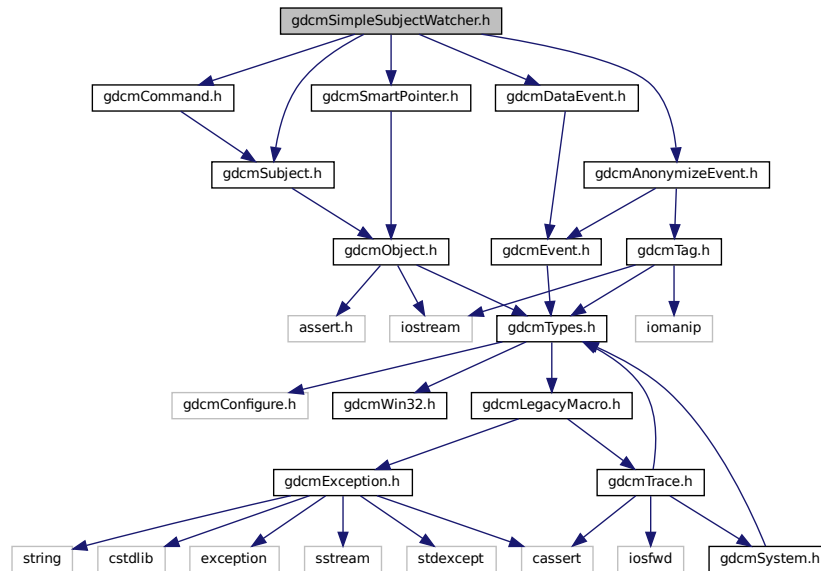
## 11.225 gdcmSimpleSubjectWatcher.h File Reference

```

#include "gdcmSubject.h"
#include "gdcmCommand.h"
#include "gdcmSmartPointer.h"
#include "gdcmAnonymizeEvent.h"
#include "gdcmDataEvent.h"

```

Include dependency graph for `gdcmSimpleSubjectWatcher.h`:



## Classes

- class `gdcm::SimpleSubjectWatcher`  
*SimpleSubjectWatcher*.

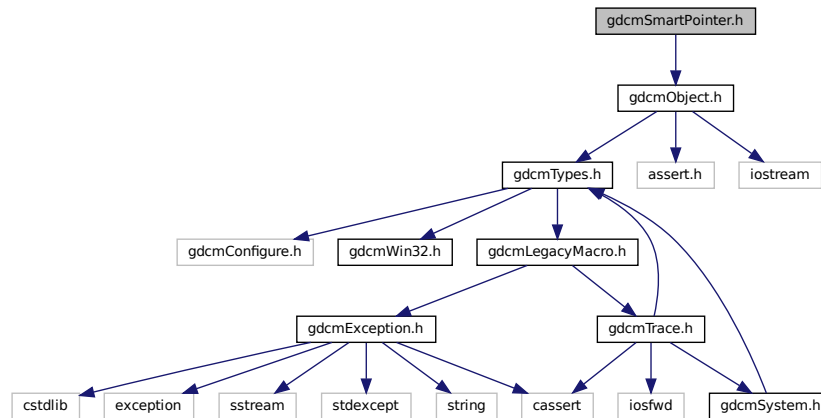
## Namespaces

- `gdcm`

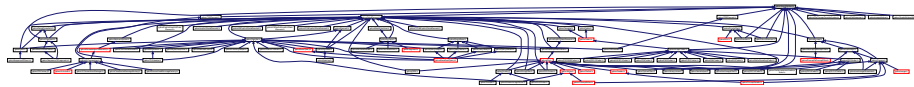
## 11.226 `gdcmSmartPointer.h` File Reference

```
#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.*

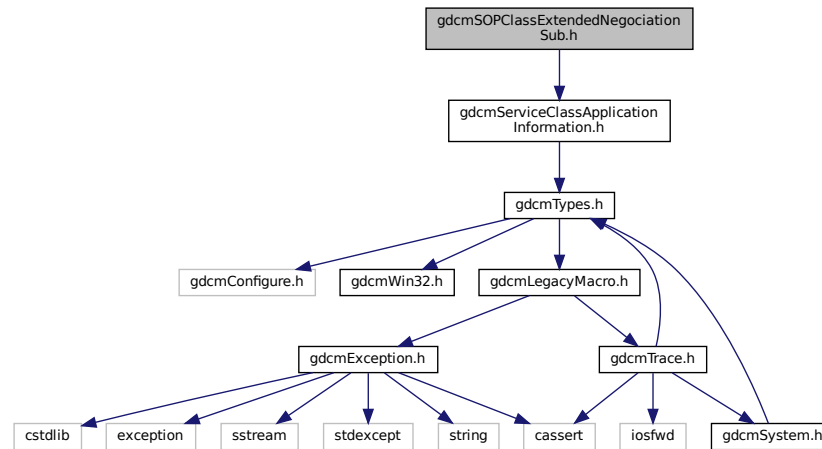
## Namespaces

- [gdcm](#)

## 11.227 gdcmSOPClassExtendedNegociationSub.h File Reference

```
#include "gdcmServiceClassApplicationInformation.h"
```

Include dependency graph for `gdcmSOPClassExtendedNegociationSub.h`:



## Classes

- class [gdcm::network::SOPClassExtendedNegociationSub](#)  
*SOPClassExtendedNegociationSub.*

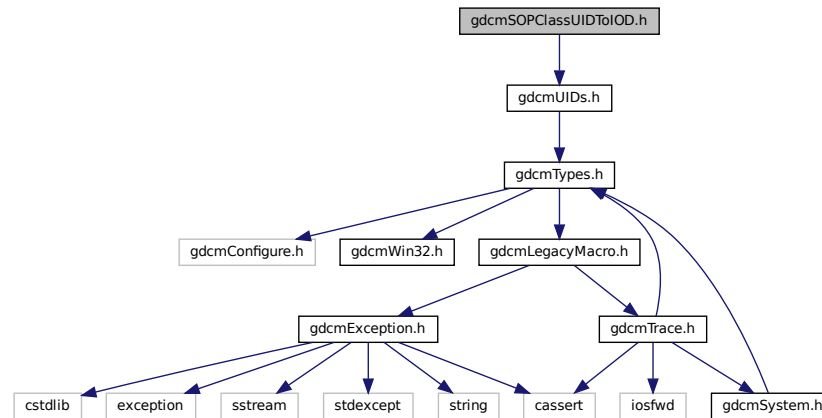
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.228 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

- [gdcm](#)

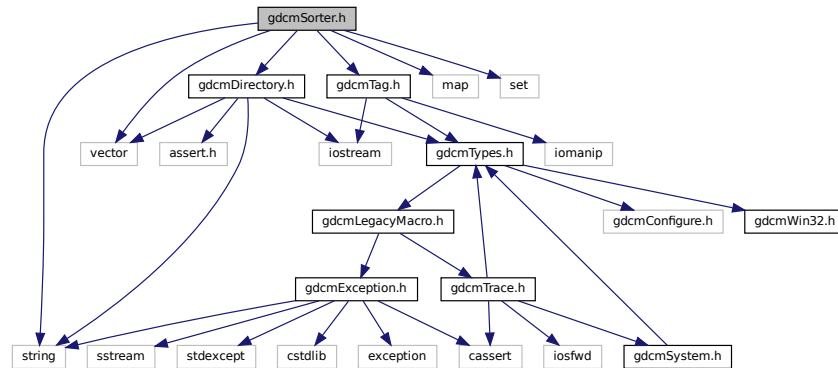
## 11.229 gdcmSorter.h File Reference

```

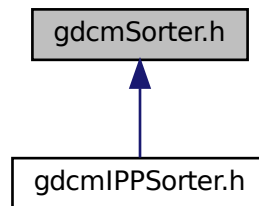
#include "gdcmDirectory.h"
#include "gdcmTag.h"
#include <vector>
#include <string>
#include <map>
#include <set>

```

Include dependency graph for `gdcmSorter.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::Sorter`  
*Sorter.*

## Namespaces

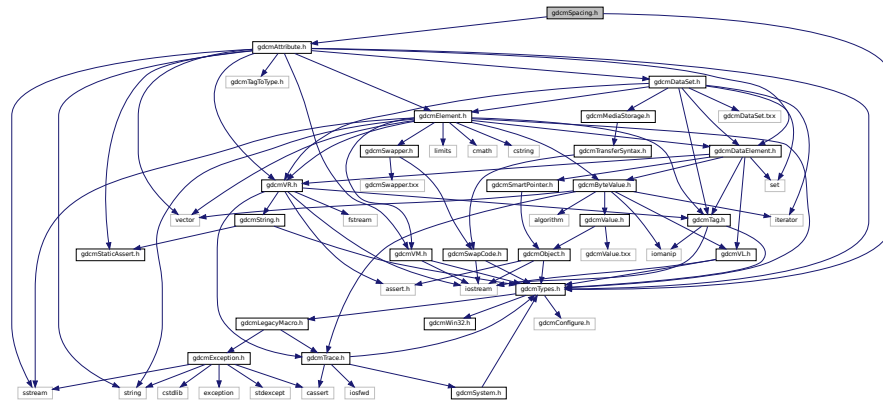
- `gdcm`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Sorter &s)`



```
#include "gdcmTypes.h"
#include "gdcmAttribute.h"
Include dependency graph for gdcmSpacing.h:
```



- class `gdcm::Spacing`  
*Class for Spacing.*

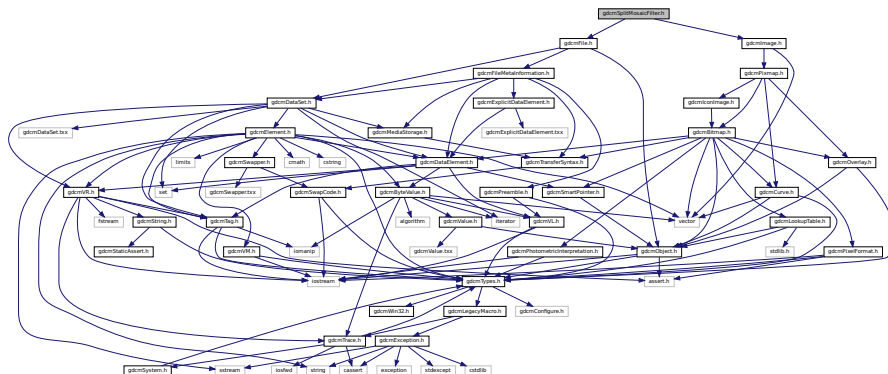
- **gdcm**

```
#include "gdcmFile.h"
```

- class `gdcm::Spectroscopy`  
*Spectroscopy* class.

- **gdcm**

```
#include "gdcmFile.h"
#include "gdcmImage.h"
Include dependency graph for gdcmSplitMosaicFilter.h:
```



## Classes

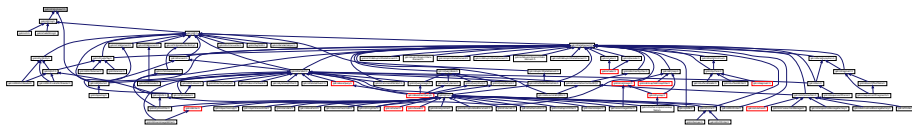
- class [gdcm::SplitMosaicFilter](#)  
*SplitMosaicFilter* class.

## Namespaces

- [gdcm](#)

## 11.233 gdcmStaticAssert.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [gdcm::static\\_assert\\_test< x >](#)
- struct [gdcm::STATIC\\_ASSERTION\\_FAILURE< true >](#)

## Namespaces

- [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.*

### 11.233.1 Macro Definition Documentation

### 11.233.1.1 GDCM\_DO\_JOIN

```
#define GDCM_DO_JOIN(  
    X,  
    Y ) GDCM_DO_JOIN2 (X, Y)
```

### 11.233.1.2 GDCM\_DO\_JOIN2

```
#define GDCM_DO_JOIN2(  
    X,  
    Y ) X##Y
```

### 11.233.1.3 GDCM\_JOIN

```
#define GDCM_JOIN(  
    X,  
    Y ) GDCM_DO_JOIN ( X, Y )
```

### 11.233.1.4 GDCM\_STATIC\_ASSERT

```
#define GDCM_STATIC_ASSERT(  
    B )
```

**Value:**

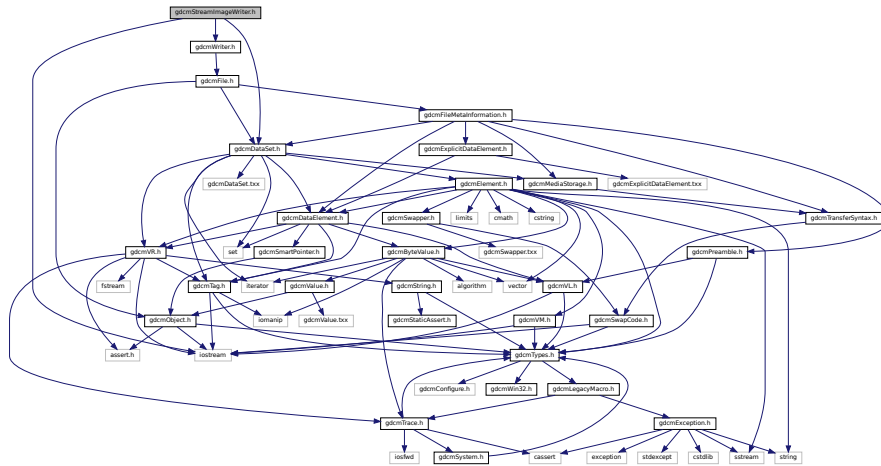
```
typedef ::gdcml::static_assert_test<\n    sizeof(::gdcml::STATIC_ASSERTION_FAILURE< (bool) ( B ) >)>\n    GDCM_JOIN(gdcml_static_assert_typedef_, __LINE__)
```

The GDCM\_JOIN + **LINE** is needed to create a uniq identifier.



```
#include "gdcmDataSet.h"
```

Include dependency graph for `gdcmStreamImageWriter.h`:



## Classes

- class `gdcm::StreamImageWriter`  
*StreamImageReader.*

## Namespaces

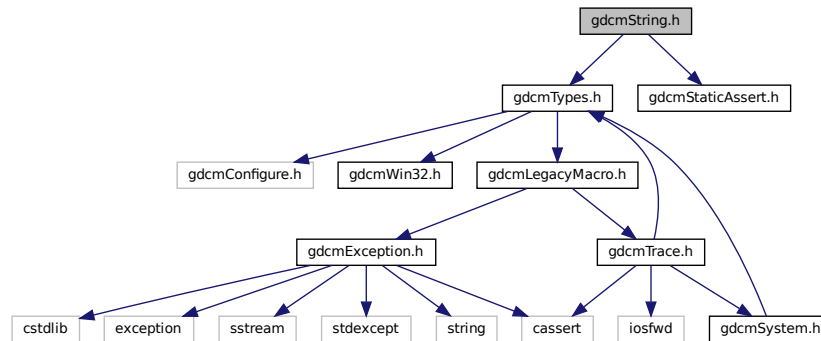
- `gdcm`

## 11.236 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 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

- `gdcm`

## Functions

- template<char TDelimiter, unsigned int TMaxLength, char TPadChar>  
`std::istream & gdcm::operator>> (std::istream &is, String< TDelimiter, TMaxLength, TPadChar > &ms)`

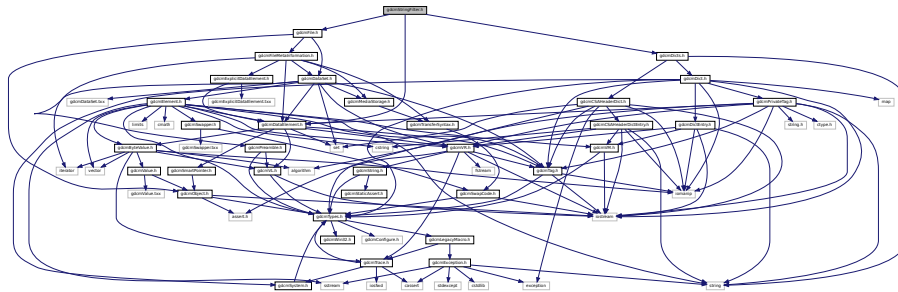
## 11.238 gdcmStringFilter.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmDicts.h"
```



```
#include "gdcmFile.h"
```

Include dependency graph for gdcmStringFilter.h:



## Classes

- class [gdcm::StringFilter](#)  
*StringFilter.*

## Namespaces

- [gdcm](#)

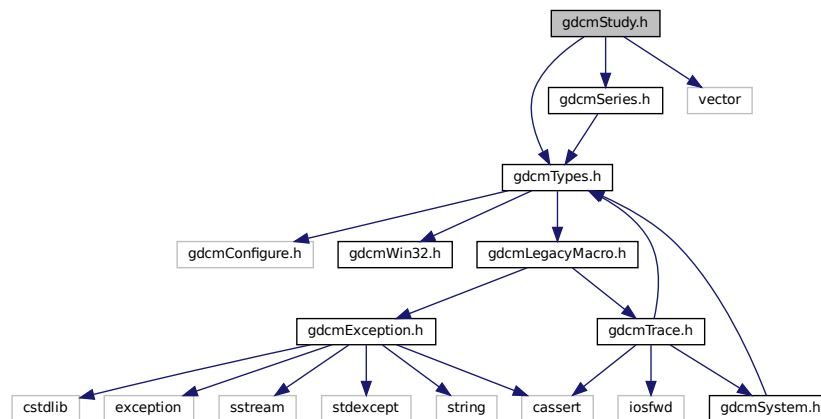
## 11.239 gdcmStudy.h File Reference

```
#include "gdcmTypes.h"
```

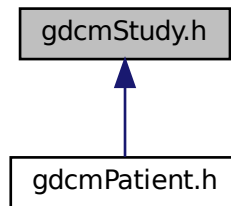
```
#include "gdcmSeries.h"
```

```
#include <vector>
```

Include dependency graph for gdcmStudy.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdc::Study](#)  
[Study](#).

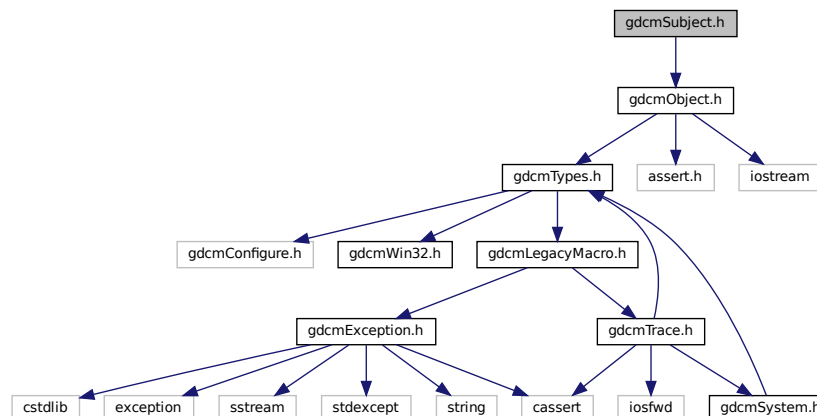
## Namespaces

- [gdc](#)

## 11.240 gdcSubject.h File Reference

```
#include "gdcObject.h"
```

Include dependency graph for gdcSubject.h:







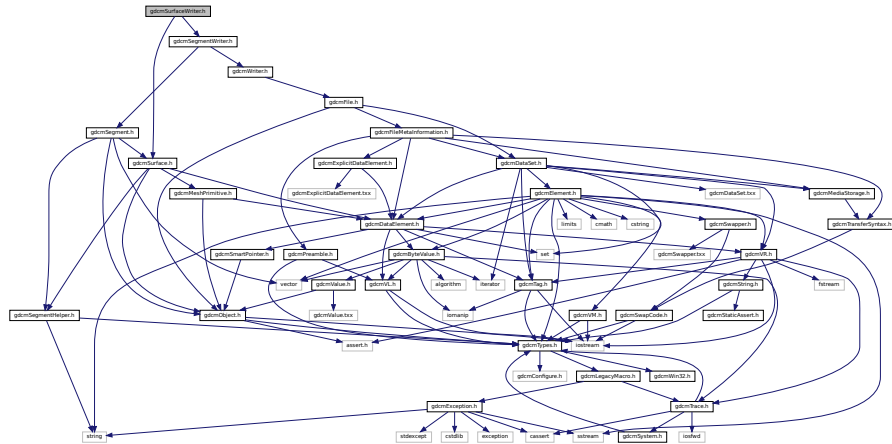


## 11.244 gdcmSurfaceWriter.h File Reference

```
#include <gdcmSegmentWriter.h>
```

```
#include <gdcmSurface.h>
```

Include dependency graph for gdcmSurfaceWriter.h:



### Classes

- class [gdcm::SurfaceWriter](#)  
*This class defines a SURFACE IE writer.*

### Namespaces

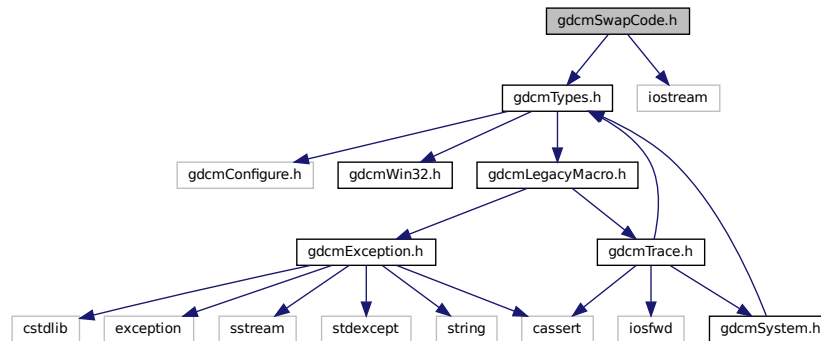
- [gdcm](#)

## 11.245 gdcmSwapCode.h File Reference

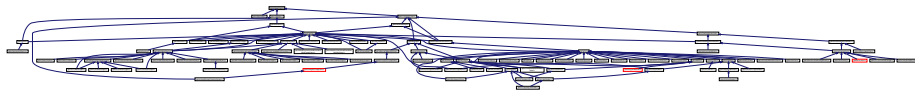
```
#include "gdcmTypes.h"
```

```
#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

- [gdcm](#)

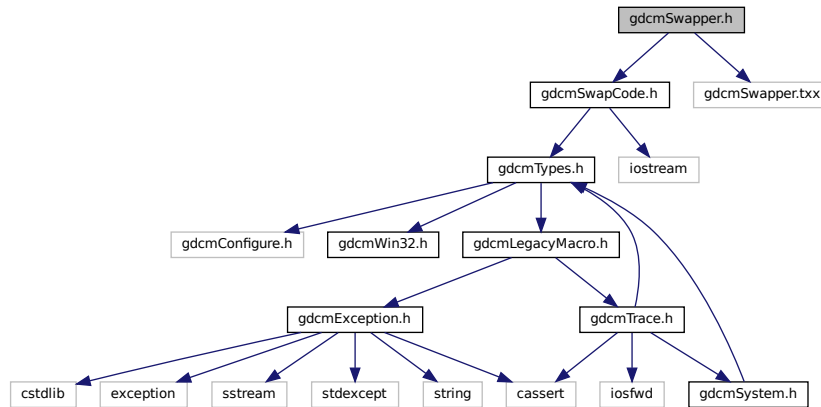
## Functions

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

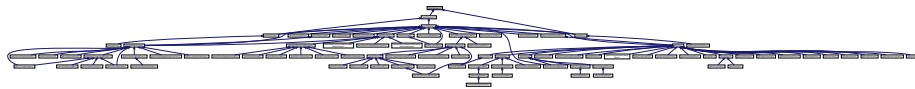
## 11.246 gdcmSwapper.h File Reference

```
#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

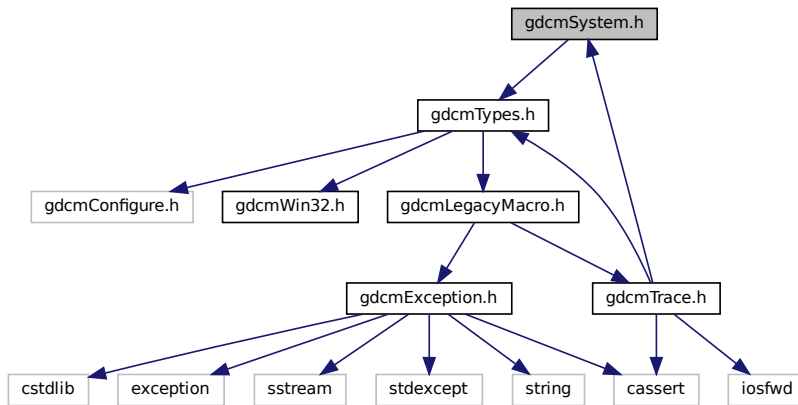
- [gdcm](#)



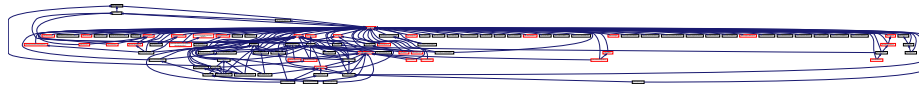
## 11.247 gdcmSystem.h File Reference

```
#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

- [gdcm](#)

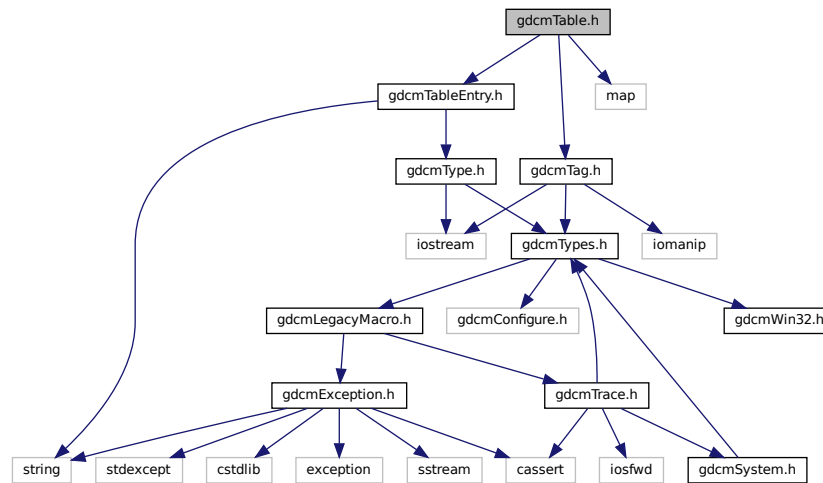
## 11.248 gdcmTable.h File Reference

```
#include "gdcmTableEntry.h"
```

```
#include "gdcmTag.h"
```

```
#include <map>
```

Include dependency graph for gdcmTable.h:



## Classes

- class [gdcm::Table](#)  
*Table.*

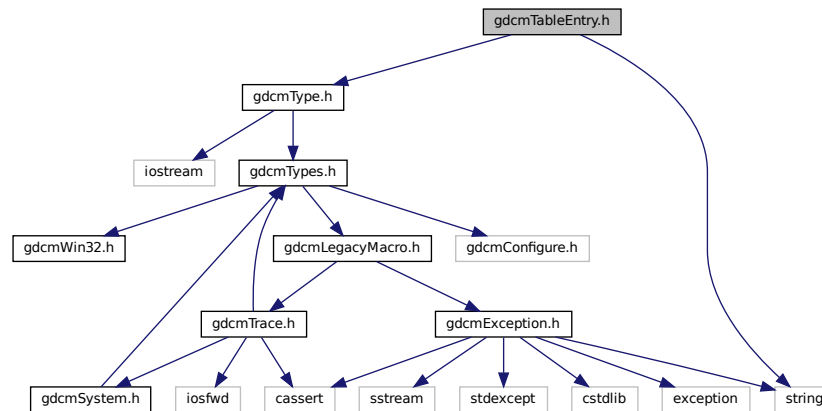
## Namespaces

- [gdcm](#)

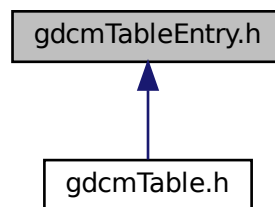
## 11.249 gdcmTableEntry.h File Reference

```
#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*.

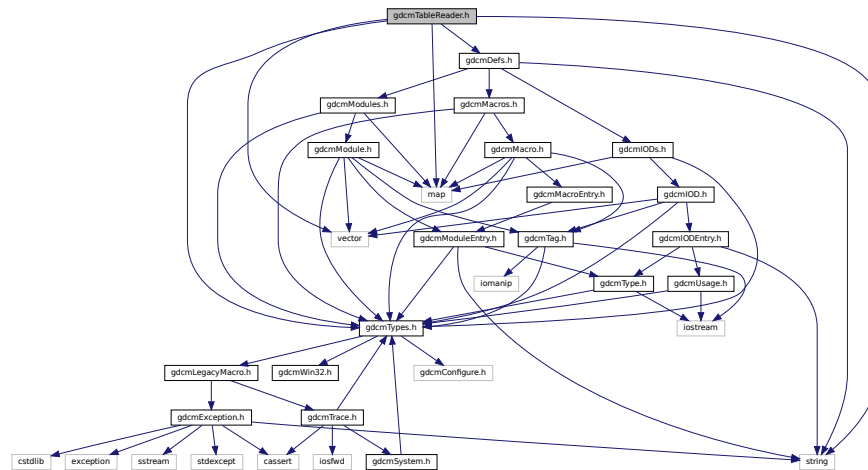
## Namespaces

- `gdcm`

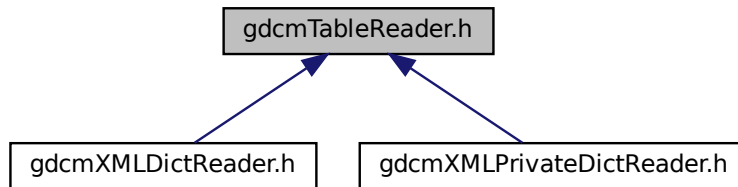
## 11.250 gdcmTableReader.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmDefs.h"
#include <string>
#include <vector>
#include <map>
```

Include dependency graph for gdcmTableReader.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [gdcm::TableReader](#)  
Class for representing a [TableReader](#).

### Namespaces

- [gdcm](#)

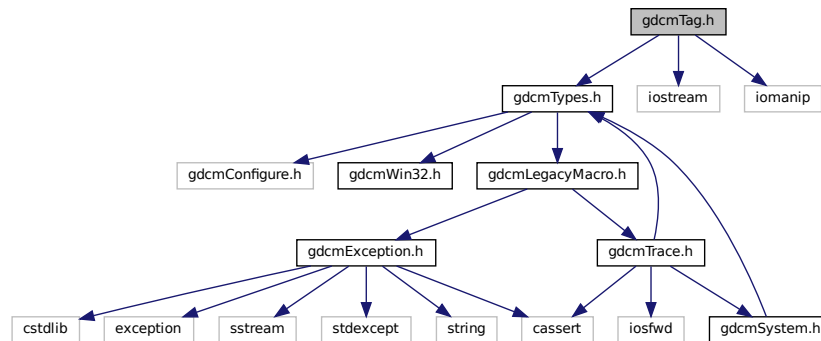
## 11.251 gdcmTag.h File Reference

```
#include "gdcmTypes.h"
```

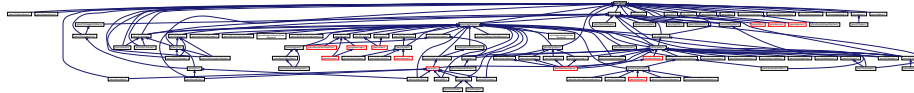
```
#include <iostream>
```

```
#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

- [gdcm](#)

### Functions

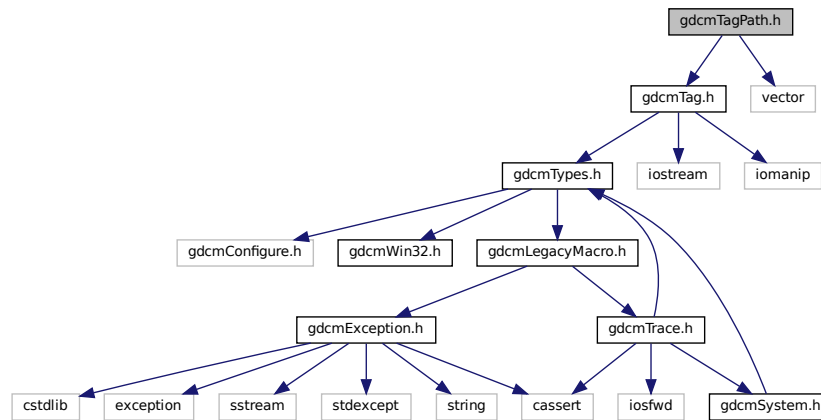
- std::ostream & [gdcm::operator<<](#) (std::ostream &\_os, const Tag &\_val)
- std::istream & [gdcm::operator>>](#) (std::istream &\_is, Tag &\_val)

## 11.252 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.*

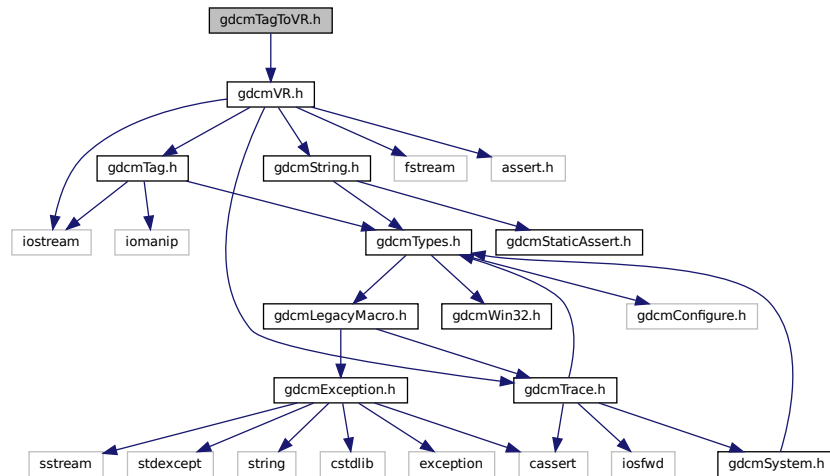
### Namespaces

- [gdcm](#)

## 11.253 gdcmTagToVR.h File Reference

```
#include "gdcmVR.h"
```

Include dependency graph for gdcmTagToVR.h:



## Namespaces

- [gdcm](#)

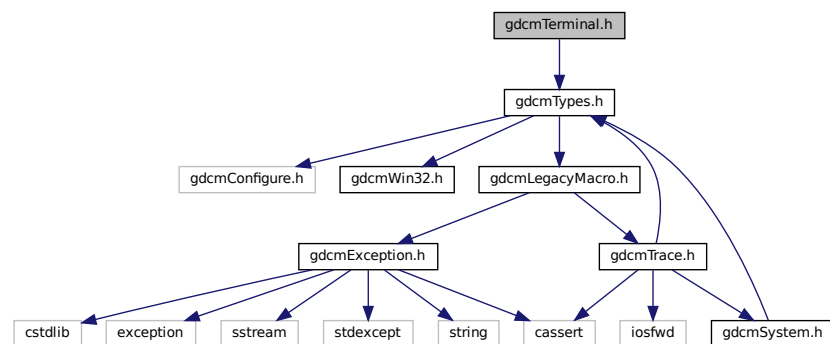
## Functions

- VR::VRType [gdcm::GetVRFromTag](#) (Tag const &tag)

## 11.254 gdcmTerminal.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmTerminal.h:



## Namespaces

- [gdc](#)
- [gdc::terminal](#)

*Class for Terminal.*

## Enumerations

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

## Functions

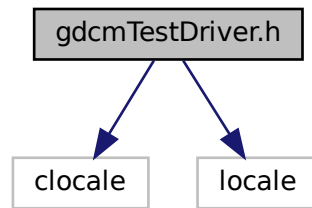
- [GDCM\\_EXPORT](#) std::string [gdc::terminal::setattribute](#) (Attribute att)
- [GDCM\\_EXPORT](#) std::string [gdc::terminal::setbgcolor](#) (Color c)
- [GDCM\\_EXPORT](#) std::string [gdc::terminal::setfgcolor](#) (Color c)
- [GDCM\\_EXPORT](#) void [gdc::terminal::setmode](#) (Mode m)

## 11.255 gdcTestDriver.h File Reference

```
#include <locale>
#include <locale>
```



Include dependency graph for gdcmTestDriver.h:

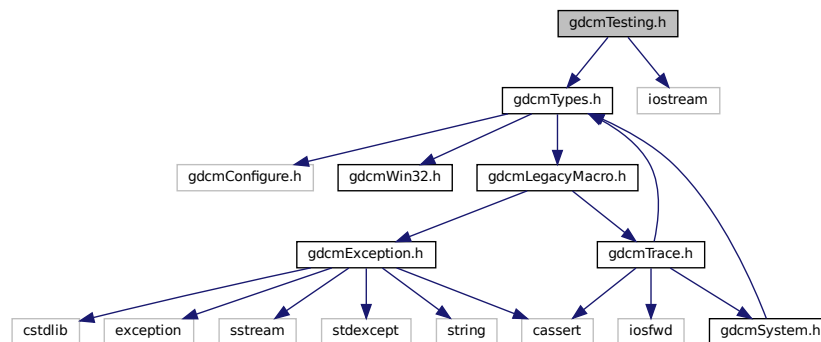


## 11.256 gdcmTesting.h File Reference

```
#include "gdcmTypes.h"
```

```
#include <iostream>
```

Include dependency graph for gdcmTesting.h:



## Classes

- class `gdcm::Testing`  
*class for testing*

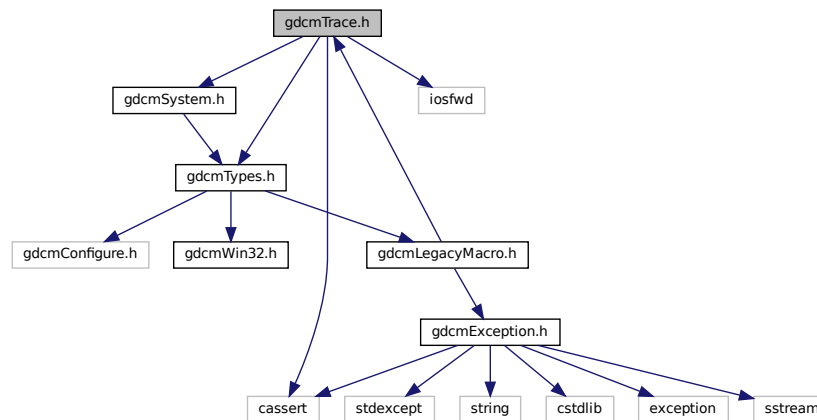
## Namespaces

- `gdcm`

## 11.257 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

- `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.

## 11.257.1 Macro Definition Documentation

### 11.257.1.1 GDCM\_FUNCTION

```
#define GDCM_FUNCTION "<unknown>"
```

### 11.257.1.2 gdcmAssertAlwaysMacro

```
#define gdcmAssertAlwaysMacro(  
    arg ) gdcmAssertMacro(arg)
```

AssertAlways.

Parameters

<code>arg</code>	argument to test An easy solution to pass also a message is to do: <code>gdcmAssertMacro( "my message" &amp;&amp; 2 &lt; 3 )</code>
------------------	---

### 11.257.1.3 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 );  
    }
```

```
}  
}
```

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>
------------	---

## 11.257.1.4 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;
    }
}
```

Debug.

## Parameters

<i>msg</i>	message part
------------	--------------

## 11.257.1.5 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;
    }
}
```

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
------------	---------------------

### 11.257.1.6 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;  
    }  
}
```

Warning.

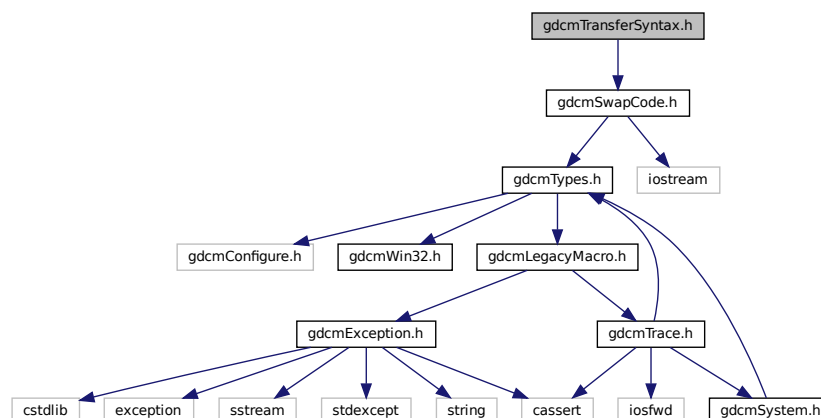
#### Parameters

<i>msg</i>	message part
------------	--------------

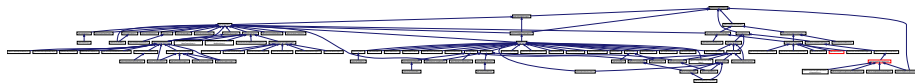
## 11.258 gdcmTransferSyntax.h File Reference

```
#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

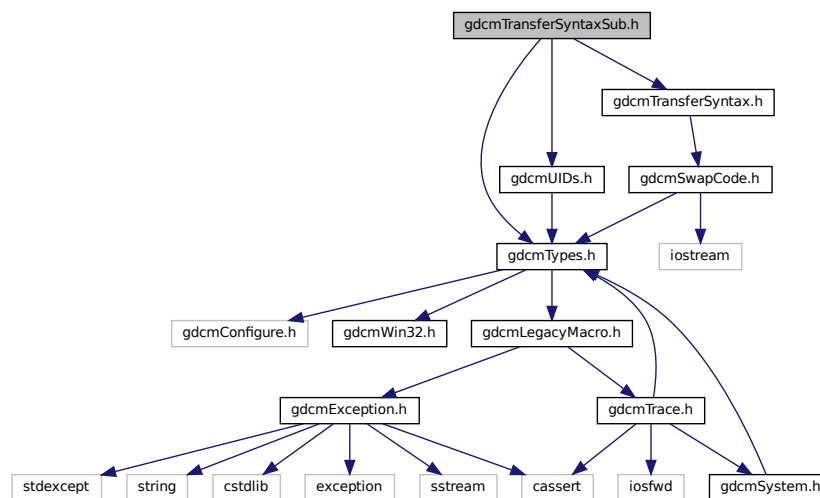
- [gdcm](#)

## Functions

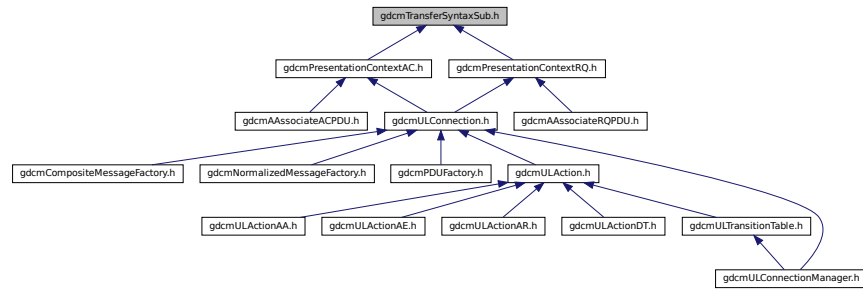
- `std::ostream & gdcm::operator<< (std::ostream &_os, const TransferSyntax &ts)`

## 11.259 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

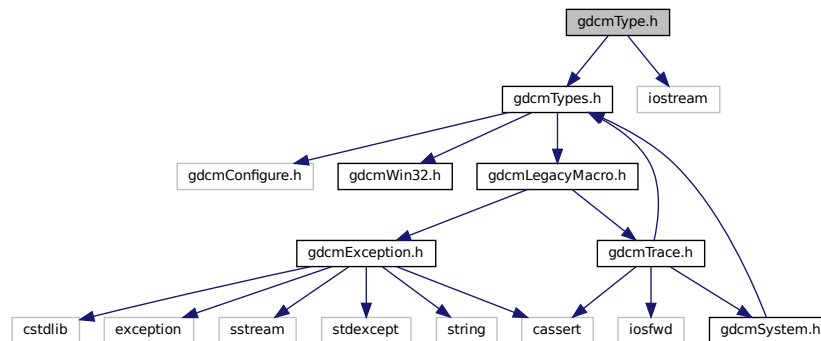
- [gdcm](#)
- [gdcm::network](#)

## 11.260 gdcmType.h File Reference

```
#include "gdcmTypes.h"
```

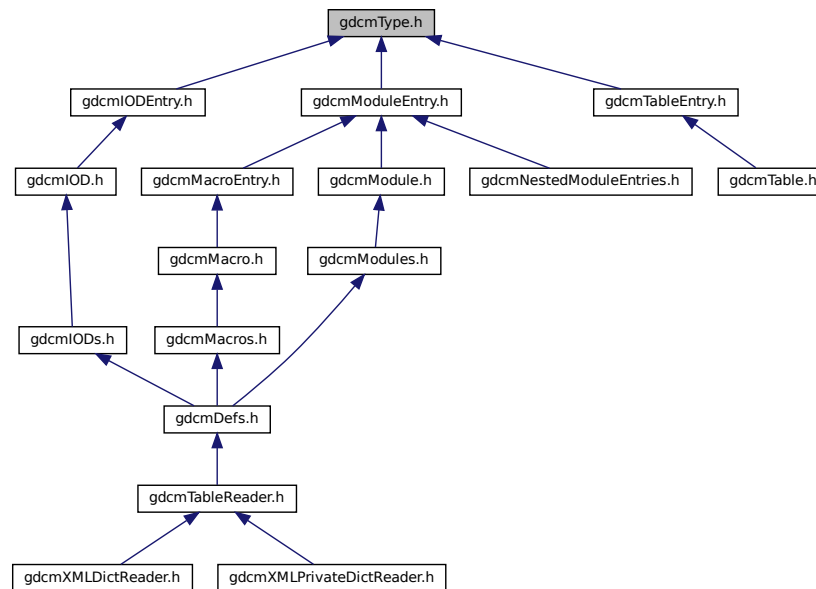
```
#include <iostream>
```

Include dependency graph for gdcmType.h:





This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::Type](#)  
*Type.*

## Namespaces

- [gdcm](#)

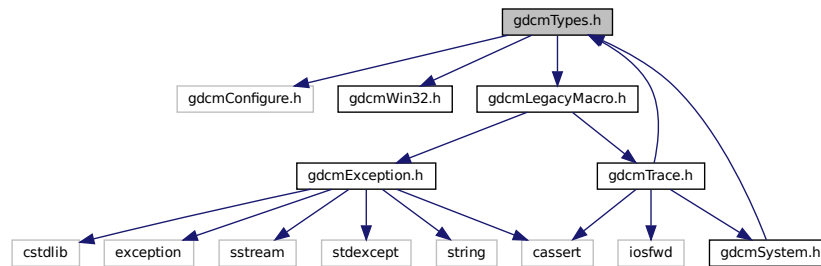
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Type &val)`

## 11.261 gdcmTypes.h File Reference

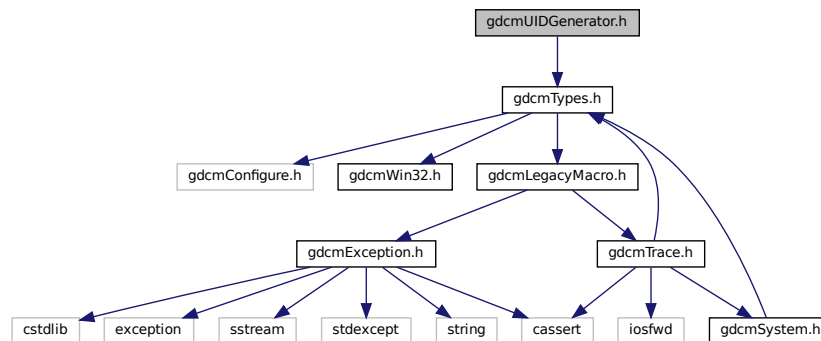
```
#include "gdcmConfigure.h"
#include "gdcmWin32.h"
```

```
#include "gdcmLegacyMacro.h"
Include dependency graph for gdcmTypes.h:
```



## 11.262 gdcmUIDGenerator.h File Reference

```
#include "gdcmTypes.h"
Include dependency graph for gdcmUIDGenerator.h:
```



### Classes

- class [gdcm::UIDGenerator](#)  
Class for generating unique UID.

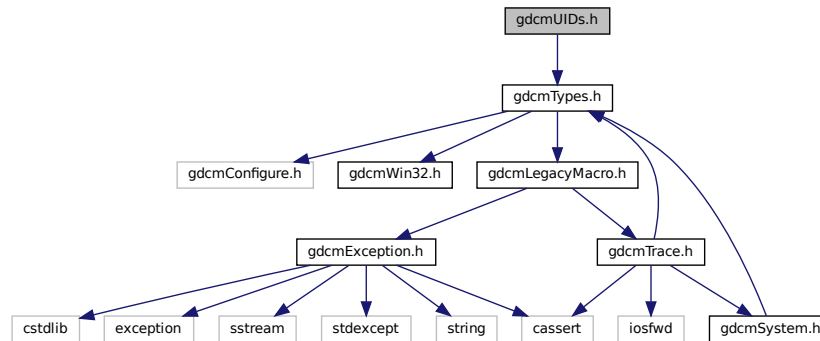
### Namespaces

- [gdcm](#)

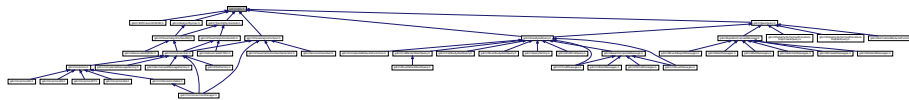
## 11.263 gdcmUIDs.h File Reference

```
#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

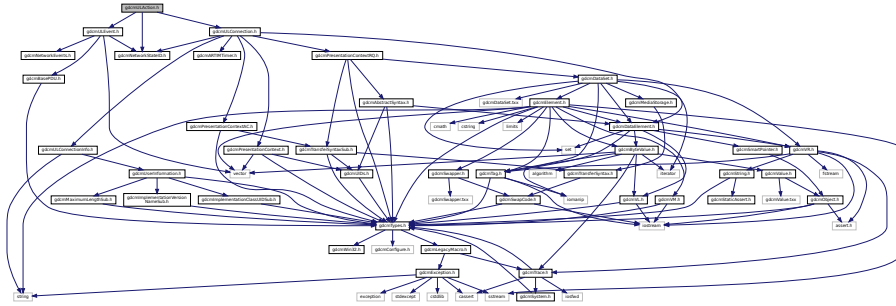
- [gdcm](#)

### Functions

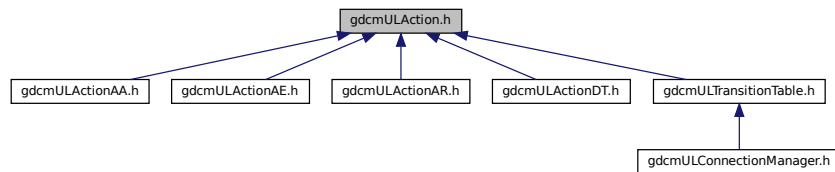
- `std::ostream & gdcm::operator<< (std::ostream &_os, const UIDs &uid)`

## 11.264 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.*

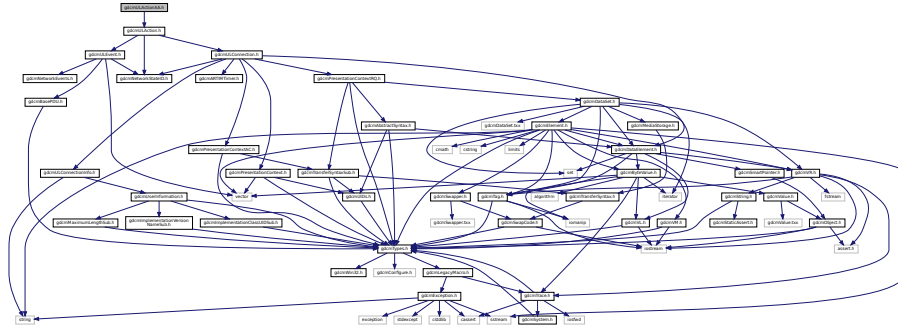
### Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.265 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](#)

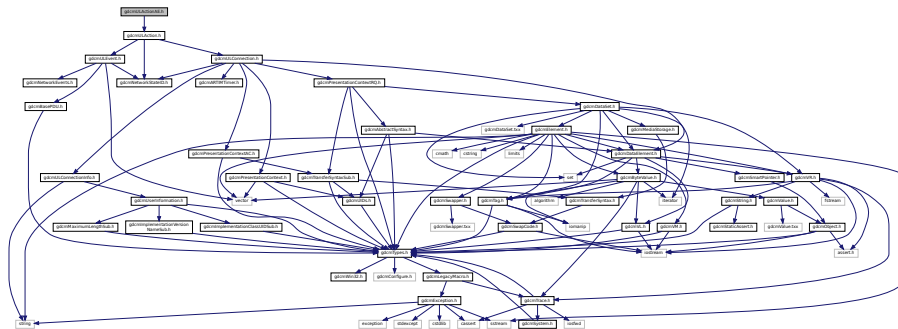
### Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.266 gdcmULActionAE.h File Reference

```
#include "gdcmULAction.h"
```

Include dependency graph for gdcmULActionAE.h:



## Classes

- class [gdcm::network::ULActionAE1](#)
- class [gdcm::network::ULActionAE2](#)
- class [gdcm::network::ULActionAE3](#)
- class [gdcm::network::ULActionAE4](#)
- class [gdcm::network::ULActionAE5](#)
- class [gdcm::network::ULActionAE6](#)
- class [gdcm::network::ULActionAE7](#)
- class [gdcm::network::ULActionAE8](#)

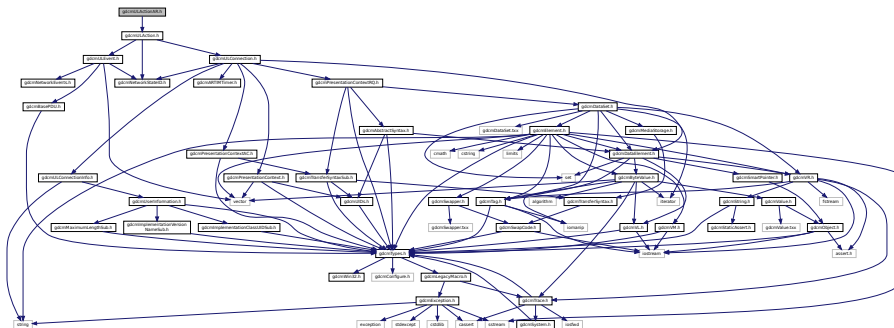
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.267 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](#)

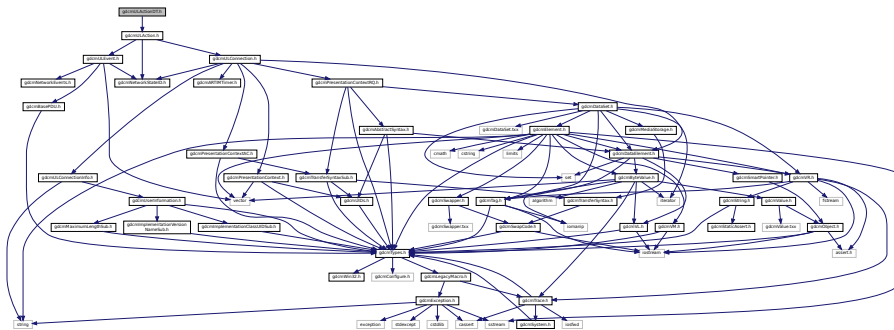
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.268 gdcmULActionDT.h File Reference

```
#include "gdcmULAction.h"
```

Include dependency graph for gdcmULActionDT.h:



## Classes

- class [gdcm::network::ULActionDT1](#)
- class [gdcm::network::ULActionDT2](#)

## Namespaces

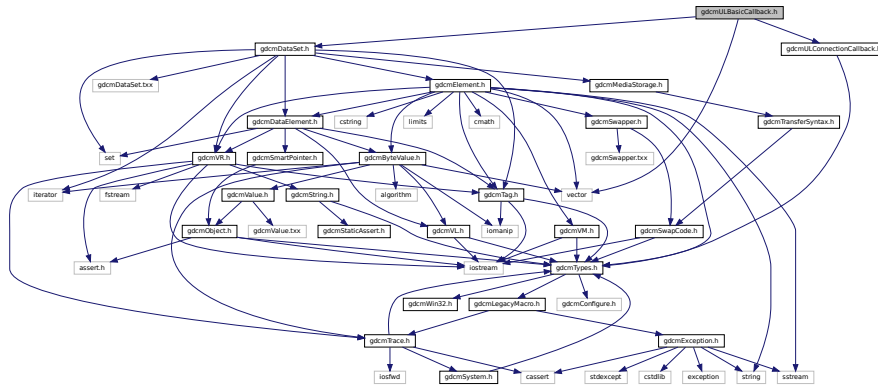
- [gdcm](#)
- [gdcm::network](#)

## 11.269 gdcmULBasicCallback.h File Reference

```
#include "gdcmULConnectionCallback.h"
```

```
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmULBasicCallback.h:



- class `gdcm::network::ULBasicCallback`  
`ULBasicCallback.`

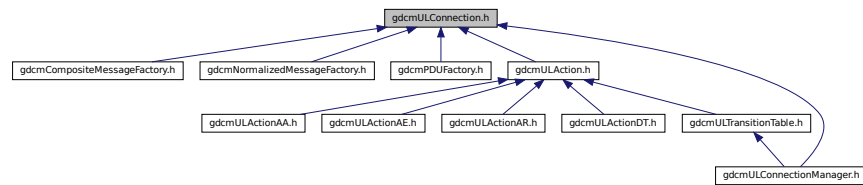
- gdc
- gdc::network

```
#include "gdcmNetworkStateID.h"
#include "gdcmARTIMTimer.h"
#include "gdcmULConnectionInfo.h"
#include "gdcmPresentationContextRQ.h"
#include "gdcmDataElement.h"
#include "gdcmPresentationContextAC.h"
#include "gdcmPresentationContext.h"
```

The diagram illustrates a highly interconnected network of nodes and edges. The nodes are represented by rectangular boxes containing text labels, which appear to be identifiers or names of entities. These nodes are distributed across the frame, with some clusters being more densely connected than others. The edges are thin, dark lines that connect different nodes, indicating relationships or interactions between them. Some edges are straight, while others are curved, suggesting different types or strengths of connections. The overall structure is complex and non-linear, typical of a large-scale network visualization.



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::network::ULConnection](#)  
*ULConnection.*

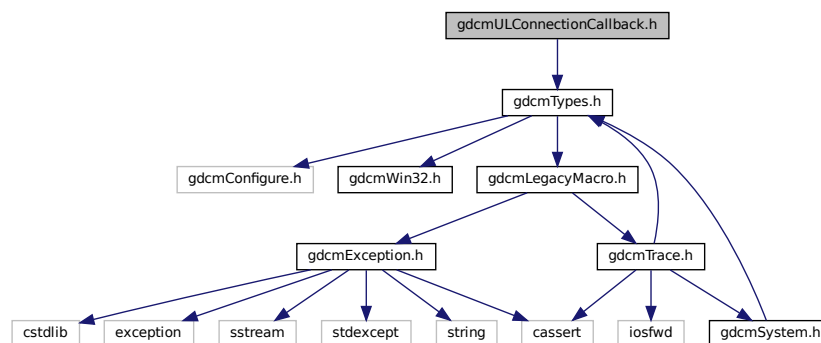
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

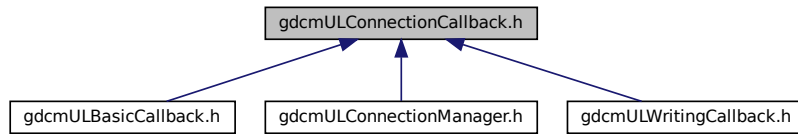
## 11.271 gdcmULConnectionCallback.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmULConnectionCallback.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcml::network::ULConnectionCallback](#)

## Namespaces

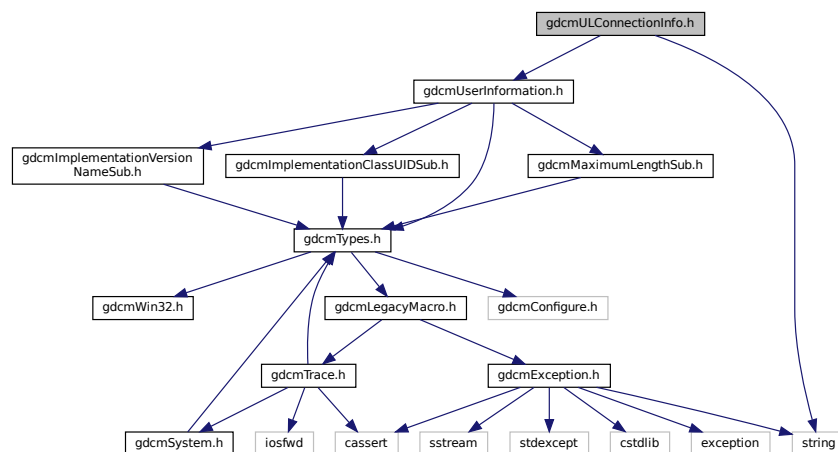
- [gdcml](#)
- [gdcml::network](#)

## 11.272 gdcmlConnectionInfo.h File Reference

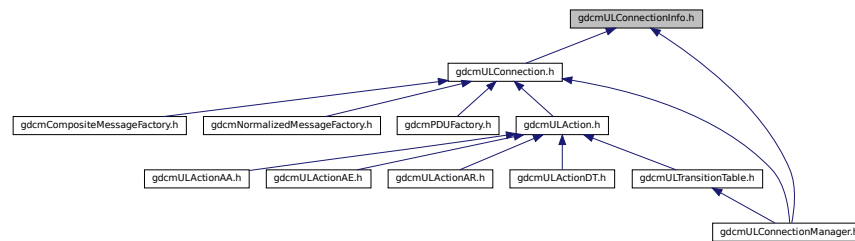
```
#include "gdcmlUserInformation.h"
```

```
#include <string>
```

Include dependency graph for `gdcmlConnectionInfo.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::network::ULConnectionInfo](#)  
*ULConnectionInfo*.

## Namespaces

- [gdcm](#)
- [gdcm::network](#)

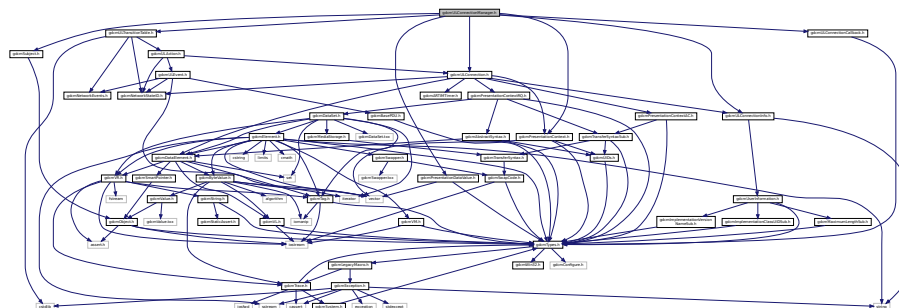
## 11.273 gdcmULConnectionManager.h File Reference

```

#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.*

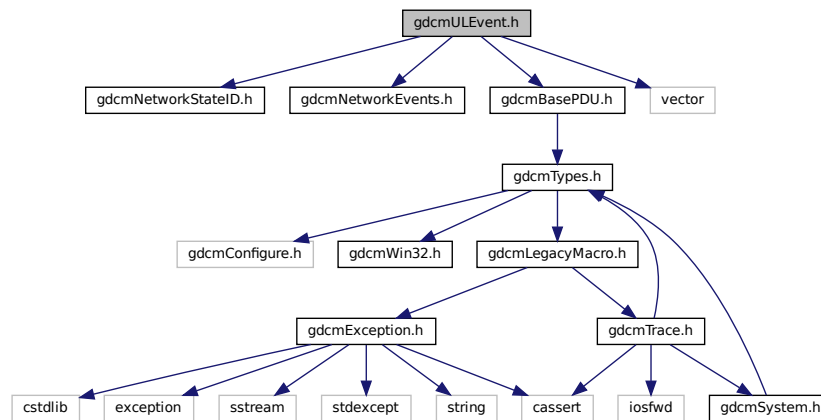
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

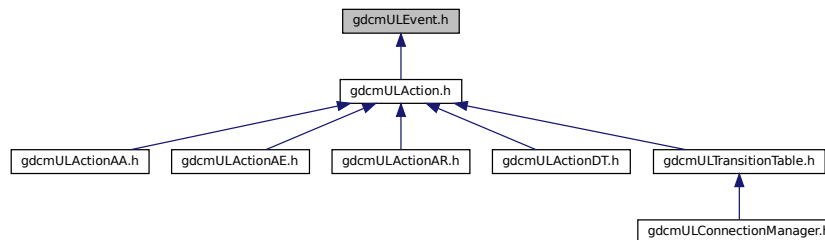
## 11.274 gdcmULEvent.h File Reference

```
#include "gdcmNetworkStateID.h"
#include "gdcmNetworkEvents.h"
#include "gdcmBasePDU.h"
#include <vector>
```

Include dependency graph for gdcmULEvent.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::network::ULEvent](#)  
*ULEvent.*

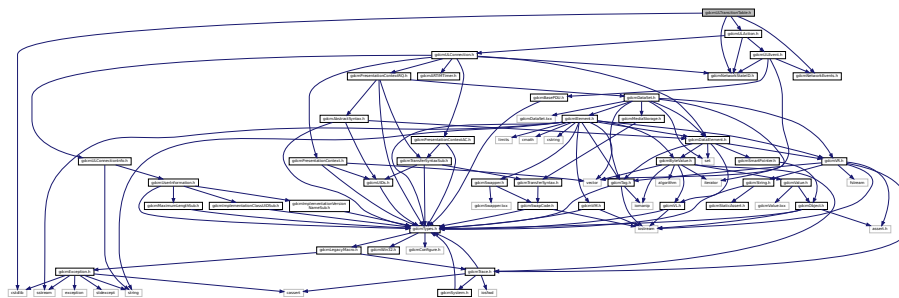
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

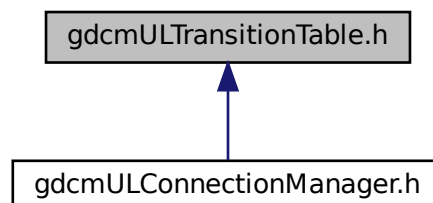
## 11.275 gdcmULTransitionTable.h File Reference

```
#include "gdcmNetworkStateID.h"  
#include "gdcmNetworkEvents.h"  
#include "gdcmULAction.h"  
#include <cstdlib>
```

Include dependency graph for gdcmULTransitionTable.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::network::TableRow](#)
- struct [gdcm::network::Transition](#)
- class [gdcm::network::ULTransitionTable](#)

*[ULTransitionTable](#) The transition table of all the ULEvents, new ULActions, and ULStates.*

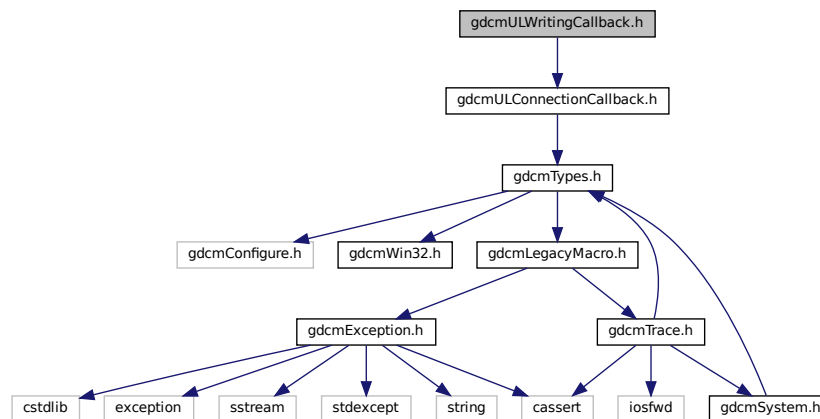
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.276 gdcmULWritingCallback.h File Reference

```
#include "gdcmULConnectionCallback.h"
```

Include dependency graph for gdcmULWritingCallback.h:



## Classes

- class [gdcm::network::ULWritingCallback](#)

## Namespaces

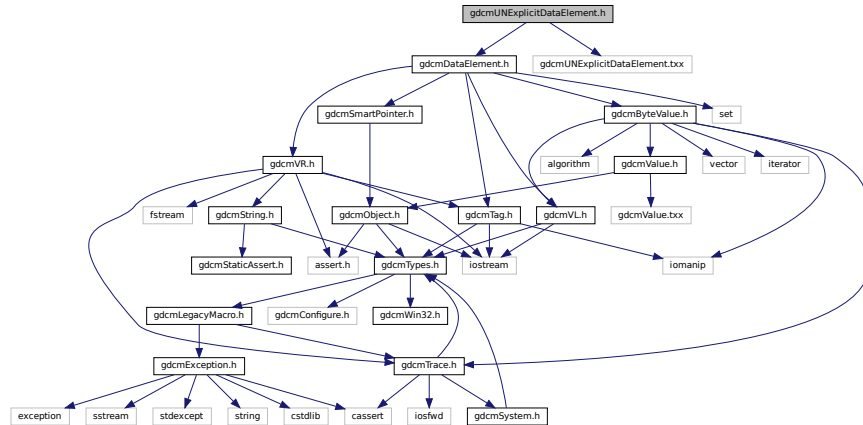
- [gdcm](#)
- [gdcm::network](#)

## 11.277 gdcmUNExplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
```

```
#include "gdcmUNExplicitDataElement.txx"
```

Include dependency graph for gdcmUNExplicitDataElement.h:



### Classes

- class [gdcm::UNExplicitDataElement](#)  
Class to read/write a [DataElement](#) as *UNExplicit Data Element*.

### Namespaces

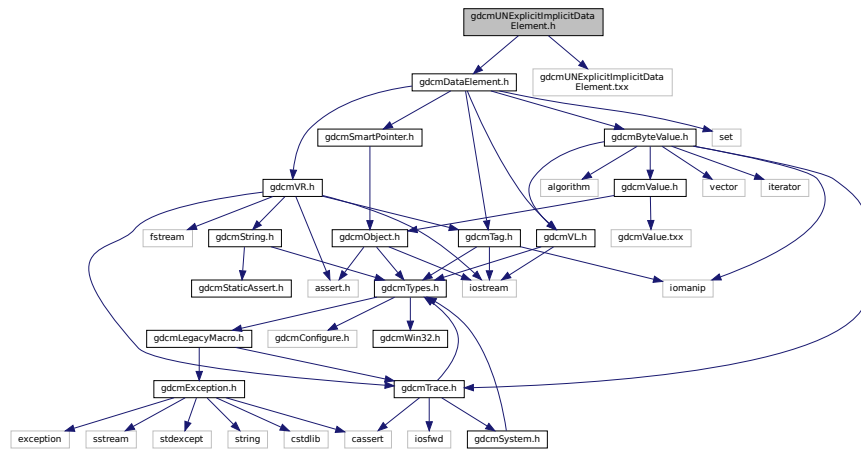
- [gdcm](#)

## 11.278 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](#).

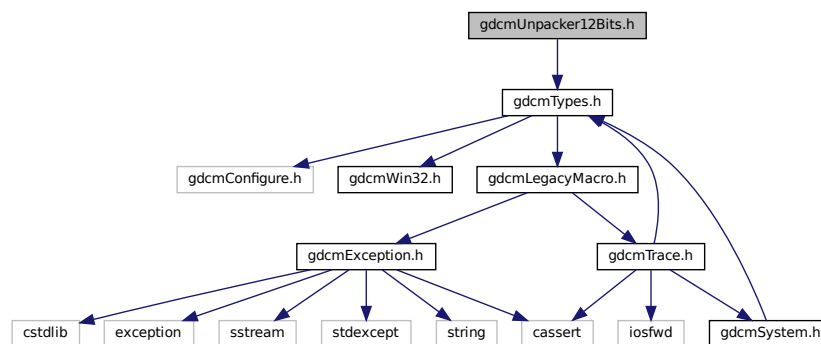
## Namespaces

- [gdcm](#)

## 11.279 gdcmUnpacker12Bits.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for `gdcmUnpacker12Bits.h`:





## Classes

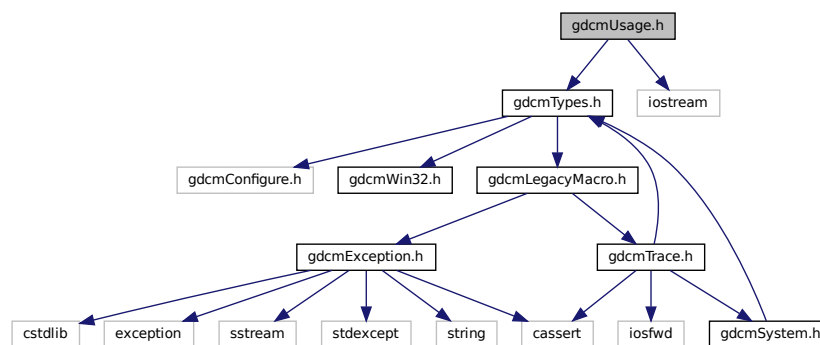
- class [gdcm::Unpacker12Bits](#)  
*Pack/Unpack 12 bits pixel into 16bits.*

## Namespaces

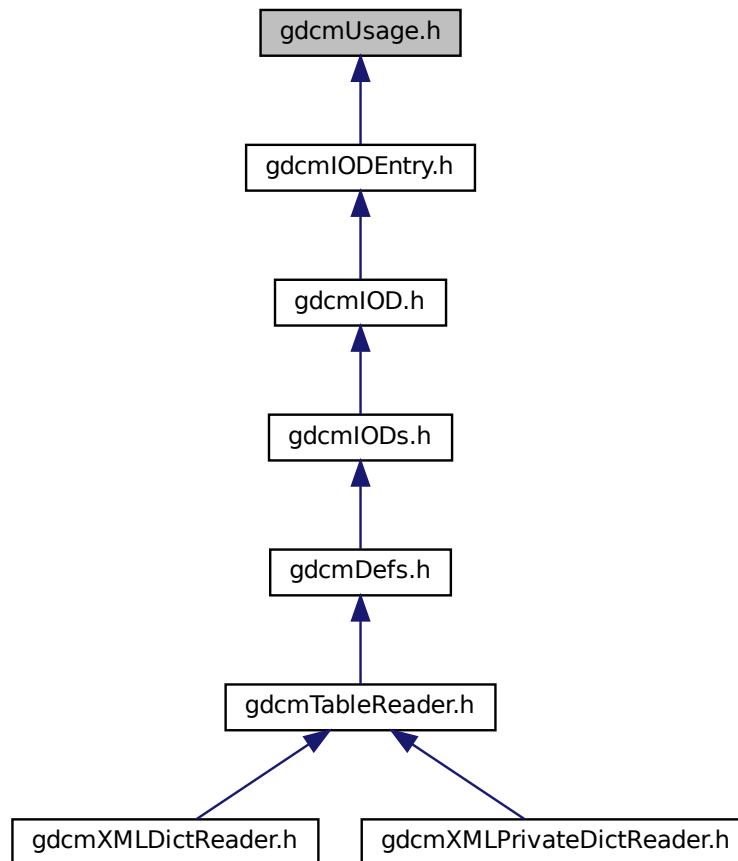
- [gdcm](#)

## 11.280 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 `gdcml::Usage`  
*Usage.*

## Namespaces

- `gdcml`

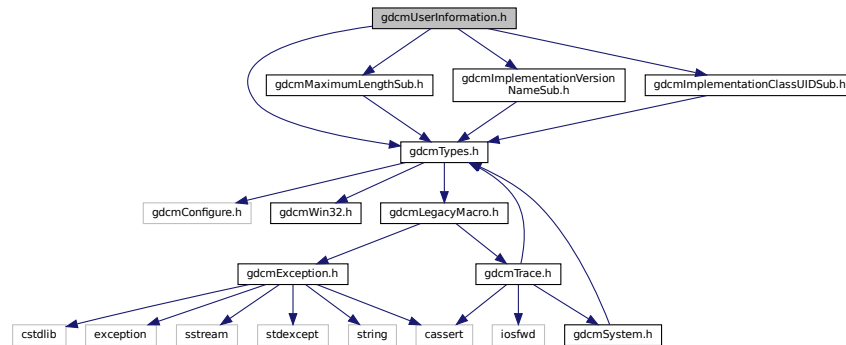
## Functions

- `std::ostream & gdcml::operator<< (std::ostream &_os, const Usage &val)`

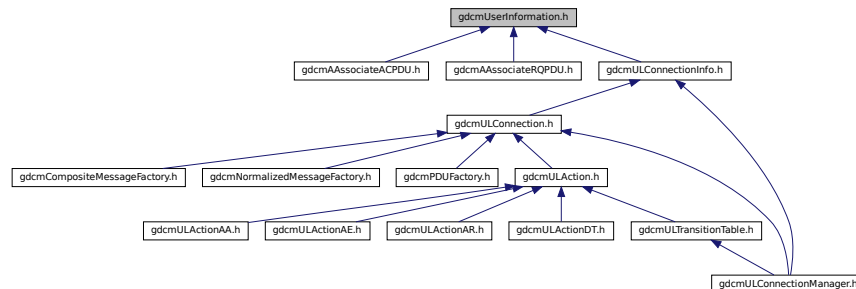
## 11.281 gdcmUserInformation.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmMaximumLengthSub.h"
#include "gdcmImplementationVersionNameSub.h"
#include "gdcmImplementationClassUIDSub.h"
```

Include dependency graph for gdcmUserInformation.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::network::UserInformation](#)  
*UserInformation.*

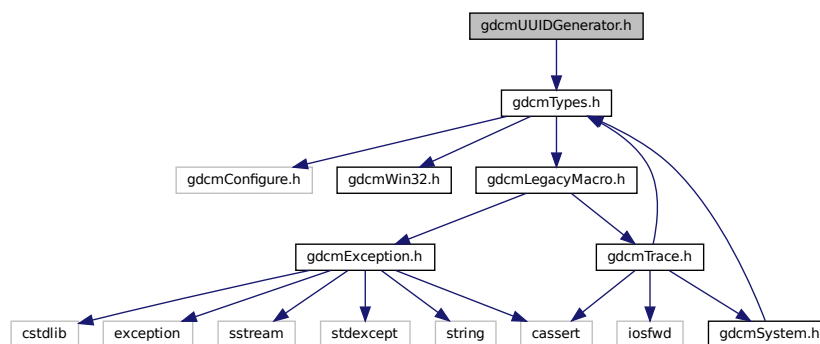
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.282 gdcmUUIDGenerator.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for `gdcmUUIDGenerator.h`:



## Classes

- class `gdcm::UUIDGenerator`  
*Class for generating unique UUID.*

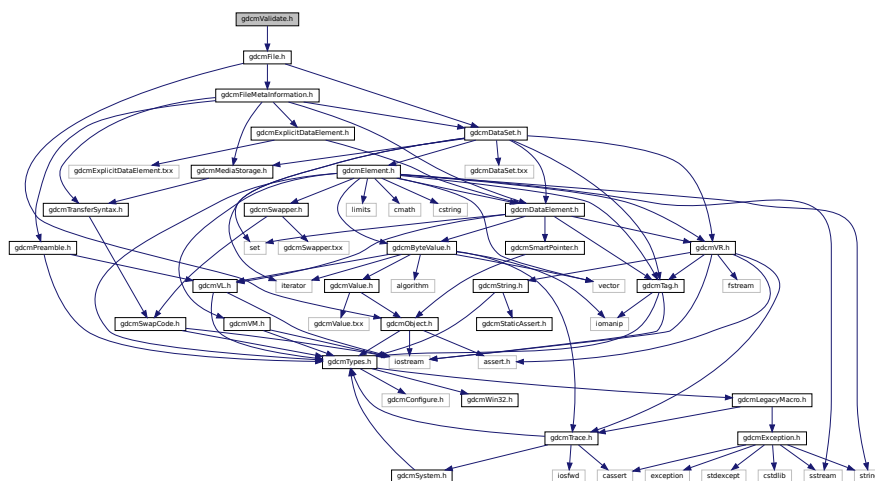
## Namespaces

- **gdcm**

## 11.283 gdcmValidate.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for `gdcmValidate.h`:



## Classes

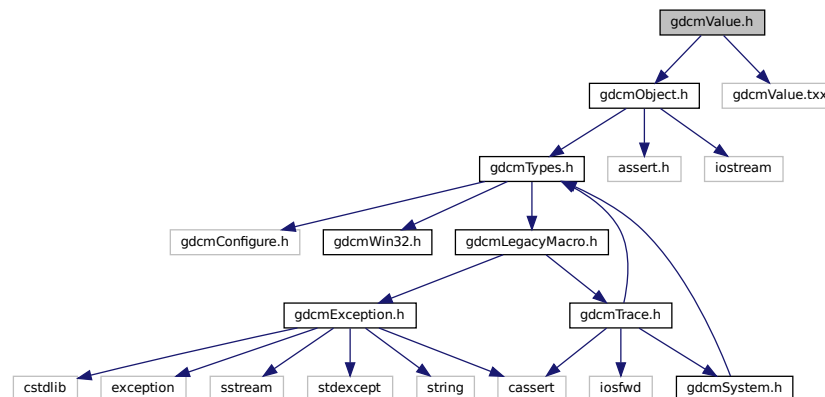
- class [gdcm::Validate](#)  
*Validate* class.

## Namespaces

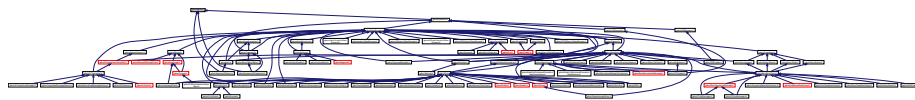
- [gdcm](#)

## 11.284 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

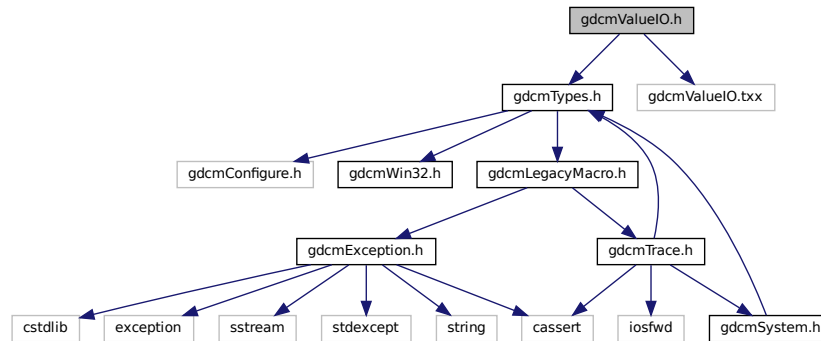
- class [gdcm::Value](#)  
Class to represent the value of a Data [Element](#).

## Namespaces

- [gdcm](#)

## 11.285 gdcmValueIO.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmValueIO.txx"
Include dependency graph for gdcmValueIO.h:
```



### Classes

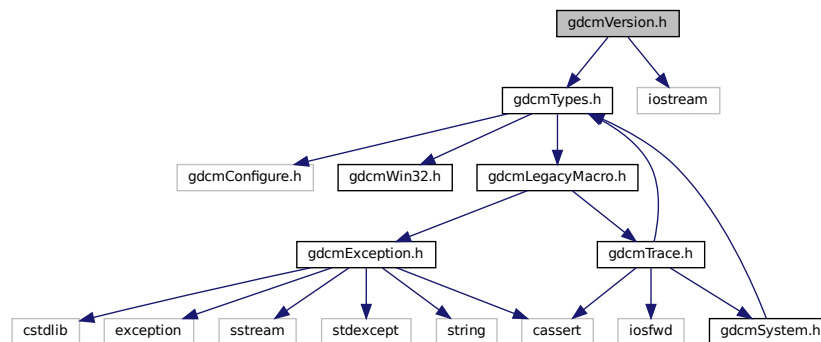
- class [gdcm::ValueIO< TDE, TSwap, TType >](#)  
*Class to dispatch template calls.*

### Namespaces

- [gdcm](#)

## 11.286 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

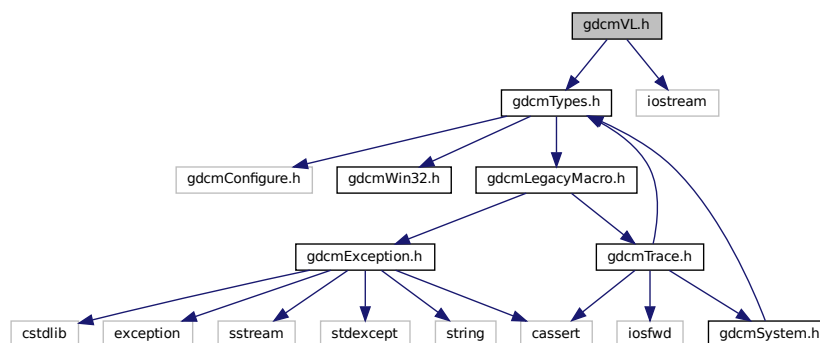
- [gdcm](#)

## Functions

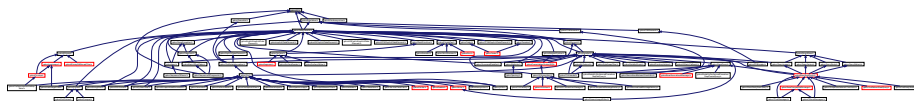
- `std::ostream & gdcm::operator<< (std::ostream &os, const Version &v)`

## 11.287 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

- [gdcm](#)

## Functions

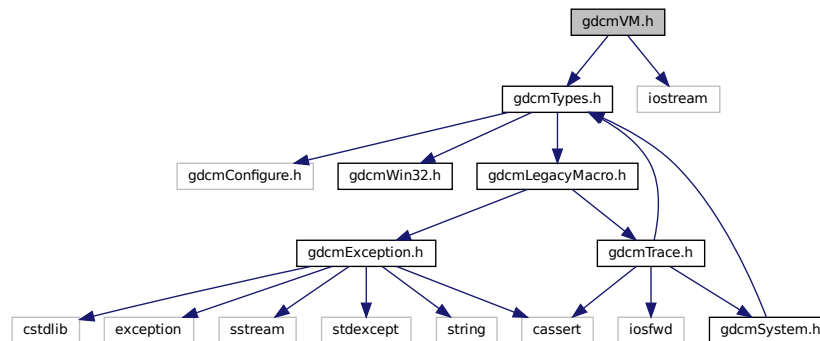
- `std::ostream & gdcm::operator<< (std::ostream &os, const VL &val)`

## 11.288 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

- [gdcm](#)



## Macros

- #define [TYPETOLENGTH](#)(type, length)

## Functions

- std::ostream & [gdcm::operator<<](#) (std::ostream &\_os, const VM &\_val)

### 11.288.1 Macro Definition Documentation

#### 11.288.1.1 TYPETOLENGTH

```
#define TYPETOLENGTH(
    type,
    length )
```

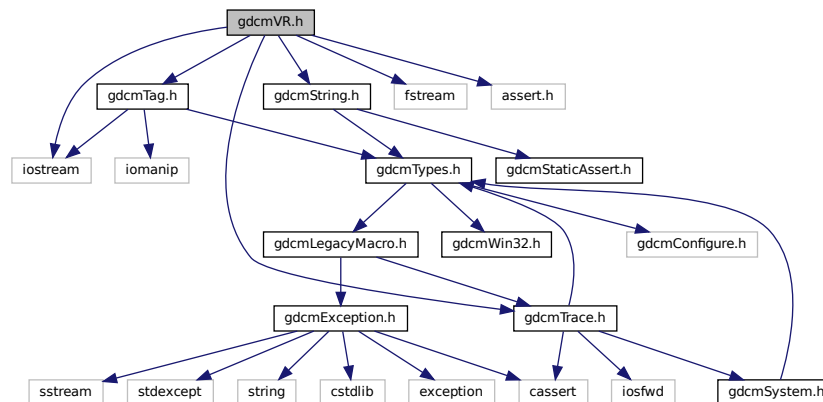
#### Value:

```
template<> struct VMToLength<VM::type> \
{ enum { Length = length }; };
```

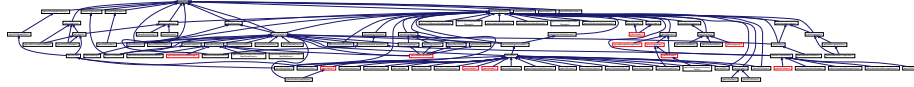
## 11.289 gdcmVR.h File Reference

```
#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

- [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)
- [gdcm::TYPETOENCODING](#) (SQ, VRBINARY, unsigned char) TYPETOENCODING(UN

## Variables

- [gdcm::VRBINARY](#)

## 11.289.1 Macro Definition Documentation

### 11.289.1.1 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; };
```

### 11.289.1.2 VRTypeTemplateCase

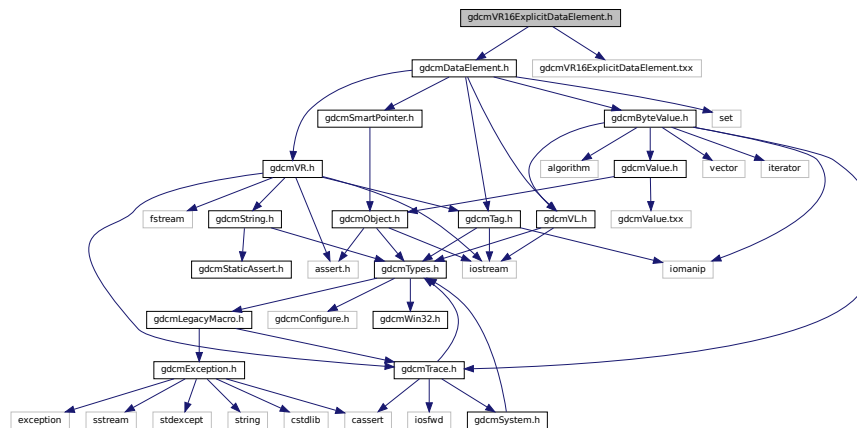
```
#define VRTypeTemplateCase(
    type )
```

#### Value:

```
case VR::type: \
    return sizeof ( VRToType<VR::type>::Type );
```

## 11.290 gdcmVR16ExplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmVR16ExplicitDataElement.txx"
Include dependency graph for gdcmVR16ExplicitDataElement.h:
```



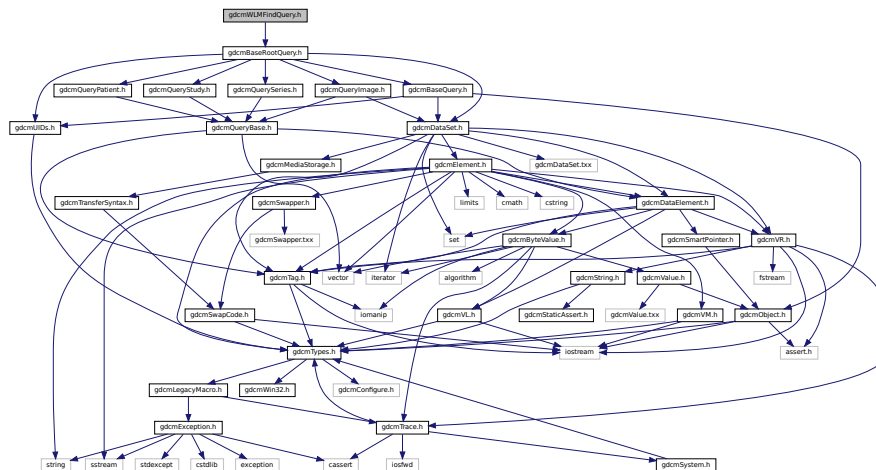


- #define GDCM\_EXPORT

### 11.292.1.1 GDCM\_EXPORT

```
#define GDCM_EXPORT
```

```
#include "gdcmBaseRootQuery.h"
Include dependency graph for gdcmWLMFindQuery.h:
```



- class `gdcm::WLMFindQuery`  
*PatientRootQuery*.

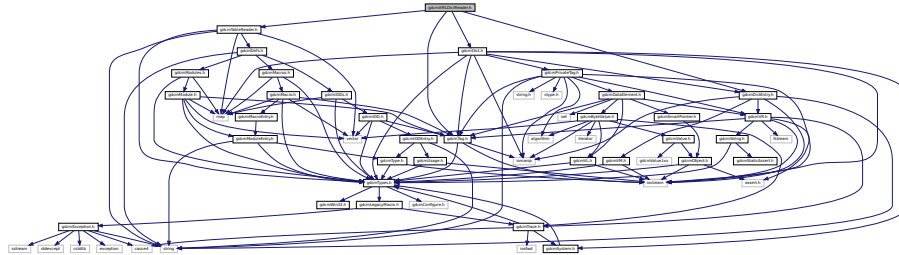
- **gdcm**



## 11.295 gdcmXMLDictReader.h File Reference

```
#include "gdcmTableReader.h"
#include "gdcmDict.h"
#include "gdcmDictEntry.h"
#include "gdcmTag.h"
```

Include dependency graph for gdcmXMLDictReader.h:



### Classes

- class [gdcm::XMLDictReader](#)  
*Class for representing a [XMLDictReader](#).*

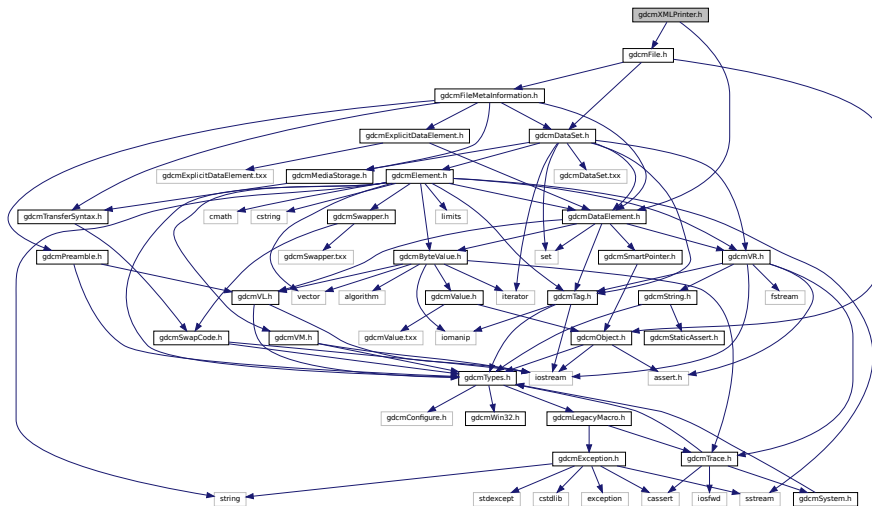
### Namespaces

- [gdcm](#)

## 11.296 gdcmXMLPrinter.h File Reference

```
#include "gdcmFile.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmXMLPrinter.h:





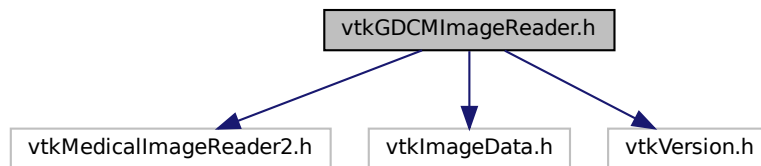


## 11.298 README.txt File Reference

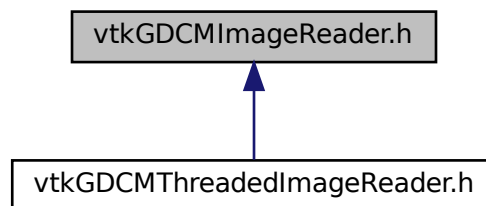
## 11.299 TestsList.txt File Reference

## 11.300 vtkGDCMImageReader.h File Reference

```
#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

- [gdc](#)

## Macros

- `#define VTK_CMYK` 8
- `#define VTK_INVERSE_LUMINANCE` 5
- `#define VTK_LOOKUP_TABLE` 6
- `#define VTK_YBR` 7

### 11.300.1 Macro Definition Documentation

#### 11.300.1.1 VTK\_CMYK

```
#define VTK_CMYK 8
```

#### 11.300.1.2 VTK\_INVERSE\_LUMINANCE

```
#define VTK_INVERSE_LUMINANCE 5
```

#### 11.300.1.3 VTK\_LOOKUP\_TABLE

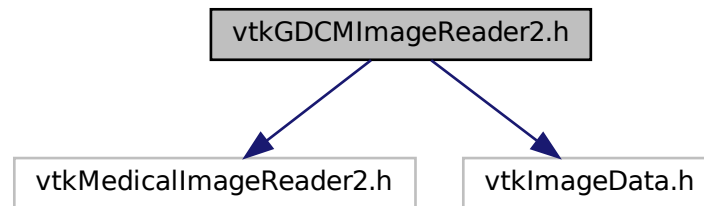
```
#define VTK_LOOKUP_TABLE 6
```

#### 11.300.1.4 VTK\_YBR

```
#define VTK_YBR 7
```

## 11.301 vtkGDCMImageReader2.h File Reference

```
#include "vtkMedicalImageReader2.h"  
#include "vtkImageData.h"  
Include dependency graph for vtkGDCMImageReader2.h:
```



### Classes

- class [vtkGDCMImageReader2](#)

### Namespaces

- [gdcmm](#)

### Macros

- `#define` [VTK\\_CMYK](#) 8
- `#define` [VTK\\_INVERSE\\_LUMINANCE](#) 5
- `#define` [VTK\\_LOOKUP\\_TABLE](#) 6
- `#define` [VTK\\_YBR](#) 7

#### 11.301.1 Macro Definition Documentation

##### 11.301.1.1 VTK\_CMYK

```
#define VTK_CMYK 8
```

### 11.301.1.2 VTK\_INVERSE\_LUMINANCE

```
#define VTK_INVERSE_LUMINANCE 5
```

### 11.301.1.3 VTK\_LOOKUP\_TABLE

```
#define VTK_LOOKUP_TABLE 6
```

### 11.301.1.4 VTK\_YBR

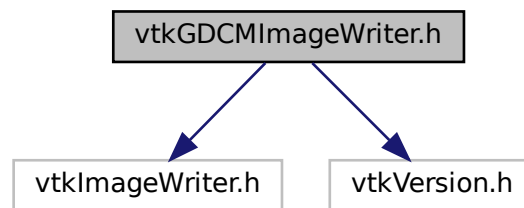
```
#define VTK_YBR 7
```

## 11.302 vtkGDCMImageWriter.h File Reference

```
#include "vtkImageWriter.h"
```

```
#include "vtkVersion.h"
```

Include dependency graph for vtkGDCMImageWriter.h:



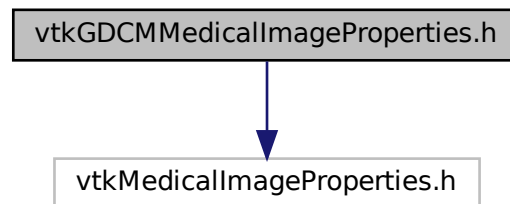
## Classes

- class [vtkGDCMImageWriter](#)

## 11.303 vtkGDCMMedicalImageProperties.h File Reference

```
#include "vtkMedicalImageProperties.h"
```

Include dependency graph for vtkGDCMMedicalImageProperties.h:



### Classes

- class [vtkGDCMMedicalImageProperties](#)

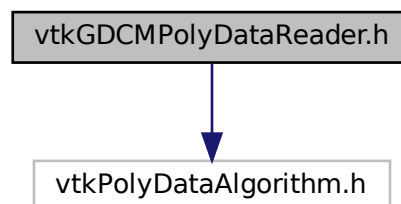
### Namespaces

- [gdcm](#)

## 11.304 vtkGDCMPolyDataReader.h File Reference

```
#include "vtkPolyDataAlgorithm.h"
```

Include dependency graph for vtkGDCMPolyDataReader.h:



## Classes

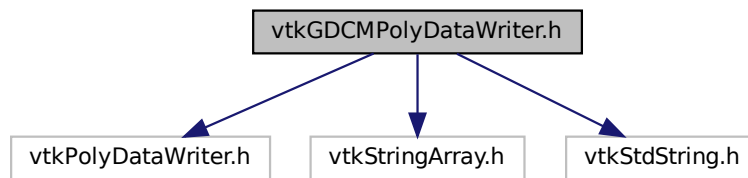
- class [vtkGDCMPolyDataReader](#)

## Namespaces

- [gdcm](#)

## 11.305 vtkGDCMPolyDataWriter.h File Reference

```
#include "vtkPolyDataWriter.h"  
#include "vtkStringArray.h"  
#include "vtkStdString.h"  
Include dependency graph for vtkGDCMPolyDataWriter.h:
```



## Classes

- class [vtkGDCMPolyDataWriter](#)

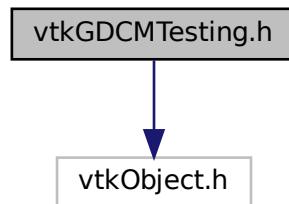
## Namespaces

- [gdcm](#)

## 11.306 vtkGDCMTesting.h File Reference

```
#include "vtkObject.h"
```

Include dependency graph for vtkGDCMTesting.h:



### Classes

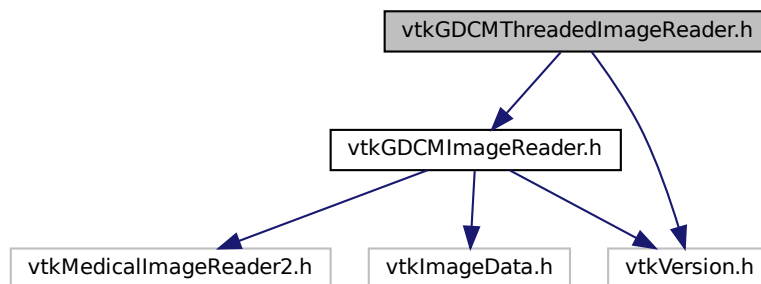
- class [vtkGDCMTesting](#)

## 11.307 vtkGDCMThreadedImageReader.h File Reference

```
#include "vtkGDCMImageReader.h"
```

```
#include "vtkVersion.h"
```

Include dependency graph for vtkGDCMThreadedImageReader.h:



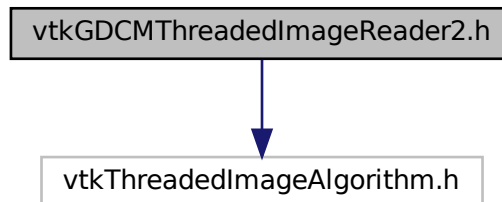
### Classes

- class [vtkGDCMThreadedImageReader](#)

## 11.308 vtkGDCMThreadedImageReader2.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkGDCMThreadedImageReader2.h:



### Classes

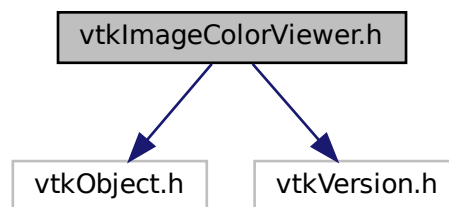
- class [vtkGDCMThreadedImageReader2](#)

## 11.309 vtkImageColorViewer.h File Reference

```
#include "vtkObject.h"
```

```
#include "vtkVersion.h"
```

Include dependency graph for vtkImageColorViewer.h:



### Classes

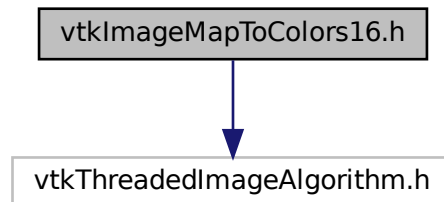
- class [vtkImageColorViewer](#)



## 11.310 vtkImageMapToColors16.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageMapToColors16.h:



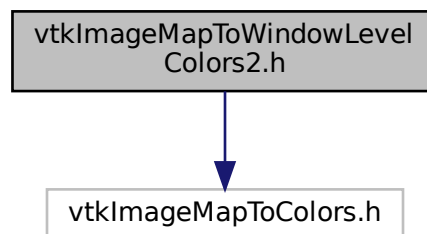
### Classes

- class [vtkImageMapToColors16](#)

## 11.311 vtkImageMapToWindowLevelColors2.h File Reference

```
#include "vtkImageMapToColors.h"
```

Include dependency graph for vtkImageMapToWindowLevelColors2.h:



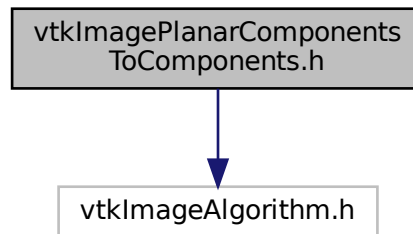
### Classes

- class [vtkImageMapToWindowLevelColors2](#)

## 11.312 vtkImagePlanarComponentsToComponents.h File Reference

```
#include "vtkImageAlgorithm.h"
```

Include dependency graph for vtkImagePlanarComponentsToComponents.h:



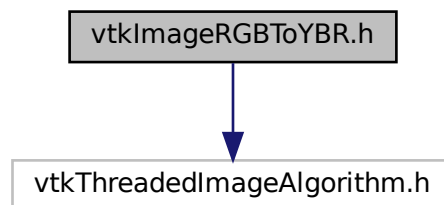
### Classes

- class [vtkImagePlanarComponentsToComponents](#)

## 11.313 vtkImageRGBToYBR.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageRGBToYBR.h:



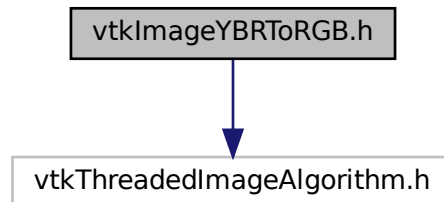
### Classes

- class [vtkImageRGBToYBR](#)

## 11.314 vtkImageYBRToRGB.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageYBRToRGB.h:



### Classes

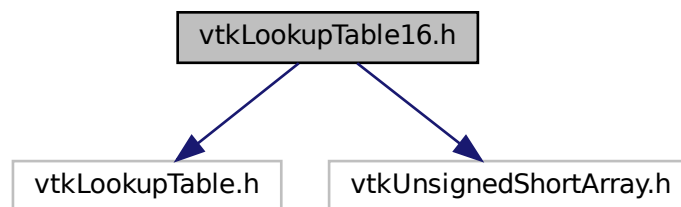
- class [vtkImageYBRToRGB](#)

## 11.315 vtkLookupTable16.h File Reference

```
#include "vtkLookupTable.h"
```

```
#include "vtkUnsignedShortArray.h"
```

Include dependency graph for vtkLookupTable16.h:



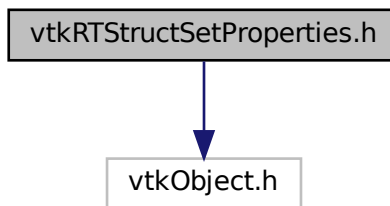
### Classes

- class [vtkLookupTable16](#)

## 11.316 vtkRTStructSetProperties.h File Reference

```
#include "vtkObject.h"
```

Include dependency graph for vtkRTStructSetProperties.h:



### Classes

- class [vtkRTStructSetProperties](#)

## Chapter 12

# Example Documentation

### 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 <string.h> // memcpy
int myfunc()
{
    char vl_str[4];
    const char raw[] = "\000\000\000\004";
    memcpy(vl_str, raw, 4);
    uint32_t vl;
    gdcm::ByteSwap<uint32_t>::SwapRangeFromSwapCodeIntoSystem((uint32_t*)(&vl_str), gdcm::SwapCode::BigEndian, 1);
    memcpy(&vl, vl_str, 4);
    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 = gdcmm::SwapCode::BigEndian;
    }
else if ( gdcmm::ByteSwap<uint16_t>::SystemIsLittleEndian() )
    {
        sc = gdcmm::SwapCode::LittleEndian;
    }
if( sc == gdcmm::SwapCode::Unknown )
    {
        std::cerr << "unk" << std::endl;
        return 1;
    }
//std::cout << "sc: " << sc << std::endl;
uint16_t t = 0x1234;
gdcmm::ByteSwap<uint16_t>::SwapFromSwapCodeIntoSystem(t, sc);
if( sc == gdcmm::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 == gdcmm::SwapCode::LittleEndian )
    {
        if( t != 0x1234 )
        {
            std::cerr << std::hex << "t: " << t << std::endl;
            return 1;
        }
    }
union { char n[2]; uint16_t tn; } u16;
memcpy(u16.n, &t, 2 );
gdcmm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem(&u16.tn, sc, 1);
uint16_t tn = u16.tn;
if( sc == gdcmm::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 == gdcmm::SwapCode::LittleEndian )
    {
        if( tn != 0x1234 )
        {
            std::cerr << std::hex << "tn: " << tn << std::endl;
            return 1;
        }
    }
gdcmm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem(&u16.tn, gdcmm::SwapCode::BigEndian, 1);
tn = u16.tn;
if( sc == gdcmm::SwapCode::LittleEndian )
    {
        if( tn != 0x3412 )
        {
            std::cerr << std::hex << "tn: " << tn << std::endl;
            return 1;
        }
    }
else if ( sc == gdcmm::SwapCode::BigEndian )
    {
        if( tn != 0x1234 )
        {
            std::cerr << std::hex << "tn: " << tn << std::endl;
            return 1;
        }
    }
}
if( myfunc() )
    {
        return 1;
    }
uint16_t array[] = { 0x1234 };
gdcmm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem(array,
    gdcmm::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() );
    }
}

```

```

{
    gdcmm::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
{
    gdcmm::Attribute<0x28,0x8> at;
    at.SetFromDataElement( ds.GetDataElement( at.GetTag() ) );
    at.SetValue( at.GetValue() * 2 );
    ds.Replace( at.GetAsDataElement() );
}
gdcmm::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:
gdcmm::ImageReader ir;
ir.SetFileName( out );
if( !ir.Read() )
{
    return 1;
}
gdcmm::Image &image = ir.GetImage();
unsigned long len = image.GetBufferLength();
const gdcmm::ByteValue *bv = ir.GetFile().GetDataSet().GetDataElement( gdcmm::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 gdcmm::SWIGDataSet

```

/*=====
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.

=====*/
/*
    Converter converter = new Converter();
    int a = converter.Convert<int>( some_int_blob );
    double b = converter.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`

```

1
14
15 import os,sys
16 import gdcm
17
18 def TestRead(filename, verbose = False):
19     r = gdcm.Reader()
20     r.SetFileName( filename )
21     success = r.Read()
22     #if verbose: print r.GetFile()
23     if verbose: print (r.GetFile().GetDataSet())
24     return success
25
26 if __name__ == "__main__":
27     success = 0
28     try:
29         filename = os.sys.argv[1]
30         success += TestRead( filename, True )
31     except:
32         # loop over all files:

```

```

33     gdcmm.Trace.DebugOff()
34     gdcmm.Trace.WarningOff()
35     t = gdcmm.Testing()
36     nfiles = t.GetNumberOfFileNames()
37     for i in range(0,nfiles):
38         filename = t.GetFileName(i)
39         success += TestRead( filename )
40
41
42     # Test succeed ?
43     sys.exit(success == 0)

```

## 12.6 DecompressJPEGFile.cs

This is a C# example on how to use [gdcmm::SequenceOfFragments](#)

```

/*=====
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.

=====*/
/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcmm/debug-gcc/bin
 * $ mono bin/DecompressJPEGFile.exe somejpegfile.jpg
 */
using System;
using gdcmm;
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 = gdcmm.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 gdcmm.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
        // gdcmm.SequenceOfFragments
        //pixeldata.SetByteValue( jstream, new gdcmm.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 gdcmm.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_FULL );
        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 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 ClinicalTrialIdentificationWorkflow
{
    public static bool ProcessOneFile( gdcm.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 gdcml.Tag(0x0012,0x0010), "MySponsorName");
ano.Replace( new gdcml.Tag(0x0012,0x0020), "MyProtocolID");
ano.Replace( new gdcml.Tag(0x0012,0x0021), "MyProtocolName");
ano.Replace( new gdcml.Tag(0x0012,0x0030), "MySiteId");
ano.Replace( new gdcml.Tag(0x0012,0x0031), "MySiteName");
ano.Replace( new gdcml.Tag(0x0012,0x0040), "MySponsorId");
ano.Replace( new gdcml.Tag(0x0012,0x0050), "MyTPId");
ano.Replace( new gdcml.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 gdcml.Tag(0x0012,0x0062), "YES");
//ano.Replace( new gdcml.Tag(0x0012,0x0063), "My Super Duper Anonymization Overload");
// We might be generating a subdirectory. Let's make sure the subdir exist:
gdcml.Filename fn = new gdcml.Filename( outfilename );
string subdir = fn.GetPath();
if( !gdcml.PosixEmulation.MakeDirectory( subdir ) )
{
    return false;
}
gdcml.FileMetaInformation fmi = ano.GetFile().GetHeader();
// The following three lines make sure to regenerate any value:
fmi.Remove( new gdcml.Tag(0x0002,0x0012) );
fmi.Remove( new gdcml.Tag(0x0002,0x0013) );
fmi.Remove( new gdcml.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)
{
    gdcml.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";
    gdcml.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
    System.Console.WriteLine( "Root dir is now: " + gdcml.UIDGenerator.GetRoot() );
    gdcml.Global global = gdcml.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( !gdcml.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:
    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( 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 GenFakeImage.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 pupose 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 pupose 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 = gdcm.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.GetFilesNames();
    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 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.TSType.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;
    }
}

```

## 12.18 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.GetFilenames();
        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 file are not guarantee 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;
    }
}

```

## 12.19 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;
}
}

```

## 12.20 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.FindElement( 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;
}
}

```

## 12.21 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;
}
}

```

## 12.22 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/rle16l0o.dcm
 * $ md5sum /tmp/frame_rgb.raw
 * 73bf61325fdb6e2830244a2b7b0c4ae2 /tmp/frame_rgb.raw
 * $ gdcimg --depth 16 --spp 3 --size 600,430 /tmp/frame_rgb.raw rgb.dcm
 * $ gdcviewer 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;
}
}

```

## 12.23 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;
}
}

```

## 12.24 FileAnonymize.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/FileAnonymize.exe input.dcm output.dcm
 */
using System;
using gdcm;
public class FileAnonymize
{
    public static int Main(string[] args)
    {
        string filename = args[0];
        string outfilename = args[1];
        gdcm.FileAnonymizer fa = new gdcm.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;
}
}

```

## 12.25 FileChangeTS.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
 *
 * 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;
}
}

```

## 12.26 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.TSType.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;
    }
}

```

## 12.27 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;
    }
}

```

## 12.28 GetArray.cs

```

/*=====
Program: GDICOM (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 rl = lut.GetLUTLength( LookupTable.LookupTableType.RED );
    //byte[] rbuf = new byte[ rl ];
    //uint rl2 = lut.GetLUT( LookupTable.LookupTableType.RED, rbuf );
    //assert rl == rl2;
    //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;
}
}

```

## 12.29 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 Image 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.infompeg?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;
}

```

## 12.30 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;
}
}

```

## 12.31 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;
    }
}

```

```

    }
}

```

## 12.32 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;
    }
}

```

## 12.33 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;
    }
}

```

## 12.34 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;
    }
}

```

## 12.35 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 gdcmlData/MR_Spectroscopy_SIEMENS_OF.dcm
 *
 */
#include "gdcmlServiceClassUser.h"
#include "gdcmlSimpleSubjectWatcher.h"
#include "gdcmlProgressEvent.h"
#include "gdcmlDirectory.h"
#include "gdcmlPresentationContextGenerator.h"
#include <QApplication>
#include <QProgressDialog>
#include <QVBoxLayout>
namespace gdcml {
/*
 * 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 gdcml
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::ostream 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();
}

```

## 12.36 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
    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();
    // Declare private tag we need to find:
    gdcm::PrivateTag pt1( 0x29,0x18, "SIEMENS CSA HEADER" );
    gdcm::PrivateTag pt2( 0x29,0x34, "SIEMENS MEDCOM HEADER" );
    gdcm::PrivateTag pt3( 0x29,0x60, "SIEMENS MEDCOM HEADER2" );
    const char str1[] = "GDCM was here 3!";
    if( !ds.FindDataElement( pt1 ) ) return 1;
    gdcm::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;
    gdcm::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;
    gdcm::DataElement de3 = ds.GetDataElement( pt3 );
    std::cout << de3 << std::endl;
    de3.SetByteValue( str3, (uint32_t)strlen(str3) );
    ds.Replace( de3 );
    gdcm::Writer writer;
    writer.SetFile( file );
    writer.SetFileName( outfile );
    if ( !writer.Write() )
    {
        return 1;
    }
    return 0;
}

```

## 12.37 ChangeSequenceUltrasound.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 "gdcmSmartPointer.h"

```

```

#include "gdcmDataSetHelper.h"
/*
  ./ChangeSequenceUltrasound gdcmData/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];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if ( ! reader.Read() )
    {
        return 1;
    }
    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();
    gdcm::Tag tsis(0x0008,0x2112); // SourceImageSequence
    if ( ds.FindDataElement( tsis ) )
    {
        const gdcm::DataElement &sis = ds.GetDataElement( tsis );
        gdcm::SmartPointer<gdcm::SequenceOfItems> sqsis = sis.GetValueAsSQ();
        if ( sqsis && sqsis->GetNumberOfItems() )
        {
            gdcm::Item &item1 = sqsis->GetItem(1);
            gdcm::DataSet &nestedds = item1.GetNestedDataSet();
            gdcm::Tag tprcs(0x0040,0xa170); // PurposeOfReferenceCodeSequence
            if( nestedds.FindDataElement( tprcs ) )
            {
                const gdcm::DataElement &prcs = nestedds.GetDataElement( tprcs );
                gdcm::SmartPointer<gdcm::SequenceOfItems> sqprcs = prcs.GetValueAsSQ();
                if ( sqprcs && sqprcs->GetNumberOfItems() )
                {
                    gdcm::Item &item2 = sqprcs->GetItem(1);
                    gdcm::DataSet &nestedds2 = item2.GetNestedDataSet();
                    // (0008,0104) LO [Uncompressed predecessor] # 24, 1 CodeMeaning
                    gdcm::Tag tcm(0x0008,0x0104);
                    if( nestedds2.FindDataElement( tcm ) )
                    {
                        gdcm::DataElement cm = nestedds2.GetDataElement( tcm );
                        std::string mystr = "GDCM was here";
                        cm.SetByteValue( mystr.c_str(), (uint32_t)mystr.size() );
                        nestedds2.Replace( cm );
                    }
                }
            }
        }
    }
    gdcm::Writer writer;
    writer.SetFile( file );
    writer.SetFileName( outfile );
    if ( !writer.Write() )
    {
        return 1;
    }
    return 0;
}

```

## 12.38 CheckBigEndianBug.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.
=====

```

```

=====*/
/*
 * WARNING: This is a dev tool, do not use !
 *
 * Usage: after a gdcmconv, 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;
}

```

## 12.39 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( outfile );
    if( !writer.Write() )
    {
        return 1;
    }
}

```

```

    return 0;
}

```

## 12.40 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();
    const gdcm::Image &image = reader.GetImage();
    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( outfile, std::ios::binary );
    //image.GetBuffer2(out);
    //out.close();
    gdcm::ImageWriter writer;
    writer.SetImage( change.GetOutput() );
    writer.SetFile( reader.GetFile() );
    writer.SetFileName( outfile );
    if( !writer.Write() )
    {
        return 1;
    }
    return 0;
}

```

## 12.41 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).
        imageQt = 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;
            }
            imageQt = 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++;
    }
    imageQt = 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( outfile );
    if( !writer.write( *imageQt ) )
    {
        return 1;
    }
    return 0;
}

```

## 12.42 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( outfile );
    if( !writer.Write() )
    {
        return 1;
    }
    delete[] buf;
    return 0;
}

```

## 12.43 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;
}

```

## 12.44 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( gdcmm::Tag(0x0010,0x20), "012345" );
anon.Empty( gdcmm::Tag(0x0010,0x30) );
anon.Empty( gdcmm::Tag(0x0010,0x40) );
anon.Empty( gdcmm::Tag(0x0008,0x20) );
anon.Empty( gdcmm::Tag(0x0008,0x30) );
anon.Empty( gdcmm::Tag(0x0008,0x90) );
anon.Empty( gdcmm::Tag(0x0020,0x10) );
anon.Empty( gdcmm::Tag(0x0020,0x11) );
anon.Empty( gdcmm::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;
}

```

## 12.45 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 );
}

```

```

gdcmm::Attribute<0x8,0x18> sopinstanceuid;
sopinstanceuid.SetFromDataSet( ds1 );
// Step 2: DERIVED object
gdcmm::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;
}
gdcmm::Writer w;
w.SetFile( r2.GetFile() );
w.SetFileName( "derived.dcm" );
if( !w.Write() )
{
    return 1;
}
return 0;
}

```

## 12.46 DiffFile.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"
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];
    gdcmm::Reader reader1;
    reader1.SetFileName( filename1 );
    if( !reader1.Read() )
    {
        return 1;
    }
    gdcmm::Reader reader2;
    reader2.SetFileName( filename2 );
    if( !reader2.Read() )
    {
        return 1;
    }
    const gdcmm::File &file1 = reader1.GetFile();
    const gdcmm::File &file2 = reader2.GetFile();
    const gdcmm::DataSet &ds1 = file1.GetDataSet();
    const gdcmm::DataSet &ds2 = file2.GetDataSet();
    gdcmm::DataSet::ConstIterator it1 = ds1.Begin();
    gdcmm::DataSet::ConstIterator it2 = ds2.Begin();
    const gdcmm::DataElement &de1 = *it1;
    const gdcmm::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;
}

```

## 12.47 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::FileNamesType > 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 discriminiat 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.GetFilesNames() );
}
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
        gdcm::DirectionCosines ref;
        gdcm::DirectionCosines dc;
        for(
            std::set< std::string >::const_iterator it = iopset.begin();
            it != iopset.end(); ++it )
        {
            ref.SetFromString( it->c_str() );
            for(
                Directory::FileNamesType::const_iterator file = files.begin();
                file != files.end(); ++file)
            {
                std::string value = s.GetValue(file->c_str(), gdcm::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::FileNamesType 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::FileNamesType seriesfiles = GetAllFileNamesFromTagToValue(
        s, s.GetFilesNames(), t2, seriesuid);
    gdcm::Scanner::ValuesType vt3 = s.GetValues(t3);
    for(
        gdcm::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;
    gdcm::Scanner::ValuesType vt2 = s.GetValues(t2);
    for(
        gdcm::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::FileNamesType >::const_iterator it = SortedFiles.begin();
        it != SortedFiles.end(); ++it )
    {
        os << "Group: " << std::endl;
        for(
            Directory::FileNamesType::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 dir1;
    if( argc < 2 )
    {
        const char *extradataroot = nullptr;
#ifdef GDCM_BUILD_TESTING
        extradataroot = gdcm::Testing::GetDataExtraRoot();
#endif
        if( !extradataroot )
        {
            return 1;
        }
        dir1 = extradataroot;
        dir1 += "/gdcmSampleData/ForSeriesTesting/VariousIncidences/ST1";
    }
    else
    {
        dir1 = argv[1];
    }
    gdcm::Directory d;
    d.Load( dir1.c_str(), 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.GetFileNames() );
    if( !b )
    {
        std::cerr << "Scanner failed" << std::endl;
        return 1;
    }
    gdcm::DiscriminateVolume dv;
    dv.ProcessIntoVolume( s );
    dv.Print( std::cout );
    return 0;
}

```

## 12.48 DumpADAC.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 goal of this example is to mimic the behavior of disp_img_header
 * see http://www.gmecorp-usa.com/IM/NM/GC/ADAC/SV/adactechtips/Released_01Q3.pdf
 */
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmAttribute.h"
#include "gdcmImageWriter.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 << "\\n";
            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::stringstream is;
    std::string dup( bv->GetPointer(), bv->GetLength() );
    is.str( dup );
    bool b = DumpADAC( is );
    if( !b ) return 1;
    return 0;
}

```

## 12.49 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 DTCOM 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 );
    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 );
    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 << gettypenamefromtype(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 ?
    (ffff,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 ?
    (ffff,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) );
        // ulgy 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());
    #if 0
        std::ofstream out2( "debug" );
        out2.write( b64.c_str(), b64.size() );
        out2.close();
    #endif
    const size_t dlen = gdcmm::Base64::GetDecodeLength(b64.c_str(), b64.size() );
    std::string decoded;
    decoded.resize( dlen );
    gdcmm::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 gdcmm::ByteValue * bv1 = del.GetByteValue();
    gdcmm::Element<gdcmm::VR::SL,gdcmm::VM::VM1> dlen = {{01}};
    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() );
    #if 1
        static int c = 0;
        std::string fn0 = gdcmm::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
    #if 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 = gdcmm::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

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( s1 == "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();
    }

    #if 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/l,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
66\05\00\00\3b\01\00\00\4a\0a\00\00\0e\00\00\00\7a\0a\00\00\95\01\00\00\08\00\00\00\1b\00\00\00\43\47\45\4e\5f\75\73\65\72\
# 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;
}

```

## 12.50 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 PrintNameValuePairMapping( 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> el1;
        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 tvaluefd1(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 );
    gdcm::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 );
    // gdcm::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 );
    gdcm::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 );
    gdcm::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 );
    gdcm::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 );
//      gdcm::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 );
//      gdcm::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 );
//      gdcm::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( gdcm::PrivateTag const & privtag, const gdcm::DataSet & ds,
gdcm::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 ni1 = sqi_values10->GetNumberOfItems();

```

```

// assert( nil == 1 );
for( size_t i1 = 1; i1 <= nil; ++i1 )
{
    gdcmm::Item &item_10 = sqi_values10->GetItem(i1);
    gdcmm::DataSet &ds10 = item_10.GetNestedDataSet();
    assert( ds10.Size() == 2 + 3 );
    // (7fel,0010)
    // (7fel,1012)
    // (7fel,1018)
    // (7fel,1020)
    // (7fel,1083)
    PrintNameValueMapping3( privtag1, privtag2, ds10, sqi_dict, " " );
    std::cout << std::endl;
    const gdcmm::PrivateTag tseq_values20(0x7fel,0x20,"GEMS_Ultrasound_MovieGroup_001");
    if( !ds10.FindDataElement( tseq_values20 ) )
    {
        assert( 0 );
        return false;
    }
    const gdcmm::DataElement& seq_values20 = ds10.GetDataElement( tseq_values20 );
    gdcmm::SmartPointer<gdcmm::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 );
        // (7fel,0010)
        // (7fel,1024)
        // (7fel,1026)
        // (7fel,1036)
        // (7fel,103a)
        // (7fel,1083) (*)
        const gdcmm::PrivateTag tseq_values20name(0x7fel,0x24,"GEMS_Ultrasound_MovieGroup_001");
        const gdcmm::PrivateTag tseq_values26(0x7fel,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(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 tseq_dict(0x7fel,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(0x7fel,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(0x7fel,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;
}
```

## 12.51 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 = 0xe000fffe;
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;
bytes.resize( l - 16 );
if( !bytes.empty() )
{
    is.read( &bytes[0], l - 16 );
}
//os << "pos:" << is.tellg() << std::endl;
if( strcmp(str, "TUSREMEASUREMENT" ) == 0 )
{
    const char *p = &bytes[0];
    uint32_t val;
    memcpy( (char*)&val, p, sizeof(val) );
    os << " " << val << std::endl;
    p += sizeof(val);
    memcpy( (char*)&val, p, sizeof(val) );
    os << " " << val << std::endl;
    p += sizeof(val);
    memcpy( (char*)&val, p, sizeof(val) );
    os << " " << val << std::endl;
    p += sizeof(val);
    memcpy( (char*)&val, p, sizeof(val) );
    os << " " << val << std::endl;
    p += sizeof(val);
    memcpy( (char*)&val, p, sizeof(val) );
    os << " " << val << std::endl;
    p += sizeof(val);
    memcpy( (char*)&val, p, sizeof(val) );
    os << " " << val << std::endl;
    p += sizeof(val);
    memcpy( (char*)&val, p, sizeof(val) );
    os << " " << val << std::endl;
    p += sizeof(val);
}
#if 0
float f;
memcpy( (char*)&f, p, sizeof(f) );
os << " " << f << std::endl;
p += sizeof(f);
#else
memcpy( (char*)&val, p, sizeof(val) );
os << " " << val << std::endl;
p += sizeof(val);
#endif
memcpy( (char*)&val, p, sizeof(val) );
os << " " << val << std::endl;
p += sizeof(val);
char str2[17];
memcpy( str2, p, 16 );
str2[16] = 0;
os << " " << str2 << std::endl;
}
#if 0
std::ofstream out( str, std::ios::binary );
out.write( (char*)&magic, sizeof( magic ) );
out.write( (char*)&l, sizeof( l ) );
out.write( str, 16 );
out.write( &bytes[0], bytes.size() );
#endif
return is;
}
static bool DumpImageHeaderInfo( std::istream & is, size_t refln )
{
    // TUSNONIMAGESTAM (5176)
    // TUSREMEASUREMEN (1352)
    // TUSBSINGLELAYOU (16)
    // TUSCLIPPARAMETE (104)
    element el;
    while( el.read( is ) )
    {
    }
    //size_t pos = is.tellg();
    //assert( pos == refln );
    (void)refln;
    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();
    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::istream is;
    std::string dup( bv->GetPointer(), bv->GetLength() );
    is.str( dup );
    bool b = DumpImageHeaderInfo( is, bv->GetLength() );
    if( !b ) return 1;
#ifdef 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;
}

```

## 12.52 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 > crcheaders;
    crcheaders.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 ) );
#if 0
            std::cout << header.val0
                << " " << header.val1[0]
                << " " << header.val1[1]
                << " " << header.val2[0]
                << " " << header.val2[1]
                << " " << header.imgsize << std::endl;
#endif
            crcheaders.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 << '_';
        //ss << crcheaders[0].imgsize; // FIXME: Assume all header are identical !
        ss << size[0];
        ss << '_';
        ss << size[1];
        ss << '_';
        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 == crcheaders[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[0];
            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[0], 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 > crcheaders;
    crcheaders.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
    crcheaders.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 << outfile;
    ss << '_';
    ss << crcheaders[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[0];
    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;
}
#endif 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> sqi2 = seq2.GetValueAsSQ();
assert( sqi2->GetNumberOfItems() >= 1 );
// FIXME: what if not in first Item ?
assert( sqi2->GetNumberOfItems() == 1 );
Item &item2 = sqi2->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( strncmp(bv->GetPointer(), "ZLib", 4) == 0 )
    {
        if( !ProcessDeflate( outfile, nslicesref, zallocref, bv2->GetPointer(),
            std::streampos(bv2->GetLength()), bv3->GetPointer(), bv3->GetLength() ) )
        {
            return 1;
        }
    }
    else if( strncmp(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;
}

```

## 12.53 DumpSiemensBase64.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.

=====*/
/*
 * https://groups.google.com/forum/#!msg/comp.protocols.dicom/2kZ21LP8EcM/WzjFrtjnAgAJ
 */
#include "gdcmlReader.h"
#include "gdcmlPrivateTag.h"
#include "gdcmlPrinter.h"
#include "gdcmlDictPrinter.h"
#include "gdcmlCSAHeader.h"
#include "gdcmlBase64.h"
#include "gdcmlExplicitDataElement.h"
#include "gdcmlSwapper.h"
#include "gdcmlPrinter.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];
    gdcml::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }
    const gdcml::DataSet& ds = reader.GetFile().GetDataSet();
    gdcml::CSAHeader csa;
    const gdcml::PrivateTag &t1 = csa.GetCSAImageHeaderInfoTag();
    if( !ds.FindElement( t1 ) ) return 1;
    csa.LoadFromDataElement( ds.GetDataElement( t1 ) );
    //const char name[] = "MRDiffusion";
    const char name[] = "MR_AS_L";
    if( csa.FindCSAElementByName( name ) )
    {
        const gdcml::CSAElement &el = csa.GetCSAElementByName( name );
        const gdcml::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 = gdcml::Base64::GetDecodeLength( str.c_str(), str.size() );
        std::vector<char> buf;
        buf.resize( dl );
        size_t dl2 = gdcml::Base64::Decode( &buf[0], buf.size(), str.c_str(), str.size() );
        (void)dl2;
        std::stringstream ss;
        ss.str( std::string( &buf[0], buf.size() ) );
        gdcml::File file;
        gdcml::DataSet &ds2 = file.GetDataSet();
        gdcml::DataElement xde;
        try
        {
            while( xde.Read<gdcml::ExplicitDataElement, gdcml::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;
}

```

## 12.54 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 gdcmm::Directory;
using gdcmm::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;
}

```

## 12.55 DumpToshibaDTI.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.

=====*/
/*
* 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[0], 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 );
    }
    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;
}

```

## 12.56 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;
}

```

## 12.57 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](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) );
    }
    gdcm::DataSet::ConstIterator it = dup.Begin();
    for( ; it != dup.End(); ++it )
    {
        ds.Insert( *it );
    }
    gdcm::Writer w;
    w.SetFile( file );
    w.SetFileName( outfile );
    if( !w.Write() )

```

```

    {
        return 1;
    }
    return 0;
}

```

## 12.58 ELSCINT1WaveToText.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 "gdcmlReader.h"
#include "gdcmlPrivateTag.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 (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:
 * 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' << buffer[i+1] <<
                '\\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' << buffer[i+74] <<
                '\\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' << " " << '\\t' << buffer[i] <<
                '\\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] <<
                '\\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' << buffer[i+74] <<
                '\\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 << '\n';
        else
        {
            if (buffer[i+75] == 0)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0 << '\n';
            if (buffer[i+75] == 16384)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0 << '\n';
            if (buffer[i+75] == 256)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0 << '\n';
            if (buffer[i+75] == -32768)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1 << '\n';
            if (buffer[i+75] == -16384)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1 << '\n';
            if (buffer[i+75] == -32512)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1 << '\n';
        }
    }
    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.GetValue();
    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;
}
```

## 12.59 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;
}

```

## 12.60 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;
}

```

## 12.61 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);

```

```

gdcmm::DataSet &nesteddds = item.GetNestedDataSet();
if( ! nesteddds.FindDataElement( gdcmm::Tag( 0x0400,0x0520) ) ) return 1;
const gdcmm::DataElement &EncryptedContent = nesteddds.GetDataElement( gdcmm::Tag( 0x0400,0x0520) );
const gdcmm::ByteValue *bv = EncryptedContent.GetByteValue();
std::ofstream of( outfilename, std::ios::binary );
of.write( bv->GetPointer(), bv->GetLength() );
of.close();
return 0;
}

```

## 12.62 ExtractIconFromFile.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.

=====*/
/*
 * This example shows how to either retrieve an Icon if present somewhere
 * in the file, or else generate one.
 */
#include "gdcmmImageReader.h"
#include "gdcmmPNMCodec.h"
#include "gdcmmIconImageFilter.h"
#include "gdcmmIconImageGenerator.h"
bool WriteIconAsPNM(const char* filename, const gdcmm::IconImage& icon)
{
    gdcmm::PNMCodec pnm;
    pnm.SetDimensions( icon.GetDimensions() );
    pnm.SetPixelFormat( icon.GetPixelFormat() );
    pnm.SetPhotometricInterpretation( icon.GetPhotometricInterpretation() );
    pnm.SetLUT( icon.GetLUT() );
    const gdcmm::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];
    gdcmm::ImageReader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read (or not image): " << filename << std::endl;
        return 1;
    }
    gdcmm::IconImageFilter iif;
    iif.SetFile( reader.GetFile() );
    bool b = iif.Extract();
    if( b )
    {
        const gdcmm::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() == gdcmm::TransferSyntax::JPEGBaselineProcess1
    || icon.GetTransferSyntax() == gdcmm::TransferSyntax::JPEGExtendedProcess2_4
    )
    {
        const gdcmm::DataElement& in = icon.GetDataElement();
        const gdcmm::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;
}

```

## 12.63 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.decod_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/MAROTTECH_CT_JP2Lossy.dcm
        //gdcmWarningMacro( "J2K start like JPEG-2000 compressed image data instead of codestream" );
        parameters.decod_format = 1; //JP2_CFMT;
        //assert(parameters.decod_format == JP2_CFMT);
    }
    else
    {
        /* JPEG-2000 codestream */
        //parameters.decod_format = J2K_CFMT;
        //assert(parameters.decod_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->tcps[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->tcps->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++)
    {
        opj_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;
            gdcm::Element<gdcm::VR::IS, gdcm::VM::VM1> el2;
            el2.SetValue(i+1);
            gdcm::DataElement rfn = el2.GetAsDataElement(); //ulr --> upper left row
            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;

```

```

    ell.SetValue(col,0);
    ell.SetValue(row,1);
    gdcm::DataElement brr = ell.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::Writer wl;
gdcm::File &file1 = wl.GetFile();
gdcm::DataSet &ds1 = file1.GetDataSet();
file1.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );
gdcm::UIDGenerator uid1;
gdcm::DataElement dea( gdcm::Tag(0x8,0x18) ); // SOP Instance UID
dea.SetVR( gdcm::VR::UI );
const char *ul = uid1.Generate();
dea.SetByteValue( ul, strlen(ul) );
ds1.Insert( dea );
gdcm::DataElement deb( gdcm::Tag(0x8,0x16) );
deb.SetVR( gdcm::VR::UI );
gdcm::MediaStorage msl( gdcm::MediaStorage::VLWholeSlideMicroscopyImageStorage );
deb.SetByteValue( msl.GetString(), strlen( msl.GetString() ) );
ds1.Insert( deb );
const char mystr1[] = "MONOCHROME2 ";
gdcm::DataElement dec( gdcm::Tag(0x28,0x04) );
//de.SetTag(gdcm::Tag(0x28,0x04));
dec.SetVR( gdcm::VR::CS );
dec.SetByteValue(mystr, strlen(mystr));
ds1.Insert( dec );
gdcm::Attribute<0x0028,0x0010> row1 = {image->y1};
//row.SetValue(512);
ds1.Insert( row1.GetAsDataElement() );
// w.SetCheckFileMetaInformation( true );
gdcm::Attribute<0x0028,0x0011> col1 = {image->x1};
ds1.Insert( col1.GetAsDataElement() );
gdcm::Attribute<0x0028,0x0008> Number_Of_Frames1 = {tccp->numresolutions};
ds1.Insert( Number_Of_Frames1.GetAsDataElement() );
gdcm::Attribute<0x0028,0x0100> ata = {8};
ds1.Insert( ata.GetAsDataElement() );
gdcm::Attribute<0x0028,0x0002> atb = {image->numcomps};
ds1.Insert( atb.GetAsDataElement() );
gdcm::Attribute<0x0028,0x0101> atc = {8};
ds1.Insert( atc.GetAsDataElement() );
gdcm::Attribute<0x0028,0x0102> atd = {7};
ds1.Insert( atd.GetAsDataElement() );
theStreamWriter.SetFile(file1);
gdcm::DataElement des( gdcm::Tag(0x0048,0x0200) );
des.SetVR(gdcm::VR::SQ);
//des.SetVR(gdcm::VM::VM1);
des.SetValue(*sq);
des.SetVLToUndefined();
ds1.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<<"\nbletoread";
// 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" << 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 << "\nable 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;
    gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new gdcm::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;
    gdcm::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 = 0xffff;
    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;
}

```

## 12.64 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://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 work was realised during the GSOC 2011 by Manoj Alwani
#include "gdcmlReader.h"
#include "gdcmlMediaStorage.h"
#include "gdcmlWriter.h"
#include "gdcmlItem.h"
#include "gdcmlImageReader.h"
#include "gdcmlAttribute.h"
#include "gdcmlFile.h"
#include "gdcmlTag.h"
#include "gdcmlTransferSyntax.h"
#include "gdcmlUIDGenerator.h"
#include "gdcmlAnonymizer.h"
#include "gdcmlStreamImageWriter.h"
#include "gdcmlImageHelper.h"
#include "gdcmlTrace.h"
int main(int, char *[])
{
    char * buffer = new char[ 256 * 256 * 3 ];
    // *p = (uint8_t*)buffer;
    char * p = buffer;
    gdcml::Trace::DebugOn();
    gdcml::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;
            }
        gdcml::Writer w;
        gdcml::File &file = w.GetFile();
        gdcml::DataSet &ds = file.GetDataSet();
        file.GetHeader().SetDataSetTransferSyntax( gdcml::TransferSyntax::ExplicitVRLittleEndian );
        gdcml::UIDGenerator uid;
        gdcml::DataElement de( gdcml::Tag(0x8,0x18) ); // SOP Instance UID
        de.SetVR( gdcml::VR::UI );
        const char *u = uid.Generate();
        de.SetByteValue( u, strlen(u) );
        ds.Insert( de );
        gdcml::DataElement del( gdcml::Tag(0x8,0x16) );
        del.SetVR( gdcml::VR::UI );
        gdcml::MediaStorage ms( gdcml::MediaStorage::VLWholeSlideMicroscopyImageStorage );
        del.SetByteValue( ms.GetString(), strlen(ms.GetString()) );
        ds.Insert( del );
        const char mystr[] = "RGB";
        gdcml::DataElement de2( gdcml::Tag(0x28,0x04) );
        //de.SetTag(gdcml::Tag(0x28,0x04));
        de2.SetVR( gdcml::VR::CS );
        de2.SetByteValue(mystr, strlen(mystr));
        ds.Insert( de2 );
        gdcml::Attribute<0x0028,0x0010> row = {256};
        //row.SetValue(512);
        ds.Insert( row.GetAsDataElement() );
        // w.SetCheckFileMetaInformation( true );

```

```

gdcmm::Attribute<0x0028,0x0011> col = {256};
ds.Insert( col.GetAsDataElement() );
gdcmm::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
ds.Insert( Number_Of_Frames.GetAsDataElement() );
gdcmm::Attribute<0x0028,0x0100> at = {8};
ds.Insert( at.GetAsDataElement() );
gdcmm::Attribute<0x0028,0x0002> at1 = {3}; //bits per pixel
ds.Insert( at1.GetAsDataElement() );
gdcmm::Attribute<0x0028,0x0101> at2 = {8};
ds.Insert( at2.GetAsDataElement() );
gdcmm::Attribute<0x0028,0x0102> at3 = {7};
ds.Insert( at3.GetAsDataElement() );
gdcmm::Attribute<0x0028,0x0006> at4 = {0};
ds.Insert( at4.GetAsDataElement() );
gdcmm::Attribute<0x0028,0x0103> at5 = {0};
ds.Insert( at5.GetAsDataElement() );
//de.SetTag(gdcmm::Tag(0x7fe0,0x0010));
//ds.Insert(de);
gdcmm::StreamImageWriter theStreamWriter;
gdcmm::SmartPointer<gdcmm::SequenceOfItems> sq = new gdcmm::SequenceOfItems();
sq->SetLengthToUndefined();
uint16_t row1 = 256;
uint16_t col1 = 256;
//std::cout << row;
gdcmm::Element<gdcmm::VR::IS,gdcmm::VM::VM1> el2;
el2.SetValue(1);
gdcmm::DataElement rfn = el2.GetAsDataElement(); //rfn ---> reference frame number
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> el1;
el1.SetValue(col1,0);
el1.SetValue(row1,1);
gdcmm::DataElement brr = el1.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::DataElement des( gdcmm::Tag(0x0048,0x0200) );
des.SetVR(gdcmm::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<<"\nbletoread";
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 theBufferSize = 2*sizeof(uint16_t)+sizeof(uint32_t);
char* tmpBuffer2 = new char[theBufferSize];
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, theBufferSize);
of.flush();
assert( of );
return 0;
}

```

## 12.65 FixBrokenJ2K.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 "gdcmSequenceOfFragments.h"
#include "gdcmFile.h"
// http://www.lost.in.ua/dicom/c.dcm
//
// -> BuggyJ2Kvvua-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( outfile );
    writer.CheckFileMetaInformationOff();
    if( !writer.Write() )
    {
        std::cerr << "Could not write" << std::endl;
    }
    // paranoid check:
    gdcm::ImageReader ireader;
    ireader.SetFileName( outfile );
    if( !ireader.Read() )
    {
        std::cerr << "file written is still not valid, please report" << std::endl;
        return 1;
    }
}

```

```

    }
    return 0;
}

```

## 12.66 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( "FixJAIBugJPEGLS" );
    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 gdc::Fragment &frag = sf->GetFragment(i);
    if( frag.IsEmpty() ) return 1;
    const gdc::ByteValue *bv = frag.GetByteValue();
    if( !bv ) return 1;
    unsigned long totalLen = bv->GetLength();
    std::vector<char> vbuffer;
    vbuffer.resize( totalLen );
    char *buffer = &vbuffer[0];
    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 << "Cant 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;
    gdc::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 << "Cant 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);
#if 0
    std::ofstream of( "/tmp/d.jls", std::ios::binary );
    of.write( &vbuffer[0], vbuffer.size() );
    of.close();
#endif
    const char *pbyteCompressed = &vbuffer[0];
    size_t cbyteCompressed = vbuffer.size(); // updated legnth
    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[0], 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() );
}
gdcmm::DataElement pixeldata( gdcmm::Tag(0x7fe0,0x0010) );
pixeldata.SetVR( gdcmm::VR::OW );
pixeldata.SetByteValue( (char*)&rgbyteOutall[0], (uint32_t)rgbyteOutall.size() );
// Add the pixel data element
reader.GetFile().GetDataSet().Replace( pixeldata );
reader.GetFile().GetHeader().SetDataSetTransferSyntax(
    gdcmm::TransferSyntax::ExplicitVRLittleEndian);
gdcmm::Writer writer;
writer.SetFileName( outfilename );
writer.SetFile( reader.GetFile() );
writer.Write();
std::cout << "Success !" << std::endl;
return 0;
}

```

## 12.67 FixOrientation.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 "gdcmmFile.h"
#include "gdcmmOrientation.h"
#include "gdcmmAttribute.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;
gdcm::File & f = reader.GetFile();
gdcm::DataSet & ds = f.GetDataSet();
at1.SetFromDataSet( ds );
#ifdef 0
at2.SetFromDataSet( ds );
const double * iop = at2.GetValues();
if( !std::equal(iop, iop + 6, axial ) )
{
    gdcm::Orientation::OrientationType type = gdcm::Orientation::GetType ( iop );
    std::cerr << "Wrong orientation: " << gdcm::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() );
gdcm::Writer writer;
writer.SetFile( f );
writer.SetFileName( outfile );
if ( !writer.Write() )
{
    return 1;
}
return 0;
}

```

## 12.68 GenAllVR.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 "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmDict.h"
#include "gdcmDictEntry.h"

```

```

#include "gdcmDicts.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDGenerator.h"
#include "gdcmFileExplicitFilter.h"
#include <cstdlib>
#include <cstring>
gdcm::Tag FindTagFromVR(gdcm::Dict const &dict, gdcm::VR const &vr)
{
    using gdcm::Dict;
    Dict::ConstIterator beg = dict.Begin();
    Dict::ConstIterator end = dict.End();
    Dict::ConstIterator it;
    for( it = beg; it != end; ++it)
    {
        const gdcm::Tag &t = it->first;
        const gdcm::DictEntry &de = it->second;
        const gdcm::VR &vr_de = de.GetVR();
        if( vr == vr_de && !de.GetRetired() && t.GetGroup() >= 0x8 )
        {
            return t;
        }
    }
    return gdcm::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 gdcm::Global &g = gdcm::Global::GetInstance();
    static const gdcm::Dicts &dicts = g.GetDicts();
    static const gdcm::Dict &pubdict = dicts.GetPublicDict();
    using gdcm::VR;
    using gdcm::Tag;
    gdcm::Writer w;
    gdcm::File &f = w.GetFile();
    gdcm::DataSet &ds = f.GetDataSet();
    gdcm::FileExplicitFilter fef;
    //fef.SetChangePrivateTags( true );
    fef.SetFile( w.GetFile() );
    if( !fef.Change() )
    {
        std::cerr << "Failed to change" << std::endl;
        return 1;
    }
    gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new gdcm::SequenceOfItems();
    sq->SetLengthToUndefined();
    // gdcm::DummyValueGenerator dv;
    const std::size_t len = 10;
    char ss[len+1];
    ss[len] = '\0';
    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 );
    // Create an item
    gdcm::Item it;
    it.SetVLToUndefined();
    gdcm::DataSet &nds = it.GetNestedDataSet();
    // nds.Insert(owner);
    // nds.Insert(de);
    // 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);
// 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;
}

```

## 12.69 GenFakeIdentifyFile.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 "gdcmTransferSyntax.h"
#include "gdcmUIDGenerator.h"
#include "gdcmAnonymizer.h"
#include <cstdlib>
#include <cstring>
gdcm::DataElement CreateFakeElement(gdcm::Tag const &tag, bool toremove)

```

```

{
    static const gdcm::Global &g = gdcm::Global::GetInstance();
    static const gdcm::Dicts &dicts = g.GetDicts();
    static const gdcm::Dict &pubdict = dicts.GetPublicDict();
    static size_t countglobal = 0;
    static std::vector<gdcm::Tag> balcptags =
        gdcm::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes();
    size_t count = countglobal % balcptags.size();
    const gdcm::DictEntry &dictentry = pubdict.GetDictEntry(tag);
    gdcm::DataElement de;
    de.SetTag( tag );
    using gdcm::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
        gdcm::Item it;
        it.SetVLToUndefined();
        gdcm::DataSet &nds = it.GetNestedDataSet();
        // Insert sequence into data set
        assert(de.GetVR() == gdcm::VR::SQ );
        gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new gdcm::SequenceOfItems();
        sq->SetLengthToUndefined();
        de.SetValue(*sq);
        de.SetVLToUndefined();
        //ds.Insert(de);
        if( !toremove )
        {
            nds.Insert( CreateFakeElement( balcptags[count], true ) );
            countglobal++;
        }
        else
        {
            gdcm::Attribute<0x0008,0x0000> at1 = { 0 }; // This element has no reason to be 'anonymized'...
            nds.Insert( at1.GetAsDataElement() );
            gdcm::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;
}

```

## 12.70 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( outfile );
    if ( !w.Write() )
    {
        return 1;
    }
    return 0;
}

```

## 12.71 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 were 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;
    }
    gdcm::File &file = reader.GetFile();
    gdcm::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
    gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new gdcm::SequenceOfItems();
    sq->SetLengthToUndefined();
    const char owner_str[] = "GDICOM CONFORMANCE TESTS";
    gdcm::DataElement owner( gdcm::Tag(0x4d4d, 0x10) );
    owner.SetByteValue(owner_str, (uint32_t)strlen(owner_str));
    owner.SetVR( gdcm::VR::LO );
    for(unsigned int idx = 0; idx < 10/* 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;
}

```

## 12.72 GenerateStandardSOPClasses.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 "gdcmDefs.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;
}

```



## 12.73 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;
}

```

## 12.74 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;
}

```

## 12.75 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;
    }
    #if 0
    {
        std::ofstream out( "/tmp/mo4", std::ios::binary );
        out.write( bv4->GetPointer(), bv4->GetLength());
        out.close();
    }
    #endif
    const ByteValue *bv5 = seq5.GetByteValue();
    #if 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[0], (uint32_t)imbuffer.size() );
gdcm::SmartPointer<gdcm::Image> im = new gdcm::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( gdcm::PhotometricInterpretation::MONOCHROME2 );
im->SetDataElement( fakedata );
gdcm::ImageWriter w;
w.SetImage( *im );
DataSet &dataset = w.GetFile().GetDataSet();
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 );
dataset.Replace( de );
de.SetTag( Tag(0x8,0x16) ); // SOP Class UID
de.SetVR( VR::UI );
gdcm::MediaStorage ms(
    gdcm::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;
}

```

## 12.76 HelloVizWorld.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.
=====*/
/*
 * Basic example for dealing with a DICOM file that contains an Image
 * (read: Pixel Data element)
 */
#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmImage.h"
#include "gdcmPhotometricInterpretation.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: " << outfilename << std::endl;
        return 1;
    }
    return 0;
}

```

## 12.77 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( outfilename );
    writer.SetFile( file );
    if( !writer.Write() )
    {
        std::cerr << "Could not write: " << outfilename << 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.
 */

```

## 12.78 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.FindElement( 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.SetVLToUndefined();
//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[0], 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[0], (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;
}

```

## 12.79 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;
}

```

## 12.80 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 and 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;
}

```

## 12.81 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.bB0CompensationValid = 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.ulEnableRFSpilling          = 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 "gdcReader.h"
#include "gdcImageReader.h"
#include "gdcImageWriter.h"
#include "gdcCSAHeader.h"
#include "gdcAttribute.h"
#include "gdcGlobal.h"
#include "gdcDicts.h"
#include <map>
#include <math.h>
int main(int argc, char *argv [])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdc::ImageReader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }
    gdc::CSAHeader csa;
    const gdc::DataSet& ds = reader.GetFile().GetDataSet();
    //const gdc::PrivateTag &t1 = csa.GetCSAImageHeaderInfoTag();
    const gdc::PrivateTag &t2 = csa.GetCSASeriesHeaderInfoTag();
    if( ds.FindDataElement( t2 ) )
    {
        csa.LoadFromDataElement( ds.GetDataElement( t2 ) );
        //csa.Print( std::cout );
    }
    if( !csa.FindCSAElementByName( "MrProtocol" ) )
    {
        return 1;
    }
    const gdc::CSAElement &csael = csa.GetCSAElementByName( "MrProtocol" );
    //std::cout << csael << std::endl;
    const gdc::ByteValue *bv = csael.GetByteValue();
    if( !bv )
    {
        return 1;
    }
    std::string str(bv->GetPointer(), bv->GetLength());
    std::istringstream 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:
#if 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;
}

```

## 12.82 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;
}

```

## 12.83 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 differents way 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 *stowner = nullptr;
    const gdcm::DictEntry &entry3 = dicts.GetDictEntry(patient_name,stowner);
    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;
}

```

## 12.84 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;
    gdcm::Writer w;
    gdcm::File & ff = w.GetFile();
    ff.GetHeader().SetDataSetTransferSyntax( gdcm::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;
}

```

## 12.85 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 "gdcmReader.h"
#include "gdcmMediaStorage.h"
typedef std::set<gdcm::DataElement> DataElementSet;
typedef DataElementSet::const_iterator ConstIterator;
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 << "Could not read: " << filename << std::endl;
        return 1;
    }
    std::stringstream strm;
    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();
    gdcm::FileMetaInformation &fmi = file.GetHeader();
    gdcm::MediaStorage ms;
    ms.SetFromFile(file);
    if( ms != gdcm::MediaStorage::MediaStorageDirectoryStorage )
    {
        std::cout << "This file is not a DICOMDIR" << std::endl;
        return 1;
    }
    if (fmi.FindDataElement( gdcm::Tag (0x0002, 0x0002)))
    {
        strm.str("");
    }
}

```

```

    fmi.GetDataElement( gdcm::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;
}

```

## 12.86 ReadAndDumpDICOMDIR2.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2017 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 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() {}
    DICOMDIRReader(const DICOMDIRReader &rhs) = delete;
    DICOMDIRReader(DICOMDIRReader &&rhs) = delete;
    DICOMDIRReader & operator = (const DICOMDIRReader &rhs) = delete;
    DICOMDIRReader & operator = (DICOMDIRReader &&rhs) = delete;
    virtual ~DICOMDIRReader() {}

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 lenght = VR field lenght,

```



```

// 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;
}

```

## 12.87 ReadAndPrintAttributes.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 small example will show how one can read and print
 * a DICOM Attribute using different technique (by tag or by name)
 */
#include "gdcmlReader.h"
#include "gdcmlGlobal.h"
#include "gdcmlDicts.h"
#include "gdcmlDict.h"
#include "gdcmlAttribute.h"
#include "gdcmlStringFilter.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 iieee double)
        Attribute<0x3004,0x000e>::ArrayType v = at.GetValue();
        std::cout << "DoseGridScaling=" << v << std::endl;
    }
    return 0;
}

```

## 12.88 ReadExplicitLengthSQIVR.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 "gdcmImplicitDataElement.h"
#include "gdcmDataSet.h"
#include "gdcmPrivateTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmByteValue.h"
#include "gdcmSequenceOfItems.h"
using namespace gdcm;
int main(int argc, char *argv[])
{
    if ( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcm::Reader r;
    r.SetFileName( filename );
    r.Read();
    //gdcm::PrivateTag pt(0x01,0x42,"ELSCINT1");
    //gdcm::Tag pt(0x88,0x200);
    gdcm::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;
}

```

## 12.89 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;
}

```

## 12.90 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)
    {
        std::cerr << "BAD ALLOC Exception caught!" << std::endl;
    }
    catch (...)
    {
        std::cerr << "Exception caught!" << std::endl;
    }
}
return 0;
}

```

## 12.91 ReadUTF8QtDir.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.
=====*/

/*
 * 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 "gdcmlReader.h"
#include "gdcmlDirectory.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;
    }
    gdcml::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.GetFiles();
    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:
    QCoreApplication 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;
}

```

## 12.92 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 gdcmlData/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 &ttv = s.GetMapping(filename);
    const gdcm::Tag *ptag = tag_array;
    for( ; ptag != tag_array + 3; ++ptag )
    {
        gdcm::StrictScanner::TagToValue::const_iterator it = ttv.find( *ptag );
        if( it != ttv.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;
}

```

## 12.93 SortImage.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 "gdcmsorter.h"
#include "gdcmscanner.h"
#include "gdcmdataset.h"
#include "gdcmdattribute.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 )
{
    gdcm::Attribute<0x0020,0x0032> at1;
    at1.Set( ds1 );
    gdcm::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(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
    gdcm::Attribute<0x0020,0x0052> at1; // FrameOfReferenceUID
    at1.Set( ds1 );
    gdcm::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];
    gdcm::Directory dir;
    unsigned int nfiles = dir.Load( dirname );
    dir.Print( std::cout );
    gdcm::Sorter sorter;
    sorter.SetSortFunction( mysort );
    sorter.Sort( dir.GetFilesNames() );
    std::cout << "Sorter:" << std::endl;
    sorter.Print( std::cout );
    gdcm::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 );
    gdcm::Scanner s;
    s.AddTag( gdcm::Tag(0x20,0x32) ); // Image Position (Patient)
    //s.AddTag( gdcm::Tag(0x20,0x37) ); // Image Orientation (Patient)
    s.Scan( dir.GetFilesNames() );
    //s.Print( std::cout );
    // Count how many different IPP there are:
    const gdcm::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;
}

```

## 12.94 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 "gdcmJPEGLSCCodec.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;
*/

```

```

gdcmm::Writer w;
gdcmm::File &file = w.GetFile();
gdcmm::DataSet &ds = file.GetDataSet();
file.GetHeader().SetDataSetTransferSyntax( gdcmm::TransferSyntax::ExplicitVRLittleEndian );
gdcmm::UIDGenerator uid;
gdcmm::DataElement de( gdcmm::Tag(0x8,0x18) ); // SOP Instance UID
de.SetVR( gdcmm::VR::UI );
const char *u = uid.Generate();
de.SetByteValue( u, strlen(u) );
ds.Insert( de );
gdcmm::DataElement de1( gdcmm::Tag(0x8,0x16) );
de1.SetVR( gdcmm::VR::UI );
gdcmm::MediaStorage ms( gdcmm::MediaStorage::VLWholeSlideMicroscopyImageStorage );
de1.SetByteValue( ms.GetString(), strlen(ms.GetString()) );
ds.Insert( de1 );
const char mystr[] = "MONOCHROME2 ";
gdcmm::DataElement de2( gdcmm::Tag(0x28,0x04) );
//de.SetTag(gdcmm::Tag(0x28,0x04));
de2.SetVR( gdcmm::VR::CS );
de2.SetByteValue( mystr, strlen(mystr) );
ds.Insert( de2 );
gdcmm::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
ds.Insert( Number_Of_Frames.GetAsDataElement() );
gdcmm::Attribute<0x0028,0x0010> row = {extent[0]/a}; //
ds.Insert( row.GetAsDataElement() );
gdcmm::Attribute<0x0028,0x0011> col = {extent[1]/a}; //
ds.Insert( col.GetAsDataElement() );
gdcmm::Attribute<0x0028,0x0100> at = {8};
ds.Insert( at.GetAsDataElement() );
gdcmm::Attribute<0x0028,0x0002> at1 = {1}; //
ds.Insert( at1.GetAsDataElement() );
gdcmm::Attribute<0x0028,0x0101> at2 = {8};
ds.Insert( at2.GetAsDataElement() );
gdcmm::Attribute<0x0028,0x0102> at3 = {7};
ds.Insert( at3.GetAsDataElement() );
/*
ds1.Remove( gdcmm::Tag(0x0028,0x0008) );

gdcmm::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 = gdcmm::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( outfile, 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, outfile, 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;
}

```

## 12.95 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::DComp( 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::TComp( 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;
}

```

## 12.96 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 "gdcmDefs.h"
#include "gdcmGlobal.h"
#include "gdcmIODs.h"
#include "gdcmIOD.h"
#include "gdcmMacros.h"
#include "gdcmIODEntry.h"
#include "gdcmModules.h"
#include "gdcmModule.h"
#include "gdcmAnonymizer.h"
#include "gdcmDicts.h"
int main(int , char *[])
{
    using namespace gdcm;
    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 = gdcm::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;
}

```

## 12.97 VolumeSorter.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 "gdcmIPPSorter.h"
#include "gdcmScanner.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"
#include "gdcmTesting.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];
    }
    gdcmm::Directory d;
    d.Load( dir1.c_str(), true ); // recursive !
    const gdcmm::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 );
    gdcmm::Scanner s0;
    const gdcmm::Tag t1(0x0020,0x000d); // Study Instance UID
    const gdcmm::Tag t2(0x0020,0x000e); // Series Instance UID
    //const gdcmm::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:
    gdcmm::Directory::FileNamesType l2 = s0.GetKeys();
    const size_t nfiles2 = l2.size();
    std::cout << nfiles2 << std::endl;
    if ( nfiles2 > nfiles )
    {
        return 1;
    }
    gdcmm::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;
}

gdcmm::Scanner s;
s.AddTag( gdcmm::Tag(0x20,0x32) ); // Image Position (Patient)
//s.AddTag( gdcmm::Tag(0x20,0x37) ); // Image Orientation (Patient)
s.Scan( d.GetFilesNames() );
//s.Print( std::cout );
const gdcmm::Scanner::ValuesType &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;
}

gdcmm::Directory::FileNamesType sorted_files = sorter.GetFilesNames();
// Which means we can take nvalues files at a time and execute gdcmm::IPPSorter on it:
gdcmm::IPPSorter ippsorter;
gdcmm::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;
}

```

## 12.98 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 &cse13 = csa.GetCSAElementByName( "PixelSpacing" );
    if( !cse13.IsEmpty() )
    {
        std::cout << cse13 << std::endl;
        gdcm::Element<gdcm::VR::DS, gdcm::VM::VM2> e13;
        e13.Set( cse13.GetValue() );
        spacing[0] = e13.GetValue(0);
        spacing[1] = e13.GetValue(1);
        std::cout << "PixelSpacing:" << e13.GetValue() << "," << e13.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] );
gdcm::PixelFormat pixeltype = gdcm::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() );
gdcm::PhotometricInterpretation pi;
pi = gdcm::PhotometricInterpretation::MONOCHROME2;
//pixeltype.SetSamplesPerPixel( );
image.SetPhotometricInterpretation( pi );
image.SetPixelFormat( pixeltype );
//image.SetIntercept( inputimage.GetIntercept() );
//image.SetSlope( inputimage.GetSlope() );
//gdcm::DataElement pixeldata( gdcm::Tag(0x7fe1,0x1010) );
//pixeldata.SetByteValue( &outbuf[0], outbuf.size() );
gdcm::PrivateTag csanonimaget(0x7fe1,0x10,"SIEMENS CSA NON-IMAGE");
const gdcm::DataElement &pixeldata = ds.GetDataElement( csanonimaget );
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;
}

```

## 12.99 iU22tomultisc.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.

=====*/
/*

```

```

* iU22 Raw Data extractor
*/
#include "gdcmReader.h"
#include "gdcmImageWriter.h"
#include "gdcmAttribute.h"
#include "gdcmPrivateTag.h"
#include <math.h>
int main(int argc, char *argv [])
{
    if( argc < 2 ) return 1;
    // IM_001
    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;
    }
    // * 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;
}

```

## 12.100 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[0];
    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 ther 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 swithing 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[0], (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;
}

```



## 12.101 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] == 0xa5 )
        {
            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 'out.rle.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 );
    gdcm::DataElement pixeldata;
    // if standard voxel data element does not exist, create it
    if( !reader.GetFile().GetDataSet().FindDataElement( tpixeldata ) )
    {
        pixeldata = gdcm::DataElement( tpixeldata, 0, gdcm::VR::OW );
    }
    else{
        pixeldata = reader.GetFile().GetDataSet().GetDataElement( tpixeldata );
    }
}

```

```

}
pixeldata.SetVR( gdcm::VR::OW );
gdcm::VL bv2l = bv2->GetLength();
gdcm::VL at1l = at1.GetValue() * at2.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[0], (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(
    gdcm::TransferSyntax::ExplicitVRLittleEndian);
gdcm::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( gdcm::Tag(0x07a1,0x1011) );
}
if( writer.GetFile().GetDataSet().FindDataElement( tprivatepixeldata ) )
{
    writer.GetFile().GetDataSet().Remove( gdcm::Tag(0x07a1,0x100a) );
}
std::string outfilename;
if (argc > 2)
    outfilename = argv[2];
else
    outfilename = "outrle.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;
}

```

## 12.102 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 );
}
}

```

## 12.103 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 );
    }
}
}

```

## 12.104 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 );
        }
    }
}

```

## 12.105 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");
            }
        }
    }
}

```

## 12.106 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" );
    }
}

```

## 12.107 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 );
}
}

```

## 12.108 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.GetFilesNames();
    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);
    }
}
}

```

## 12.109 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 r1 = lut.GetLUTLength( LookupTable.LookupTableType.RED );
            byte[] rbuf = new byte[ (int)r1 ];
            long r12 = lut.GetLUT( LookupTable.LookupTableType.RED, rbuf );
            assert r1 == r12;
            long g1 = lut.GetLUTLength( LookupTable.LookupTableType.GREEN );
            byte[] gbuf = new byte[ (int)g1 ];
            long g12 = lut.GetLUT( LookupTable.LookupTableType.GREEN, gbuf );
            assert g1 == g12;
            long b1 = lut.GetLUTLength( LookupTable.LookupTableType.BLUE );
            byte[] bbuf = new byte[ (int)b1 ];
            long b12 = lut.GetLUT( LookupTable.LookupTableType.BLUE, bbuf );
            assert b1 == b12;
            colorModel = new IndexColorModel(8, (int)r1, 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" );
}

```

## 12.110 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, "" );
    }
}

```

## 12.111 AddPrivateAttribute.py

```

1
14
15 """
16 Usage:

```

```

17
18 python AddPrivateAttribute.py input.dcm output.dcm
19
20
21 """
22
23 import sys
24 import gdcm
25
26 if __name__ == "__main__":
27
28     file1 = sys.argv[1]
29     file2 = sys.argv[2]
30
31     r = gdcm.Reader()
32     r.SetFileName( file1 )
33     if not r.Read():
34         sys.exit(1)
35
36     f = r.GetFile()
37     ds = f.GetDataSet()
38
39     # Create a dataelement
40     de = gdcm.DataElement(gdcm.Tag(0x0051, 0x1011))
41     de.SetByteStringValue("p2")
42     de.SetVR(gdcm.VR(gdcm.VR.SH))
43
44     ds.Insert(de)
45
46     w = gdcm.Writer()
47     w.SetFile( f )
48     w.SetFileName( file2 )
49     if not w.Write():
50         sys.exit(1)

```

## 12.112 ConvertMPL.py

```

1
14
15 """
16 display a DICOM image with matplotlib via numpy
17
18 Caveats:
19 - Does not support UINT12/INT12
20
21 Usage:
22
23 python ConvertNumpy.py "IM000000"
24
25 Thanks:
26 plotting example - Ray Schumacher 2009
27 """
28
29 import gdcm
30 import numpy
31 from pylab import *
32
33
34 def get_gdcm_to_numpy_typemap():
35     """Returns the GDCM Pixel Format to numpy array type mapping."""
36     _gdcm_np = {gdcm.PixelFormat.UINT8 :numpy.int8,
37                 gdcm.PixelFormat.INT8 :numpy.uint8,
38                 gdcm.PixelFormat.UINT16 :numpy.uint16,
39                 gdcm.PixelFormat.INT16 :numpy.int16,
40                 gdcm.PixelFormat.UINT32 :numpy.uint32,
41                 gdcm.PixelFormat.INT32 :numpy.int32,
42                 gdcm.PixelFormat.FLOAT32 :numpy.float32,
43                 gdcm.PixelFormat.FLOAT64 :numpy.float64 }
44     return _gdcm_np
45
46 def get_numpy_array_type(gdcm_pixel_format):
47     """Returns a numpy array typecode given a GDCM Pixel Format."""
48     return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]
49
50 def gdcm_to_numpy(image):
51     """Converts a GDCM image to a numpy array.
52     """

```

```

53     pf = image.GetPixelFormat().GetScalarType()
54     print 'pf', pf
55     print image.GetPixelFormat().GetScalarTypeAsString()
56     assert pf in get_gdcm_to_numpy_typemap().keys(), \
57         "Unsupported array type %s"%pf
58     d = image.GetDimension(0), image.GetDimension(1)
59     print 'Image Size: %d x %d' % (d[0], d[1])
60     dtype = get_numpy_array_type(pf)
61     gdcm_array = image.GetBuffer()
62
63     result = numpy.frombuffer(gdcm_array, dtype=dtype).astype(float)
64
65
66
67
68     result.shape = d
69     return result
70
71 if __name__ == "__main__":
72     import sys
73     r = gdcm.ImageReader()
74     filename = sys.argv[1]
75     r.SetFileName( filename )
76     if not r.Read(): sys.exit(1)
77     numpy_array = gdcm_to_numpy( r.GetImage() )
78
79     subplot(111)# one plot, on left
80     title(filename)
81
82     imshow(numpy_array, interpolation='bilinear', cmap=cm.jet)
83
84     subplots_adjust(bottom=0.1, right=0.8, top=0.9)
85     cax = axes([0.85, 0.1, 0.075, 0.8])
86     colorbar(cax=cax)
87     title('values')
88     get_current_fig_manager().window.title('plot')
89     show()

```

## 12.113 ConvertNumpy.py

```

1
14
15 """
16 This module add support for converting a gdcm.Image to a numpy array.
17
18 Caveats:
19 - Does not support UINT12/INT12
20
21 Removed:
22 - float16 is defined in GDCM API but no implementation exist for it ...
23 """
24
25 import gdcm
26 import numpy
27
28 def get_gdcm_to_numpy_typemap():
29     """Returns the GDCM Pixel Format to numpy array type mapping."""
30     _gdcm_np = {gdcm.PixelFormat.UINT8 :numpy.uint8,
31                 gdcm.PixelFormat.INT8 :numpy.int8,
32                 #gdcm.PixelFormat.UINT12 :numpy.uint12,
33                 #gdcm.PixelFormat.INT12 :numpy.int12,
34                 gdcm.PixelFormat.UINT16 :numpy.uint16,
35                 gdcm.PixelFormat.INT16 :numpy.int16,
36                 gdcm.PixelFormat.UINT32 :numpy.uint32,
37                 gdcm.PixelFormat.INT32 :numpy.int32,
38                 #gdcm.PixelFormat.FLOAT16:numpy.float16,
39                 gdcm.PixelFormat.FLOAT32:numpy.float32,
40                 gdcm.PixelFormat.FLOAT64:numpy.float64 }
41     return _gdcm_np
42
43 def get_numpy_array_type(gdcm_pixel_format):
44     """Returns a numpy array typecode given a GDCM Pixel Format."""
45     return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]
46
47 def gdcm_to_numpy(image):
48     """Converts a GDCM image to a numpy array.
49     """
50     pf = image.GetPixelFormat()
51
52     assert pf.GetScalarType() in get_gdcm_to_numpy_typemap().keys(), \

```

```

53         "Unsupported array type %s"%pf
54
55     shape = image.GetDimension(0) * image.GetDimension(1), pf.GetSamplesPerPixel()
56     if image.GetNumberOfDimensions() == 3:
57         shape = shape[0] * image.GetDimension(2), shape[1]
58
59     dtype = get_numpy_array_type(pf.GetScalarType())
60     gdcarray = image.GetBuffer()
61     result = numpy.frombuffer(gdcarray, dtype=dtype)
62     result.shape = shape
63     return result
64
65 if __name__ == "__main__":
66     import sys
67     r = gdc.ImageReader()
68     filename = sys.argv[1]
69     r.SetFileName( filename )
70     if not r.Read():
71         sys.exit(1)
72
73     numpy_array = gdc_to_numpy( r.GetImage() )
74     print numpy_array

```

## 12.114 ConvertPIL.py

```

1
14
15 """
16 save a DICOM image with PIL via numpy
17
18 Caveats:
19 - Does not support UINT12/INT12
20
21 Usage:
22
23 python ConvertNumpy.py "IM000000"
24
25 Thanks:
26 plotting example - Ray Schumacher 2009
27 """
28
29 import gdc
30 import numpy
31 from PIL import Image, ImageOps
32
33
34 def get_gdc_to_numpy_typemap():
35     """Returns the GDCM Pixel Format to numpy array type mapping."""
36     _gdc_np = {gdc.PixelFormat.UINT8 :numpy.int8,
37                gdc.PixelFormat.INT8 :numpy.uint8,
38                gdc.PixelFormat.UINT16 :numpy.uint16,
39                gdc.PixelFormat.INT16 :numpy.int16,
40                gdc.PixelFormat.UINT32 :numpy.uint32,
41                gdc.PixelFormat.INT32 :numpy.int32,
42                gdc.PixelFormat.FLOAT32:numpy.float32,
43                gdc.PixelFormat.FLOAT64:numpy.float64 }
44     return _gdc_np
45
46 def get_numpy_array_type(gdc_pixel_format):
47     """Returns a numpy array typecode given a GDCM Pixel Format."""
48     return get_gdc_to_numpy_typemap()[gdc_pixel_format]
49
50 def gdc_to_numpy(image):
51     """Converts a GDCM image to a numpy array.
52     """
53     pf = image.GetPixelFormat().GetScalarType()
54     print 'pf', pf
55     print image.GetPixelFormat().GetScalarTypeAsString()
56     assert pf in get_gdc_to_numpy_typemap().keys(), \
57         "Unsupported array type %s"%pf
58     d = image.GetDimension(0), image.GetDimension(1)
59     print 'Image Size: %d x %d' % (d[0], d[1])
60     dtype = get_numpy_array_type(pf)
61     gdcarray = image.GetBuffer()
62     result = numpy.frombuffer(gdcarray, dtype=dtype)
63     maxV = float(result[result.argmax()])
64

```



```

67     result = numpy.log(result+50)
68     maxV = float(result[result.argmax()])
69     result = result*(2.**8/maxV)
70     result.shape = d
71     return result
72
73 if __name__ == "__main__":
74     import sys
75     r = gdcm.ImageReader()
76     filename = sys.argv[1]
77     r.SetFileName( filename )
78     if not r.Read(): sys.exit(1)
79     numpy_array = gdcm_to_numpy( r.GetImage() )
80
81     pilImage = Image.frombuffer('L',
82                                numpy_array.shape,
83                                numpy_array.astype(numpy.uint8),
84                                'raw','L',0,1)
85
86     pilImage = ImageOps.autocontrast(pilImage, cutoff=.1)
87     pilImage.save(sys.argv[1]+' .jpg')

```

## 12.115 CreateRAWStorage.py

```

1
14
15 """
16     <uid value="1.2.840.10008.5.1.4.1.1.66" name="Raw Data Storage" type="SOP Class" part="PS 3.4"
17     retired="false"/>
18 """
19 import gdcm
20 import sys,os
21
22 if __name__ == "__main__":
23     r = gdcm.Reader()
24     # Will require Testing...
25     dataroot = gdcm.Testing.GetDataRoot()
26     filename = os.path.join( dataroot, '012345.002.050.dcm' )
27     r.SetFileName( filename )
28     r.Read()
29     f = r.GetFile()
30     ds = f.GetDataSet()
31
32     uid = "1.2.840.10008.5.1.4.1.1.66"
33     # f = gdcm.File()
34     # ds = f.GetDataSet()
35     de = gdcm.DataElement( gdcm.Tag(0x0008,0x0016) )
36     de.SetByteStringValue( uid )
37     vr = gdcm.VR( gdcm.VR.UI )
38     de.SetVR( vr )
39     ds.Replace( de )
40
41     ano = gdcm.Anonymizer()
42     ano.SetFile( r.GetFile() )
43     ano.RemovePrivateTags()
44     ano.RemoveGroupLength()
45     taglist = [
46         gdcm.Tag(0x0008,0x0008),
47         gdcm.Tag(0x0008,0x0022),
48         gdcm.Tag(0x0008,0x0032),
49         gdcm.Tag(0x0008,0x2111),
50         gdcm.Tag(0x0008,0x1150),
51         gdcm.Tag(0x0008,0x1155),
52         gdcm.Tag(0x0008,0x0100),
53         gdcm.Tag(0x0008,0x0102),
54         gdcm.Tag(0x0008,0x0104),
55         gdcm.Tag(0x0040,0xa170),
56         gdcm.Tag(0x0008,0x2112),
57         gdcm.Tag(0x0008,0x0100),
58         gdcm.Tag(0x0008,0x0102),
59         gdcm.Tag(0x0008,0x0104),
60         gdcm.Tag(0x0008,0x9215),
61         gdcm.Tag(0x0018,0x0010),
62         gdcm.Tag(0x0018,0x0022),
63         gdcm.Tag(0x0018,0x0050),
64         gdcm.Tag(0x0018,0x0060),

```

```

65     gdcM.Tag(0x0018,0x0088),
66     gdcM.Tag(0x0018,0x0090),
67     gdcM.Tag(0x0018,0x1040),
68     gdcM.Tag(0x0018,0x1100),
69     gdcM.Tag(0x0018,0x1110),
70     gdcM.Tag(0x0018,0x1111),
71     gdcM.Tag(0x0018,0x1120),
72     gdcM.Tag(0x0018,0x1130),
73     gdcM.Tag(0x0018,0x1150),
74     gdcM.Tag(0x0018,0x1151),
75     gdcM.Tag(0x0018,0x1152),
76     gdcM.Tag(0x0018,0x1160),
77     gdcM.Tag(0x0018,0x1190),
78     gdcM.Tag(0x0018,0x1210),
79     gdcM.Tag(0x0020,0x0012),
80     gdcM.Tag(0x0020,0x0032),
81     gdcM.Tag(0x0020,0x0037),
82     gdcM.Tag(0x0020,0x1041),
83     gdcM.Tag(0x0020,0x4000),
84     gdcM.Tag(0x0028,0x0002),
85     gdcM.Tag(0x0028,0x0004),
86     gdcM.Tag(0x0028,0x0010),
87     gdcM.Tag(0x0028,0x0011),
88     gdcM.Tag(0x0028,0x0030),
89     gdcM.Tag(0x0028,0x0100),
90     gdcM.Tag(0x0028,0x0101),
91     gdcM.Tag(0x0028,0x0102),
92     gdcM.Tag(0x0028,0x0103),
93     gdcM.Tag(0x0028,0x1052),
94     gdcM.Tag(0x0028,0x1053),
95     gdcM.Tag(0x0028,0x2110),
96     gdcM.Tag(0x0028,0x2112),
97     gdcM.Tag(0x7Fe0,0x0010),
98     gdcM.Tag(0x0018,0x0020),
99     gdcM.Tag(0x0018,0x0021),
100    gdcM.Tag(0x0018,0x0023),
101    gdcM.Tag(0x0018,0x0025),
102    gdcM.Tag(0x0018,0x0080),
103    gdcM.Tag(0x0018,0x0081),
104    gdcM.Tag(0x0018,0x0083),
105    gdcM.Tag(0x0018,0x0084),
106    gdcM.Tag(0x0018,0x0085),
107    gdcM.Tag(0x0018,0x0086),
108    gdcM.Tag(0x0018,0x0087),
109    gdcM.Tag(0x0018,0x0091),
110    gdcM.Tag(0x0018,0x0093),
111    gdcM.Tag(0x0018,0x0094),
112    gdcM.Tag(0x0018,0x0095),
113    gdcM.Tag(0x0018,0x1088),
114    gdcM.Tag(0x0018,0x1090),
115    gdcM.Tag(0x0018,0x1094),
116    gdcM.Tag(0x0018,0x1250),
117    gdcM.Tag(0x0018,0x1251),
118    gdcM.Tag(0x0018,0x1310),
119    gdcM.Tag(0x0018,0x1312),
120    gdcM.Tag(0x0018,0x1314),
121    gdcM.Tag(0x0018,0x1315),
122    gdcM.Tag(0x0018,0x1316),
123    gdcM.Tag(0x0020,0x0110),
124    gdcM.Tag(0x0028,0x0120),
125    gdcM.Tag(0x0028,0x1050),
126    gdcM.Tag(0x0028,0x1051)
127 ]
128 for tag in taglist:
129     #print tag
130     ano.Remove( tag )
131
132 # special handling
133 gen = gdcM.UIDGenerator()
134 ano.Replace( gdcM.Tag(0x0008,0x9123), gen.Generate() )
135 #ano.Empty( gdcM.Tag(0x0040,0x0555) )
136
137
138 #
139 # uid = gen.Generate()
140 # de.SetTag( gdcM.Tag(0x0008,0x0018) )
141 # de.SetByteStringValue( uid )
142 # ds.Insert( de )
143
144 # init FMI now:
145 #fmi = f.GetHeader()

```

```

146     #ts = gdcm.TransferSyntax()
147     #print ts
148     #fmi.SetDataSetTransferSyntax( ts ) # default
149     #print fmi.GetDataSetTransferSyntax()
150     #de.SetTag( gdcm.Tag(0x0002,0x0010) )
151     #uid = "1.2.840.10008.1.2"
152     #de.SetByteStringValue( uid )
153     #fmi.Insert( de )
154     # f.SetHeader( r.GetFile().GetHeader() )
155
156     writer = gdcm.Writer()
157     writer.SetFile( ano.GetFile() )
158     writer.SetFileName( "rawstorage.dcm" );
159     writer.Write()

```

## 12.116 DecompressImage.py

```

1
14
15 """
16 Usage:
17
18     python DecompressImage.py gdcmData/012345.002.050.dcm decompress.dcm
19 """
20
21 import gdcm
22 import sys
23
24 if __name__ == "__main__":
25
26     file1 = sys.argv[1]
27     file2 = sys.argv[2]
28
29     r = gdcm.ImageReader()
30     r.SetFileName( file1 )
31     if not r.Read():
32         sys.exit(1)
33
34     # check GetFragment API:
35     pd = r.GetFile().GetDataSet().GetDataElement(gdcm.Tag(0x7fe0, 0x0010))
36     frags = pd.GetSequenceOfFragments();
37     frags.GetFragment(0);
38
39     ir = r.GetImage()
40     w = gdcm.ImageWriter()
41     image = w.GetImage()
42
43     image.SetNumberOfDimensions( ir.GetNumberOfDimensions() );
44     dims = ir.GetDimensions();
45     print ir.GetDimension(0);
46     print ir.GetDimension(1);
47     print "Dims:",dims
48
49     # Just for fun:
50     dircos = ir.GetDirectionCosines()
51     t = gdcm.Orientation.GetType(tuple(dircos))
52     l = gdcm.Orientation.GetLabel(t)
53     print "Orientation label:",l
54
55     image.SetDimension(0, ir.GetDimension(0) );
56     image.SetDimension(1, ir.GetDimension(1) );
57
58     pixeltype = ir.GetPixelFormat();
59     image.SetPixelFormat( pixeltype );
60
61     pi = ir.GetPhotometricInterpretation();
62     image.SetPhotometricInterpretation( pi );
63
64     pixeldata = gdcm.DataElement( gdcm.Tag(0x7fe0,0x0010) )
65     str1 = ir.GetBuffer()
66     #print ir.GetBufferLength()
67     pixeldata.SetByteStringValue( str1 )
68     image.SetDataElement( pixeldata )
69
70     w.SetFileName( file2 )
71     w.SetFile( r.GetFile() )
72     w.SetImage( image )

```

```

73 if not w.Write():
74     sys.exit(1)

```

## 12.117 DumbAnonymizer.py

```

1
14
15 """
16 This example shows how one can use the gdcm.Anonymizer in 'dumb' mode.
17 This class becomes really handy when one knows which particular tag to fill in.
18
19 Usage:
20
21 python DumbAnonymizer.py gdcmData/012345.002.050.dcm out.dcm
22
23 """
24
25 import gdcm
26
27 # http://www.oid-info.com/get/1.3.6.1.4.17434
28 THERALYS_ORG_ROOT = "1.3.6.1.4.17434"
29
30 tag_rules={
31     # Value
32     (0x0012,0x0010):("Value","MySponsorName"),
33     (0x0012,0x0020):("Value","MyProtocolID"),
34     (0x0012,0x0021):("Value","MyProtocolName"),
35     (0x0012,0x0062):("Value","YES"),
36     (0x0012,0x0063):("Value","MyDeidentificationMethod"),
37
38     # Method
39     (0x0002,0x0003):("Method","GenerateMSOPIId"),
40     (0x0008,0x1155):("Method","GenerateMSOPIId"),
41     (0x0008,0x0018):("Method","GenerateMSOPIId"),
42     (0x0010,0x0010):("Method","GetSponsorInitials"),
43     (0x0010,0x0020):("Method","GetSponsorId"),
44     (0x0012,0x0030):("Method","GetSiteId"),
45     (0x0012,0x0031):("Method","GetSiteName"),
46     (0x0012,0x0040):("Method","GetSponsorId"),
47     (0x0012,0x0050):("Method","GetTPIId"),
48     (0x0018,0x0022):("Method","KeepIfExist"),
49     (0x0018,0x1315):("Method","KeepIfExist"),
50     (0x0020,0x000d):("Method","GenerateStudyId"),
51     (0x0020,0x000e):("Method","GenerateSeriesId"),
52     (0x0020,0x1002):("Method","GetNumberOfFrames"),
53     (0x0020,0x0020):("Method","GetPatientOrientation"),
54
55     # Other:
56     (0x0012,0x0051):("Patient Field","Type Examen"),
57     (0x0018,0x1250):("Sequence Field","Receive Coil"),
58     (0x0018,0x0088):("Sequence Field","Spacing Between Slice"),
59     (0x0018,0x0095):("Sequence Field","Pixel Bandwidth"),
60     (0x0018,0x0082):("Sequence Field","Inversion Time"),
61 }
62
63 class MyAnon:
64     def __init__(self):
65         self.studyuid = None
66         self.seriesuid = None
67         generator = gdcm.UIDGenerator()
68         if not self.studyuid:
69             self.studyuid = generator.Generate()
70         if not self.seriesuid:
71             self.seriesuid = generator.Generate()
72     def GetSponsorInitials(self):
73         return "dummy^foobar"
74     def GenerateStudyId(self):
75         return self.studyuid
76     def GenerateSeriesId(self):
77         return self.seriesuid
78     #def GenerateMSOPIId(self):
79     def GenerateMSOPIId(self):
80         generator = gdcm.UIDGenerator()
81         return generator.Generate()
82     def GetSiteId(self):
83         return "MySiteId"
84     def GetSiteName(self):
85         return "MySiteName"

```

```

85 def GetSponsorId(self):
86     return "MySponsorId"
87 def GetTPid(self):
88     return "MyTP"
89
90 if __name__ == "__main__":
91     import sys
92     gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "DumbAnonymizer" )
93     gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT )
94
95     r = gdcm.Reader()
96     filename = sys.argv[1]
97     r.SetFileName( filename )
98     if not r.Read(): sys.exit(1)
99
100     obj = MyAnon()
101
102     w = gdcm.Writer()
103     ano = gdcm.Anonymizer()
104     ano.SetFile( r.GetFile() )
105     ano.RemoveGroupLength()
106     for tag,rule in tag_rules.items():
107         if rule[0] == 'Value':
108             print tag,rule
109             ano.Replace( gdcm.Tag( tag[0], tag[1] ), rule[1] )
110         elif rule[0] == 'Method':
111             print tag,rule
112             # result = locals()[rule[1]]()
113             methodname = rule[1]
114             if hasattr(obj, methodname):
115                 _member = getattr(obj, methodname)
116                 result = _member()
117                 ano.Replace( gdcm.Tag( tag[0], tag[1] ), result )
118             else:
119                 print "Problem with: ", methodname
120
121     outfilename = sys.argv[2]
122     w.SetFileName( outfilename )
123     w.SetFile( ano.GetFile() )
124     if not w.Write(): sys.exit(1)

```

## 12.118 ExtractImageRegion.py

```

1
14
15 """
16
17 This small code shows how to use the gdcm.ImageRegionReader API
18 In this example we are taking each frame by frame and dump them to
19 /tmp/frame.raw.
20
21 Usage:
22 $ ExtractImageRegion.py input.dcm
23
24 Example:
25 $ ExtractImageRegion.py gdcmData/012345.002.050.dcm
26 $ md5sum /tmp/frame.raw
27 d594a5e2fde12f32b6633ca859b4d4a6 /tmp/frame.raw
28 $ gdcmInfo --md5sum gdcmData/012345.002.050.dcm
29 [...]
30 md5sum: d594a5e2fde12f32b6633ca859b4d4a6
31 """
32
33 import gdcm
34
35 if __name__ == "__main__":
36     import sys
37     filename = sys.argv[1]
38
39     file_size = gdcm.System.FileSize(filename);
40
41     # instantiate the reader:
42     reader = gdcm.ImageRegionReader();
43     reader.SetFileName( filename );
44
45     # pull DICOM info:
46     if not reader.ReadInformation():

```

```

47     sys.exit(1)
48
49     # store current offset:
50     cur_pos = reader.GetStreamCurrentPosition();
51
52     remaining = file_size - cur_pos;
53
54     print("Remaining bytes to read (Pixel Data): %d" % remaining );
55
56     # Get file infos
57     f = reader.GetFile();
58
59     # get some info about image
60     dims = gdcm.ImageHelper.GetDimensionsValue(f);
61     print(dims)
62     pf = gdcm.ImageHelper.GetPixelFormatValue (f);
63     pixelsize = pf.GetPixelSize();
64     pi = gdcm.ImageHelper.GetPhotometricInterpretationValue(f);
65     print( pi );
66
67     # buffer to get the pixels
68     buffer = bytearray( dims[0] * dims[1] * pixelsize )
69
70     # define a simple box region.
71     box = gdcm.BoxRegion();
72     for z in range(0, dims[2]):
73         # Define that I want the image 0, full size (dimx x dimy pixels)
74         # and do that for each z:
75         box.SetDomain(0, dims[0] - 1, 0, dims[1] - 1, z, z);
76         #print( box.toString() );
77         reader.SetRegion( box );
78
79         # reader will try to load the uncompressed image region into buffer.
80         # the call returns an error when buffer.Length is too small. For instance
81         # one can call:
82         # uint buf_len = reader.ComputeBufferLength(); // take into account pixel size
83         # to get the exact size of minimum buffer
84         if reader.ReadIntoBuffer(buffer):
85             open('/tmp/frame.raw', 'wb').write(buffer)
86         else:
87             #throw new Exception("can't read pixels error");
88             sys.exit(1)

```

## 12.119 FindAllPatientName.py

```

1
14 """
15 This example shows how one can use the gdcm.CompositeNetworkFunctions class
16 for executing a C-FIND query
17 It will print the list of patient name found
18
19 Usage:
20
21 python FindAllPatientName.py
22
23 """
24
25 import gdcm
26
27 # Patient Name
28 tag = gdcm.Tag(0x10,0x10)
29 de = gdcm.DataElement(tag)
30
31 # Search all patient name where string match 'F*'
32 de.SetByteStringValue('F*')
33
34 ds = gdcm.DataSet()
35 ds.Insert(de)
36
37 cnf = gdcm.CompositeNetworkFunctions()
38 theQuery = cnf.ConstructQuery (gdcm.ePatientRootType,gdcm.ePatient,ds)
39
40 #print theQuery.ValidateQuery()
41
42 # prepare the variable for output
43 ret = gdcm.DataSetArrayType()
44

```

```

45 # Execute the C-FIND query
46 cnf.CFind('dicom.example.com',11112,theQuery,ret,'GDCM_PYTHON','ANY-SCP')
47
48 for i in range(0,ret.size()):
49     print "Patient #",i
50     print ret[i]

```

## 12.120 FixCommaBug.py

```

1
14
15 """
16 Using LC_NUMERIC set to something not compatible with "C" it is possible to write out "," instead of
17 "." as required by the DICOM standard
18 Issue is still current (IMHO) with gdcm 2.0.9
19 """
20
21 import gdcm
22 import sys
23
24 filename = sys.argv[1]
25 outname = sys.argv[2]
26
27 # read
28 r = gdcm.Reader()
29 r.SetFileName( filename )
30 if not r.Read():
31     print "not valid"
32     sys.exit(1)
33
34 file = r.GetFile()
35 dataset = file.GetDataSet()
36
37 ano = gdcm.Anonymizer()
38 ano.SetFile( file )
39
40 tags = [
41     gdcm.Tag(0x0018,0x1164),
42     gdcm.Tag(0x0018,0x0088),
43     gdcm.Tag(0x0018,0x0050),
44     gdcm.Tag(0x0028,0x0030),
45 ]
46
47 for tag in tags:
48     print tag
49     if dataset.FindDataElement( tag ):
50         pixelspacing = dataset.GetDataElement( tag )
51         #print pixelspacing
52         bv = pixelspacing.GetByteValue()
53         str = bv.GetBuffer()
54         #print bv.GetLength()
55         #print len(str)
56         new_str = str.replace(",",".")
57         # Need to explicitly pass bv.GetLength() to remove any trailing garbage
58         ano.Replace( tag, new_str, bv.GetLength() )
59
60 #print dataset
61
62 w = gdcm.Writer()
63 w.SetFile( file )
64 w.SetFileName( outname )
65 if not w.Write():
66     print "Cannot write"
67     sys.exit(1)
68
69 # paranoid:
70 image_reader = gdcm.ImageReader()
71 image_reader.SetFileName( outname )
72 if not image_reader.Read():
73     print "there is still a comma"
74     sys.exit(1)
75
76 print "Success!"
77 sys.exit(0) # success

```

## 12.121 GetPortionCSAHeader.py

```

1
14
15 """
16 Usage:
17
18 python GetPortionCSAHeader.py input.dcm
19
20 Footnote:
21 SIEMENS is not publishing any information on the CSA header. So any info extracted
22 is at your own risk.
23 """
24
25 import sys
26 import gdcm
27
28 if __name__ == "__main__":
29
30     file = sys.argv[1]
31
32     r = gdcm.Reader()
33     r.SetFileName( file )
34     if not r.Read():
35         sys.exit(1)
36
37     ds = r.GetFile().GetDataSet()
38     csa_t1 = gdcm.CSAHeader()
39     csa_t2 = gdcm.CSAHeader()
40     #print csa
41     t1 = csa_t1.GetCSAImageHeaderInfoTag();
42     print t1
43     t2 = csa_t2.GetCSASeriesHeaderInfoTag();
44     print t2
45     # Let's do it for t1:
46     if ds.FindDataElement( t1 ):
47         csa_t1.LoadFromDataElement( ds.GetDataElement( t1 ) )
48         print csa_t1
49
50     # Now let's pretend we are only interested in B_value and DiffusionGradientDirection entries:
51     bvalues = csa_t1.GetCSAElementByName( "B_value" ) # WARNING: it is case sensitive !
52     print bvalues
53
54     diffgraddir = csa_t1.GetCSAElementByName( "DiffusionGradientDirection" ) # WARNING: it is case sensitive !
55     print diffgraddir
56
57     # repeat for t2 if you like it:
58     if ds.FindDataElement( t2 ):
59         csa_t2.LoadFromDataElement( ds.GetDataElement( t2 ) )
60         # print csa_t2
61
62     gdt = csa_t2.GetCSAElementByName( "GradientDelayTime" )
63     print gdt
64
65     bv = gdt.GetByteValue();
66     #print bv
67     str = bv.GetPointer()
68     print str.split("\\")

```

## 12.122 HelloWorld.py

```

1
14
15 """
16 Hello World !
17 """
18
19 import gdcm
20 import sys
21
22 if __name__ == "__main__":
23
24     # verbosity:
25     #gdcm.Trace.DebugOn()
26     #gdcm.Trace.WarningOn()
27     #gdcm.Trace.ErrorOn()
28

```



```

29 # Get the filename from the command line
30 filename = sys.argv[1]
31
32 # Instantiate a gdcm.Reader
33 # This is the main class to handle any type of DICOM object
34 # You should check for gdcm.ImageReader for reading specifically DICOM Image file
35 r = gdcm.Reader()
36 r.SetFileName( filename )
37 # If the reader fails to read the file, we should stop !
38 if not r.Read():
39     print "Not a valid DICOM file"
40     sys.exit(1)
41
42 # Get the DICOM File structure
43 file = r.GetFile()
44
45 # Get the DataSet part of the file
46 dataset = file.GetDataSet()
47
48 # Ok let's print it !
49 print dataset
50
51 # Use StringFilter to print a particular Tag:
52 sf = gdcm.StringFilter()
53 sf.SetFile(r.GetFile())
54
55 # Check if Attribute exist
56 print dataset.FindElement( gdcm.Tag(0x0028,0x0010))
57
58 # Let's print it as string pair:
59 print sf.ToStringPair(gdcm.Tag(0x0028,0x0010))

```

## 12.123 ManipulateFile.py

```

1
14
15 """
16 Usage:
17
18 python ManipulateFile.py input.dcm output.dcm
19
20 Footnote:
21 GDCM 1.2.x would create incorrect Multiframe MR Image Storage file. Try to recover from
22 the issues to recreate a MultiframeGrayscaleByteSecondaryCaptureImageStorage file.
23 e.g:
24
25 python ManipulateFile.py Insight/Testing/Temporary/itkGDCMImageIOTest5-j2k.dcm manipulated.dcm
26 """
27
28 import sys
29 import gdcm
30
31 if __name__ == "__main__":
32
33     file1 = sys.argv[1]
34     file2 = sys.argv[2]
35
36     r = gdcm.Reader()
37     r.SetFileName( file1 )
38     if not r.Read():
39         sys.exit(1)
40
41     ano = gdcm.Anonymizer()
42     ano.SetFile( r.GetFile() )
43     ano.RemovePrivateTags()
44     ano.Remove( gdcm.Tag(0x0032,0x1030) )
45     ano.Remove( gdcm.Tag(0x008,0x14) )
46     ano.Remove( gdcm.Tag(0x008,0x1111) )
47     ano.Remove( gdcm.Tag(0x008,0x1120) )
48     ano.Remove( gdcm.Tag(0x008,0x1140) )
49     ano.Remove( gdcm.Tag(0x10,0x21b0) )
50     ano.Empty( gdcm.Tag(0x10,0x10) )
51     ano.Empty( gdcm.Tag(0x10,0x20) )
52     ano.Empty( gdcm.Tag(0x10,0x30) )
53     ano.Empty( gdcm.Tag(0x20,0x10) )
54     ano.Empty( gdcm.Tag(0x32,0x1032) )
55     ano.Empty( gdcm.Tag(0x32,0x1033) )

```

```

56 ano.Empty( gdcm.Tag(0x40,0x241) )
57 ano.Empty( gdcm.Tag(0x40,0x254) )
58 ano.Empty( gdcm.Tag(0x40,0x253) )
59 ano.Empty( gdcm.Tag(0x40,0x1001) )
60 ano.Empty( gdcm.Tag(0x8,0x80) )
61 ano.Empty( gdcm.Tag(0x8,0x50) )
62 ano.Empty( gdcm.Tag(0x8,0x1030) )
63 ano.Empty( gdcm.Tag(0x8,0x103e) )
64 ano.Empty( gdcm.Tag(0x18,0x1030) )
65 ano.Empty( gdcm.Tag(0x38,0x300) )
66 g = gdcm.UIDGenerator()
67 ano.Replace( gdcm.Tag(0x0008,0x0018), g.Generate() )
68 ano.Replace( gdcm.Tag(0x0020,0x00d), g.Generate() )
69 ano.Replace( gdcm.Tag(0x0020,0x00e), g.Generate() )
70 ano.Replace( gdcm.Tag(0x0020,0x052), g.Generate() )
71 #ano.Replace( gdcm.Tag(0x0008,0x0016), "1.2.840.10008.5.1.4.1.1.7.2" )
72 """
73 ano.Remove( gdcm.Tag(0x0018,0x0020) ) # ScanningSequence
74 ano.Remove( gdcm.Tag(0x0018,0x0021) ) # SequenceVariant
75 ano.Remove( gdcm.Tag(0x0018,0x0022) ) # ScanOptions
76 ano.Remove( gdcm.Tag(0x0018,0x0023) ) # MRAcquisitionType
77 ano.Remove( gdcm.Tag(0x0018,0x0050) ) # SliceThickness
78 ano.Remove( gdcm.Tag(0x0018,0x0080) ) # RepetitionTime
79 ano.Remove( gdcm.Tag(0x0018,0x0081) ) # EchoTime
80 ano.Remove( gdcm.Tag(0x0018,0x0088) ) # SpacingBetweenSlices
81 ano.Remove( gdcm.Tag(0x0018,0x0091) ) # EchoTrainLength
82 ano.Remove( gdcm.Tag(0x0018,0x1164) ) # ImagerPixelSpacing
83
84 ano.Remove( gdcm.Tag(0x0020,0x0032) ) # Image Position (Patient)
85 ano.Remove( gdcm.Tag(0x0020,0x0037) ) # Image Orientation (Patient)
86 ano.Remove( gdcm.Tag(0x0020,0x0052) ) # Frame of Reference UID
87 ano.Remove( gdcm.Tag(0x0020,0x1040) ) # Position Reference Indicator
88
89 ano.Replace( gdcm.Tag(0x0028,0x0301), "NO" ) # Burned In Annotation
90
91 ano.Empty( gdcm.Tag(0x0020,0x0020) )
92
93 ano.Remove( gdcm.Tag(0x7fe0,0x0000) )
94
95 #ano.Empty( gdcm.Tag(0x0028,0x0009) ) # Frame Increment Pointer
96
97 #ano.Empty( gdcm.Tag(0x0028,0x1052) ) #<entry group="0028" element="1052" vr="DS" vm="1" name="Rescale
Intercept"/>
98 #ano.Empty( gdcm.Tag(0x0028,0x1053) ) #<entry group="0028" element="1053" vr="DS" vm="1" name="Rescale
Slope"/>
99 #ano.Replace( gdcm.Tag(0x0028,0x1054), "US" ) #<entry group="0028" element="1054" vr="LO" vm="1"
name="Rescale Type"/>
100
101 ano.Replace( gdcm.Tag(0x2050, 0x0020), "IDENTITY")
102 """
103
104 w = gdcm.Writer()
105 w.SetFile( ano.GetFile() )
106 w.SetFileName( file2 )
107 if not w.Write():
108     sys.exit(1)

```

## 12.124 ManipulateSequence.py

```

1
14
15 """
16 Usage:
17
18 python ManipulateSequence.py input.dcm output.dcm
19
20 This was tested using:
21
22 python ManipulateSequence.py gdcmData/D_CLUNIE_CT1_J2KI.dcm myoutput.dcm
23
24 This is a dummy example on how to modify a value set in a nested-nested dataset
25
26 WARNING:
27 Do not use as-is in production, this is just an example
28 This example works in an undefined length Item only (you need to explicitly recompute the length otherwise)
29 """
30

```

```

31 import sys
32 import gdcm
33
34 if __name__ == "__main__":
35
36     file1 = sys.argv[1]
37     file2 = sys.argv[2]
38
39     r = gdcm.Reader()
40     r.SetFileName( file1 )
41     if not r.Read():
42         sys.exit(1)
43
44     f = r.GetFile()
45     ds = f.GetDataSet()
46     tsis = gdcm.Tag(0x0008,0x2112) # SourceImageSequence
47     if ds.FindDataElement( tsis ):
48         sis = ds.GetDataElement( tsis )
49         #sqsis = sis.GetSequenceOfItems()
50         # GetValueAsSQ handle more cases
51         sqsis = sis.GetValueAsSQ()
52         if sqsis.GetNumberOfItems():
53             item1 = sqsis.GetItem(1)
54             nestedds = item1.GetNestedDataSet()
55             tprcs = gdcm.Tag(0x0040,0xa170) # PurposeOfReferenceCodeSequence
56             if nestedds.FindDataElement( tprcs ):
57                 prcs = nestedds.GetDataElement( tprcs )
58                 sqprcs = prcs.GetSequenceOfItems()
59                 if sqprcs.GetNumberOfItems():
60                     item2 = sqprcs.GetItem(1)
61                     nestedds2 = item2.GetNestedDataSet()
62                     # (0008,0104) LO [Uncompressed predecessor] # 24, 1 CodeMeaning
63                     tcm = gdcm.Tag(0x0008,0x0104)
64                     if nestedds2.FindDataElement( tcm ):
65                         cm = nestedds2.GetDataElement( tcm )
66                         mystr = "GDCM was here"
67                         cm.SetByteStringValue( mystr )
68
69     w = gdcm.Writer()
70     w.SetFile( f )
71     w.SetFileName( file2 )
72     if not w.Write():
73         sys.exit(1)

```

## 12.125 MergeFile.py

```

1
14
15 """
16 Usage:
17
18 python MergeFile.py input1.dcm input2.dcm
19
20 It will produce a 'merge.dcm' output file, which contains all meta information from input1.dcm
21 and copy the Stored Pixel values from input2.dcm
22 This script even works when input2.dcm is a Secondary Capture and does not contains information
23 such as IOP and IPP...
24 """
25
26 import sys
27 import gdcm
28
29 if __name__ == "__main__":
30
31     file1 = sys.argv[1]
32     file2 = sys.argv[2]
33
34     r1 = gdcm.ImageReader()
35     r1.SetFileName( file1 )
36     if not r1.Read():
37         sys.exit(1)
38
39     r2 = gdcm.ImageReader()
40     r2.SetFileName( file2 )
41     if not r2.Read():
42         sys.exit(1)
43

```

```

44 # Image from r2 could be Secondary Capture and thus would not contains neither IPP nor IOP
45 # Instead always prefer to only copy the Raw Data Element.
46 # Warning ! Image need to be identical ! Only the value of Stored Pixel can be different.
47 r1.GetImage().SetDataElement( r2.GetImage().GetDataElement() )
48
49 w = gdcm.ImageWriter()
50 w.SetFile( r1.GetFile() )
51 #w.SetImage( r2.GetImage() ) # See comment above
52 w.SetImage( r1.GetImage() )
53
54 w.SetFileName( "merge.dcm" )
55 if not w.Write():
56     sys.exit(1)
57
58 sys.exit(0)

```

## 12.126 NewSequence.py

```

1
14
15 """
16 Usage:
17
18     python NewSequence.py input.dcm output.dcm
19
20
21 Thanks to Robert Irie for code
22 """
23
24 import sys
25 import gdcm
26
27 if __name__ == "__main__":
28
29     file1 = sys.argv[1]
30     file2 = sys.argv[2]
31
32     r = gdcm.Reader()
33     r.SetFileName( file1 )
34     if not r.Read():
35         sys.exit(1)
36
37     f = r.GetFile()
38     ds = f.GetDataSet()
39     #tisis = gdcm.Tag(0x0008,0x2112) # SourceImageSequence
40
41     # Create a dataelement
42     de = gdcm.DataElement(gdcm.Tag(0x0010, 0x2180))
43     de.SetByteStringValue("Occupation")
44     de.SetVR(gdcm.VR(gdcm.VR.SH))
45
46     # Create an item
47     it=gdcm.Item()
48     it.SetVLToUndefined() # Needed to not popup error message
49     #it.InsertDataElement(de)
50     nds=it.GetNestedDataSet()
51     nds.Insert(de)
52
53     # Create a Sequence
54     sq=gdcm.SequenceOfItems().New()
55     sq.SetLengthToUndefined()
56     sq.AddItem(it)
57
58     # Insert sequence into data set
59     des=gdcm.DataElement(gdcm.Tag(0x0400,0x0550))
60     des.SetVR(gdcm.VR(gdcm.VR.SQ))
61     des.SetValue(sq.__ref__())
62     des.SetVLToUndefined()
63
64     ds.Insert(des)
65
66     w = gdcm.Writer()
67     w.SetFile( f )
68     w.SetFileName( file2 )
69     if not w.Write():
70         sys.exit(1)

```

## 12.127 PhilipsPrivateRescaleInterceptSlope.py

```

1
14
15 """
16 Usage:
17
18 python
19 """
20
21 import gdcmm
22 import sys
23
24 filename = sys.argv[1]
25 tmpfile = "/tmp/philips_rescaled.dcm"
26
27
28 # Need to access some private tags, read the file :
29 reader = gdcmm.Reader()
30 reader.SetFileName( filename )
31 if not reader.Read():
32     sys.exit(1)
33
34 ds = reader.GetFile().GetDataSet()
35
36 #print ds
37 # (2005,1409)      DS      4      0.0
38 # (2005,140a)      DS     16     1.52283272283272
39
40 # (2005,0014)      LO     26     Philips MR Imaging DD 005
41 tag1 = gdcmm.PrivateTag(0x2005,0x09,"Philips MR Imaging DD 005")
42 tag2 = gdcmm.PrivateTag(0x2005,0x0a,"Philips MR Imaging DD 005")
43 print tag1
44 print tag2
45
46 # make sure to do a copy, we want the private tag to remain
47 # otherwise gdcmm gives us a reference
48 e11 = gdcmm.DataElement( ds.GetDataElement( tag1 ) )
49 print e11
50 e12 = gdcmm.DataElement( ds.GetDataElement( tag2 ) )
51 print e12
52
53 # (0028,1052) DS [-1000]          # 6, 1 RescaleIntercept
54 # (0028,1053) DS [1]            # 2, 1 RescaleSlope
55
56 e11.SetTag( gdcmm.Tag(0x0028,0x1052) )
57 e12.SetTag( gdcmm.Tag(0x0028,0x1053) )
58
59 ds.Insert( e11 )
60 ds.Insert( e12 )
61
62 w = gdcmm.Writer()
63 w.SetCheckFileMetaInformation( False )
64 w.SetFileName( tmpfile )
65 w.SetFile( reader.GetFile() )
66 if not w.Write():
67     sys.exit(1)
68
69 print "success"

```

## 12.128 PlaySound.py

```

1
14
15 """
16 Usage:
17
18 python PlaySound.py input.dcm
19 """
20
21 import gdcmm
22 import sys
23
24 #filename = "/home/mmalaterre/Creatis/gdcmDataExtra/gdcmNonImageData/audio_from_rafael_sanguinetti.dcm"
25 filename = sys.argv[1]
26 print filename
27

```

```

28 r = gdcM.Reader()
29 r.SetFileName( filename )
30 if not r.Read():
31     sys.exit(1)
32
33 ds = r.GetFile().GetDataSet()
34
35 waveformtag = gdcM.Tag(0x5400,0x0100)
36 waveformsq = ds.GetDataElement( waveformtag )
37 #print waveformsq
38
39 #print dir(waveformsq)
40
41 items = waveformsq.GetSequenceOfItems()
42
43 if not items.GetNumberOfItems():
44     sys.exit(1)
45
46 item = items.GetItem(1)
47 #print item
48
49 waveformds = item.GetNestedDataSet()
50 #print waveformds
51
52 waveformdatatag = gdcM.Tag(0x5400,0x01010)
53 waveformdata = waveformds.GetDataElement( waveformdatatag )
54
55 #print waveformdata.GetPointer()
56 bv = waveformdata.GetByteValue()
57 print dir(bv)
58
59 #print bv.GetPointer()
60 print bv.GetLength()
61 l = 116838
62
63 file='test.wav'
64 myfile = open(file, "wb")
65 s = bv.GetPointer()
66 for i in range(0, l):
67     myfile.write(s[i])
68 myfile.close()
69
70 # http://mail.python.org/pipermail/python-list/2004-October/288905.html
71 if sys.platform.startswith('win'):
72     from winsound import PlaySound, SND_FILENAME, SND_ASYNC
73     PlaySound(file, SND_FILENAME|SND_ASYNC)
74 elif sys.platform.find('linux')>-1:
75     from wave import open as waveOpen
76     from ossaudiodev import open as ossOpen
77     s = waveOpen(file,'rb')
78     (nc,sw,fr,nf,comptype, compname) = s.getparams( )
79     dsp = ossOpen('/dev/dsp','w')
80     try:
81         from ossaudiodev import AFMT_S16_NE
82     except ImportError:
83         if byteorder == "little":
84             AFMT_S16_NE = ossaudiodev.AFMT_S16_LE
85         else:
86             AFMT_S16_NE = ossaudiodev.AFMT_S16_BE
87     dsp.setparameters(AFMT_S16_NE, nc, fr)
88     data = s.readframes(nf)
89     s.close()
90     dsp.write(data)
91     dsp.close()

```

## 12.129 PrivateDict.py

```

1
14
15 """
16 """
17
18 import gdcM
19 import sys,os
20
21 if __name__ == "__main__":
22     #gdcM.Trace.DebugOn()

```

```

23 globInst = gdcm.Global.GetInstance()
24 # Try to load Part3.xml file
25 # This file is too big for being accessible directly at runtime.
26 globInst.LoadResourcesFiles()
27
28
29 # Get a private tag from the runtime dicts. LoadResourcesFiles could
30 # have failed but this has no impact on the private dict
31
32 d = globInst.GetDicts()
33 print d.GetDictEntry( gdcm.Tag(0x0029,0x0010) ,"SIEMENS CSA HEADER" )
34 pd = d.GetPrivateDict()
35 print pd.GetDictEntry( gdcm.PrivateTag(0x0029,0x0010,"SIEMENS CSA HEADER") )

```

## 12.130 ReWriteSCAsMR.py

```

1
14
15 """
16 GDCM 1.x would write out MR Image Storage as Secondary Capture Object while still setting Rescale
17 Slope/Intercept
18 and saving the Pixel Spacing in (0028,0030)
19 """
20 import gdcm
21 import sys,os
22
23 def CheckSecondaryCaptureObjectIsMRImageStorage(r):
24     ds = r.GetFile().GetDataSet()
25     # Check Source Image Sequence
26     if ds.FindDataElement( gdcm.Tag(0x0008,0x2112) ):
27         sis = ds.GetDataElement( gdcm.Tag(0x0008,0x2112) )
28         sqsis = sis.GetSequenceOfItems()
29         if sqsis.GetNumberOfItems():
30             item1 = sqsis.GetItem(1)
31             nestedds = item1.GetNestedDataSet()
32             if nestedds.FindDataElement( gdcm.Tag(0x0008,0x1150) ):
33                 ReferencedSOPClassUID = nestedds.GetDataElement( gdcm.Tag(0x0008,0x1150) )
34                 raw = ReferencedSOPClassUID.GetByteValue().GetPointer()
35                 uids = gdcm.UIDs()
36                 # what is the actual object we are looking at ?
37                 ms = gdcm.MediaStorage()
38                 ms.SetFromDataSet( ds )
39                 msuid = ms.GetString()
40                 uids.SetFromUID( msuid )
41                 msuidname = uids.GetName() # real Media Storage Name
42                 uids.SetFromUID( raw )
43                 sqmsuidname = uids.GetName() # Source Image Sequence Media Storage Name
44                 # If object is SC and Source derivation is MRImageStorage then we can assume 'Pixel Spacing' is
45                 correct
46                 if( sqmsuidname == 'MR Image Storage' and msuidname == 'Secondary Capture Image Storage' ):
47                     return True
48             # in all other case simply return the currentspacing:
49             return False
50 if __name__ == "__main__":
51     r = gdcm.ImageReader()
52     filename = sys.argv[1]
53     r.SetFileName( filename )
54     if not r.Read():
55         sys.exit(1)
56     f = r.GetFile()
57
58     if( CheckSecondaryCaptureObjectIsMRImageStorage(r) ):
59         # Special handling of the spacing:
60         # GDCM 1.2.0 would not rewrite correctly DICOM Object and would always set them as 'Secondary Capture
61         # Image Storage'
62         # while we would rather have 'MR Image Storage'
63         gdcm.ImageHelper.SetForcePixelSpacing( True )
64         mrspacing = gdcm.ImageHelper.GetSpacingValue( r.GetFile() )
65         # TODO: I cannot do simply the following:
66         #image.SetSpacing( mrspacing )
67         image.SetSpacing(0, mrspacing[0] )
68         image.SetSpacing(1, mrspacing[1] )
69         image.SetSpacing(2, mrspacing[2] )
70         gdcm.ImageHelper.SetForceRescaleInterceptSlope( True )
71         ris = gdcm.ImageHelper.GetRescaleInterceptSlopeValue( r.GetFile() )

```

```

71     image.SetIntercept( ris[0] )
72     image.SetSlope( ris[1] )
73
74     outfilename = sys.argv[2]
75     w = gdcm.ImageWriter()
76     w.SetFileName( outfilename )
77     w.SetFile( r.GetFile() )
78     w.SetImage( image )
79     if not w.Write():
80         sys.exit(1)
81
82     sys.exit(0)

```

## 12.131 ReadAndDumpDICOMDIR.py

```

1
23
24
25
26 import sys
27 import gdcm
28
29 if __name__ == "__main__":
30     # Check arguments
31     if (len(sys.argv) < 2):
32         # No filename passed
33         print "No input filename found"
34         quit()
35
36     filename = sys.argv[1]
37
38
39     # Read file
40     reader = gdcm.Reader()
41     reader.SetFileName(filename)
42     if (not reader.Read()):
43         print "Unable to read %s" % (filename)
44         quit()
45
46     file = reader.GetFile()
47
48     # Retrieve header information
49     fileMetaInformation = file.GetHeader()
50     print fileMetaInformation
51
52     # Retrieve data set
53     dataSet = file.GetDataSet()
54     #print dataSet
55
56     # Check media storage
57     mediaStorage = gdcm.MediaStorage()
58     mediaStorage.SetFromFile(file)
59     if (gdcm.MediaStorage.GetMSType(str(mediaStorage)) != gdcm.MediaStorage.MediaStorageDirectoryStorage):
60         # File is not a DICOMDIR
61         print "This file is not a DICOMDIR (Media storage type: %s)" % (str(mediaStorage))
62         quit()
63
64     # Check Media Storage SOP Class
65     if (fileMetaInformation.FindDataElement(gdcm.Tag(0x0002, 0x0002))):
66         sopClassUid = str(fileMetaInformation.GetDataElement(gdcm.Tag(0x0002, 0x0002)).GetValue())
67         # Check SOP UID
68         if (sopClassUid != "1.2.840.10008.1.3.10"):
69             # File is not a DICOMDIR
70             print "This file is not a DICOMDIR"
71     else:
72         # Not present
73         print "Media Storage SOP Class not present"
74         quit()
75
76     # Iterate through the DICOMDIR data set
77     iterator = dataSet.GetDES().begin()
78     while (not iterator.equal(dataSet.GetDES().end())):
79         dataElement = iterator.next()
80
81         # Check the element tag
82         if (dataElement.GetTag() == gdcm.Tag(0x004, 0x1220)):
83             # The 'Directory Record Sequence' element

```



```

84         sequence = dataElement.GetValueAsSQ()
85
86         # Loop through the sequence items
87         itemNr = 1
88         while (itemNr < sequence.GetNumberOfItems()):
89             item = sequence.GetItem(itemNr)
90
91             # Check the element tag
92             if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
93                 # The 'Directory Record Type' element
94                 value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())
95
96                 # PATIENT
97                 while (value.strip() == "PATIENT"):
98                     print value.strip()
99                     # Print patient name
100                    if (item.FindDataElement(gdcm.Tag(0x0010, 0x0010))):
101                        value = str(item.GetDataElement(gdcm.Tag(0x0010, 0x0010)).GetValue())
102                        print value
103
104                    # Print patient ID
105                    if (item.FindDataElement(gdcm.Tag(0x0010, 0x0020))):
106                        value = str(item.GetDataElement(gdcm.Tag(0x0010, 0x0020)).GetValue())
107                        print value
108
109                    # Next
110                    itemNr = itemNr + 1
111                    item = sequence.GetItem(itemNr)
112                    if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
113                        value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())
114
115                    # STUDY
116                    while (value.strip() == "STUDY"):
117                        print value.strip()
118
119                        # Print study UID
120                        if (item.FindDataElement(gdcm.Tag(0x0020, 0x000d))):
121                            value = str(item.GetDataElement(gdcm.Tag(0x0020, 0x000d)).GetValue())
122                            print value
123
124                        # Print study date
125                        if (item.FindDataElement(gdcm.Tag(0x0008, 0x0020))):
126                            value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x0020)).GetValue())
127                            print value
128
129                        # Print study description
130                        if (item.FindDataElement(gdcm.Tag(0x0008, 0x1030))):
131                            value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x1030)).GetValue())
132                            print value
133
134                        # Next
135                        itemNr = itemNr + 1
136                        item = sequence.GetItem(itemNr)
137                        if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
138                            value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())
139
140                        # SERIES
141                        while (value.strip() == "SERIES"):
142                            print value.strip()
143
144                            # Print series UID
145                            if (item.FindDataElement(gdcm.Tag(0x0020, 0x000e))):
146                                value = str(item.GetDataElement(gdcm.Tag(0x0020, 0x000e)).GetValue())
147                                print value
148
149                            # Print series modality
150                            if (item.FindDataElement(gdcm.Tag(0x0008, 0x0060))):
151                                value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x0060)).GetValue())
152                                print "Modality"
153                                print value
154
155                            # Print series description
156                            if (item.FindDataElement(gdcm.Tag(0x0008, 0x103e))):
157                                value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x103e)).GetValue())
158                                print "Description"
159                                print value
160
161                            # Next
162                            itemNr = itemNr + 1
163                            item = sequence.GetItem(itemNr)
164                            if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):

```

```

165         value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())
166
167         # IMAGE
168         while (value.strip() == "IMAGE"):
169             print value.strip()
170
171         # Print image UID
172         if (item.FindDataElement(gdcm.Tag(0x0004, 0x1511))):
173             value = str(item.GetDataElement(gdcm.Tag(0x0004,
174                 0x1511)).GetValue())
175             print value
176
177         # Next
178         if (itemNr < sequence.GetNumberOfItems()):
179             itemNr = itemNr + 1
180         else:
181             break
182
183         item = sequence.GetItem(itemNr)
184         if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
185             value = str(item.GetDataElement(gdcm.Tag(0x0004,
186                 0x1430)).GetValue())
187             # Next
188             itemNr = itemNr + 1

```

## 12.132 RemovePrivateTags.py

```

1
14
15 """
16 Usage:
17
18 python RemovePrivateTags.py input.dcm output.dcm
19 """
20
21 import sys
22 import gdcm
23
24
25 if __name__ == "__main__":
26
27     file1 = sys.argv[1]
28     file2 = sys.argv[2]
29
30     # Instanciate the reader.
31     r = gdcm.Reader()
32     r.SetFileName( file1 )
33     if not r.Read():
34         sys.exit(1)
35
36     # Remove private tags
37     ano = gdcm.Anonymizer()
38     ano.SetFile( r.GetFile() )
39     if not ano.RemovePrivateTags():
40         sys.exit(1)
41
42     # Write DICOM file
43     w = gdcm.Writer()
44     w.SetFile( ano.GetFile() )
45     #w.CheckFileMetaInformationOff() # Do not attempt to check meta header
46     w.SetFileName( file2 )
47     if not w.Write():
48         sys.exit(1)
49
50     # It is usually a good idea to exit the script with an error, as gdcm does not remove partial (incorrect)
51     # DICOM file
52     # (application level)

```

## 12.133 ScanDirectory.py

```

1
14

```

```

15 import gdcmm
16 import sys,os
17
18 class ProgressWatcher(gdcm.SimpleSubjectWatcher):
19     def ShowProgress(self, sender, event):
20         pe = gdcm.ProgressEvent.Cast(event)
21         print pe.GetProgress()
22     def EndFilter(self):
23         print "Yay ! I am done"
24
25 if __name__ == "__main__":
26     directory = sys.argv[1]
27
28     # Define the set of tags we are interested in
29     t1 = gdcm.Tag(0x8,0x8);
30     t2 = gdcm.Tag(0x10,0x10);
31
32     # Iterate over directory
33     d = gdcm.Directory();
34     nfiles = d.Load( directory );
35     if(nfiles == 0): sys.exit(1);
36     # System.Console.WriteLine( "Files:\n" + d.toString() );
37
38     filenames = d.GetFilenames()
39
40     # Get rid of any Warning while parsing the DICOM files
41     gdcm.Trace.WarningOff()
42
43     # instantiate Scanner:
44     sp = gdcm.Scanner.New();
45     s = sp.__ref__()
46     w = ProgressWatcher(s, 'Watcher')
47
48     s.AddTag( t1 );
49     s.AddTag( t2 );
50     b = s.Scan( filenames );
51     if(not b): sys.exit(1);
52
53     print "success" ;
54     #print s
55
56     pttv = gdcm.PythonTagToValue( s.GetMapping( filenames[1] ) )
57     pttv.Start()
58     # iterate until the end:
59     while( not pttv.IsAtEnd() ):
60         # get current value for tag and associated value:
61         # if tag was not found, then it was simply not added to the internal std::map
62         # Warning value can be None
63         tag = pttv.GetCurrentTag()
64         value = pttv.GetCurrentValue()
65         print tag,"->",value
66         # increment iterator
67         pttv.Next()
68
69     sys.exit(0)

```

## 12.134 SortImage.py

```

1
14
15 """
16 Usage:
17
18 python SortImage.py dirname
19 """
20
21 import gdcm
22 import sys
23
24 def PrintProgress(object, event):
25     assert event == "ProgressEvent"
26     print "Progress:", object.GetProgress()
27
28 def MySort(ds1, ds2):
29     # compare ds1
30     return False
31

```

```

32 if __name__ == "__main__":
33
34     dirname = sys.argv[1]
35     d = gdcm.Directory()
36     d.Load( dirname )
37
38     print d
39
40     sorter = gdcm.Sorter()
41     sorter.SetSortFunction( MySort )
42     #sorter.AddObserver( "ProgressEvent", PrintProgress )
43     sorter.Sort( d.GetFilenames() )
44
45     print "Sorter:"
46     print sorter

```

## 12.135 WriteBuffer.py

```

1
14
15 """
16 Usage:
17
18 http://chuckhahm.com/Ischem/Zurich/XX_0134
19
20 (2005,1132) SQ (Sequence with undefined length #=8)          # u/l, 1 Unknown Tag & Data
21   (fffe,e000) na (Item with undefined length #=9)          # u/l, 1 Item
22     (2005,0011) LO [Philips MR Imaging DD 002]              # 26, 1 PrivateCreator
23     (2005,1137) PN [PDF_CONTROL_GEN_PARS]                   # 20, 1 Unknown Tag & Data
24     (2005,1138) PN (no value available)                     # 0, 0 Unknown Tag & Data
25     (2005,1139) PN [IEEE_PDF]                               # 8, 1 Unknown Tag & Data
26     (2005,1140) PN (no value available)                     # 0, 0 Unknown Tag & Data
27     (2005,1141) PN (no value available)                     # 0, 0 Unknown Tag & Data
28     (2005,1143) SL 3103                                       # 4, 1 Unknown Tag & Data
29     (2005,1144) OW 0566\0000\013b\0000\0a4a\0000\000e\0000\0a7a\0000\0195\0000\0008... # 3104, 1 Unknown Tag
    & Data
30     (2005,1147) CS [Y]                                         # 2, 1 Unknown Tag & Data
31   (fffe,e00d) na (ItemDelimitationItem)                    # 0, 0 ItemDelimitationItem
32   (fffe,e000) na (Item with undefined length #=9)          # u/l, 1 Item
33     (2005,0011) LO [Philips MR Imaging DD 002]              # 26, 1 PrivateCreator
34     (2005,1137) PN [PDF_CONTROL_PREP_PARS]                   # 22, 1 Unknown Tag & Data
35     (2005,1138) PN (no value available)                     # 0, 0 Unknown Tag & Data
36     (2005,1139) PN [IEEE_PDF]                               # 8, 1 Unknown Tag & Data
37     (2005,1140) PN (no value available)                     # 0, 0 Unknown Tag & Data
38     (2005,1141) PN (no value available)                     # 0, 0 Unknown Tag & Data
39     (2005,1143) SL 7934                                       # 4, 1 Unknown Tag & Data
40     (2005,1144) OW 19b6\0000\005f\0000\1b2a\0000\00f3\0000\1eee\0000\0000\0000\0008... # 7934, 1 Unknown Tag
    & Data
41     (2005,1147) CS [Y]                                         # 2, 1 Unknown Tag & Data
42   (fffe,e00d) na (ItemDelimitationItem)                    # 0, 0 ItemDelimitationItem
43 ...
44 """
45
46 import sys
47 import gdcm
48
49 if __name__ == "__main__":
50
51     file1 = sys.argv[1]
52     file2 = sys.argv[2]
53
54     r = gdcm.Reader()
55     r.SetFileName( file1 )
56     if not r.Read():
57         sys.exit(1)
58
59     fg = gdcm.FileNameGenerator()
60     f = r.GetFile()
61     ds = f.GetDataSet()
62     tsis = gdcm.Tag(0x2005,0x1132) #
63     if ds.FindDataElement( tsis ):
64         sis = ds.GetDataElement( tsis )
65         #sqsis = sis.GetSequenceOfItems()
66         # GetValueAsSQ handle more cases
67         sqsis = sis.GetValueAsSQ()
68         if sqsis.GetNumberOfItems():
69             nitems = sqsis.GetNumberOfItems();

```

```

70     fg.SetNumberOfFileNames( nitems )
71     fg.SetPrefix( file2 )
72     if not fg.Generate():
73         print "problem"
74     sys.exit(1)
75     for i in range(0,nitems):
76         item1 = sqsis.GetItem(i+1) # Item start at 1
77         nestedds = item1.GetNestedDataSet()
78         tprcs = gdcm.Tag(0x2005,0x1144) #
79         if nestedds.FindDataElement( tprcs ):
80             prcs = nestedds.GetDataElement( tprcs )
81             bv = prcs.GetByteValue()
82             print bv
83             f = open( fg.GetFilename(i) , "w" )
84             f.write( bv.WriteBuffer() )

```

## 12.136 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 vtkgdcm.vtkImageData ConnectActivizToSWIG(Kitware.VTK.vtkImageData imgin)

```

```

    {
        HandleRef rawCppThis = imgin.GetCppThis();
        vtkgdcm.vtkImageData imgout = new vtkgdcm.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
        vtkgdcm.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;
    }
}

```

## 12.137 HelloActiviz2.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;
/*
 * Usage:
 * $ export MONO_PATH=/usr/lib/cli/Activiz.NET:/usr/lib/cli/Kitware.mummy.Runtime-1.0
 * $ mono ./bin/HelloActiviz2.exe gdcmData/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;
}
}
```

## 12.138 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;
    }
}

```

## 12.139 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.Write(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;
    }
}

```

## 12.140 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.Write( "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.Write( "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;
    }
}

```

## 12.141 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() == vtkgdc.vtkgdc.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;
    }
}

```

## 12.142 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 vtkgdc;
/*
 * 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;
    }
}

```

## 12.143 MetaImageMD5Activiz.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;
}
}

```

## 12.144 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 cstor / dstor 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;
    }
}

```

## 12.145 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 "gdcmIPPSorter.h"
#ifdef vtkFloatingPointType
#define vtkFloatingPointType double
#else
#define vtkFloatingPointType float
#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 ipzspacing = 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], ippszspacing );
v16->Update();
v16->GetOutput()->Print( std::cout );
return 0;
}

```

## 12.146 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 "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;
}

```

## 12.147 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 "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 );
#ifdef 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;
}

```

## 12.148 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 "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;
}

```

## 12.149 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"
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( outfile );
    //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;
}

```

## 12.150 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 "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::ostringstream 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;
}

```

## 12.151 CreateFakeRTDOSE.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 "gdcmTrace.h"
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.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;
    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 automagically 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;
}

```

## 12.152 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 <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::FileNamesType 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";
        gdcm::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];
            sprintf(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;
}

```

## 12.153 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 "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;
}

```

## 12.154 gdcmortoplanes.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 "gdcmSystem.h"
#include "gdcmDirectory.h"
#include "gdcmIPPSorter.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], ippzspacing );
    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();
    }
    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
    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(v16->GetOutputPort());
    #else
        planeWidgetY->SetInput(v16->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(v16->GetOutputPort());
    #else
        planeWidgetZ->SetInput(v16->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;
}

```

## 12.155 gdcmreslice.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 "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"
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
}

```

```

#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 ); // cant 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;
}

```

## 12.156 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 "gdcmReader.h"
#include "gdcmAttribute.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];
    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,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\36.71]
# 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 &tbeamsq = 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 &tcompensatorsq = 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 &tcompensatorthicknessdata = 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 &tcompensatorrows = 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 &tcompensatorcols = 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 &tcompensatorpixelspacing = 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 gdcm::DataElement &compensatorposition = nestedds2.GetDataElement( at4.GetTag() );
at4.SetFromDataElement( compensatorposition );
std::cout << at4.GetValue(0) << std::endl;
vtkDoubleArray *d = vtkDoubleArray::New();
d->SetArray( (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/1,1 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\37.4\43.0\37.1\44.7\36] # 1934,2-2n Block Data
  (fffe,e00d)
  (fffe,e0dd)

*/
gdcm::Tag tblocksq(0x300a,0x03a6);
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] = 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;
}

```

## 12.157 gdcmrtpplan.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 "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 "gdcmsReader.h"
#include "gdcmsAttribute.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 * outfilename = argv[2];
    gdcms::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }
    gdcms::MediaStorage ms;
    ms.SetFromFile( reader.GetFile() );
    if( ms != gdcms::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\36.71]
          # 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)
      (ffff,e00d)
      (ffff,e0dd)
    */
    const gdcms::DataSet& ds = reader.GetFile().GetDataSet();

```

```

gdcM::Tag tbeamsq(0x300a,0x00b0);
if( !ds.FindDataElement( tbeamsq ) )
{
    return 1;
}
const gdcM::DataElement &beamsq = ds.GetDataElement( tbeamsq );
//std::cout << beamsq << std::endl;
gdcM::SmartPointer<gdcM::SequenceOfItems> sqi = beamsq.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 &compensatorsq = nestedds.GetDataElement( tcompensatorsq );
//     //std::cout << compensatorsq << std::endl;
//     gdcM::SmartPointer<gdcM::SequenceOfItems> ssqi = compensatorsq.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 &compensatorthicknessdata = nestedds2.GetDataElement( tcompensatorthicknessdata );
//     // std::cout << compensatorthicknessdata << std::endl;
//     gdcM::Attribute<0x300a,0x00ec> at;
//     at.SetFromDataElement( compensatorthicknessdata );
//     const double* pts = at.GetValues();
//     //         (300a,00e7) IS [35]                                     # 2,1 Compensator Rows
//     gdcM::Attribute<0x300a,0x00e7> at1;
//     const gdcM::DataElement &compensatorrows = nestedds2.GetDataElement( at1.GetTag() );
//     at1.SetFromDataElement( compensatorrows );
//     std::cout << at1.GetValue() << std::endl;
//     //         (300a,00e8) IS [37]                                     # 2,1 Compensator Columns
//     gdcM::Attribute<0x300a,0x00e8> at2;
//     const gdcM::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
//     gdcM::Attribute<0x300a,0x00e9> at3;
//     const gdcM::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
//     gdcM::Attribute<0x300a,0x00ea> at4;
//     const gdcM::DataElement &compensatorposition = nestedds2.GetDataElement( at4.GetTag() );
//     at4.SetFromDataElement( compensatorposition );
//     std::cout << at4.GetValue(0) << std::endl;
//     vtkDoubleArray *d = vtkDoubleArray::New();
//     d->SetArray( (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);
//     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.
        # 1934,2-2n Block Data
    (fffe,e00d)
    (fffe,e000) na (Item with undefined length)
    (fffe,e00d)
    (fffe,e0dd)
*/
gdcmm::Tag tblocksq(0x300a,0x00f4);
if( !nestedds.FindDataElement( tblocksq ) )
{
    return 1;
}
const gdcmm::DataElement &blocksq = nestedds.GetDataElement( tblocksq );
//std::cout << "blocksq" << std::endl;
gdcmm::SmartPointer<gdcmm::SequenceOfItems> sssqi = blocksq.GetValueAssSQ();
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] = 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

```

```

#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;
}

```

## 12.158 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"
// 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)
{
    #if (VTK_MAJOR_VERSION >= 6)
        append->AddInputConnection( reader->GetOutputPort(i) );
    #else
        append->AddInput( reader->GetOutput(i) );
    #endif
}
vtkPolyDataWriter *writer = vtkPolyDataWriter::New();
#if (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() );
#if (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;
}

```

## 12.159 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"
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 ); // cant get it to work
cube->GetAssembly()->SetUserTransform( transform ); // cant 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;
}

```

## 12.160 gdcmvolume.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 "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"
// gdcmvolume gdcmData/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;
}

```

## 12.161 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"
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;
}

```

## 12.162 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 <sstream>
#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 "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();
        _imageView->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.GetFilesNames();
        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();
        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();
        _imageView->GetRenderer()->AddActor(_annotation);
        // Add the cut shape actor to the renderer.
        _imageView->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());
#ifdef 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());
    }

```

```

        _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());
#ifdef (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;
}

```

## 12.163 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"
/*
 * 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;
}

```

## 12.164 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 <pthread.h>
struct threadparams
{
    const char **filenames;
    size_t nfiles;
    char *scalarpainter;
}

```

```

// 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;
#ifdef 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);
#ifdef VTK_MAJOR_VERSION >= 6
    int numscal = pixeltype.GetSamplesPerPixel();
    switch( pixeltype )

```

```

    {
    case gdcmm::PixelFormat::INT8:
        output->AllocateScalars( VTK_SIGNED_CHAR, numscal );
        break;
    case gdcmm::PixelFormat::UINT8:
        output->AllocateScalars( VTK_UNSIGNED_CHAR, numscal );
        break;
    case gdcmm::PixelFormat::INT16:
        output->AllocateScalars( VTK_SHORT, numscal );
        break;
    case gdcmm::PixelFormat::UINT16:
        output->AllocateScalars( VTK_UNSIGNED_SHORT, numscal );
        break;
    case gdcmm::PixelFormat::INT32:
        output->AllocateScalars( VTK_INT, numscal );
        break;
    case gdcmm::PixelFormat::UINT32:
        output->AllocateScalars( VTK_UNSIGNED_INT, numscal );
        break;
    default:
        assert(0);
    }
#else
    switch( pixeltype )
    {
    case gdcmm::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 gdcmm::PixelFormat::UINT8:
        output->SetScalarType ( VTK_UNSIGNED_CHAR );
        break;
    case gdcmm::PixelFormat::INT16:
        output->SetScalarType ( VTK_SHORT );
        break;
    case gdcmm::PixelFormat::UINT16:
        output->SetScalarType ( VTK_UNSIGNED_SHORT );
        break;
    case gdcmm::PixelFormat::INT32:
        output->SetScalarType ( VTK_INT );
        break;
    case gdcmm::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 && gdcmm::System::FileIsDirectory( argv[1] ) )
    {
        gdcmm::Directory d;
        d.Load( argv[1] );
        gdcmm::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;
}

```

## 12.165 AWTMedical3.java

```

/*=====
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.

=====*/

```

```

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.GetFilesNames());
        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.GetFilesNames().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.GetFilesNames();
            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 creatin 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 saggitalColors = new vtkImageMapToColors();
saggitalColors.SetInput(theImageData);
saggitalColors.SetLookupTable(bwLut);
vtkImageActor saggital = new vtkImageActor();
saggital.SetInput(saggitalColors.GetOutput());
saggital.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(saggital);
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);
}
}
}

```

## 12.166 HelloVTKWorld.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.

=====*/
// We are required to call the package 'vtk' eventhough I (MM) would have preferred
// an import statement along the line of:
// import vtkgdcml.*;
import vtk.*;
/*
 * Compilation:
 * CLASSPATH=vtkgdcml.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:vtkgdcml.jar:gdcml.jar:. java HelloVTKWorld gdcmlData/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("vtkgdcmlJava");
        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() == vtkgdcml.vtkgdcml.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( dircos );
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 );
}
}
```

## 12.167 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=vtkgdc.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:vtkgdc.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("vtkgdc.jar");
    }
    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();
}
}

```

## 12.168 MPRViewer.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=vtkgdc.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:vtkgdc.jar:gdcm.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("vtkgdc.jar");
    }
    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();
}
}

```

## 12.169 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 = oz+(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 );
}
}

```

## 12.170 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' eventhough I (MM) would have preferred
// an import statement along the line of:
// import vtkgdcm.*;
import vtk.*;
/*
 * Usage:
 * export LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:.
 * java -classpath 'pwd'/vtkgdcm.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("vtkgdcmJava");
        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() );
    }
}

```

```

    }
}

```

## 12.171 CastConvertPhilips.py

```

1
14
15 """
16 Usage:
17
18 python --public /path/to/directory/
19 or
20 python --private /path/to/directory/
21
22 python --public --extension bak /path/to/directory/
23
24 rename -f 's/\.bak$//' *.bak
25
26 TODO:
27 http://docs.python.org/library/optparse.html#module-optparse
28 """
29
30 import vtkgdc
31 import vtk
32 import sys
33 import gdc
34
35 def ProcessOneFilePublic(filename, outfilename, tmpfile):
36     gdc.ImageHelper.SetForceRescaleInterceptSlope(True)
37     vtkreader = vtkgdc.vtkGDCMImageReader()
38     vtkreader.SetFileName( filename )
39     vtkreader.Update()
40
41     cast = vtk.vtkImageCast()
42     cast.SetInput( vtkreader.GetOutput() )
43     cast.SetOutputScalarTypeToUnsignedShort()
44
45     # vtkGDCMImageWriter does not support Sequence, so let's write a tmp file first:
46     # Some operation will actually be discarded (we simply need a temp storage)
47     vtkwriter = vtkgdc.vtkGDCMImageWriter()
48     vtkwriter.SetFileName( tmpfile )
49     vtkwriter.SetMedicalImageProperties( vtkreader.GetMedicalImageProperties() )
50     vtkwriter.SetDirectionCosines( vtkreader.GetDirectionCosines() )
51     print "Format:", vtkreader.GetImageFormat()
52     vtkwriter.SetImageFormat( vtkreader.GetImageFormat() )
53     vtkwriter.SetInput( cast.GetOutput() )
54     #vtkwriter.Update()
55     vtkwriter.Write()
56
57     # ok now rewrite the exact same file as the original (keep all info)
58     # but use the Pixel Data Element from the written file
59     tmpreader = gdc.ImageReader()
60     tmpreader.SetFileName( tmpfile )
61     if not tmpreader.Read():
62         sys.exit(1)
63
64     reader = gdc.Reader()
65     reader.SetFileName( filename )
66     if not reader.Read():
67         sys.exit(1)
68
69     # Make sure to remove Slope/Rescale to avoid re-execution
70     ds = reader.GetFile().GetDataSet()
71     tags = [
72         gdc.Tag(0x0028,0x1052),
73         gdc.Tag(0x0028,0x1053),
74         gdc.Tag(0x0028,0x1053),
75     ]
76     for tag in tags:
77         ds.Remove( tag )
78
79     writer = gdc.ImageWriter()
80     writer.SetFileName( outfilename )
81     # Pass image from vtk written file
82     writer.SetImage( tmpreader.GetImage() )
83     # pass dataset from initial 'reader'
84     writer.SetFile( reader.GetFile() )

```

```

85     if not writer.Write():
86         sys.exit(1)
87
88 def ProcessOneFilePrivate(filename, outfilename, tmpfile):
89     vtkreader = vtkgdcmm.vtkGDCMImageReader()
90     vtkreader.SetFileName( filename )
91     vtkreader.Update()
92
93
94     # (2005,1409)      DS      4      0.0
95     # (2005,140a)     DS      16     1.52283272283272
96
97     # (2005,0014)     LO      26     Philips MR Imaging DD 005
98     tag1 = gdcmm.PrivateTag(0x2005,0x09,"Philips MR Imaging DD 005")
99     tag2 = gdcmm.PrivateTag(0x2005,0x0a,"Philips MR Imaging DD 005")
100
101
102
103     # Need to access some private tags, reread the file (for now):
104     reader = gdcmm.Reader()
105     reader.SetFileName( filename )
106     if not reader.Read():
107         sys.exit(1)
108
109     ds = reader.GetFile().GetDataSet()
110
111     el1 = ds.GetDataElement( tag1 )
112     el2 = ds.GetDataElement( tag2 )
113
114
115     #pf = gdcmm.PythonFilter()
116     #pf.SetFile( reader.GetFile() )
117     #print el1.GetTag()
118
119     print el1.GetByteValue()
120     v1 = eval( el1.GetByteValue().GetBuffer() )
121     print el2.GetByteValue()
122     v2 = eval( el2.GetByteValue().GetBuffer() )
123
124     print v1
125     shift = v1
126     print v2
127     scale = v2
128
129     ss = vtk.vtkImageShiftScale()
130     ss.SetInput( vtkreader.GetOutput() )
131     # because VTK image shift / scale convention is inverted from DICOM make sure shift is 0
132     assert shift == 0
133     ss.SetShift( shift )
134     ss.SetScale( scale )
135     ss.SetOutputScalarTypeToUnsignedShort ()
136     ss.Update()
137
138     # vtkGDCMImageWriter does not support Sequence, so let's write a tmp file first:
139     # Some operation will actually be discarded (we simply need a temp storage)
140     vtkwriter = vtkgdcmm.vtkGDCMImageWriter()
141     vtkwriter.SetFileName( tmpfile )
142     vtkwriter.SetMedicalImageProperties( vtkreader.GetMedicalImageProperties() )
143     vtkwriter.SetDirectionCosines( vtkreader.GetDirectionCosines() )
144     vtkwriter.SetImageFormat( reader.GetImageFormat() )
145     # do not pass shift/scale again
146     vtkwriter.SetInput( ss.GetOutput() )
147     #vtkwriter.Update()
148     vtkwriter.Write()
149
150     # ok now rewrite the exact same file as the original (keep all info)
151     # but use the Pixel Data Element from the written file
152     tmpreader = gdcmm.ImageReader()
153     tmpreader.SetFileName( tmpfile )
154     if not tmpreader.Read():
155         sys.exit(1)
156
157     writer = gdcmm.ImageWriter()
158     writer.SetFileName( outfilename )
159     # Pass image from vtk written file
160     writer.SetImage( tmpreader.GetImage() )
161     # pass dataset from initial 'reader'
162     writer.SetFile( reader.GetFile() )
163     if not writer.Write():
164         sys.exit(1)
165

```

```
166 if __name__ == "__main__":
167
168     gdcm.Trace.DebugOff()
169     gdcm.Trace.WarningOff()
170     #filename = sys.argv[1]
171     #outfilename = sys.argv[2]
172     tmpfile = "/tmp/philips_rescaled.dcm"
173     #ProcessOneFile( filename, outfilename, tmpfile )
174     rescaletype = sys.argv[1]
175     assert rescaletype == "--public" or rescaletype == "--private"
176     dirname = sys.argv[2]
177     d = gdcm.Directory()
178     d.Load( dirname )
179
180     for f in d.GetFilenames():
181         #print f
182         ProcessOneFilePublic( f, f + ".bak", tmpfile )
183
184
185 print "success"
```

## 12.172 headsq2dcm.py

```
1
14
15 """
16 Usage:
17 python headsq2dcm.py -D /path/to/VTKData
18 """
19
20 import vtk
21 import vtkgdcm
22 from vtk.util.misc import vtkGetDataRoot
23 VTK_DATA_ROOT = vtkGetDataRoot()
24
25 reader = vtk.vtkVolume16Reader()
26 reader.SetDataDimensions(64, 64)
27 reader.SetDataByteOrderToLittleEndian()
28 reader.SetFilePrefix(VTK_DATA_ROOT + "/Data/headsq/quarter")
29 reader.SetImageRange(1, 93)
30 reader.SetDataSpacing(3.2, 3.2, 1.5)
31
32 cast = vtk.vtkImageCast()
33 cast.SetInput( reader.GetOutput() )
34 cast.SetOutputScalarTypeToUnsignedChar()
35
36 # By default this is creating a Multiframe Grayscale Word Secondary Capture Image Storage
37 writer = vtkgdcm.vtkGDCMImageWriter()
38 writer.SetFileName( "headsq.dcm" )
39 writer.SetInput( reader.GetOutput() )
40 # cast -> Multiframe Grayscale Byte Secondary Capture Image Storage
41 #writer.SetInput( cast.GetOutput() )
42 writer.SetFileDimensionality( 3 )
43 writer.Write()
```

# Index

- ~ASN1
  - gdcmm::ASN1, [131](#)
- ~AnonymizeEvent
  - gdcmm::AnonymizeEvent, [109](#)
- ~Anonymizer
  - gdcmm::Anonymizer, [113](#)
- ~Attribute
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, [155](#)
- ~AudioCodec
  - gdcmm::AudioCodec, [167](#)
- ~BaseCompositeMessage
  - gdcmm::network::BaseCompositeMessage, [172](#)
- ~BaseNormalizedMessage
  - gdcmm::network::BaseNormalizedMessage, [174](#)
- ~BasePDU
  - gdcmm::network::BasePDU, [176](#)
- ~BaseQuery
  - gdcmm::BaseQuery, [179](#)
- ~BaseRootQuery
  - gdcmm::BaseRootQuery, [185](#)
- ~Bitmap
  - gdcmm::Bitmap, [197](#)
- ~BitmapToBitmapFilter
  - gdcmm::BitmapToBitmapFilter, [211](#)
- ~BoxRegion
  - gdcmm::BoxRegion, [214](#)
- ~ByteSwapFilter
  - gdcmm::ByteSwapFilter, [222](#)
- ~ByteValue
  - gdcmm::ByteValue, [225](#)
- ~CAPICryptographicMessageSyntax
  - gdcmm::CAPICryptographicMessageSyntax, [234](#)
- ~CSAHeader
  - gdcmm::CSAHeader, [289](#)
- ~Coder
  - gdcmm::Coder, [250](#)
- ~Command
  - gdcmm::Command, [259](#)
- ~CommandDataSet
  - gdcmm::CommandDataSet, [261](#)
- ~CryptoFactory
  - gdcmm::CryptoFactory, [274](#)
- ~CryptographicMessageSyntax
  - gdcmm::CryptographicMessageSyntax, [276](#)
- ~Curve
  - gdcmm::Curve, [305](#)
- ~DICOMDIRGenerator
  - gdcmm::DICOMDIRGenerator, [356](#)
- ~DataEvent
  - gdcmm::DataEvent, [325](#)
- ~DataSetEvent
  - gdcmm::DataSetEvent, [342](#)
- ~Decoder
  - gdcmm::Decoder, [345](#)
- ~Defs
  - gdcmm::Defs, [349](#)
- ~DeltaEncodingCodec
  - gdcmm::DeltaEncodingCodec, [353](#)
- ~DictConverter
  - gdcmm::DictConverter, [365](#)
- ~DictPrinter
  - gdcmm::DictPrinter, [375](#)
- ~Dicts
  - gdcmm::Dicts, [377](#)
- ~DirectionCosines
  - gdcmm::DirectionCosines, [383](#)
- ~Directory
  - gdcmm::Directory, [387](#)
- ~Dumper
  - gdcmm::Dumper, [395](#)
- ~Element
  - gdcmm::Element< TVR, VM::VM1\_n >, [404](#)
- ~EmptyMaskGenerator
  - gdcmm::EmptyMaskGenerator, [423](#)
- ~Event
  - gdcmm::Event, [434](#)
- ~Exception
  - gdcmm::Exception, [437](#)
- ~File
  - gdcmm::File, [447](#)
- ~FileAnonymizer
  - gdcmm::FileAnonymizer, [452](#)
- ~FileChangeTransferSyntax
  - gdcmm::FileChangeTransferSyntax, [456](#)
- ~FileDecompressLookupTable
  - gdcmm::FileDecompressLookupTable, [459](#)
- ~FileDerivation
  - gdcmm::FileDerivation, [462](#)
- ~FileExplicitFilter

- gdcm::FileExplicitFilter, [466](#)
- ~FileMetaInformation
  - gdcm::FileMetaInformation, [471](#)
- ~FileNameEvent
  - gdcm::FileNameEvent, [483](#)
- ~FileStreamer
  - gdcm::FileStreamer, [493](#)
- ~FilenameGenerator
  - gdcm::FilenameGenerator, [487](#)
- ~Global
  - gdcm::Global, [509](#)
- ~GroupDict
  - gdcm::GroupDict, [513](#)
- ~IconImageFilter
  - gdcm::IconImageFilter, [515](#)
- ~IconImageGenerator
  - gdcm::IconImageGenerator, [519](#)
- ~Image
  - gdcm::Image, [524](#)
- ~ImageApplyLookupTable
  - gdcm::ImageApplyLookupTable, [531](#)
- ~ImageChangePhotometricInterpretation
  - gdcm::ImageChangePhotometricInterpretation, [534](#)
- ~ImageChangePlanarConfiguration
  - gdcm::ImageChangePlanarConfiguration, [538](#)
- ~ImageChangeTransferSyntax
  - gdcm::ImageChangeTransferSyntax, [542](#)
- ~ImageCodec
  - gdcm::ImageCodec, [548](#)
- ~ImageConverter
  - gdcm::ImageConverter, [559](#)
- ~ImageFragmentSplitter
  - gdcm::ImageFragmentSplitter, [563](#)
- ~ImageReader
  - gdcm::ImageReader, [573](#)
- ~ImageRegionReader
  - gdcm::ImageRegionReader, [577](#)
- ~ImageToImageFilter
  - gdcm::ImageToImageFilter, [581](#)
- ~ImageWriter
  - gdcm::ImageWriter, [584](#)
- ~JPEG12Codec
  - gdcm::JPEG12Codec, [614](#)
- ~JPEG16Codec
  - gdcm::JPEG16Codec, [617](#)
- ~JPEG2000Codec
  - gdcm::JPEG2000Codec, [621](#)
- ~JPEG8Codec
  - gdcm::JPEG8Codec, [628](#)
- ~JPEGCodec
  - gdcm::JPEGCodec, [632](#)
- ~JPEGLSCodec
  - gdcm::JPEGLSCodec, [641](#)
- ~JSON
  - gdcm::JSON, [646](#)
- ~KAKADUCodec
  - gdcm::KAKADUCodec, [649](#)
- ~LookupTable
  - gdcm::LookupTable, [657](#)
- ~MemberCommand
  - gdcm::MemberCommand< T >, [686](#)
- ~MeshPrimitive
  - gdcm::MeshPrimitive, [692](#)
- ~ModuleEntry
  - gdcm::ModuleEntry, [707](#)
- ~MrProtocol
  - gdcm::MrProtocol, [718](#)
- ~Object
  - gdcm::Object, [748](#)
- ~OpenSSLCryptographicMessageSyntax
  - gdcm::OpenSSLCryptographicMessageSyntax, [753](#)
- ~OpenSSL7CryptographicMessageSyntax
  - gdcm::OpenSSL7CryptographicMessageSyntax, [758](#)
- ~Orientation
  - gdcm::Orientation, [761](#)
- ~Overlay
  - gdcm::Overlay, [767](#)
- ~PDBHeader
  - gdcm::PDBHeader, [788](#)
- ~PDFCodec
  - gdcm::PDFCodec, [792](#)
- ~PGXCodec
  - gdcm::PGXCodec, [800](#)
- ~PNMCodec
  - gdcm::PNMCodec, [834](#)
- ~PVRGCodec
  - gdcm::PVRGCodec, [875](#)
- ~ParseException
  - gdcm::ParseException, [775](#)
- ~Parser
  - gdcm::Parser, [778](#)
- ~Pixmap
  - gdcm::Pixmap, [817](#)
- ~PixmapReader
  - gdcm::PixmapReader, [823](#)
- ~PixmapToPixmapFilter
  - gdcm::PixmapToPixmapFilter, [827](#)
- ~PixmapWriter
  - gdcm::PixmapWriter, [830](#)
- ~Preamble
  - gdcm::Preamble, [838](#)
- ~Printer
  - gdcm::Printer, [861](#)
- ~PrivateDict
  - gdcm::PrivateDict, [865](#)
- ~ProgressEvent
  - gdcm::ProgressEvent, [872](#)



- ~PythonFilter
  - gdcm::PythonFilter, [878](#)
- ~QueryBase
  - gdcm::QueryBase, [880](#)
- ~RAWCodec
  - gdcm::RAWCodec, [895](#)
- ~RLECodec
  - gdcm::RLECodec, [917](#)
- ~Reader
  - gdcm::Reader, [901](#)
- ~Region
  - gdcm::Region, [908](#)
- ~Rescaler
  - gdcm::Rescaler, [912](#)
- ~SHA1
  - gdcm::SHA1, [987](#)
- ~Scanner
  - gdcm::Scanner, [926](#)
- ~Segment
  - gdcm::Segment, [935](#)
- ~SegmentReader
  - gdcm::SegmentReader, [947](#)
- ~SegmentWriter
  - gdcm::SegmentWriter, [950](#)
- ~SegmentedPaletteColorLookupTable
  - gdcm::SegmentedPaletteColorLookupTable, [943](#)
- ~SerieHelper
  - gdcm::SerieHelper, [971](#)
- ~ServiceClassUser
  - gdcm::ServiceClassUser, [980](#)
- ~SimpleMemberCommand
  - gdcm::SimpleMemberCommand< T >, [991](#)
- ~SimpleSubjectWatcher
  - gdcm::SimpleSubjectWatcher, [994](#)
- ~SmartPointer
  - gdcm::SmartPointer< ObjectType >, [1001](#)
- ~Sorter
  - gdcm::Sorter, [1009](#)
- ~Spacing
  - gdcm::Spacing, [1014](#)
- ~SplitMosaicFilter
  - gdcm::SplitMosaicFilter, [1016](#)
- ~StreamImageReader
  - gdcm::StreamImageReader, [1022](#)
- ~StreamImageWriter
  - gdcm::StreamImageWriter, [1027](#)
- ~StrictScanner
  - gdcm::StrictScanner, [1035](#)
- ~StringFilter
  - gdcm::StringFilter, [1047](#)
- ~Subject
  - gdcm::Subject, [1053](#)
- ~Surface
  - gdcm::Surface, [1058](#)
- ~SurfaceReader
  - gdcm::SurfaceReader, [1075](#)
- ~SurfaceWriter
  - gdcm::SurfaceWriter, [1078](#)
- ~Table
  - gdcm::Table, [1093](#)
- ~TableEntry
  - gdcm::TableEntry, [1095](#)
- ~TableReader
  - gdcm::TableReader, [1097](#)
- ~TableRow
  - gdcm::network::TableRow, [1101](#)
- ~TagPath
  - gdcm::TagPath, [1113](#)
- ~Testing
  - gdcm::Testing, [1116](#)
- ~Trace
  - gdcm::Trace, [1124](#)
- ~Transition
  - gdcm::network::Transition, [1138](#)
- ~ULAction
  - gdcm::network::ULAction, [1184](#)
- ~ULBasicCallback
  - gdcm::network::ULBasicCallback, [1224](#)
- ~ULConnection
  - gdcm::network::ULConnection, [1227](#)
- ~ULConnectionCallback
  - gdcm::network::ULConnectionCallback, [1232](#)
- ~ULConnectionManager
  - gdcm::network::ULConnectionManager, [1238](#)
- ~ULEvent
  - gdcm::network::ULEvent, [1245](#)
- ~ULWritingCallback
  - gdcm::network::ULWritingCallback, [1249](#)
- ~UserInformation
  - gdcm::network::UserInformation, [1260](#)
- ~Validate
  - gdcm::Validate, [1264](#)
- ~Value
  - gdcm::Value, [1267](#)
- ~Version
  - gdcm::Version, [1271](#)
- ~Writer
  - gdcm::Writer, [1417](#)
- ~XMLDictReader
  - gdcm::XMLDictReader, [1421](#)
- ~XMLPrinter
  - gdcm::XMLPrinter, [1425](#)
- ~XMLPrivateDictReader
  - gdcm::XMLPrivateDictReader, [1428](#)
- ~vtkGDCMImageReader
  - vtkGDCMImageReader, [1297](#)
- ~vtkGDCMImageReader2
  - vtkGDCMImageReader2, [1312](#)

- ~vtkGDCMImageWriter
  - vtkGDCMImageWriter, [1326](#)
- ~vtkGDCMMedicalImageProperties
  - vtkGDCMMedicalImageProperties, [1334](#)
- ~vtkGDCMPolyDataReader
  - vtkGDCMPolyDataReader, [1338](#)
- ~vtkGDCMPolyDataWriter
  - vtkGDCMPolyDataWriter, [1343](#)
- ~vtkGDCMTesting
  - vtkGDCMTesting, [1348](#)
- ~vtkGDCMThreadedImageReader
  - vtkGDCMThreadedImageReader, [1352](#)
- ~vtkGDCMThreadedImageReader2
  - vtkGDCMThreadedImageReader2, [1356](#)
- ~vtkImageColorViewer
  - vtkImageColorViewer, [1366](#)
- ~vtkImageMapToColors16
  - vtkImageMapToColors16, [1379](#)
- ~vtkImageMapToWindowLevelColors2
  - vtkImageMapToWindowLevelColors2, [1385](#)
- ~vtkImagePlanarComponentsToComponents
  - vtkImagePlanarComponentsToComponents, [1389](#)
- ~vtkImageRGBToYBR
  - vtkImageRGBToYBR, [1391](#)
- ~vtkImageYBRToRGB
  - vtkImageYBRToRGB, [1394](#)
- ~vtkLookupTable16
  - vtkLookupTable16, [1397](#)
- ~vtkRTStructSetProperties
  - vtkRTStructSetProperties, [1401](#)
- AAAbortPDU
  - gdcm::network::AAAbortPDU, [88](#)
- AAAssociateACPDU
  - gdcm::network::AAAssociateACPDU, [91](#)
  - gdcm::network::AAAssociateRQPDU, [103](#)
- AAAssociateRJPDU
  - gdcm::network::AAAssociateRJPDU, [95](#)
- AAAssociateRQPDU
  - gdcm::network::AAAssociateACPDU, [94](#)
  - gdcm::network::AAAssociateRQPDU, [99](#)
- AbstractMultiDimensionalImageModel
  - gdcm::UIDs, [1171](#)
- AbstractSyntax
  - gdcm::network::AbstractSyntax, [105](#)
  - gdcm::PresentationContext, [844](#)
- AcquisitionContextSRStorage
  - gdcm::UIDs, [1170](#)
- ActiveComponent
  - vtkImageMapToColors16, [1383](#)
- Add
  - gdcm::GroupDict, [513](#)
- add1
  - gdcm, [63](#)
- AddAcceptedPresentationContext
  - gdcm::network::ULConnection, [1227](#)
- AddContourReferencedFrameOfReference
  - vtkRTStructSetProperties, [1401](#)
- AddCSAHeaderDictEntry
  - gdcm::CSAHeaderDict, [294](#)
- AddDerivationDescription
  - gdcm::FileDerivation, [462](#)
- AddDictEntry
  - gdcm::Dict, [361](#)
  - gdcm::PrivateDict, [865](#)
- AddFile
  - gdcm::FileSet, [490](#)
  - gdcm::SerieHelper, [971](#)
- AddFileName
  - gdcm::SerieHelper, [972](#)
- AddFragment
  - gdcm::SequenceOfFragments, [955](#)
- AddFromFile
  - gdcm::PresentationContextGenerator, [849](#)
- AddGroupLength
  - gdcm::DictConverter, [365](#)
- AddImageDirectoryRecord
  - gdcm::DICOmdirGenerator, [357](#)
- AddInput
  - vtkImageColorViewer, [1366](#)
- AddInputConnection
  - vtkImageColorViewer, [1366](#)
- AddIOD
  - gdcm::IODs, [601](#)
- AddIODEntry
  - gdcm::IOD, [595](#)
- AddItem
  - gdcm::SequenceOfItems, [964](#)
- AddMacro
  - gdcm::Macros, [668](#)
  - gdcm::Module, [703](#)
- AddMacroEntry
  - gdcm::Macro, [665](#)
- AddModule
  - gdcm::Modules, [711](#)
- AddModuleEntry
  - gdcm::Module, [703](#)
  - gdcm::NestedModuleEntries, [731](#)
- AddNewUndefinedLengthItem
  - gdcm::SequenceOfItems, [964](#)
- AddObserver
  - gdcm::Subject, [1053](#)
- AddPatientDirectoryRecord
  - gdcm::DICOmdirGenerator, [357](#)
- AddPresentationContext
  - gdcm::network::AAAssociateRQPDU, [99](#)
  - gdcm::PresentationContextGenerator, [849](#)
- AddPresentationContextAC

- gdcm::network::AAAssociateACPDU, [92](#)
- AddPresentationDataValue
  - gdcm::network::PDataTFPDU, [782](#)
- AddPrimitiveData
  - gdcm::MeshPrimitive, [692](#)
- AddPrivateTag
  - gdcm::Scanner, [926](#)
  - gdcm::StrictScanner, [1035](#)
- AddPurposeOfReferenceCodeSequence
  - gdcm::FileDerivation, [462](#)
- AddQueryDataSet
  - gdcm::BaseQuery, [179](#)
- AddReference
  - gdcm::FileDerivation, [462](#)
- AddReferencedFrameOfReference
  - vtkRTStructSetProperties, [1402](#)
- AddRestriction
  - gdcm::SerieHelper, [972](#)
- AddRoleSelectionSub
  - gdcm::network::UserInformation, [1261](#)
- AddSegment
  - gdcm::SegmentWriter, [951](#)
- AddSelect
  - gdcm::Sorter, [1009](#)
- AddSeriesDirectoryRecord
  - gdcm::DICOMDIRGenerator, [357](#)
- AddSkipTag
  - gdcm::Scanner, [927](#)
  - gdcm::StrictScanner, [1036](#)
- AddSOPClassExtendedNegociationSub
  - gdcm::network::UserInformation, [1261](#)
- AddSourceImageSequence
  - gdcm::FileDerivation, [462](#)
- AddStructureSetROI
  - vtkRTStructSetProperties, [1402](#)
- AddStructureSetROIObservation
  - vtkRTStructSetProperties, [1402](#)
- AddStudyDirectoryRecord
  - gdcm::DICOMDIRGenerator, [357](#)
- AddSurface
  - gdcm::Segment, [935](#)
- AddTag
  - gdcm::Scanner, [927](#)
  - gdcm::StrictScanner, [1036](#)
- AddTransferSyntax
  - gdcm::network::PresentationContextRQ, [853](#)
  - gdcm::PresentationContext, [843](#)
- AdultMouseAnatomyOntology
  - gdcm::UIDs, [1168](#)
- AdvancedBlendingPresentationStateStorage
  - gdcm::UIDs, [1169](#)
- AE
  - gdcm::VR, [1284](#)
- AECComp
  - gdcm, [58](#)
- AES128\_CIPHER
  - gdcm::CryptographicMessageSyntax, [276](#)
- AES192\_CIPHER
  - gdcm::CryptographicMessageSyntax, [276](#)
- AES256\_CIPHER
  - gdcm::CryptographicMessageSyntax, [276](#)
- AffectedSOPClassUID
  - gdcm::network::CEchoRQ, [238](#)
- ALGOType
  - gdcm::Segment, [934](#)
- ALGOType\_END
  - gdcm::Segment, [935](#)
- Allocate
  - gdcm::LookupTable, [658](#)
- AmbulatoryECGWaveformStorage
  - gdcm::MediaStorage, [677](#)
  - gdcm::UIDs, [1165](#)
- AnatomicRegion
  - gdcm::Segment, [940](#)
- AnatomicRegionModifiers
  - gdcm::Segment, [940](#)
- AnonymizeEvent
  - gdcm::AnonymizeEvent, [108](#), [109](#)
- Anonymizer
  - gdcm::Anonymizer, [113](#)
- Append
  - gdcm::ByteValue, [226](#)
  - gdcm::Global, [509](#)
- AppendFrameEncode
  - gdcm::ImageCodec, [548](#)
  - gdcm::JPEG2000Codec, [621](#)
  - gdcm::JPEGCodec, [632](#)
  - gdcm::JPEGLSCodec, [641](#)
  - gdcm::RLECodec, [917](#)
- AppendImplementationClassUID
  - gdcm::FileMetaInformation, [471](#)
- AppendRowEncode
  - gdcm::ImageCodec, [548](#)
  - gdcm::JPEG2000Codec, [621](#)
  - gdcm::JPEGCodec, [633](#)
  - gdcm::JPEGLSCodec, [641](#)
  - gdcm::RLECodec, [917](#)
- AppendToDataElement
  - gdcm::FileStreamer, [493](#)
- AppendToGroupDataElement
  - gdcm::FileStreamer, [493](#)
- ApplicationContext
  - gdcm::network::ApplicationContext, [120](#)
- Apply
  - gdcm::ImageApplyLookupTable, [531](#)
- ApplyInverseVideo
  - vtkGDCMImageReader, [1305](#)
  - vtkGDCMImageReader2, [1320](#)

- ApplyLookupTable
  - vtkGDCMImageReader, [1306](#)
  - vtkGDCMImageReader2, [1320](#)
- ApplyPlanarConfiguration
  - vtkGDCMImageReader, [1306](#)
  - vtkGDCMImageReader2, [1320](#)
- ApplyShiftScale
  - vtkGDCMImageReader, [1306](#)
  - vtkGDCMImageReader2, [1321](#)
- ApplyYBRToRGB
  - vtkGDCMImageReader, [1306](#)
  - vtkGDCMImageReader2, [1321](#)
- Area
  - gdcm::BoxRegion, [214](#)
  - gdcm::Region, [908](#)
- AResourceRPPDU
  - gdcm::network::AResourceRPPDU, [125](#)
- AResourceRQPDU
  - gdcm::network::AResourceRQPDU, [127](#)
- AreOverlaysInPixelData
  - gdcm::Bitmap, [197](#)
  - gdcm::Pixmap, [817](#)
- ARGB
  - gdcm::PhotometricInterpretation, [804](#)
- ArrayIncludeMacroType
  - gdcm::Macro, [665](#)
  - gdcm::Module, [703](#)
- ArrayType
  - gdcm::Attribute< Group, Element, TVR, TVM >, [136](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [145](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [154](#)
- ArterialPulseWaveformStorage
  - gdcm::UIDs, [1169](#)
- ARTIMTimer
  - gdcm::network::ARTIMTimer, [129](#)
- AS
  - gdcm::VR, [1284](#)
- ASComp
  - gdcm, [58](#)
- ASN1
  - gdcm::ASN1, [131](#)
- AsynchronousOperationsWindowSub
  - gdcm::network::AsynchronousOperationsWindowSub, [133](#)
- AT
  - gdcm::VR, [1284](#)
- Attribute
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [155](#)
  - gdcm::terminal, [85](#)
- Audio
  - gdcm::MediaStorage, [679](#)
- AudioCodec
  - gdcm::AudioCodec, [167](#)
- AudioSRStorageTrialRetired
  - gdcm::UIDs, [1166](#)
- AUTOMATIC
  - gdcm::Segment, [935](#)
- AutoPixelMinMax
  - gdcm::IconImageGenerator, [519](#)
- AutorefractionMeasurementsStorage
  - gdcm::UIDs, [1169](#)
- AXIAL
  - gdcm::Orientation, [761](#)
- backslash
  - gdcm, [63](#)
- BadBigEndian
  - gdcm::SwapCode, [1081](#)
- BadLittleEndian
  - gdcm::SwapCode, [1081](#)
- BALCPPProtect
  - gdcm::Anonymizer, [113](#)
- Base64
  - gdcm::Base64, [169](#)
- BaseQuery
  - gdcm::BaseQuery, [179](#)
- BaseRootQuery
  - gdcm::BaseRootQuery, [185](#)
- BasicAnnotationBoxSOPClass
  - gdcm::UIDs, [1164](#)
- BasicApplicationLevelConfidentialityProfile
  - gdcm::Anonymizer, [113](#)
- BasicCodedEntry
  - gdcm::SegmentHelper::BasicCodedEntry, [189](#)
- BasicCodedEntryVector
  - gdcm::Segment, [934](#)
- BasicColorImageBoxSOPClass
  - gdcm::UIDs, [1164](#)
- BasicColorPrintManagementMetaSOPClass
  - gdcm::UIDs, [1164](#)
- BasicFilmBoxSOPClass
  - gdcm::UIDs, [1164](#)
- BasicFilmSessionSOPClass
  - gdcm::UIDs, [1164](#)
- BasicGrayscaleImageBoxSOPClass
  - gdcm::UIDs, [1164](#)
- BasicGrayscalePrintManagementMetaSOPClass
  - gdcm::UIDs, [1164](#)
- BasicOffsetTable
  - gdcm::BasicOffsetTable, [192](#)
- BasicPrintImageOverlayBoxSOPClassRetired
  - gdcm::UIDs, [1164](#)
- BasicStructuredDisplayStorage
  - gdcm::UIDs, [1170](#)
- BasicStudyContentNotificationSOPClassRetired

- gdcm::UIDs, [1163](#)
- BasicTextSR
  - gdcm::MediaStorage, [677](#)
- BasicTextSRStorage
  - gdcm::UIDs, [1166](#)
- BasicVoiceAudioWaveformStorage
  - gdcm::MediaStorage, [677](#)
  - gdcm::UIDs, [1165](#)
- Begin
  - gdcm::CSAHeaderDict, [294](#)
  - gdcm::DataSet, [331](#)
  - gdcm::Dict, [361](#)
  - gdcm::IODs, [601](#)
  - gdcm::Scanner, [927](#)
  - gdcm::SequenceOfFragments, [956](#)
  - gdcm::SequenceOfItems, [964](#)
  - gdcm::StrictScanner, [1036](#)
- BigEndian
  - gdcm::SwapCode, [1081](#)
- Bitmap
  - gdcm::Bitmap, [197](#)
  - gdcm::JPEG2000Codec, [626](#)
  - gdcm::PixelFormat, [814](#)
- BitmapToBitmapFilter
  - gdcm::BitmapToBitmapFilter, [211](#)
- BitSample
  - gdcm::JPEGCodec, [638](#)
  - gdcm::LookupTable, [662](#)
- black
  - gdcm::terminal, [85](#)
- BlendingSoftcopyPresentationStateStorageSOPClass
  - gdcm::UIDs, [1165](#)
- blink
  - gdcm::terminal, [85](#)
- BLUE
  - gdcm::LookupTable, [657](#)
- blue
  - gdcm::terminal, [85](#)
- BOOL\_FUNCTION\_PFILE\_PFILE\_POINTER
  - gdcm, [58](#)
- BoundingBox
  - gdcm::BoxRegion, [214](#)
- BoxRegion
  - gdcm::BoxRegion, [214](#)
- BreakConnection
  - gdcm::network::ULConnectionManager, [1239](#)
- BreakConnectionNow
  - gdcm::network::ULConnectionManager, [1239](#)
- BreastImagingRelevantPatientInformationQuery
  - gdcm::UIDs, [1167](#)
- BreastProjectionXRayImageStorageForPresentation
  - gdcm::MediaStorage, [678](#)
  - gdcm::UIDs, [1169](#)
- BreastProjectionXRayImageStorageForProcessing
  - gdcm::MediaStorage, [678](#)
  - gdcm::UIDs, [1169](#)
- BreastTomosynthesisImageStorage
  - gdcm::MediaStorage, [678](#)
  - gdcm::UIDs, [1168](#)
- bright
  - gdcm::terminal, [85](#)
- Build
  - vtkLookupTable16, [1397](#)
- ByteBuffer
  - gdcm::ByteBuffer, [218](#)
- bytes
  - gdcm::Tag, [1112](#)
- ByteSwap
  - gdcm::ByteSwapFilter, [222](#)
- ByteSwapFilter
  - gdcm::ByteSwapFilter, [221](#), [222](#)
- ByteValue
  - gdcm::ByteValue, [225](#)
- C\_CANCEL\_RQ
  - gdcm::network::DIMSE, [381](#)
- C\_ECHO\_RQ
  - gdcm::network::DIMSE, [381](#)
- C\_ECHO\_RSP
  - gdcm::network::DIMSE, [381](#)
- C\_FIND\_RQ
  - gdcm::network::DIMSE, [381](#)
- C\_FIND\_RSP
  - gdcm::network::DIMSE, [381](#)
- C\_GET\_RQ
  - gdcm::network::DIMSE, [381](#)
- C\_GET\_RSP
  - gdcm::network::DIMSE, [381](#)
- C\_MOVE\_RQ
  - gdcm::network::DIMSE, [381](#)
- C\_MOVE\_RSP
  - gdcm::network::DIMSE, [381](#)
- C\_STORE\_RQ
  - gdcm::network::DIMSE, [381](#)
- C\_STORE\_RSP
  - gdcm::network::DIMSE, [381](#)
- CALIBRATED
  - gdcm::Spacing, [1014](#)
- CanCode
  - gdcm::AudioCodec, [167](#)
  - gdcm::Coder, [251](#)
  - gdcm::ImageCodec, [549](#)
  - gdcm::JPEG2000Codec, [621](#)
  - gdcm::JPEGCodec, [633](#)
  - gdcm::JPEGLSCCodec, [641](#)
  - gdcm::KAKADUCCodec, [649](#)
  - gdcm::PDFCodec, [792](#)
  - gdcm::PGXCodec, [801](#)

- gdcm::PNMCodec, [834](#)
- gdcm::PVRGCodec, [875](#)
- gdcm::RAWCodec, [895](#)
- gdcm::RLECodec, [917](#)
- CanDecode
  - gdcm::AudioCodec, [167](#)
  - gdcm::Decoder, [346](#)
  - gdcm::DeltaEncodingCodec, [353](#)
  - gdcm::ImageCodec, [549](#)
  - gdcm::JPEG2000Codec, [622](#)
  - gdcm::JPEGCodec, [633](#)
  - gdcm::JPEGLSCodec, [641](#)
  - gdcm::KAKADUCodec, [649](#)
  - gdcm::PDFCodec, [792](#)
  - gdcm::PGXCodec, [801](#)
  - gdcm::PNMCodec, [835](#)
  - gdcm::PVRGCodec, [876](#)
  - gdcm::RAWCodec, [896](#)
  - gdcm::RLECodec, [917](#)
- CanDisplay
  - gdcm::VR, [1285](#)
- CanEmptyTag
  - gdcm::Anonymizer, [114](#)
- CanRead
  - gdcm::Reader, [901](#)
- CanReadFile
  - vtkGDCMImageReader, [1297](#)
  - vtkGDCMImageReader2, [1312](#)
- CanReadImage
  - gdcm::StreamImageReader, [1022](#)
- CanStoreLossy
  - gdcm::TransferSyntax, [1132](#)
- CanWriteFile
  - gdcm::StreamImageWriter, [1027](#)
- CAPI
  - gdcm::CryptoFactory, [273](#)
- CAPICryptoFactory
  - gdcm::CAPICryptoFactory, [232](#)
- CAPICryptographicMessageSyntax
  - gdcm::CAPICryptographicMessageSyntax, [234](#)
- CardiacElectrophysiologyWaveformStorage
  - gdcm::MediaStorage, [677](#)
  - gdcm::UIDs, [1165](#)
- CardiacRelevantPatientInformationQuery
  - gdcm::UIDs, [1167](#)
- CEcho
  - gdcm::CompositeNetworkFunctions, [266](#)
- CFind
  - gdcm::CompositeNetworkFunctions, [266](#)
- Change
  - gdcm::FileChangeTransferSyntax, [456](#)
  - gdcm::FileDecompressLookupTable, [459](#)
  - gdcm::FileExplicitFilter, [466](#)
  - gdcm::ImageChangePhotometricInterpretation, [534](#)
  - gdcm::ImageChangePlanarConfiguration, [538](#)
  - gdcm::ImageChangeTransferSyntax, [543](#)
- ChangeFMI
  - gdcm::FileExplicitFilter, [466](#)
- ChangeMonochrome
  - gdcm::ImageChangePhotometricInterpretation, [535](#)
- ChangeRGB2YBR
  - gdcm::ImageChangePhotometricInterpretation, [535](#)
- ChangeYBR2RGB
  - gdcm::ImageChangePhotometricInterpretation, [535](#)
- CharacterDataHandler
  - gdcm::TableReader, [1097](#)
  - gdcm::XMLDictReader, [1422](#)
  - gdcm::XMLPrivateDictReader, [1428](#)
- CheckDataElement
  - gdcm::FileStreamer, [494](#)
- CheckEvent
  - gdcm::AnonymizeEvent, [109](#)
  - gdcm::DataEvent, [326](#)
  - gdcm::DataSetEvent, [342](#)
  - gdcm::Event, [434](#)
  - gdcm::FileNameEvent, [483](#)
  - gdcm::ProgressEvent, [872](#)
- CheckFileMetaInformationOff
  - gdcm::Writer, [1417](#)
- CheckFileMetaInformationOn
  - gdcm::Writer, [1417](#)
- CheckTemplateFileName
  - gdcm::FileStreamer, [494](#)
- ChestCADSRStorage
  - gdcm::UIDs, [1166](#)
- CipherTypes
  - gdcm::CryptographicMessageSyntax, [275](#)
- Clamp
  - gdcm, [63](#)
- clean
  - gdcm, [63](#)
- CleanupUnusedBits
  - gdcm::ImageCodec, [549](#)
- Clear
  - gdcm::Bitmap, [197](#)
  - gdcm::ByteValue, [226](#)
  - gdcm::DataElement, [313](#)
  - gdcm::DataSet, [331](#)
  - gdcm::IOD, [595](#)
  - gdcm::IODs, [601](#)
  - gdcm::Item, [609](#)
  - gdcm::LookupTable, [658](#)
  - gdcm::Macro, [666](#)
  - gdcm::Macros, [668](#)
  - gdcm::Module, [703](#)
  - gdcm::Modules, [711](#)
  - gdcm::Preamble, [838](#)
  - gdcm::SequenceOfFragments, [956](#)



- gdcM::SequenceOfItems, 964
- gdcM::SerieHelper, 972
- gdcM::Value, 1267
- vtkGDCMMedicalImageProperties, 1334
- vtkRTStructSetProperties, 1402
- ClearInternalUIDs
  - gdcM::Anonymizer, 114
- ClearSkipTags
  - gdcM::Scanner, 927
  - gdcM::StrictScanner, 1036
- ClearTags
  - gdcM::Scanner, 927
  - gdcM::StrictScanner, 1036
- Clone
  - gdcM::BoxRegion, 215
  - gdcM::ImageCodec, 549
  - gdcM::JPEG2000Codec, 622
  - gdcM::JPEGCodec, 633
  - gdcM::JPEGLSCodec, 642
  - gdcM::KAKADUCodec, 649
  - gdcM::PGXCodec, 801
  - gdcM::PNMCodec, 835
  - gdcM::PVRGCodec, 876
  - gdcM::RAWCodec, 896
  - gdcM::Region, 909
  - gdcM::RLECodec, 918
- CM
  - gdcM::SegmentHelper::BasicCodedEntry, 190
- cMaxEventID
  - gdcM::network, 83
- cMaxStateID
  - gdcM::network, 83
- CMove
  - gdcM::CompositeNetworkFunctions, 267
- CMYK
  - gdcM::PhotometricInterpretation, 804
- Code
  - gdcM::Coder, 251
  - gdcM::JPEG2000Codec, 622
  - gdcM::JPEGCodec, 634
  - gdcM::JPEGLSCodec, 642
  - gdcM::JSON, 646
  - gdcM::KAKADUCodec, 650
  - gdcM::PVRGCodec, 876
  - gdcM::RAWCodec, 896
  - gdcM::RLECodec, 918
- CodeMeaning
  - gdcM::RealWorldValueMappingContent, 906
- CodeString
  - gdcM::CodeString, 254, 255
- CodeValue
  - gdcM::RealWorldValueMappingContent, 906
- ColonCADSRStorage
  - gdcM::UIDs, 1170
- Color
  - gdcM::terminal, 85
- ColorArray
  - gdcM::SurfaceHelper, 1071
- ColorPaletteQueryRetrieveInformationModelFIND
  - gdcM::UIDs, 1171
- ColorPaletteQueryRetrieveInformationModelGET
  - gdcM::UIDs, 1171
- ColorPaletteQueryRetrieveInformationModelMOVE
  - gdcM::UIDs, 1171
- ColorPaletteStorage
  - gdcM::UIDs, 1171
- ColorSoftcopyPresentationStateStorageSOPClass
  - gdcM::UIDs, 1165
- Command
  - gdcM::Command, 258, 259
- CommandDataSet
  - gdcM::CommandDataSet, 261
- CommandTypes
  - gdcM::network::DIMSE, 381
- Compatible
  - gdcM::VM, 1280
  - gdcM::VR, 1286
- Component
  - gdcM::PersonName, 798
- CompOperators
  - gdcM, 60
- CompositeInstanceRetrieveWithoutBulkDataGET
  - gdcM::UIDs, 1170
- CompositeInstanceRootRetrieveGET
  - gdcM::UIDs, 1170
- CompositeInstanceRootRetrieveMOVE
  - gdcM::UIDs, 1170
- CompositingPlanarMPRVolumetricPresentationStateStorage
  - gdcM::UIDs, 1169
- Comprehensive3DSRStorage
  - gdcM::UIDs, 1170
- ComprehensiveSR
  - gdcM::MediaStorage, 677
- ComprehensiveSRStorage
  - gdcM::UIDs, 1166
- ComprehensiveSRStorageTrialRetired
  - gdcM::UIDs, 1166
- CompressionTypes
  - vtkGDCMImageWriter, 1326
- Compute
  - gdcM::EquipmentManufacturer, 432
  - gdcM::MD5, 672
  - gdcM::SHA1, 987
- ComputeBoundingBox
  - gdcM::BoxRegion, 215
  - gdcM::Region, 909
- ComputeBufferLength
  - gdcM::ImageRegionReader, 577

- ComputeByteLength
  - gdcm::SequenceOfFragments, [956](#)
- ComputeDataElement
  - gdcm::DataSet, [331](#)
- ComputeDataSetMediaStorageSOPClass
  - gdcm::FileMetaInformation, [471](#)
- ComputeDataSetTransferSyntax
  - gdcm::FileMetaInformation, [471](#)
- ComputeDistAlongNormal
  - gdcm::DirectionCosines, [383](#)
- ComputedRadiographyImageStorage
  - gdcm::MediaStorage, [676](#)
  - gdcm::UIDs, [1164](#)
- ComputeFile
  - gdcm::MD5, [672](#)
  - gdcm::SHA1, [987](#)
- ComputeFileMD5
  - gdcm::Testing, [1117](#)
- ComputeGroupLength
  - gdcm::DataSet, [331](#)
- ComputeInterceptSlopePixelType
  - gdcm::Rescaler, [912](#)
- ComputeLength
  - gdcm::ByteValue, [226](#)
  - gdcm::Fragment, [506](#)
  - gdcm::SequenceOfFragments, [956](#)
  - gdcm::SequenceOfItems, [965](#)
- ComputeLossyFlag
  - gdcm::Bitmap, [197](#)
- ComputeMD5
  - gdcm::Testing, [1117](#)
- ComputeMediaStorageFromModality
  - gdcm::ImageHelper, [565](#)
- ComputeMOSAICDimensions
  - gdcm::SplitMosaicFilter, [1016](#)
- ComputeMOSAICSliceNormal
  - gdcm::SplitMosaicFilter, [1016](#)
- ComputeMOSAICSlicePosition
  - gdcm::SplitMosaicFilter, [1017](#)
- ComputeNumberOfSurfaces
  - gdcm::SurfaceWriter, [1078](#)
- ComputeOffsetTable
  - gdcm::JPEGCodec, [634](#)
- ComputePixelAspectRatioFromPixelSpacing
  - gdcm::Spacing, [1014](#)
- ComputePixelTypeFromMinMax
  - gdcm::Rescaler, [912](#)
- ComputeSpacingFromImagePositionPatient
  - gdcm::ImageHelper, [565](#)
- ComputeTargetMediaStorage
  - gdcm::ImageWriter, [584](#)
- ComputeVR
  - gdcm::DataSetHelper, [344](#)
- ComputeZSpacing
  - gdcm::IPPSorter, [606](#)
- ConcatenatePDVBlobs
  - gdcm::network::PresentationDataValue, [856](#)
- ConcatenatePDVBlobsAsExplicit
  - gdcm::network::PresentationDataValue, [856](#)
- CONDENSED\_STYLE
  - gdcm::Printer, [861](#)
- Conditional
  - gdcm::Usage, [1257](#)
- CONSOLE
  - gdcm::terminal, [86](#)
- const
  - gdcm::SOPClassUIDToIOD, [1005](#)
- const\_iterator
  - gdcm::CodeString, [253](#)
  - gdcm::LO, [652](#)
  - gdcm::String< TDelimiter, TMaxLength, TPadChar  
>, [1043](#)
- const\_reference
  - gdcm::CodeString, [253](#)
  - gdcm::LO, [652](#)
  - gdcm::String< TDelimiter, TMaxLength, TPadChar  
>, [1043](#)
- const\_reverse\_iterator
  - gdcm::CodeString, [253](#)
  - gdcm::LO, [652](#)
  - gdcm::String< TDelimiter, TMaxLength, TPadChar  
>, [1043](#)
- ConstCharWrapper
  - gdcm::ConstCharWrapper, [269](#)
- ConstIterator
  - gdcm::CSAHeaderDict, [293](#)
  - gdcm::DataSet, [330](#)
  - gdcm::Dict, [360](#)
  - gdcm::Scanner, [925](#)
  - gdcm::SequenceOfFragments, [955](#)
  - gdcm::SequenceOfItems, [963](#)
  - gdcm::StrictScanner, [1034](#)
- Construct
  - gdcm::BaseRootQuery, [185](#)
- ConstructAbortPDU
  - gdcm::network::PDUFactory, [794](#)
- ConstructCEchoRQ
  - gdcm::network::CompositeMessageFactory, [263](#)
- ConstructCFindRQ
  - gdcm::network::CompositeMessageFactory, [263](#)
- ConstructCMoveRQ
  - gdcm::network::CompositeMessageFactory, [263](#)
- ConstructCStoreRQ
  - gdcm::network::CompositeMessageFactory, [264](#)
- ConstructCStoreRSP
  - gdcm::network::CompositeMessageFactory, [264](#)
- ConstructFromString
  - gdcm::TagPath, [1113](#)



- ConstructFromTagList
    - gdcmm::TagPath, 1113
  - ConstructNAction
    - gdcmm::network::NormalizedMessageFactory, 739
  - ConstructNCreate
    - gdcmm::network::NormalizedMessageFactory, 739
  - ConstructNDelete
    - gdcmm::network::NormalizedMessageFactory, 739
  - ConstructNEventReport
    - gdcmm::network::NormalizedMessageFactory, 740
  - ConstructNGet
    - gdcmm::network::NormalizedMessageFactory, 740
  - ConstructNSet
    - gdcmm::network::NormalizedMessageFactory, 740
  - ConstructorType
    - gdcmm::Dicts, 377
  - ConstructPDU
    - gdcmm::network::PDUFactory, 794
  - ConstructPDV
    - gdcmm::network::BaseCompositeMessage, 172
    - gdcmm::network::BaseNormalizedMessage, 174
    - gdcmm::network::CEchoRQ, 238
    - gdcmm::network::CFindRQ, 243
    - gdcmm::network::CMoveRQ, 247
    - gdcmm::network::CStoreRQ, 301
    - gdcmm::network::CStoreRSP, 302
    - gdcmm::network::NActionRQ, 721
    - gdcmm::network::NCreateRQ, 724
    - gdcmm::network::NDeleteRQ, 727
    - gdcmm::network::NEventReportRQ, 733
    - gdcmm::network::NGetRQ, 736
    - gdcmm::network::NSetRQ, 744
  - ConstructPDVByDataSet
    - gdcmm::network::CEchoRSP, 240
    - gdcmm::network::CFindCancelRQ, 241
    - gdcmm::network::CFindRSP, 244
    - gdcmm::network::CMoveCancelRq, 245
    - gdcmm::network::CMoveRSP, 248
    - gdcmm::network::NActionRSP, 722
    - gdcmm::network::NCreateRSP, 725
    - gdcmm::network::NDeleteRSP, 728
    - gdcmm::network::NEventReportRSP, 734
    - gdcmm::network::NGetRSP, 737
    - gdcmm::network::NSetRSP, 746
  - ConstructQuery
    - gdcmm::CompositeNetworkFunctions, 267, 268
    - gdcmm::NormalizedNetworkFunctions, 741
  - ConstructReleasePDU
    - gdcmm::network::PDUFactory, 794
  - ContentAssessmentResultsStorage
    - gdcmm::UIDs, 1170
  - Convert
    - gdcmm::DictConverter, 365
    - gdcmm::ImageConverter, 560
  - ConvertRGBToPaletteColor
    - gdcmm::IconImageGenerator, 519
  - ConvertToCXX
    - gdcmm::DictConverter, 366
  - ConvertToUNC
    - gdcmm::System, 1085
  - ConvertToXML
    - gdcmm::DictConverter, 366
  - CornealTopographyMapStorage
    - gdcmm::UIDs, 1170
  - CORONAL
    - gdcmm::Orientation, 761
  - Create
    - gdcmm::Preamble, 838
  - CreateCEchoPDU
    - gdcmm::network::PDUFactory, 794
  - CreateCFindPDU
    - gdcmm::network::PDUFactory, 794
  - CreateCMovePDU
    - gdcmm::network::PDUFactory, 794
  - CreateCMSProvider
    - gdcmm::CAPICryptoFactory, 233
    - gdcmm::CryptoFactory, 274
    - gdcmm::OpenSSLCryptoFactory, 751
    - gdcmm::OpenSSL7CryptoFactory, 756
  - CreateCStoreRQPDU
    - gdcmm::network::PDUFactory, 795
  - CreateCStoreRSPPDU
    - gdcmm::network::PDUFactory, 795
  - CreateDefaultUniqueSeriesIdentifier
    - gdcmm::SerieHelper, 972
  - CreateNActionPDU
    - gdcmm::network::PDUFactory, 795
  - CreateNCreatePDU
    - gdcmm::network::PDUFactory, 795
  - CreateNDeletePDU
    - gdcmm::network::PDUFactory, 795
  - CreateNEventReportPDU
    - gdcmm::network::PDUFactory, 795
  - CreateNGetPDU
    - gdcmm::network::PDUFactory, 796
  - CreateNSetPDU
    - gdcmm::network::PDUFactory, 796
  - CreateUniqueSeriesIdentifier
    - gdcmm::SerieHelper, 973
- Cross
    - gdcmm::DirectionCosines, 383
  - CrossDot
    - gdcmm::DirectionCosines, 383
  - CryptoFactory
    - gdcmm::CryptoFactory, 274
  - CryptographicMessageSyntax
    - gdcmm::CryptographicMessageSyntax, 276
  - CryptoLib

- gdcm::CryptoFactory, [273](#)
- CS
  - gdcm::VR, [1284](#)
- CSAElement
  - gdcm::CSAElement, [280](#)
- CSAHeader
  - gdcm::CSAHeader, [288](#)
  - gdcm::DataSet, [339](#)
- CSAHeaderDict
  - gdcm::CSAHeaderDict, [293](#), [294](#)
- CSAHeaderDictEntry
  - gdcm::CSAHeaderDictEntry, [296](#)
- CSAHeaderType
  - gdcm::CSAHeader, [288](#)
- CSANonImageStorage
  - gdcm::MediaStorage, [677](#)
- CSComp
  - gdcm, [58](#)
- CSD
  - gdcm::SegmentHelper::BasicCodedEntry, [190](#)
- CStore
  - gdcm::CompositeNetworkFunctions, [268](#)
- CSV
  - gdcm::SegmentHelper::BasicCodedEntry, [190](#)
- CT\_private\_ELE
  - gdcm::TransferSyntax, [1131](#)
- CTDefinedProcedureProtocolStorage
  - gdcm::UIDs, [1170](#)
- CTImageStorage
  - gdcm::MediaStorage, [676](#)
  - gdcm::UIDs, [1164](#)
- CTPerformedProcedureProtocolStorage
  - gdcm::UIDs, [1170](#)
- Curve
  - gdcm::Curve, [304](#), [305](#)
  - vtkGDCMImageReader, [1306](#)
  - vtkGDCMImageReader2, [1321](#)
- Curves
  - gdcm::Pixmap, [820](#)
- CV
  - gdcm::SegmentHelper::BasicCodedEntry, [191](#)
- CXX
  - gdcm::Printer, [861](#)
- cyan
  - gdcm::terminal, [85](#)
- DA
  - gdcm::VR, [1284](#)
- DAComp
  - gdcm, [58](#)
- DataElement
  - gdcm::DataElement, [312](#), [313](#)
  - gdcm::Value, [1268](#)
- DataElementSet
  - gdcm::DataSet, [330](#)
- DataElementType
  - gdcm::ModuleEntry, [709](#)
- DataEvent
  - gdcm::DataEvent, [325](#), [326](#)
- DataField
  - gdcm::CSAElement, [285](#)
- DataPtr
  - gdcm::CSAElement, [280](#)
- DATASET\_FORMAT
  - gdcm::CSAHeader, [288](#)
- DataSetEvent
  - gdcm::DataSetEvent, [342](#)
- DataSetHandled
  - gdcm::network::ULConnectionCallback, [1233](#)
- DataSetHandles
  - gdcm::network::ULConnectionCallback, [1233](#)
- DataSetMS
  - gdcm::FileMetaInformation, [477](#)
- DataSetTS
  - gdcm::FileMetaInformation, [477](#)
- DataWasPassed
  - vtkImageMapToColors16, [1383](#)
- dCor
  - gdcm::MrProtocol::Vector3, [1270](#)
- DebugOff
  - gdcm::Trace, [1124](#)
- DebugOn
  - gdcm::Trace, [1124](#)
- Decode
  - gdcm::AudioCodec, [168](#)
  - gdcm::Base64, [169](#)
  - gdcm::Curve, [305](#)
  - gdcm::Decoder, [346](#)
  - gdcm::DeltaEncodingCodec, [353](#)
  - gdcm::ImageCodec, [550](#)
  - gdcm::JPEG2000Codec, [622](#)
  - gdcm::JPEGCodec, [634](#)
  - gdcm::JPEGLSCoDec, [642](#)
  - gdcm::JSON, [646](#)
  - gdcm::KAKADUCoDec, [650](#)
  - gdcm::LookupTable, [658](#)
  - gdcm::PDFCodec, [792](#)
  - gdcm::PVRGCodec, [876](#)
  - gdcm::RAWCodec, [896](#)
  - gdcm::RLECodec, [918](#)
- Decode8
  - gdcm::LookupTable, [658](#)
- DecodeByStreams
  - gdcm::Decoder, [346](#)
  - gdcm::ImageCodec, [550](#)
  - gdcm::JPEG12Codec, [614](#)
  - gdcm::JPEG16Codec, [617](#)
  - gdcm::JPEG2000Codec, [623](#)

- gdcmm::JPEG8Codec, 628
- gdcmm::JPEGCodec, 634
- gdcmm::RAWCodec, 897
- gdcmm::RLECodec, 918
- DecodeBytes
  - gdcmm::RAWCodec, 897
- DecodeExtent
  - gdcmm::JPEG2000Codec, 623
  - gdcmm::JPEGCodec, 634
  - gdcmm::JPEGLSCodec, 643
  - gdcmm::RLECodec, 919
- Decompress
  - gdcmm::Overlay, 767
- Decrypt
  - gdcmm::CAPICryptographicMessageSyntax, 235
  - gdcmm::CryptographicMessageSyntax, 276
  - gdcmm::OpenSSLCryptographicMessageSyntax, 753
  - gdcmm::OpenSSL7CryptographicMessageSyntax, 758
- DeepCopy
  - vtkRTStructSetProperties, 1402
- DEFAULT
  - gdcmm::CryptoFactory, 273
- Default
  - gdcmm::FileMetaInformation, 471
- DefinedProcedureProtocolInformationModelFIND
  - gdcmm::UIDs, 1170
- DefinedProcedureProtocolInformationModelGET
  - gdcmm::UIDs, 1170
- DefinedProcedureProtocolInformationModelMOVE
  - gdcmm::UIDs, 1170
- DefinedTerms
  - gdcmm::DefinedTerms, 347
- DefinePixelExtent
  - gdcmm::StreamImageReader, 1022
  - gdcmm::StreamImageWriter, 1027
- DefineProperBufferLength
  - gdcmm::StreamImageReader, 1022
  - gdcmm::StreamImageWriter, 1027
- DeflatedExplicitVRLittleEndian
  - gdcmm::TransferSyntax, 1131
  - gdcmm::UIDs, 1162
- DeformableSpatialRegistrationStorage
  - gdcmm::UIDs, 1165
- Defs
  - gdcmm::Defs, 348, 349
- DeleteDirectory
  - gdcmm::System, 1085
- DeltaEncodingCodec
  - gdcmm::DeltaEncodingCodec, 353
- Derive
  - gdcmm::FileDerivation, 463
- DES3\_CIPHER
  - gdcmm::CryptographicMessageSyntax, 276
- Description
  - gdcmm::ModuleEntry, 707
- DescriptionField
  - gdcmm::ModuleEntry, 709
- DetachedInterpretationManagementSOPClassRetired
  - gdcmm::UIDs, 1164
- DetachedPatientManagementMetaSOPClassRetired
  - gdcmm::UIDs, 1163
- DetachedPatientManagementSOPClass
  - gdcmm::MediaStorage, 677
- DetachedPatientManagementSOPClassRetired
  - gdcmm::UIDs, 1163
- DetachedResultsManagementMetaSOPClassRetired
  - gdcmm::UIDs, 1163
- DetachedResultsManagementSOPClassRetired
  - gdcmm::UIDs, 1163
- DetachedStudyManagementMetaSOPClassRetired
  - gdcmm::UIDs, 1163
- DetachedStudyManagementSOPClass
  - gdcmm::MediaStorage, 677
- DetachedStudyManagementSOPClassRetired
  - gdcmm::UIDs, 1163
- DetachedVisitManagementSOPClass
  - gdcmm::MediaStorage, 677
- DetachedVisitManagementSOPClassRetired
  - gdcmm::UIDs, 1163
- DetailSRStorageTrialRetired
  - gdcmm::UIDs, 1166
- DETECTOR
  - gdcmm::Spacing, 1014
- DetermineEventByPDU
  - gdcmm::network::PDUFactory, 796
- dicomAETitle
  - gdcmm::UIDs, 1167
- dicomApplicationCluster
  - gdcmm::UIDs, 1167
- DICOMApplicationContextName
  - gdcmm::UIDs, 1163
- dicomAssociationAcceptor
  - gdcmm::UIDs, 1167
- dicomAssociationInitiator
  - gdcmm::UIDs, 1167
- dicomAuthorizedNodeCertificateReference
  - gdcmm::UIDs, 1167
- dicomConfigurationRoot
  - gdcmm::UIDs, 1168
- DICOMContentMappingResource
  - gdcmm::UIDs, 1171
- DICOMControlledTerminology
  - gdcmm::UIDs, 1163
- dicomDescription
  - gdcmm::UIDs, 1167
- dicomDevice
  - gdcmm::UIDs, 1168

dicomDeviceName  
     gdcm::UIDs, [1167](#)  
 dicomDeviceSerialNumber  
     gdcm::UIDs, [1168](#)  
 dicomDevicesRoot  
     gdcm::UIDs, [1168](#)  
 DICOMDIR  
     gdcm::DICOMDIR, [354](#)  
 DICOMDIRGenerator  
     gdcm::DICOMDIRGenerator, [356](#)  
 dicomHostname  
     gdcm::UIDs, [1167](#)  
 dicomInstalled  
     gdcm::UIDs, [1167](#)  
 dicomInstitutionAddress  
     gdcm::UIDs, [1168](#)  
 dicomInstitutionDepartmentName  
     gdcm::UIDs, [1168](#)  
 dicomInstitutionName  
     gdcm::UIDs, [1168](#)  
 dicomIssuerOfPatientID  
     gdcm::UIDs, [1168](#)  
 dicomManufacturer  
     gdcm::UIDs, [1167](#)  
 dicomManufacturerModelName  
     gdcm::UIDs, [1167](#)  
 dicomNetworkAE  
     gdcm::UIDs, [1168](#)  
 dicomNetworkConnection  
     gdcm::UIDs, [1168](#)  
 dicomNetworkConnectionReference  
     gdcm::UIDs, [1167](#)  
 dicomPort  
     gdcm::UIDs, [1167](#)  
 dicomPreferredCalledAETitle  
     gdcm::UIDs, [1167](#)  
 dicomPreferredCallingAETitle  
     gdcm::UIDs, [1168](#)  
 dicomPrimaryDeviceType  
     gdcm::UIDs, [1167](#)  
 dicomRelatedDeviceReference  
     gdcm::UIDs, [1167](#)  
 dicomSoftwareVersion  
     gdcm::UIDs, [1167](#)  
 dicomSOPClass  
     gdcm::UIDs, [1167](#)  
 dicomStationName  
     gdcm::UIDs, [1168](#)  
 dicomSupportedCharacterSet  
     gdcm::UIDs, [1168](#)  
 dicomThisNodeCertificateReference  
     gdcm::UIDs, [1167](#)  
 dicomTLSCyphersuite  
     gdcm::UIDs, [1167](#)  
 dicomTransferCapability  
     gdcm::UIDs, [1168](#)  
 dicomTransferRole  
     gdcm::UIDs, [1167](#)  
 dicomTransferSyntax  
     gdcm::UIDs, [1167](#)  
 DICOMUIDRegistry  
     gdcm::UIDs, [1163](#)  
 dicomUniqueAETitle  
     gdcm::UIDs, [1168](#)  
 dicomUniqueAETitlesRegistryRoot  
     gdcm::UIDs, [1168](#)  
 dicomVendorData  
     gdcm::UIDs, [1167](#)  
 DICOS2DAITStorage  
     gdcm::UIDs, [1170](#)  
 DICOS3DAITStorage  
     gdcm::UIDs, [1170](#)  
 DICOSCTImageStorage  
     gdcm::UIDs, [1170](#)  
 DICOSDigitalXRayImageStorageForPresentation  
     gdcm::UIDs, [1170](#)  
 DICOSDigitalXRayImageStorageForProcessing  
     gdcm::UIDs, [1170](#)  
 DICOSQuadrupoleResonanceQRStorage  
     gdcm::UIDs, [1170](#)  
 DICOSThreatDetectionReportStorage  
     gdcm::UIDs, [1170](#)  
 Dict  
     gdcm::Dict, [360](#), [361](#)  
     gdcm::DictEntry, [372](#)  
 DICT\_DEBUG  
     gdcm::DictConverter, [365](#)  
 DICT\_DEFAULT  
     gdcm::DictConverter, [365](#)  
 DICT\_XML  
     gdcm::DictConverter, [365](#)  
 DictConverter  
     gdcm::DictConverter, [365](#)  
 DictEntry  
     gdcm::DictEntry, [369](#)  
 DictPrinter  
     gdcm::DictPrinter, [374](#)  
 Dicts  
     gdcm::CSAHeaderDict, [295](#)  
     gdcm::Dict, [363](#)  
     gdcm::Dicts, [377](#)  
     gdcm::PrivateDict, [867](#)  
 difference\_type  
     gdcm::CodeString, [253](#)  
     gdcm::LO, [652](#)  
     gdcm::String< TDelimiter, TMaxLength, TPadChar  
         >, [1043](#)  
 DigitalIntraoralXRayImageStorageForPresentation

- gdcmm::UIDs, [1164](#)
- DigitalIntraoralXRayImageStorageForPresentation
  - gdcmm::MediaStorage, [676](#)
- DigitalIntraoralXRayImageStorageForProcessing
  - gdcmm::MediaStorage, [676](#)
  - gdcmm::UIDs, [1164](#)
- DigitalMammographyImageStorageForPresentation
  - gdcmm::MediaStorage, [676](#)
- DigitalMammographyImageStorageForProcessing
  - gdcmm::MediaStorage, [676](#)
- DigitalMammographyXRayImageStorageForPresentation
  - gdcmm::UIDs, [1164](#)
- DigitalMammographyXRayImageStorageForProcessing
  - gdcmm::UIDs, [1164](#)
- DigitalXRayImageStorageForPresentation
  - gdcmm::MediaStorage, [676](#)
  - gdcmm::UIDs, [1164](#)
- DigitalXRayImageStorageForProcessing
  - gdcmm::MediaStorage, [676](#)
  - gdcmm::UIDs, [1164](#)
- dim
  - gdcmm::terminal, [85](#)
- Dimensions
  - gdcmm::Bitmap, [208](#)
  - gdcmm::ImageCodec, [557](#)
- DirCosTolerance
  - gdcmm::IPPSorter, [606](#)
- DirectionCosines
  - gdcmm::DirectionCosines, [383](#)
  - vtkGDCMImageReader, [1306](#)
  - vtkGDCMImageReader2, [1321](#)
- Directory
  - gdcmm::Directory, [387](#)
- DisplaySystemSOPClass
  - gdcmm::UIDs, [1169](#)
- DisplaySystemSOPInstance
  - gdcmm::UIDs, [1169](#)
- DoByteSwap
  - gdcmm::ImageCodec, [550](#)
- DolconImage
  - gdcmm::PixmapWriter, [830](#)
- DolInvertMonochrome
  - gdcmm::ImageCodec, [550](#)
- DoOverlayCleanup
  - gdcmm::ImageCodec, [551](#)
- DoPaddedCompositePixelCode
  - gdcmm::ImageCodec, [551](#)
- DoPlanarConfiguration
  - gdcmm::ImageCodec, [551](#)
- doround
  - gdcmm, [64](#)
- DoSimpleCopy
  - gdcmm::ImageCodec, [551](#)
- Dot
  - gdcmm::DirectionCosines, [384](#)
- DoYBR
  - gdcmm::ImageCodec, [551](#)
- DoYBRFull422
  - gdcmm::ImageCodec, [551](#)
- DropDuplicatePositions
  - gdcmm::IPPSorter, [607](#)
- DS
  - gdcmm::VR, [1284](#)
- dSag
  - gdcmm::MrProtocol::Vector3, [1270](#)
- DT
  - gdcmm::VR, [1284](#)
- DTComp
  - gdcmm, [58](#)
- dTra
  - gdcmm::MrProtocol::Vector3, [1270](#)
- Dumper
  - gdcmm::Dumper, [394](#)
- DuplicateAttributeError
  - gdcmm::Parser, [778](#)
- eAABORTPDUReturnedOpen
  - gdcmm::network, [81](#)
- eAABORTRequest
  - gdcmm::network, [81](#)
- eAASSOCIATE\_RQPDUReturned
  - gdcmm::network, [81](#)
- eAASSOCIATERequestLocalUser
  - gdcmm::network, [81](#)
- eAASSOCIATEResponseAccept
  - gdcmm::network, [81](#)
- eAASSOCIATEResponseReject
  - gdcmm::network, [81](#)
- eArabic
  - gdcmm, [61](#)
- eARELEASE\_RPPDUReturned
  - gdcmm::network, [81](#)
- eARELEASE\_RQPDUReturnedOpen
  - gdcmm::network, [81](#)
- eARELEASERequest
  - gdcmm::network, [81](#)
- eARELEASEResponse
  - gdcmm::network, [81](#)
- eARTIMTimerExpired
  - gdcmm::network, [81](#)
- eASSOCIATE\_ACPDUReturned
  - gdcmm::network, [81](#)
- eASSOCIATE\_RJPDUReturned
  - gdcmm::network, [81](#)
- ECG12leadWaveformStorage
  - gdcmm::UIDs, [1165](#)
- ECharSet
  - gdcmm, [61](#)

- eCreateMMPS
  - gdcm, [62](#)
- eCyrillic
  - gdcm, [61](#)
- EddyCurrentImageStorage
  - gdcm::UIDs, [1170](#)
- EddyCurrentMultiframeImageStorage
  - gdcm::UIDs, [1170](#)
- EDGE
  - gdcm::MeshPrimitive, [692](#)
- eEventDoesNotExist
  - gdcm::network, [81](#)
- EEventID
  - gdcm::network, [81](#)
- eFind
  - gdcm, [62](#)
- eGB18030
  - gdcm, [61](#)
- eGreek
  - gdcm, [61](#)
- eHebrew
  - gdcm, [61](#)
- eImage
  - gdcm, [62](#)
- eJapanese
  - gdcm, [61](#)
- eJapaneseKanjiMultibyte
  - gdcm, [61](#)
- eJapaneseSupplementaryKanjiMultibyte
  - gdcm, [61](#)
- eKoreanHangulHanjaMultibyte
  - gdcm, [61](#)
- eLatin1
  - gdcm, [61](#)
- eLatin2
  - gdcm, [61](#)
- eLatin3
  - gdcm, [61](#)
- eLatin4
  - gdcm, [61](#)
- eLatin5
  - gdcm, [61](#)
- elem
  - gdcm::SerieHelper, [974](#)
- Element
  - gdcm::Element< TVR, VM::VM1\_n >, [404](#)
- eMove
  - gdcm, [62](#)
- Empty
  - gdcm::Anonymizer, [114](#)
  - gdcm::BoxRegion, [215](#)
  - gdcm::DataElement, [313](#)
  - gdcm::FileAnonymizer, [452](#)
  - gdcm::Region, [909](#)
- EmptyMaskGenerator
  - gdcm::EmptyMaskGenerator, [423](#)
- EncapsulatedCDASStorage
  - gdcm::MediaStorage, [677](#)
  - gdcm::UIDs, [1166](#)
- EncapsulatedDocument
  - gdcm::EncapsulatedDocument, [425](#)
- EncapsulatedPDFStorage
  - gdcm::MediaStorage, [677](#)
  - gdcm::UIDs, [1166](#)
- EncapsulatedSTLStorage
  - gdcm::UIDs, [1170](#)
- Encode
  - gdcm::Base64, [169](#)
- EncodeBuffer
  - gdcm::JPEG12Codec, [615](#)
  - gdcm::JPEG16Codec, [618](#)
  - gdcm::JPEG8Codec, [628](#)
  - gdcm::JPEGCodec, [635](#)
- EncodeBytes
  - gdcm::System, [1086](#)
- Encrypt
  - gdcm::CAPICryptographicMessageSyntax, [235](#)
  - gdcm::CryptographicMessageSyntax, [276](#)
  - gdcm::OpenSSLCryptographicMessageSyntax, [753](#)
  - gdcm::OpenSSL7CryptographicMessageSyntax, [758](#)
- End
  - gdcm::CSAHeaderDict, [294](#)
  - gdcm::DataSet, [331](#), [332](#)
  - gdcm::Dict, [361](#)
  - gdcm::IODs, [601](#)
  - gdcm::Scanner, [927](#)
  - gdcm::SequenceOfFragments, [956](#), [957](#)
  - gdcm::SequenceOfItems, [965](#)
  - gdcm::StrictScanner, [1036](#)
- EndElement
  - gdcm::TableReader, [1097](#)
  - gdcm::XMLDictReader, [1422](#)
  - gdcm::XMLPrivateDictReader, [1429](#)
- EndElementHandler
  - gdcm::Parser, [777](#)
- EndFilter
  - gdcm::SimpleSubjectWatcher, [994](#)
- EndWith
  - gdcm::Filename, [479](#)
- EnhancedCTImageStorage
  - gdcm::MediaStorage, [676](#)
  - gdcm::UIDs, [1164](#)
- EnhancedMRColorImageStorage
  - gdcm::MediaStorage, [678](#)
  - gdcm::UIDs, [1171](#)
- EnhancedMRImageStorage
  - gdcm::MediaStorage, [677](#)

- gdcm::UIDs, [1164](#)
- EnhancedPETImageStorage
  - gdcm::MediaStorage, [678](#)
  - gdcm::UIDs, [1170](#)
- EnhancedSR
  - gdcm::MediaStorage, [677](#)
- EnhancedSRStorage
  - gdcm::UIDs, [1166](#)
- EnhancedUSVolumeStorage
  - gdcm::MediaStorage, [678](#)
  - gdcm::UIDs, [1168](#)
- EnhancedXAImageStorage
  - gdcm::MediaStorage, [678](#)
  - gdcm::UIDs, [1165](#)
- EnhancedXRFImageStorage
  - gdcm::UIDs, [1165](#)
- ENQueryType
  - gdcm, [61](#)
- EnumeratedValues
  - gdcm::EnumeratedValues, [430](#)
- ePatient
  - gdcm, [62](#)
- ePatientRootType
  - gdcm, [62](#)
- ePDATArequest
  - gdcm::network, [81](#)
- ePDATATFPDU
  - gdcm::network, [81](#)
- EQueryLevel
  - gdcm, [62](#)
- EQueryType
  - gdcm, [62](#)
- ERootType
  - gdcm, [62](#)
- ErrorOff
  - gdcm::Trace, [1125](#)
- ErrorOn
  - gdcm::Trace, [1125](#)
- ErrorType
  - gdcm::Parser, [777](#)
- eSeries
  - gdcm, [62](#)
- eSetMMPS
  - gdcm, [62](#)
- eSta10ReleaseCollisionAc
  - gdcm::network, [83](#)
- eSta11ReleaseCollisionRq
  - gdcm::network, [83](#)
- eSta12ReleaseCollisionAcLocal
  - gdcm::network, [83](#)
- eSta13AwaitingClose
  - gdcm::network, [83](#)
- eSta1Idle
  - gdcm::network, [83](#)
- eSta2Open
  - gdcm::network, [83](#)
- eSta3WaitLocalAssoc
  - gdcm::network, [83](#)
- eSta4LocalAssocDone
  - gdcm::network, [83](#)
- eSta5WaitRemoteAssoc
  - gdcm::network, [83](#)
- eSta6TransferReady
  - gdcm::network, [83](#)
- eSta7WaitRelease
  - gdcm::network, [83](#)
- eSta8WaitLocalRelease
  - gdcm::network, [83](#)
- eSta9ReleaseCollisionRqLocal
  - gdcm::network, [83](#)
- EstablishConnection
  - gdcm::network::ULConnectionManager, [1239](#)
- EstablishConnectionMove
  - gdcm::network::ULConnectionManager, [1239](#)
- eStaDoesNotExist
  - gdcm::network, [83](#)
- EStateID
  - gdcm::network, [81](#)
- eStudy
  - gdcm, [62](#)
- eStudyRootType
  - gdcm, [62](#)
- eThai
  - gdcm, [61](#)
- eTransportConnConfirmLocal
  - gdcm::network, [81](#)
- eTransportConnectionClosed
  - gdcm::network, [81](#)
- eTransportConnIndicLocal
  - gdcm::network, [81](#)
- eUnrecognizedPDURceived
  - gdcm::network, [81](#)
- eUTF8
  - gdcm, [61](#)
- Event
  - gdcm::Event, [434](#)
- eWLMFind
  - gdcm, [62](#)
- Exception
  - gdcm::Exception, [437](#)
- Execute
  - gdcm::Command, [259](#)
  - gdcm::EmptyMaskGenerator, [423](#)
  - gdcm::MemberCommand< T >, [687](#)
  - gdcm::SimpleMemberCommand< T >, [991](#)
- ExecuteData
  - vtkGDCMImageReader, [1297](#)
  - vtkGDCMThreadedImageReader, [1352](#)



- ExecuteInformation
  - vtkGDCMImageReader, [1298](#)
  - vtkGDCMThreadedImageReader, [1352](#)
- ExecuteQuery
  - gdcmm::StringFilter, [1048](#)
- Explicit
  - gdcmm::TransferSyntax, [1130](#)
- ExplicitVRBigEndian
  - gdcmm::TransferSyntax, [1131](#)
  - gdcmm::UIDs, [1162](#)
- ExplicitVRLittleEndian
  - gdcmm::TransferSyntax, [1131](#)
  - gdcmm::UIDs, [1162](#)
- Explore
  - gdcmm::Directory, [387](#)
- ExtensibleSRStorage
  - gdcmm::UIDs, [1170](#)
- Extract
  - gdcmm::IconImageFilter, [516](#)
- ExtractIconImages
  - gdcmm::IconImageFilter, [516](#)
- ExtractVeprolconImages
  - gdcmm::IconImageFilter, [516](#)
- F
  - gdcmm::Printer, [864](#)
  - gdcmm::Reader, [905](#)
  - gdcmm::Validate, [1265](#)
  - gdcmm::XMLPrinter, [1426](#)
- FACET
  - gdcmm::MeshPrimitive, [692](#)
- FallColorPaletteSOPInstance
  - gdcmm::UIDs, [1168](#)
- FD
  - gdcmm::VR, [1284](#)
- Fiducials
  - gdcmm::Fiducials, [445](#)
- File
  - gdcmm::File, [447](#)
- FileAnonymizer
  - gdcmm::FileAnonymizer, [452](#)
- FileChangeTransferSyntax
  - gdcmm::FileChangeTransferSyntax, [456](#)
  - gdcmm::ImageCodec, [557](#)
- FileDecompressLookupTable
  - gdcmm::FileDecompressLookupTable, [459](#)
- FileDerivation
  - gdcmm::FileDerivation, [462](#)
- FileExists
  - gdcmm::System, [1086](#)
- FileExplicitFilter
  - gdcmm::FileExplicitFilter, [466](#)
- FilesDirectory
  - gdcmm::System, [1086](#)
- FilesSymlink
  - gdcmm::System, [1086](#)
- FileList
  - gdcmm, [58](#)
- FileMetaInformation
  - gdcmm::FileMetaInformation, [470](#), [471](#)
- FileName
  - vtkGDCMPolyDataReader, [1341](#)
- Filename
  - gdcmm::Filename, [478](#)
- filename
  - gdcmm::FileWithName, [498](#)
- FileNameEvent
  - gdcmm::FileNameEvent, [483](#)
- FilenameGenerator
  - gdcmm::FilenameGenerator, [486](#)
- FileNameOrdering
  - gdcmm::SerieHelper, [973](#)
- FileNames
  - vtkGDCMImageReader, [1306](#)
- Filenames
  - gdcmm::Sorter, [1011](#)
- FilenamesType
  - gdcmm::DICOmdirGenerator, [356](#)
  - gdcmm::Directory, [387](#)
  - gdcmm::FilenameGenerator, [486](#)
- FilenameType
  - gdcmm::DICOmdirGenerator, [356](#)
  - gdcmm::Directory, [387](#)
  - gdcmm::FilenameGenerator, [486](#)
- FileSet
  - gdcmm::FileSet, [490](#)
- FileSize
  - gdcmm::System, [1087](#)
- FileStreamer
  - gdcmm::FileStreamer, [493](#)
- FileType
  - gdcmm::FileSet, [489](#)
- FileTime
  - gdcmm::System, [1087](#)
- FileType
  - gdcmm::FileSet, [490](#)
- FileWithName
  - gdcmm::FileWithName, [497](#)
- Fill
  - gdcmm::ByteValue, [226](#)
- FillFromDataSet
  - gdcmm::FileMetaInformation, [472](#)
- FillMedicalImageInformation
  - vtkGDCMImageReader, [1298](#)
  - vtkGDCMImageReader2, [1312](#)
  - vtkGDCMPolyDataReader, [1338](#)
- FindContext
  - gdcmm::network::ULConnection, [1227](#)



- FindCSAElementByName
  - gdcm::CSAHeader, [289](#)
- FindDataElement
  - gdcm::DataSet, [332](#)
  - gdcm::Item, [610](#)
  - gdcm::SequenceOfItems, [965](#)
- FindDictEntry
  - gdcm::PrivateDict, [865](#)
- FindMacroEntry
  - gdcm::Macro, [666](#)
- FindModuleEntryInMacros
  - gdcm::Module, [704](#)
- FindMrProtocolByName
  - gdcm::MrProtocol, [719](#)
- FindNextDataElement
  - gdcm::DataSet, [332](#)
- FindPatientRootQuery
  - gdcm::FindPatientRootQuery, [499](#)
- FindPDBElementByName
  - gdcm::PDBHeader, [789](#)
- FindStudyRootQuery
  - gdcm::FindStudyRootQuery, [502](#)
- FirstRender
  - vtkImageColorViewer, [1376](#)
- FL
  - gdcm::VR, [1284](#)
- FLOAT16
  - gdcm::PixelFormat, [808](#)
- FLOAT32
  - gdcm::PixelFormat, [808](#)
- FLOAT64
  - gdcm::PixelFormat, [808](#)
- ForceRescale
  - vtkGDCMImageReader, [1307](#)
  - vtkGDCMImageReader2, [1321](#)
- FormatDateTime
  - gdcm::System, [1087](#)
- Fragment
  - gdcm::Fragment, [506](#)
- FragmentVector
  - gdcm::SequenceOfFragments, [955](#)
- FromString
  - gdcm::StringFilter, [1048](#)
- FUJI
  - gdcm::EquipmentManufacturer, [431](#)
- FujiPrivateCRImageStorage
  - gdcm::MediaStorage, [678](#)
- FujiPrivateMammoCRImageStorage
  - gdcm::MediaStorage, [678](#)
- gdcm, [43](#)
  - add1, [63](#)
  - AESComp, [58](#)
  - ASComp, [58](#)
  - backslash, [63](#)
  - BOOL\_FUNCTION\_PFILE\_PFILE\_POINTER, [58](#)
  - Clamp, [63](#)
  - clean, [63](#)
  - CompOperators, [60](#)
  - CSComp, [58](#)
  - DAComp, [58](#)
  - doround, [64](#)
  - DTComp, [58](#)
  - eArabic, [61](#)
  - ECharSet, [61](#)
  - eCreateMMPS, [62](#)
  - eCyrillic, [61](#)
  - eFind, [62](#)
  - eGB18030, [61](#)
  - eGreek, [61](#)
  - eHebrew, [61](#)
  - eImage, [62](#)
  - eJapanese, [61](#)
  - eJapaneseKanjiMultibyte, [61](#)
  - eJapaneseSupplementaryKanjiMultibyte, [61](#)
  - eKoreanHangulHanjaMultibyte, [61](#)
  - eLatin1, [61](#)
  - eLatin2, [61](#)
  - eLatin3, [61](#)
  - eLatin4, [61](#)
  - eLatin5, [61](#)
  - eMove, [62](#)
  - ENQueryType, [61](#)
  - ePatient, [62](#)
  - ePatientRootType, [62](#)
  - EQueryLevel, [62](#)
  - EQueryType, [62](#)
  - ERootType, [62](#)
  - eSeries, [62](#)
  - eSetMMPS, [62](#)
  - eStudy, [62](#)
  - eStudyRootType, [62](#)
  - eThai, [61](#)
  - eUTF8, [61](#)
  - eWLMFind, [62](#)
  - FileList, [58](#)
  - GDCM\_DIFFERENT, [61](#)
  - GDCM\_EQUAL, [61](#)
  - GDCM\_GREATER, [61](#)
  - GDCM\_GREATEROREQUAL, [61](#)
  - GDCM\_LESS, [61](#)
  - GDCM\_LESSOREQUAL, [61](#)
  - GetVRFromTag, [64](#)
  - GlobalInstance, [76](#)
  - IconImage, [59](#)
  - LD\_ALL, [63](#)
  - LD\_NOSEQ, [63](#)
  - LD\_NOSHADOW, [63](#)

- LD\_NOSHADOWSEQ, 63
- LOComp, 59
- LodModeType, 62
- LTComp, 59
- MacroEntry, 59
- NestedMacroEntries, 59
- operator!=, 64
- operator<<, 64–74
- operator>>, 74, 75
- operator==, 74
- PNComp, 59
- Round, 75
- roundat, 75
- SHComp, 59
- STComp, 60
- TMComp, 60
- TYPETOENCODING, 75
- UCComp, 60
- UIComp, 60
- URComp, 60
- UTComp, 60
- VRBINARY, 76
- x16printf, 76
- gdcmm::AbortEvent, 103
- gdcmm::AnonymizeEvent, 107
  - ~AnonymizeEvent, 109
  - AnonymizeEvent, 108, 109
  - CheckEvent, 109
  - GetEventName, 109
  - GetTag, 109
  - MakeObject, 109
  - operator=, 110
  - Self, 108
  - SetTag, 110
  - Superclass, 108
- gdcmm::Anonymizer, 110
  - ~Anonymizer, 113
  - Anonymizer, 113
  - BALCPPProtect, 113
  - BasicApplicationLevelConfidentialityProfile, 113
  - CanEmptyTag, 114
  - ClearInternalUIDs, 114
  - Empty, 114
  - GetBasicApplicationLevelConfidentialityProfileAttributes, 114
  - GetCryptographicMessageSyntax, 115
  - GetFile, 115
  - New, 115
  - RecurseDataSet, 115
  - Remove, 115
  - RemoveGroupLength, 115
  - RemovePrivateTags, 116
  - RemoveRetired, 116
  - Replace, 116
  - SetCryptographicMessageSyntax, 117
  - SetFile, 117
- gdcmm::AnyEvent, 118
- gdcmm::ApplicationEntity, 121
  - Internal, 123
  - IsValid, 122
  - MaxLength, 123
  - MaxNumberOfComponents, 123
  - Padding, 123
  - Print, 122
  - Separator, 123
  - SetBlob, 122
  - Squeeze, 123
- gdcmm::ASN1, 130
  - ~ASN1, 131
  - ASN1, 131
  - operator=, 132
  - ParseDump, 132
  - ParseDumpFile, 132
  - TestPBKDF2, 132
- gdcmm::Attribute< Group, Element, TVR, TVM >, 134
  - ArrayType, 136
  - GDCM\_STATIC\_ASSERT, 137
  - GetAsDataElement, 137
  - GetDictVM, 137
  - GetDictVR, 138
  - GetNumberOfValues, 138
  - GetTag, 138
  - GetValue, 138, 139
  - GetValues, 139
  - GetVM, 139
  - GetVR, 139
  - Internal, 143
  - operator!=, 140
  - operator<, 140
  - operator==, 140
  - operator[], 140
  - Print, 141
  - Set, 141
  - SetByteValue, 141
  - SetByteValueNoSwap, 141
  - SetFromDataElement, 142
  - SetFromDataSet, 142
  - SetValue, 142
  - SetValues, 143
  - VMType, 137
- gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 144
  - ArrayType, 145
  - GDCM\_STATIC\_ASSERT, 146
  - GetAsDataElement, 146
  - GetDictVM, 147
  - GetDictVR, 147
  - GetNumberOfValues, 147
  - GetTag, 147

- GetValue, [147](#)
- GetValues, [148](#)
- GetVM, [148](#)
- GetVR, [148](#)
- Internal, [150](#)
- operator!=, [148](#)
- operator<, [148](#)
- operator==, [148](#)
- Print, [149](#)
- Set, [149](#)
- SetByteValue, [149](#)
- SetByteValueNoSwap, [149](#)
- SetFromDataElement, [149](#)
- SetFromDataSet, [150](#)
- SetValue, [150](#)
- VMType, [146](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1\_3 >, [151](#)
  - GetVM, [151](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1\_8 >, [152](#)
  - GetVM, [153](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, [153](#)
  - ~Attribute, [155](#)
  - ArrayType, [154](#)
  - Attribute, [155](#)
  - GDCM\_STATIC\_ASSERT, [155](#)
  - GetAsDataElement, [155](#)
  - GetDictVM, [156](#)
  - GetDictVR, [156](#)
  - GetNumberOfValues, [156](#)
  - GetTag, [156](#)
  - GetValue, [156](#)
  - GetValues, [157](#)
  - GetVM, [157](#)
  - GetVR, [157](#)
  - operator[], [157](#)
  - Print, [157](#)
  - Set, [158](#)
  - SetByteValue, [158](#)
  - SetFromDataElement, [158](#)
  - SetFromDataSet, [158](#)
  - SetNumberOfValues, [158](#)
  - SetValue, [159](#)
  - SetValues, [159](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM2\_2n >, [160](#)
  - GetVM, [161](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM2\_n >, [161](#)
  - GetVM, [162](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM3\_3n >, [163](#)
  - GetVM, [164](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM3\_n >, [164](#)
  - GetVM, [165](#)
- gdcmm::AudioCodec, [166](#)
  - ~AudioCodec, [167](#)
  - AudioCodec, [167](#)
  - CanCode, [167](#)
  - CanDecode, [167](#)
  - Decode, [168](#)
- gdcmm::Base64, [168](#)
  - Base64, [169](#)
  - Decode, [169](#)
  - Encode, [169](#)
  - GetDecodeLength, [170](#)
  - GetEncodeLength, [170](#)
  - operator=, [170](#)
- gdcmm::BaseQuery, [177](#)
  - ~BaseQuery, [179](#)
  - AddQueryDataSet, [179](#)
  - BaseQuery, [179](#)
  - GetAbstractSyntaxUID, [180](#)
  - GetQueryDataSet, [180](#)
  - GetSOPInstanceUID, [180](#)
  - mDataSet, [182](#)
  - mSopInstanceUID, [182](#)
  - Print, [180](#)
  - QueryFactory, [182](#)
  - SetSearchParameter, [180](#), [181](#)
  - SetSOPInstanceUID, [181](#)
  - ValidateQuery, [181](#)
  - ValidDataSet, [181](#)
  - WriteHelpFile, [182](#)
  - WriteQuery, [182](#)
- gdcmm::BaseRootQuery, [183](#)
  - ~BaseRootQuery, [185](#)
  - BaseRootQuery, [185](#)
  - Construct, [185](#)
  - GetQueryLevelFromQueryRoot, [185](#)
  - GetQueryLevelFromString, [185](#)
  - GetQueryLevelString, [185](#)
  - GetTagListByLevel, [186](#)
  - InitializeDataSet, [186](#)
  - mHelpDescription, [187](#)
  - mImage, [187](#)
  - mPatient, [187](#)
  - mRootType, [187](#)
  - mSeries, [187](#)
  - mStudy, [187](#)
  - QueryFactory, [187](#)
  - ValidateQuery, [186](#)
- gdcmm::BasicOffsetTable, [191](#)
  - BasicOffsetTable, [192](#)
  - operator<<, [193](#)

Read, [193](#)  
gdcmm::Bitmap, [194](#)  
    ~Bitmap, [197](#)  
    AreOverlaysInPixelData, [197](#)  
    Bitmap, [197](#)  
    Clear, [197](#)  
    ComputeLossyFlag, [197](#)  
    Dimensions, [208](#)  
    GetBuffer, [198](#)  
    GetBuffer2, [198](#)  
    GetBufferLength, [198](#)  
    GetColumns, [198](#)  
    GetDataElement, [198](#), [199](#)  
    GetDimension, [199](#)  
    GetDimensions, [199](#)  
    GetLUT, [199](#)  
    GetNeedByteSwap, [200](#)  
    GetNumberOfDimensions, [200](#)  
    GetPhotometricInterpretation, [200](#)  
    GetPixelFormat, [200](#), [201](#)  
    GetPlanarConfiguration, [201](#)  
    GetRows, [201](#)  
    GetTransferSyntax, [201](#)  
    ImageChangeTransferSyntax, [207](#)  
    IsEmpty, [201](#)  
    IsLossy, [202](#)  
    IsTransferSyntaxCompatible, [202](#)  
    LossyFlag, [208](#)  
    LUT, [208](#)  
    LUTPtr, [197](#)  
    NeedByteSwap, [208](#)  
    NumberOfDimensions, [208](#)  
    PF, [208](#)  
    PI, [208](#)  
    PixelData, [209](#)  
    PixmapReader, [207](#)  
    PlanarConfiguration, [209](#)  
    Print, [202](#)  
    SetColumns, [202](#)  
    SetDataElement, [202](#)  
    SetDimension, [203](#)  
    SetDimensions, [203](#)  
    SetLossyFlag, [203](#)  
    SetLUT, [203](#)  
    SetNeedByteSwap, [204](#)  
    SetNumberOfDimensions, [204](#)  
    SetPhotometricInterpretation, [204](#)  
    SetPixelFormat, [204](#)  
    SetPlanarConfiguration, [205](#)  
    SetRows, [205](#)  
    SetTransferSyntax, [205](#)  
    TryJPEG2000Codec, [205](#)  
    TryJPEG2000Codec2, [206](#)  
    TryJPEGCodec, [206](#)  
    TryJPEGCodec2, [206](#)  
    TryJPEGLSCodec, [206](#)  
    TryKAKADUCodec, [206](#)  
    TryPVRGCodec, [206](#)  
    TryRAWCodec, [207](#)  
    TryRLECodec, [207](#)  
    TS, [209](#)  
    UnusedBitsPresentInPixelData, [207](#)  
gdcmm::BitmapToBitmapFilter, [209](#)  
    ~BitmapToBitmapFilter, [211](#)  
    BitmapToBitmapFilter, [211](#)  
    GetOutput, [211](#)  
    GetOutputAsBitmap, [211](#)  
    Input, [212](#)  
    Output, [212](#)  
    SetInput, [211](#)  
gdcmm::BoxRegion, [212](#)  
    ~BoxRegion, [214](#)  
    Area, [214](#)  
    BoundingBox, [214](#)  
    BoxRegion, [214](#)  
    Clone, [215](#)  
    ComputeBoundingBox, [215](#)  
    Empty, [215](#)  
    GetXMax, [215](#)  
    GetXMin, [215](#)  
    GetYMax, [216](#)  
    GetYMin, [216](#)  
    GetZMax, [216](#)  
    GetZMin, [216](#)  
    IsValid, [216](#)  
    operator=, [216](#)  
    Print, [217](#)  
    SetDomain, [217](#)  
gdcmm::ByteBuffer, [217](#)  
    ByteBuffer, [218](#)  
    Get, [218](#)  
    GetStart, [218](#)  
    ShiftEnd, [218](#)  
    UpdatePosition, [218](#)  
gdcmm::ByteSwap< T >, [219](#)  
    Swap, [219](#)  
    SwapFromSwapCodeIntoSystem, [220](#)  
    SwapRange, [220](#)  
    SwapRangeFromSwapCodeIntoSystem, [220](#)  
    SystemIsBigEndian, [220](#)  
    SystemIsLittleEndian, [221](#)  
gdcmm::ByteSwapFilter, [221](#)  
    ~ByteSwapFilter, [222](#)  
    ByteSwap, [222](#)  
    ByteSwapFilter, [221](#), [222](#)  
    operator=, [222](#)  
    SetByteSwapTag, [222](#)  
gdcmm::ByteValue, [223](#)

- ~ByteValue, [225](#)
- Append, [226](#)
- ByteValue, [225](#)
- Clear, [226](#)
- ComputeLength, [226](#)
- Fill, [226](#)
- GetBuffer, [226](#)
- GetLength, [226](#)
- GetPointer, [227](#)
- GetVoidPointer, [227](#)
- IsEmpty, [228](#)
- IsPrintable, [228](#)
- operator const std::vector< char > &, [228](#)
- operator=, [228](#)
- operator==, [228](#), [229](#)
- Print, [229](#)
- PrintASCII, [229](#)
- PrintASCIIXML, [229](#)
- PrintGroupLength, [229](#)
- PrintHex, [229](#)
- PrintHexXML, [230](#)
- PrintPNXML, [230](#)
- Read, [230](#)
- SetLength, [230](#)
- SetLengthOnly, [230](#)
- Write, [231](#)
- WriteBuffer, [231](#)
- gdcmm::CAPICryptoFactory, [232](#)
  - CAPICryptoFactory, [232](#)
  - CreateCMSProvider, [233](#)
- gdcmm::CAPICryptographicMessageSyntax, [233](#)
  - ~CAPICryptographicMessageSyntax, [234](#)
  - CAPICryptographicMessageSyntax, [234](#)
  - Decrypt, [235](#)
  - Encrypt, [235](#)
  - GetCipherType, [235](#)
  - GetInitialized, [235](#)
  - ParseCertificateFile, [236](#)
  - ParseKeyFile, [236](#)
  - SetCipherType, [236](#)
  - SetPassword, [236](#)
- gdcmm::Codec, [249](#)
- gdcmm::Coder, [250](#)
  - ~Coder, [250](#)
  - CanCode, [251](#)
  - Code, [251](#)
  - InternalCode, [251](#)
- gdcmm::CodeString, [252](#)
  - CodeString, [254](#), [255](#)
  - const\_iterator, [253](#)
  - const\_reference, [253](#)
  - const\_reverse\_iterator, [253](#)
  - difference\_type, [253](#)
  - GetAsString, [255](#)
  - IsValid, [255](#)
  - iterator, [253](#)
  - operator!=, [256](#)
  - operator<<, [256](#)
  - operator==, [256](#)
  - pointer, [254](#)
  - reference, [254](#)
  - reverse\_iterator, [254](#)
  - Size, [256](#)
  - size\_type, [254](#)
  - TrimInternal, [256](#)
  - value\_type, [254](#)
- gdcmm::Command, [257](#)
  - ~Command, [259](#)
  - Command, [258](#), [259](#)
  - Execute, [259](#)
  - operator=, [259](#)
- gdcmm::CommandDataSet, [260](#)
  - ~CommandDataSet, [261](#)
  - CommandDataSet, [261](#)
  - Insert, [261](#)
  - operator<<, [262](#)
  - Read, [262](#)
  - Replace, [262](#)
  - Write, [262](#)
- gdcmm::CompositeNetworkFunctions, [264](#)
  - CEcho, [266](#)
  - CFind, [266](#)
  - CMove, [267](#)
  - ConstructQuery, [267](#), [268](#)
  - CStore, [268](#)
  - KeyValuePairArrayType, [265](#)
  - KeyValuePairType, [265](#)
- gdcmm::ConstCharWrapper, [269](#)
  - ConstCharWrapper, [269](#)
  - operator const char \*, [269](#)
- gdcmm::CP246ExplicitDataElement, [270](#)
  - GetLength, [271](#)
  - Read, [271](#)
  - ReadPreValue, [271](#)
  - ReadValue, [271](#)
  - ReadWithLength, [272](#)
- gdcmm::CryptoFactory, [272](#)
  - ~CryptoFactory, [274](#)
  - CAPI, [273](#)
  - CreateCMSProvider, [274](#)
  - CryptoFactory, [274](#)
  - CryptoLib, [273](#)
  - DEFAULT, [273](#)
  - GetFactoryInstance, [274](#)
  - OPENSSL, [273](#)
  - OPENSSL7, [273](#)
- gdcmm::CryptographicMessageSyntax, [275](#)
  - ~CryptographicMessageSyntax, [276](#)

- AES128\_CIPHER, 276
- AES192\_CIPHER, 276
- AES256\_CIPHER, 276
- CipherTypes, 275
- CryptographicMessageSyntax, 276
- Decrypt, 276
- DES3\_CIPHER, 276
- Encrypt, 276
- GetCipherType, 277
- operator=, 277
- ParseCertificateFile, 277
- ParseKeyFile, 277
- SetCipherType, 277
- SetPassword, 278
- gdcmm::CSAElement, 278
  - CSAElement, 280
  - DataField, 285
  - DataPtr, 280
  - GetByteValue, 281
  - GetKey, 281
  - GetName, 281
  - GetNoOfItems, 281
  - GetSyngoDT, 281
  - GetValue, 282
  - GetVM, 282
  - GetVR, 282
  - IsEmpty, 282
  - KeyField, 285
  - NameField, 285
  - NoOfItemsField, 285
  - operator<, 283
  - operator<<, 285
  - operator=, 283
  - operator==, 283
  - SetByteValue, 283
  - SetKey, 283
  - SetName, 284
  - SetNoOfItems, 284
  - SetSyngoDT, 284
  - SetValue, 284
  - SetVM, 284
  - SetVR, 284
  - SyngoDTField, 286
  - ValueMultiplicityField, 286
  - VRField, 286
- gdcmm::CSAHeader, 286
  - ~CSAHeader, 289
  - CSAHeader, 288
  - CSAHeaderType, 288
  - DATASET\_FORMAT, 288
  - FindCSAElementByName, 289
  - GetCSADatInfo, 289
  - GetCSAEEnd, 289
  - GetCSAElementByName, 289
  - GetCSAImageHeaderInfoTag, 290
  - GetCSASeriesHeaderInfoTag, 290
  - GetDataSet, 290
  - GetFormat, 291
  - GetInterfile, 291
  - GetMrProtocol, 291
  - INTERFILE, 288
  - LoadFromDataElement, 291
  - NOMAGIC, 288
  - operator<<, 292
  - Print, 291
  - SV10, 288
  - UNKNOWN, 288
  - ZEROED\_OUT, 288
- gdcmm::CSAHeaderDict, 292
  - AddCSAHeaderDictEntry, 294
  - Begin, 294
  - ConstIterator, 293
  - CSAHeaderDict, 293, 294
  - Dicts, 295
  - End, 294
  - GetCSAHeaderDictEntry, 294
  - IsEmpty, 294
  - Iterator, 293
  - LoadDefault, 295
  - MapCSAHeaderDictEntry, 293
  - operator<<, 295
  - operator=, 295
- gdcmm::CSAHeaderDictEntry, 295
  - CSAHeaderDictEntry, 296
  - GetDescription, 297
  - GetName, 297
  - GetVM, 297
  - GetVR, 297
  - operator<, 297
  - operator<<, 298
  - SetDescription, 298
  - SetName, 298
  - SetVM, 298
  - SetVR, 298
- gdcmm::CSAHeaderDictException, 299
- gdcmm::Curve, 303
  - ~Curve, 305
  - Curve, 304, 305
  - Decode, 305
  - GetAsPoints, 305
  - GetCurveDataDescriptor, 305
  - GetDataValueRepresentation, 305
  - GetDimensions, 306
  - GetGroup, 306
  - GetNumberOfCurves, 306
  - GetNumberOfPoints, 306
  - GetTypeOfData, 306
  - GetTypeOfDataDescription, 306

- IsEmpty, [306](#)
- Print, [307](#)
- SetCoordinateStartValue, [307](#)
- SetCoordinateStepValue, [307](#)
- SetCurve, [307](#)
- SetCurveDataDescriptor, [307](#)
- SetCurveDescription, [307](#)
- SetDataValueRepresentation, [308](#)
- SetDimensions, [308](#)
- SetGroup, [308](#)
- SetNumberOfPoints, [308](#)
- SetTypeOfData, [308](#)
- Update, [308](#)
- gdcmm::DataElement, [309](#)
  - Clear, [313](#)
  - DataElement, [312](#), [313](#)
  - Empty, [313](#)
  - GetByteValue, [313](#)
  - GetLength, [313](#)
  - GetSequenceOfFragments, [314](#)
  - GetTag, [314](#)
  - GetValue, [314](#), [315](#)
  - GetValueAsSQ, [315](#)
  - GetVL, [315](#), [316](#)
  - GetVR, [316](#)
  - IsEmpty, [316](#)
  - IsUndefinedLength, [316](#)
  - operator<, [317](#)
  - operator<=, [321](#)
  - operator=, [317](#)
  - operator==, [317](#)
  - Read, [317](#)
  - ReadOrSkip, [317](#)
  - ReadPreValue, [318](#)
  - ReadValue, [318](#)
  - ReadValueWithLength, [318](#)
  - ReadWithLength, [318](#)
  - SetByteValue, [318](#)
  - SetTag, [319](#)
  - SetValue, [319](#)
  - SetValueFieldLength, [320](#)
  - SetVL, [320](#)
  - SetVLToUndefined, [320](#)
  - SetVR, [320](#)
  - TagField, [321](#)
  - ValueField, [322](#)
  - ValueLengthField, [322](#)
  - ValuePtr, [312](#)
  - VRField, [322](#)
  - Write, [321](#)
- gdcmm::DataElementException, [323](#)
- gdcmm::DataEvent, [323](#)
  - ~DataEvent, [325](#)
  - CheckEvent, [326](#)
  - DataEvent, [325](#), [326](#)
  - GetData, [326](#)
  - GetDataLength, [326](#)
  - GetEventName, [326](#)
  - MakeObject, [326](#)
  - operator=, [327](#)
  - Self, [325](#)
  - SetData, [327](#)
  - Superclass, [325](#)
- gdcmm::DataSet, [327](#)
  - Begin, [331](#)
  - Clear, [331](#)
  - ComputeDataElement, [331](#)
  - ComputeGroupLength, [331](#)
  - ConstIterator, [330](#)
  - CSAHeader, [339](#)
  - DataElementSet, [330](#)
  - End, [331](#), [332](#)
  - FindDataElement, [332](#)
  - FindNextDataElement, [332](#)
  - GetDataElement, [333](#)
  - GetDEEnd, [333](#)
  - GetDES, [334](#)
  - GetLength, [334](#)
  - GetMediaStorage, [334](#)
  - GetPrivateCreator, [334](#)
  - Insert, [334](#)
  - InsertDataElement, [335](#)
  - IsEmpty, [335](#)
  - Iterator, [330](#)
  - operator<=, [339](#)
  - operator(), [335](#)
  - operator=, [335](#)
  - operator[], [336](#)
  - Print, [336](#)
  - Read, [336](#)
  - ReadNested, [336](#)
  - ReadSelectedPrivateTags, [336](#)
  - ReadSelectedPrivateTagsWithLength, [337](#)
  - ReadSelectedTags, [337](#)
  - ReadSelectedTagsWithLength, [337](#)
  - ReadUpToTag, [337](#)
  - ReadUpToTagWithLength, [337](#)
  - ReadWithLength, [338](#)
  - Remove, [338](#)
  - Replace, [338](#)
  - ReplaceEmpty, [338](#)
  - Size, [339](#)
  - SizeType, [330](#)
  - Write, [339](#)
- gdcmm::DataSetEvent, [340](#)
  - ~DataSetEvent, [342](#)
  - CheckEvent, [342](#)
  - DataSetEvent, [342](#)

- GetDataSet, [343](#)
- GetEventName, [343](#)
- m\_DataSet, [343](#)
- MakeObject, [343](#)
- operator=, [343](#)
- Self, [342](#)
- Superclass, [342](#)
- gdcmm::DataSetHelper, [344](#)
  - ComputeVR, [344](#)
- gdcmm::Decoder, [345](#)
  - ~Decoder, [345](#)
  - CanDecode, [346](#)
  - Decode, [346](#)
  - DecodeByStreams, [346](#)
- gdcmm::DefinedTerms, [347](#)
  - DefinedTerms, [347](#)
- gdcmm::Defs, [347](#)
  - ~Defs, [349](#)
  - Defs, [348](#), [349](#)
  - GetIODFromFile, [349](#)
  - GetIODNameFromMediaStorage, [349](#)
  - GetIODs, [349](#)
  - GetMacros, [350](#)
  - GetModules, [350](#)
  - GetTypeFromTag, [350](#)
  - Global, [351](#)
  - IsEmpty, [350](#)
  - LoadDefaults, [350](#)
  - LoadFromFile, [351](#)
  - operator=, [351](#)
  - Verify, [351](#)
- gdcmm::DeltaEncodingCodec, [352](#)
  - ~DeltaEncodingCodec, [353](#)
  - CanDecode, [353](#)
  - Decode, [353](#)
  - DeltaEncodingCodec, [353](#)
- gdcmm::DICOMDIR, [354](#)
  - DICOMDIR, [354](#)
- gdcmm::DICOMDIRGenerator, [355](#)
  - ~DICOMDIRGenerator, [356](#)
  - AddImageDirectoryRecord, [357](#)
  - AddPatientDirectoryRecord, [357](#)
  - AddSeriesDirectoryRecord, [357](#)
  - AddStudyDirectoryRecord, [357](#)
  - DICOMDIRGenerator, [356](#)
  - FileNamesType, [356](#)
  - FilenameType, [356](#)
  - Generate, [357](#)
  - GetFile, [357](#)
  - GetScanner, [358](#)
  - SetDescriptor, [358](#)
  - SetFile, [358](#)
  - SetFileNames, [358](#)
  - SetRootDirectory, [358](#)
- gdcmm::Dict, [359](#)
  - AddDictEntry, [361](#)
  - Begin, [361](#)
  - ConstIterator, [360](#)
  - Dict, [360](#), [361](#)
  - Dicts, [363](#)
  - End, [361](#)
  - GetDictEntry, [361](#)
  - GetDictEntryByKeyword, [362](#)
  - GetDictEntryByName, [362](#)
  - GetKeywordFromTag, [362](#)
  - IsEmpty, [362](#)
  - Iterator, [360](#)
  - LoadDefault, [363](#)
  - MapDictEntry, [360](#)
  - operator<<, [363](#)
  - operator=, [363](#)
- gdcmm::DictConverter, [363](#)
  - ~DictConverter, [365](#)
  - AddGroupLength, [365](#)
  - Convert, [365](#)
  - ConvertToCXX, [366](#)
  - ConvertToXML, [366](#)
  - DICT\_DEBUG, [365](#)
  - DICT\_DEFAULT, [365](#)
  - DICT\_XML, [365](#)
  - DictConverter, [365](#)
  - GetDictName, [366](#)
  - GetInputFilename, [366](#)
  - GetOutputFilename, [366](#)
  - GetOutputType, [366](#)
  - OutputTypes, [365](#)
  - Readuint16, [366](#)
  - ReadVM, [367](#)
  - ReadVR, [367](#)
  - SetDictName, [367](#)
  - SetInputFileName, [367](#)
  - SetOutputFileName, [367](#)
  - SetOutputType, [367](#)
  - WriteFooter, [368](#)
  - WriteHeader, [368](#)
- gdcmm::DictEntry, [368](#)
  - Dict, [372](#)
  - DictEntry, [369](#)
  - GetKeyword, [369](#)
  - GetName, [370](#)
  - GetRetired, [370](#)
  - GetVM, [370](#)
  - GetVR, [370](#)
  - IsUnique, [371](#)
  - operator<<, [372](#)
  - SetElementXX, [371](#)
  - SetGroupXX, [371](#)
  - SetKeyword, [371](#)



- SetName, [371](#)
- SetRetired, [372](#)
- SetVM, [372](#)
- SetVR, [372](#)
- gdcm::DictPrinter, [373](#)
  - ~DictPrinter, [375](#)
  - DictPrinter, [374](#)
  - Print, [375](#)
  - PrintDataElement2, [375](#)
  - PrintDataSet2, [375](#)
- gdcm::Dicts, [376](#)
  - ~Dicts, [377](#)
  - ConstructorType, [377](#)
  - Dicts, [377](#)
  - GEMS, [377](#)
  - GetConstructorString, [378](#)
  - GetCSAHeaderDict, [378](#)
  - GetDictEntry, [378](#)
  - GetPrivateDict, [378](#), [379](#)
  - GetPublicDict, [379](#)
  - Global, [379](#)
  - IsEmpty, [379](#)
  - LoadDefaults, [379](#)
  - operator<<, [380](#)
  - operator=, [379](#)
  - PHILIPS, [377](#)
  - SIEMENS, [377](#)
- gdcm::DirectionCosines, [382](#)
  - ~DirectionCosines, [383](#)
  - ComputeDistAlongNormal, [383](#)
  - Cross, [383](#)
  - CrossDot, [383](#)
  - DirectionCosines, [383](#)
  - Dot, [384](#)
  - IsValid, [384](#)
  - Normalize, [384](#)
  - operator const double \*, [385](#)
  - Print, [385](#)
  - SetFromString, [385](#)
- gdcm::Directory, [385](#)
  - ~Directory, [387](#)
  - Directory, [387](#)
  - Explore, [387](#)
  - FilenameType, [387](#)
  - FilenameType, [387](#)
  - GetDirectories, [388](#)
  - GetFileNames, [388](#)
  - GetToplevel, [388](#)
  - Load, [388](#)
  - operator<<, [389](#)
  - Print, [389](#)
- gdcm::DirectoryHelper, [390](#)
  - GetCTImageSeriesUIDs, [390](#)
  - GetFileNamesFromSeriesUIDs, [390](#)
- GetFrameOfReference, [391](#)
- GetMRImageSeriesUIDs, [391](#)
- GetRTStructSeriesUIDs, [391](#)
- GetSeriesUIDsBySOPClassUID, [391](#)
- GetSOPClassUID, [391](#)
- GetStringValueFromTag, [392](#)
- LoadImageFromFiles, [392](#)
- RetrieveSOPInstanceUIDFromIndex, [392](#)
- RetrieveSOPInstanceUIDFromZPosition, [392](#)
- gdcm::DummyValueGenerator, [392](#)
  - Generate, [393](#)
- gdcm::Dumper, [393](#)
  - ~Dumper, [395](#)
  - Dumper, [394](#)
- gdcm::Element< TVR, TVM >, [395](#)
  - GetAsDataElement, [397](#)
  - GetLength, [397](#)
  - GetValue, [398](#)
  - GetValues, [398](#)
  - GetVM, [398](#)
  - GetVR, [398](#)
  - Internal, [400](#)
  - operator[], [399](#)
  - Print, [399](#)
  - Read, [399](#)
  - Set, [399](#)
  - SetFromDataElement, [399](#)
  - SetNoSwap, [400](#)
  - SetValue, [400](#)
  - Type, [397](#)
  - Write, [400](#)
- gdcm::Element< TVR, VM::VM1\_2 >, [401](#)
  - Parent, [402](#)
  - SetLength, [402](#)
- gdcm::Element< TVR, VM::VM1\_n >, [403](#)
  - ~Element, [404](#)
  - Element, [404](#)
  - GetAsDataElement, [404](#)
  - GetLength, [405](#)
  - GetValue, [405](#)
  - GetVM, [405](#)
  - GetVR, [405](#)
  - operator=, [405](#)
  - operator[], [406](#)
  - Print, [406](#)
  - Read, [406](#)
  - Set, [406](#)
  - SetArray, [406](#)
  - SetFromDataElement, [407](#)
  - SetLength, [407](#)
  - SetNoSwap, [407](#)
  - SetValue, [407](#)
  - Type, [404](#)
  - Write, [407](#)

WriteASCII, 408  
 gdcmm::Element< TVR, VM::VM2\_2n >, 408  
   Parent, 409  
   SetLength, 409  
 gdcmm::Element< TVR, VM::VM2\_n >, 410  
   Parent, 411  
   SetLength, 411  
 gdcmm::Element< TVR, VM::VM3\_3n >, 412  
   Parent, 413  
   SetLength, 413  
 gdcmm::Element< TVR, VM::VM3\_n >, 414  
   Parent, 415  
   SetLength, 415  
 gdcmm::Element< VR::AS, VM::VM5 >, 416  
   GetLength, 416  
   Internal, 416  
   Print, 416  
 gdcmm::Element< VR::OB, VM::VM1 >, 417  
 gdcmm::Element< VR::OW, VM::VM1 >, 418  
 gdcmm::ElementDisableCombinations< TVR, TVM >, 420  
 gdcmm::ElementDisableCombinations< VR::OB, VM::VM1\_n  
   >, 421  
 gdcmm::ElementDisableCombinations< VR::OW, VM::VM1\_n  
   >, 421  
 gdcmm::EmptyMaskGenerator, 421  
   ~EmptyMaskGenerator, 423  
   EmptyMaskGenerator, 423  
   Execute, 423  
   SetInputDirectory, 423  
   SetOutputDirectory, 424  
   SetSOPClassUIDMode, 424  
   SOPClassUIDMode, 423  
   UseGrayscaleSecondaryImageStorage, 423  
   UseOriginalSOPClassUID, 423  
 gdcmm::EncapsulatedDocument, 424  
   EncapsulatedDocument, 425  
 gdcmm::EncodingImplementation< T >, 425  
 gdcmm::EncodingImplementation< VR::VRASCII >, 426  
   Read, 426  
   ReadComputeLength, 426  
   ReadNoSwap, 426  
   Write, 427  
 gdcmm::EncodingImplementation< VR::VRBINARY >, 427  
   Read, 428  
   ReadComputeLength, 428  
   ReadNoSwap, 428  
   Write, 428  
 gdcmm::EndEvent, 429  
 gdcmm::EnumeratedValues, 430  
   EnumeratedValues, 430  
 gdcmm::EquipmentManufacturer, 430  
   Compute, 432  
   FUJI, 431  
   GEMS, 431  
   HITACHI, 431  
   KODAK, 431  
   MARCONI, 431  
   PMS, 431  
   SIEMENS, 431  
   TOSHIBA, 431  
   Type, 431  
   UNKNOWN, 431  
 gdcmm::Event, 432  
   ~Event, 434  
   CheckEvent, 434  
   Event, 434  
   GetEventName, 434  
   MakeObject, 435  
   operator=, 435  
   Print, 435  
 gdcmm::Exception, 436  
   ~Exception, 437  
   Exception, 437  
   GetDescription, 437  
   what, 437  
 gdcmm::ExitEvent, 438  
 gdcmm::ExplicitDataElement, 439  
   GetLength, 441  
   Read, 441  
   ReadPreValue, 441  
   ReadValue, 441  
   ReadWithLength, 441  
   Write, 441  
 gdcmm::ExplicitImplicitDataElement, 442  
   GetLength, 444  
   Read, 444  
   ReadPreValue, 444  
   ReadValue, 444  
   ReadWithLength, 444  
 gdcmm::Fiducials, 445  
   Fiducials, 445  
 gdcmm::File, 445  
   ~File, 447  
   File, 447  
   GetDataSet, 448  
   GetHeader, 448  
   operator<<, 450  
   Read, 449  
   SetDataSet, 449  
   SetHeader, 449  
   Write, 449  
 gdcmm::FileAnonymizer, 450  
   ~FileAnonymizer, 452  
   Empty, 452  
   FileAnonymizer, 452  
   Remove, 452  
   Replace, 453  
   SetInputFileName, 453

- SetOutputFileName, 453
- Write, 454
- gdcmm::FileChangeTransferSyntax, 454
  - ~FileChangeTransferSyntax, 456
  - Change, 456
  - FileChangeTransferSyntax, 456
  - GetCodec, 456
  - New, 456
  - SetInputFileName, 457
  - SetOutputFileName, 457
  - SetTransferSyntax, 457
- gdcmm::FileDecompressLookupTable, 458
  - ~FileDecompressLookupTable, 459
  - Change, 459
  - FileDecompressLookupTable, 459
  - GetFile, 460
  - GetPixmap, 460
  - SetFile, 460
  - SetPixmap, 460
- gdcmm::FileDerivation, 461
  - ~FileDerivation, 462
  - AddDerivationDescription, 462
  - AddPurposeOfReferenceCodeSequence, 462
  - AddReference, 462
  - AddSourceImageSequence, 462
  - Derive, 463
  - FileDerivation, 462
  - GetFile, 463
  - SetAppendDerivationHistory, 463
  - SetDerivationCodeSequenceCodeValue, 463
  - SetDerivationDescription, 464
  - SetFile, 464
  - SetPurposeOfReferenceCodeSequenceCodeValue, 464
- gdcmm::FileExplicitFilter, 465
  - ~FileExplicitFilter, 466
  - Change, 466
  - ChangeFMI, 466
  - FileExplicitFilter, 466
  - GetFile, 466
  - ProcessDataSet, 466
  - SetChangePrivateTags, 467
  - SetFile, 467
  - SetRecomputeItemLength, 467
  - SetRecomputeSequenceLength, 467
  - SetUseVRUN, 467
- gdcmm::FileMetaInformation, 468
  - ~FileMetaInformation, 471
  - AppendImplementationClassUID, 471
  - ComputeDataSetMediaStorageSOPClass, 471
  - ComputeDataSetTransferSyntax, 471
  - DataSetMS, 477
  - DataSetTS, 477
  - Default, 471
  - FileMetaInformation, 470, 471
  - FillFromDataSet, 472
  - GetDataSetTransferSyntax, 472
  - GetFileMetaInformationVersion, 472
  - GetFullLength, 472
  - GetGDCMImplementationClassUID, 472
  - GetGDCMImplementationVersionName, 472
  - GetGDCMSourceApplicationEntityTitle, 473
  - GetImplementationClassUID, 473
  - GetImplementationVersionName, 473
  - GetMediaStorage, 473
  - GetMediaStorageAsString, 473
  - GetMetaInformationTS, 473
  - GetPreamble, 473, 474
  - GetSourceApplicationEntityTitle, 474
  - Insert, 474
  - IsValid, 474
  - MetaInformationTS, 477
  - operator<<, 476
  - Read, 474
  - ReadCompat, 474
  - ReadCompatInternal, 475
  - Replace, 475
  - SetDataSetTransferSyntax, 475
  - SetImplementationClassUID, 475
  - SetImplementationVersionName, 476
  - SetPreamble, 476
  - SetSourceApplicationEntityTitle, 476
  - Write, 476
- gdcmm::Filename, 477
  - EndWith, 479
  - Filename, 478
  - GetExtension, 479
  - GetFileName, 479
  - GetName, 479
  - GetPath, 479
  - IsEmpty, 479
  - IsIdentical, 480
  - Join, 480
  - operator const char \*, 480
  - ToUnixSlashes, 480
  - ToWindowsSlashes, 480
- gdcmm::FileNameEvent, 481
  - ~FileNameEvent, 483
  - CheckEvent, 483
  - FileNameEvent, 483
  - GetEventName, 484
  - GetFileName, 484
  - MakeObject, 484
  - operator=, 484
  - Self, 483
  - SetFileName, 484
  - Superclass, 483
- gdcmm::FilenameGenerator, 485

- ~FilenameGenerator, [487](#)
- FilenameGenerator, [486](#)
- FilenamesType, [486](#)
- FilenameType, [486](#)
- Generate, [487](#)
- GetFilename, [487](#)
- GetFilenames, [487](#)
- GetNumberOfFilenames, [487](#)
- GetPattern, [488](#)
- GetPrefix, [488](#)
- SetNumberOfFilenames, [488](#)
- SetPattern, [488](#)
- SetPrefix, [488](#)
- SizeType, [486](#)
- gdcm::FileSet, [489](#)
  - AddFile, [490](#)
  - FileSet, [490](#)
  - FilesType, [489](#)
  - FileType, [490](#)
  - GetFiles, [490](#)
  - operator<<, [491](#)
  - SetFiles, [491](#)
- gdcm::FileStreamer, [491](#)
  - ~FileStreamer, [493](#)
  - AppendToDataElement, [493](#)
  - AppendToGroupDataElement, [493](#)
  - CheckDataElement, [494](#)
  - CheckTemplateFileName, [494](#)
  - FileStreamer, [493](#)
  - New, [494](#)
  - ReserveDataElement, [494](#)
  - ReserveGroupDataElement, [494](#)
  - SetOutputFileName, [495](#)
  - SetTemplateFileName, [495](#)
  - StartDataElement, [495](#)
  - StartGroupDataElement, [495](#)
  - StopDataElement, [496](#)
  - StopGroupDataElement, [496](#)
- gdcm::FileWithName, [496](#)
  - filename, [498](#)
  - FileWithName, [497](#)
- gdcm::FindPatientRootQuery, [498](#)
  - FindPatientRootQuery, [499](#)
  - GetAbstractSyntaxUID, [500](#)
  - GetTagListByLevel, [500](#)
  - InitializeDataSet, [500](#)
  - QueryFactory, [501](#)
  - ValidateQuery, [500](#)
- gdcm::FindStudyRootQuery, [501](#)
  - FindStudyRootQuery, [502](#)
  - GetAbstractSyntaxUID, [503](#)
  - GetTagListByLevel, [503](#)
  - InitializeDataSet, [503](#)
  - QueryFactory, [504](#)
  - ValidateQuery, [503](#)
- gdcm::Fragment, [504](#)
  - ComputeLength, [506](#)
  - Fragment, [506](#)
  - GetLength, [506](#)
  - operator<<, [507](#)
  - Read, [506](#)
  - ReadBacktrack, [506](#)
  - ReadPreValue, [506](#)
  - ReadValue, [507](#)
  - Write, [507](#)
- gdcm::Global, [507](#)
  - ~Global, [509](#)
  - Append, [509](#)
  - GetDefs, [509](#)
  - GetDicts, [509](#), [510](#)
  - GetInstance, [510](#)
  - Global, [508](#), [509](#)
  - LoadResourcesFiles, [510](#)
  - Locate, [510](#)
  - operator<<, [511](#)
  - operator=, [511](#)
  - Prepend, [511](#)
- gdcm::GroupDict, [511](#)
  - ~GroupDict, [513](#)
  - Add, [513](#)
  - GetAbbreviation, [513](#)
  - GetName, [513](#)
  - GroupDict, [512](#)
  - GroupStringVector, [512](#)
  - Insert, [513](#)
  - operator<<, [514](#)
  - Size, [513](#)
- gdcm::IconImageFilter, [514](#)
  - ~IconImageFilter, [515](#)
  - Extract, [516](#)
  - ExtractIconImages, [516](#)
  - ExtractVeprolIconImages, [516](#)
  - GetFile, [516](#)
  - GetIconImage, [516](#)
  - GetNumberOfIconImages, [517](#)
  - IconImageFilter, [515](#)
  - SetFile, [517](#)
- gdcm::IconImageGenerator, [517](#)
  - ~IconImageGenerator, [519](#)
  - AutoPixelMinMax, [519](#)
  - ConvertRGBToPaletteColor, [519](#)
  - Generate, [519](#)
  - GetIconImage, [519](#)
  - GetPixmap, [520](#)
  - IconImageGenerator, [518](#)
  - SetOutputDimensions, [520](#)
  - SetOutsideValuePixel, [520](#)
  - SetPixelMinMax, [520](#)

- SetPixmap, [521](#)
- gdcmm::ignore\_char, [521](#)
  - ignore\_char, [521](#)
  - m\_char, [522](#)
- gdcmm::Image, [522](#)
  - ~Image, [524](#)
  - GetDirectionCosines, [525](#)
  - GetIntercept, [525](#)
  - GetOrigin, [525](#)
  - GetSlope, [526](#)
  - GetSpacing, [526](#)
  - Image, [524](#)
  - Print, [526](#)
  - SetDirectionCosines, [526](#), [527](#)
  - SetIntercept, [527](#)
  - SetOrigin, [527](#), [528](#)
  - SetSlope, [528](#)
  - SetSpacing, [528](#)
- gdcmm::ImageApplyLookupTable, [529](#)
  - ~ImageApplyLookupTable, [531](#)
  - Apply, [531](#)
  - ImageApplyLookupTable, [531](#)
  - SetRGB8, [531](#)
- gdcmm::ImageChangePhotometricInterpretation, [532](#)
  - ~ImageChangePhotometricInterpretation, [534](#)
  - Change, [534](#)
  - ChangeMonochrome, [535](#)
  - ChangeRGB2YBR, [535](#)
  - ChangeYBR2RGB, [535](#)
  - GetPhotometricInterpretation, [535](#)
  - ImageChangePhotometricInterpretation, [534](#)
  - RGB2YBR, [535](#)
  - SetPhotometricInterpretation, [535](#)
  - YBR2RGB, [536](#)
- gdcmm::ImageChangePlanarConfiguration, [536](#)
  - ~ImageChangePlanarConfiguration, [538](#)
  - Change, [538](#)
  - GetPlanarConfiguration, [538](#)
  - ImageChangePlanarConfiguration, [538](#)
  - RGBPixelsToRGBPlanes, [539](#)
  - RGBPlanesToRGBPixels, [539](#)
  - SetPlanarConfiguration, [539](#)
- gdcmm::ImageChangeTransferSyntax, [540](#)
  - ~ImageChangeTransferSyntax, [542](#)
  - Change, [543](#)
  - GetTransferSyntax, [543](#)
  - ImageChangeTransferSyntax, [542](#)
  - SetCompressIconImage, [543](#)
  - SetForce, [543](#)
  - SetTransferSyntax, [543](#)
  - SetUserCodec, [544](#)
  - TryJPEG2000Codec, [544](#)
  - TryJPEGCodec, [544](#)
  - TryJPEGLSCodec, [544](#)
  - TryRAWCodec, [545](#)
  - TryRLECodec, [545](#)
- gdcmm::ImageCodec, [545](#)
  - ~ImageCodec, [548](#)
  - AppendFrameEncode, [548](#)
  - AppendRowEncode, [548](#)
  - CanCode, [549](#)
  - CanDecode, [549](#)
  - CleanupUnusedBits, [549](#)
  - Clone, [549](#)
  - Decode, [550](#)
  - DecodeByStreams, [550](#)
  - Dimensions, [557](#)
  - DoByteSwap, [550](#)
  - DoInvertMonochrome, [550](#)
  - DoOverlayCleanup, [551](#)
  - DoPaddedCompositePixelCode, [551](#)
  - DoPlanarConfiguration, [551](#)
  - DoSimpleCopy, [551](#)
  - DoYBR, [551](#)
  - DoYBRFull422, [551](#)
  - FileChangeTransferSyntax, [557](#)
  - GetDimensions, [552](#)
  - GetHeaderInfo, [552](#)
  - GetLossyFlag, [552](#)
  - GetLUT, [552](#)
  - GetNeedByteSwap, [552](#)
  - GetNumberOfDimensions, [552](#)
  - GetPhotometricInterpretation, [553](#)
  - GetPixelFormat, [553](#)
  - GetPlanarConfiguration, [553](#)
  - ImageChangePhotometricInterpretation, [557](#)
  - ImageCodec, [548](#)
  - IsFrameEncoder, [553](#)
  - IsLossy, [553](#)
  - IsRowEncoder, [554](#)
  - IsValid, [554](#)
  - LossyFlag, [557](#)
  - LUT, [557](#)
  - LUTPtr, [548](#)
  - NeedByteSwap, [557](#)
  - NeedOverlayCleanup, [558](#)
  - NumberOfDimensions, [558](#)
  - PF, [558](#)
  - PI, [558](#)
  - PlanarConfiguration, [558](#)
  - RequestPaddedCompositePixelCode, [558](#)
  - RequestPlanarConfiguration, [558](#)
  - SetDimensions, [554](#)
  - SetLossyFlag, [554](#)
  - SetLUT, [555](#)
  - SetNeedByteSwap, [555](#)
  - SetNeedOverlayCleanup, [555](#)
  - SetNumberOfDimensions, [555](#)

- SetPhotometricInterpretation, 555
- SetPixelFormat, 556
- SetPlanarConfiguration, 556
- StartEncode, 556
- StopEncode, 556
- gdcm::ImageConverter, 559
  - ~ImageConverter, 559
  - Convert, 560
  - GetOutput, 560
  - ImageConverter, 559
  - SetInput, 560
- gdcm::ImageFragmentSplitter, 561
  - ~ImageFragmentSplitter, 563
  - GetFragmentSizeMax, 563
  - ImageFragmentSplitter, 563
  - SetForce, 563
  - SetFragmentSizeMax, 563
  - Split, 564
- gdcm::ImageHelper, 564
  - ComputeMediaStorageFromModality, 565
  - ComputeSpacingFromImagePositionPatient, 565
  - GetDimensionsValue, 566
  - GetDirectionCosinesFromDataSet, 566
  - GetDirectionCosinesValue, 566
  - GetForcePixelSpacing, 566
  - GetForceRescaleInterceptSlope, 566
  - GetLUT, 567
  - GetOriginValue, 567
  - GetPhotometricInterpretationValue, 567
  - GetPixelFormatValue, 567
  - GetPlanarConfigurationValue, 567
  - GetPMSRescaleInterceptSlope, 567
  - GetPointerFromElement, 568
  - GetRealWorldValueMappingContent, 568
  - GetRescaleInterceptSlopeValue, 568
  - GetSpacingTagFromMediaStorage, 568
  - GetSpacingValue, 568
  - GetZSpacingTagFromMediaStorage, 569
  - SetDimensionsValue, 569
  - SetDirectionCosinesValue, 569
  - SetForcePixelSpacing, 569
  - SetForceRescaleInterceptSlope, 569
  - SetOriginValue, 570
  - SetPMSRescaleInterceptSlope, 570
  - SetRescaleInterceptSlopeValue, 570
  - SetSpacingValue, 570
- gdcm::ImageReader, 571
  - ~ImageReader, 573
  - GetImage, 573
  - ImageReader, 573
  - Read, 574
  - ReadACRNEMAIImage, 574
  - ReadImage, 574
- gdcm::ImageRegionReader, 575
  - ~ImageRegionReader, 577
  - ComputeBufferLength, 577
  - GetRegion, 578
  - ImageRegionReader, 577
  - Read, 578
  - ReadInformation, 578
  - ReadIntoBuffer, 578
  - SetRegion, 579
- gdcm::ImageToImageFilter, 579
  - ~ImageToImageFilter, 581
  - GetInput, 581
  - GetOutput, 581
  - ImageToImageFilter, 581
- gdcm::ImageWriter, 582
  - ~ImageWriter, 584
  - ComputeTargetMediaStorage, 584
  - GetImage, 585
  - ImageWriter, 584
  - Write, 585
- gdcm::ImplicitDataElement, 590
  - GetLength, 591
  - Read, 591
  - ReadPreValue, 591
  - ReadValue, 591
  - ReadValueWithLength, 592
  - ReadWithLength, 592
  - Write, 592
- gdcm::InitializeEvent, 593
- gdcm::IOD, 594
  - AddIODEntry, 595
  - Clear, 595
  - GetIODEntry, 595
  - GetNumberOfIODs, 595
  - GetTypeFromTag, 596
  - IOD, 595
  - MapIODEntry, 594
  - operator<<, 596
  - SizeType, 595
- gdcm::IODEntry, 596
  - GetIE, 597
  - GetName, 598
  - GetRef, 598
  - GetUsage, 598
  - GetUsageType, 598
  - IODEntry, 597
  - operator<<, 599
  - SetIE, 598
  - SetName, 598
  - SetRef, 598
  - SetUsage, 599
- gdcm::IODs, 599
  - AddIOD, 601
  - Begin, 601
  - Clear, 601

- End, [601](#)
- GetIOD, [602](#)
- IODMapType, [600](#)
- IODMapTypeConstIterator, [600](#)
- IODName, [600](#)
- IODs, [601](#)
- operator<<, [602](#)
- gdcmm::IPPSorter, [602](#)
  - ComputeZSpacing, [606](#)
  - DirCosTolerance, [606](#)
  - DropDuplicatePositions, [607](#)
  - GetDirectionCosinesTolerance, [604](#)
  - GetZSpacing, [604](#)
  - GetZSpacingTolerance, [605](#)
  - IPPSorter, [604](#)
  - SetComputeZSpacing, [605](#)
  - SetDirectionCosinesTolerance, [605](#)
  - SetDropDuplicatePositions, [605](#)
  - SetZSpacingTolerance, [606](#)
  - Sort, [606](#)
  - ZSpacing, [607](#)
  - ZTolerance, [607](#)
- gdcmm::Item, [607](#)
  - Clear, [609](#)
  - FindDataElement, [610](#)
  - GetDataElement, [610](#)
  - GetLength, [610](#)
  - GetNestedDataSet, [610](#)
  - InsertDataElement, [610](#)
  - Item, [609](#)
  - operator<<, [611](#)
  - Read, [611](#)
  - SetNestedDataSet, [611](#)
  - Write, [611](#)
- gdcmm::IterationEvent, [612](#)
- gdcmm::JPEG12Codec, [613](#)
  - ~JPEG12Codec, [614](#)
  - DecodeByStreams, [614](#)
  - EncodeBuffer, [615](#)
  - GetHeaderInfo, [615](#)
  - InternalCode, [615](#)
  - IsStateSuspension, [615](#)
  - JPEG12Codec, [614](#)
- gdcmm::JPEG16Codec, [616](#)
  - ~JPEG16Codec, [617](#)
  - DecodeByStreams, [617](#)
  - EncodeBuffer, [618](#)
  - GetHeaderInfo, [618](#)
  - InternalCode, [618](#)
  - IsStateSuspension, [618](#)
  - JPEG16Codec, [617](#)
- gdcmm::JPEG2000Codec, [619](#)
  - ~JPEG2000Codec, [621](#)
  - AppendFrameEncode, [621](#)
  - AppendRowEncode, [621](#)
  - Bitmap, [626](#)
  - CanCode, [621](#)
  - CanDecode, [622](#)
  - Clone, [622](#)
  - Code, [622](#)
  - Decode, [622](#)
  - DecodeByStreams, [623](#)
  - DecodeExtent, [623](#)
  - GetHeaderInfo, [623](#)
  - GetQuality, [623](#)
  - GetRate, [624](#)
  - ImageRegionReader, [626](#)
  - IsFrameEncoder, [624](#)
  - IsRowEncoder, [624](#)
  - JPEG2000Codec, [621](#)
  - SetNumberOfResolutions, [624](#)
  - SetNumberOfThreadsForDecompression, [624](#)
  - SetQuality, [625](#)
  - SetRate, [625](#)
  - SetReversible, [625](#)
  - SetTileSize, [625](#)
  - StartEncode, [625](#)
  - StopEncode, [625](#)
- gdcmm::JPEG8Codec, [626](#)
  - ~JPEG8Codec, [628](#)
  - DecodeByStreams, [628](#)
  - EncodeBuffer, [628](#)
  - GetHeaderInfo, [629](#)
  - InternalCode, [629](#)
  - IsStateSuspension, [629](#)
  - JPEG8Codec, [628](#)
- gdcmm::JPEGCodec, [630](#)
  - ~JPEGCodec, [632](#)
  - AppendFrameEncode, [632](#)
  - AppendRowEncode, [633](#)
  - BitSample, [638](#)
  - CanCode, [633](#)
  - CanDecode, [633](#)
  - Clone, [633](#)
  - Code, [634](#)
  - ComputeOffsetTable, [634](#)
  - Decode, [634](#)
  - DecodeByStreams, [634](#)
  - DecodeExtent, [634](#)
  - EncodeBuffer, [635](#)
  - GetHeaderInfo, [635](#)
  - GetLossless, [635](#)
  - GetQuality, [635](#)
  - ImageRegionReader, [638](#)
  - IsFrameEncoder, [636](#)
  - IsRowEncoder, [636](#)
  - IsStateSuspension, [636](#)
  - IsValid, [636](#)



- JPEGCodec, [632](#)
- Quality, [638](#)
- SetBitSample, [636](#)
- SetLossless, [636](#)
- SetPixelFormat, [637](#)
- SetQuality, [637](#)
- StartEncode, [637](#)
- StopEncode, [637](#)
- gdcmm::JPEGLSCodec, [638](#)
  - ~JPEGLSCodec, [641](#)
  - AppendFrameEncode, [641](#)
  - AppendRowEncode, [641](#)
  - CanCode, [641](#)
  - CanDecode, [641](#)
  - Clone, [642](#)
  - Code, [642](#)
  - Decode, [642](#)
  - DecodeExtent, [643](#)
  - GetBufferLength, [643](#)
  - GetHeaderInfo, [643](#)
  - GetLossless, [643](#)
  - ImageRegionReader, [645](#)
  - IsFrameEncoder, [643](#)
  - IsRowEncoder, [644](#)
  - JPEGLSCodec, [640](#)
  - SetBufferLength, [644](#)
  - SetLossless, [644](#)
  - SetLossyError, [644](#)
  - StartEncode, [644](#)
  - StopEncode, [644](#)
- gdcmm::JSON, [645](#)
  - ~JSON, [646](#)
  - Code, [646](#)
  - Decode, [646](#)
  - GetPrettyPrint, [646](#)
  - JSON, [645](#)
  - PrettyPrintOff, [646](#)
  - PrettyPrintOn, [647](#)
  - SetPrettyPrint, [647](#)
- gdcmm::KAKADUCodec, [647](#)
  - ~KAKADUCodec, [649](#)
  - CanCode, [649](#)
  - CanDecode, [649](#)
  - Clone, [649](#)
  - Code, [650](#)
  - Decode, [650](#)
  - KAKADUCodec, [649](#)
- gdcmm::LO, [651](#)
  - const\_iterator, [652](#)
  - const\_reference, [652](#)
  - const\_reverse\_iterator, [652](#)
  - difference\_type, [652](#)
  - IsValid, [654](#)
  - iterator, [652](#)
  - LO, [654](#)
  - pointer, [653](#)
  - reference, [653](#)
  - reverse\_iterator, [653](#)
  - size\_type, [653](#)
  - Superclass, [653](#)
  - value\_type, [653](#)
- gdcmm::LookupTable, [655](#)
  - ~LookupTable, [657](#)
  - Allocate, [658](#)
  - BitSample, [662](#)
  - BLUE, [657](#)
  - Clear, [658](#)
  - Decode, [658](#)
  - Decode8, [658](#)
  - GetBitSample, [659](#)
  - GetBufferAsRGBA, [659](#)
  - GetLUT, [659](#)
  - GetLUTDescriptor, [659](#)
  - GetLUTLength, [660](#)
  - GetPointer, [660](#)
  - GRAY, [657](#)
  - GREEN, [657](#)
  - IncompleteLUT, [663](#)
  - InitializeBlueLUT, [660](#)
  - Initialized, [660](#)
  - InitializeGreenLUT, [660](#)
  - InitializeLUT, [661](#)
  - InitializeRedLUT, [661](#)
  - Internal, [663](#)
  - IsRGB8, [661](#)
  - LookupTable, [657](#)
  - LookupTableType, [657](#)
  - Print, [661](#)
  - RED, [657](#)
  - SetBlueLUT, [661](#)
  - SetGreenLUT, [662](#)
  - SetLUT, [662](#)
  - SetRedLUT, [662](#)
  - UNKNOWN, [657](#)
  - WriteBufferAsRGBA, [662](#)
- gdcmm::Macro, [664](#)
  - AddMacroEntry, [665](#)
  - ArrayIncludeMacrosType, [665](#)
  - Clear, [666](#)
  - FindMacroEntry, [666](#)
  - GetMacroEntry, [666](#)
  - GetName, [666](#)
  - Macro, [665](#)
  - MapModuleEntry, [665](#)
  - operator<<, [667](#)
  - SetName, [666](#)
  - Verify, [666](#)
- gdcmm::Macros, [667](#)



- AddMacro, [668](#)
- Clear, [668](#)
- GetMacro, [669](#)
- IsEmpty, [669](#)
- Macros, [668](#)
- ModuleMapType, [668](#)
- operator<<, [669](#)
- gdcmm::MD5, [671](#)
  - Compute, [672](#)
  - ComputeFile, [672](#)
- gdcmm::MediaStorage, [672](#)
  - AmbulatoryECGWaveformStorage, [677](#)
  - Audio, [679](#)
  - BasicTextSR, [677](#)
  - BasicVoiceAudioWaveformStorage, [677](#)
  - BreastProjectionXRayImageStorageForPresentation, [678](#)
  - BreastProjectionXRayImageStorageForProcessing, [678](#)
  - BreastTomosynthesisImageStorage, [678](#)
  - CardiacElectrophysiologyWaveformStorage, [677](#)
  - ComprehensiveSR, [677](#)
  - ComputedRadiographylImageStorage, [676](#)
  - CSANonImageStorage, [677](#)
  - CTImageStorage, [676](#)
  - DetachedPatientManagementSOPClass, [677](#)
  - DetachedStudyManagementSOPClass, [677](#)
  - DetachedVisitManagementSOPClass, [677](#)
  - DigitalIntraoralXRayImageStorageForPresentation, [676](#)
  - DigitalIntraoralXRayImageStorageForProcessing, [676](#)
  - DigitalMammographyImageStorageForPresentation, [676](#)
  - DigitalMammographyImageStorageForProcessing, [676](#)
  - DigitalXRayImageStorageForPresentation, [676](#)
  - DigitalXRayImageStorageForProcessing, [676](#)
  - EncapsulatedCDASStorage, [677](#)
  - EncapsulatedPDFStorage, [677](#)
  - EnhancedCTImageStorage, [676](#)
  - EnhancedMRColorImageStorage, [678](#)
  - EnhancedMRIImageStorage, [677](#)
  - EnhancedPETImageStorage, [678](#)
  - EnhancedSR, [677](#)
  - EnhancedUSVolumeStorage, [678](#)
  - EnhancedXAImageStorage, [678](#)
  - FujiPrivateCRImageStorage, [678](#)
  - FujiPrivateMammoCRImageStorage, [678](#)
  - GeneralECGWaveformStorage, [677](#)
  - GeneralElectricMagneticResonanceImageStorage, [678](#)
  - GEPrivate3DModelStorage, [678](#)
  - GetModality, [679](#)
  - GetModalityDimension, [679](#)
  - GetMSString, [680](#)
  - GetMSType, [680](#)
  - GetNumberOfModality, [680](#)
  - GetNumberOfMSString, [680](#)
  - GetNumberOfMSType, [680](#)
  - GetString, [681](#)
  - GrayscaleSoftcopyPresentationStateStorageSOP-Class, [677](#)
  - GuessFromModality, [681](#)
  - HangingProtocolStorage, [678](#)
  - HardcopyColorImageStorage, [678](#)
  - HardcopyGrayscaleImageStorage, [677](#)
  - HemodynamicWaveformStorage, [677](#)
  - IsImage, [681](#)
  - IsUndefined, [681](#)
  - IVOCTForPresentation, [678](#)
  - IVOCTForProcessing, [678](#)
  - KeyObjectSelectionDocument, [678](#)
  - LeadECGWaveformStorage, [677](#)
  - LegacyConvertedEnhancedCTImageStorage, [678](#)
  - LegacyConvertedEnhancedMRIImageStorage, [678](#)
  - LegacyConvertedEnhancedPETImageStorage, [678](#)
  - MammographyCADSR, [678](#)
  - MediaStorage, [679](#)
  - MediaStorageDirectoryStorage, [676](#)
  - ModalityPerformedProcedureStepSOPClass, [678](#)
  - MRIImageStorage, [676](#)
  - MRSpectroscopyStorage, [677](#)
  - MS\_END, [678](#)
  - MSType, [676](#)
  - MultiframeGrayscaleByteSecondaryCaptureImageStorage, [677](#)
  - MultiframeGrayscaleWordSecondaryCaptureImageStorage, [677](#)
  - MultiframeSingleBitSecondaryCaptureImageStorage, [677](#)
  - MultiframeTrueColorSecondaryCaptureImageStorage, [677](#)
  - NoObject, [679](#)
  - NuclearMedicineImageStorage, [677](#)
  - NuclearMedicineImageStorageRetired, [677](#)
  - ObjectEnd, [679](#)
  - ObjectType, [679](#)
  - operator MSType, [682](#)
  - operator<<, [683](#)
  - OphthalmicPhotography16BitImageStorage, [678](#)
  - OphthalmicPhotography8BitImageStorage, [678](#)
  - OphthalmicTomographyImageStorage, [678](#)
  - PDF, [679](#)
  - PETImageStorage, [677](#)
  - Philips3D, [677](#)
  - PhilipsPrivateMRSyntheticImageStorage, [678](#)
  - RawDataStorage, [677](#)

- RTDoseStorage, [677](#)
- RTImageStorage, [677](#)
- RTIonBeamsTreatmentRecordStorage, [678](#)
- RTIonPlanStorage, [678](#)
- RTPlanStorage, [677](#)
- RTStructureSetStorage, [677](#)
- RTTreatmentSummaryRecordStorage, [678](#)
- SecondaryCaptureImageStorage, [677](#)
- Segmentation, [679](#)
- SegmentationStorage, [678](#)
- SetFromDataSet, [682](#)
- SetFromFile, [682](#)
- SetFromHeader, [682](#)
- SetFromModality, [682](#)
- SetFromSourceImageSequence, [683](#)
- SpacialFiducialsStorage, [677](#)
- SpacialRegistrationStorage, [677](#)
- StandaloneCurveStorage, [677](#)
- StandaloneModalityLUTStorage, [677](#)
- StandaloneOverlayStorage, [677](#)
- StandaloneVOILUTStorage, [677](#)
- StudyComponentManagementSOPClass, [677](#)
- SurfaceSegmentationStorage, [678](#)
- ToshibaPrivateDataStorage, [678](#)
- UltrasoundImageStorage, [676](#)
- UltrasoundImageStorageRetired, [676](#)
- UltrasoundMultiFrameImageStorage, [676](#)
- UltrasoundMultiFrameImageStorageRetired, [676](#)
- URI, [679](#)
- Video, [679](#)
- VideoEndoscopicImageStorage, [678](#)
- VideoPhotographicImageStorage, [678](#)
- VLEndoscopicImageStorage, [678](#)
- VLMicroscopicImageStorage, [678](#)
- VLPhotographicImageStorage, [678](#)
- VLWholeSlideMicroscopyImageStorage, [678](#)
- Waveform, [679](#)
- XRay3DAngiographicImageStorage, [678](#)
- XRay3DCraniofacialImageStorage, [678](#)
- XRayAngiographicBiPlanarImageStorageRetired, [677](#)
- XRayAngiographicImageStorage, [677](#)
- XRayRadiationDoseSR, [678](#)
- XRayRadiofluoroscopicImageStorage, [677](#)
- gdcm::MemberCommand< T >, [683](#)
- ~MemberCommand, [686](#)
- Execute, [687](#)
- m\_ConstMemberFunction, [688](#)
- m\_MemberFunction, [688](#)
- m\_This, [689](#)
- MemberCommand, [686](#)
- New, [687](#)
- operator=, [687](#)
- Self, [685](#)
- SetCallbackFunction, [688](#)
- TConstMemberFunctionPointer, [686](#)
- TMemberFunctionPointer, [686](#)
- gdcm::MeshPrimitive, [689](#)
- ~MeshPrimitive, [692](#)
- AddPrimitiveData, [692](#)
- EDGE, [692](#)
- FACET, [692](#)
- GetMPType, [692](#)
- GetMPTypeString, [692](#)
- GetNumberOfPrimitivesData, [693](#)
- GetPrimitiveData, [693](#)
- GetPrimitivesData, [693](#)
- GetPrimitiveType, [694](#)
- LINE, [692](#)
- MeshPrimitive, [692](#)
- MPType, [691](#)
- MPType\_END, [692](#)
- PrimitiveData, [694](#)
- PrimitivesData, [691](#)
- PrimitiveType, [695](#)
- SetPrimitiveData, [694](#)
- SetPrimitivesData, [694](#)
- SetPrimitiveType, [694](#)
- TRIANGLE, [692](#)
- TRIANGLE\_FAN, [692](#)
- TRIANGLE\_STRIP, [692](#)
- VERTEX, [692](#)
- gdcm::ModalityPerformedProcedureStepCreateQuery, [695](#)
- GetAbstractSyntaxUID, [697](#)
- GetRequiredDataSet, [697](#)
- ModalityPerformedProcedureStepCreateQuery, [697](#)
- QueryFactory, [697](#)
- ValidateQuery, [697](#)
- gdcm::ModalityPerformedProcedureStepSetQuery, [698](#)
- GetAbstractSyntaxUID, [700](#)
- GetRequiredDataSet, [700](#)
- ModalityPerformedProcedureStepSetQuery, [700](#)
- QueryFactory, [700](#)
- ValidateQuery, [700](#)
- gdcm::ModifiedEvent, [701](#)
- gdcm::Module, [702](#)
- AddMacro, [703](#)
- AddModuleEntry, [703](#)
- ArrayIncludeMacrosType, [703](#)
- Clear, [703](#)
- FindModuleEntryInMacros, [704](#)
- GetModuleEntryInMacros, [704](#)
- GetName, [704](#)
- MapModuleEntry, [703](#)
- Module, [703](#)
- operator<<, [704](#)
- SetName, [704](#)
- Verify, [704](#)

- gdcmmoduleentry, 705
  - ~ModuleEntry, 707
  - DataElementType, 709
  - Description, 707
  - DescriptionField, 709
  - GetDescription, 707
  - GetName, 708
  - GetType, 708
  - ModuleEntry, 707
  - Name, 709
  - operator<<, 708
  - SetDescription, 708
  - SetName, 708
  - SetType, 708
- gdcmmodule, 709
  - AddModule, 711
  - Clear, 711
  - GetModule, 711
  - IsEmpty, 711
  - ModuleMapType, 710
  - Modules, 710
  - operator<<, 711
- gdcmmovepatientrootquery, 712
  - GetAbstractSyntaxUID, 713
  - GetTagListByLevel, 713
  - InitializeDataSet, 713
  - MovePatientRootQuery, 713
  - QueryFactory, 714
  - ValidateQuery, 714
- gdcmmovestudyrootquery, 715
  - GetAbstractSyntaxUID, 716
  - GetTagListByLevel, 716
  - InitializeDataSet, 716
  - MoveStudyRootQuery, 716
  - QueryFactory, 717
  - ValidateQuery, 717
- gdcmmrprotocol, 717
  - ~MrProtocol, 718
  - FindMrProtocolByName, 719
  - GetMrProtocolByName, 719
  - GetSliceArray, 719
  - GetVersion, 719
  - Load, 719
  - MrProtocol, 718
  - operator<<, 720
  - Print, 719
- gdcmmrprotocol::slice, 997
  - Normal, 997
  - Position, 997
- gdcmmrprotocol::slicearray, 998
  - Slices, 998
- gdcmmrprotocol::vector3, 1270
  - dCor, 1270
  - dSag, 1270
  - dTra, 1270
- gdcmmoduleentries, 729
  - AddModuleEntry, 731
  - GetModuleEntry, 731
  - GetNumberOfModuleEntries, 731
  - NestedModuleEntries, 730
  - operator<<, 731
  - SizeType, 730
- gdcmmodule::network, 77
  - cMaxEventID, 83
  - cMaxStateID, 83
  - eAABORTPDUReturnReceivedOpen, 81
  - eAABORTRequest, 81
  - eAASSOCIATE\_RQPDUReturnReceived, 81
  - eAASSOCIATERequestLocalUser, 81
  - eAASSOCIATEResponseAccept, 81
  - eAASSOCIATEResponseReject, 81
  - eARELEASE\_RPPDUReturnReceived, 81
  - eARELEASE\_RQPDUReturnReceivedOpen, 81
  - eARELEASERequest, 81
  - eARELEASEResponse, 81
  - eARTIMTimerExpired, 81
  - eASSOCIATE\_ACPDUReturnReceived, 81
  - eASSOCIATE\_RJPDUReturnReceived, 81
  - eEventDoesNotExist, 81
  - EEventID, 81
  - ePDATArequest, 81
  - ePDATATFPDU, 81
  - eSta10ReleaseCollisionAc, 83
  - eSta11ReleaseCollisionRq, 83
  - eSta12ReleaseCollisionAcLocal, 83
  - eSta13AwaitingClose, 83
  - eSta1Idle, 83
  - eSta2Open, 83
  - eSta3WaitLocalAssoc, 83
  - eSta4LocalAssocDone, 83
  - eSta5WaitRemoteAssoc, 83
  - eSta6TransferReady, 83
  - eSta7WaitRelease, 83
  - eSta8WaitLocalRelease, 83
  - eSta9ReleaseCollisionRqLocal, 83
  - eStaDoesNotExist, 83
  - EStateID, 81
  - eTransportConnConfirmLocal, 81
  - eTransportConnectionClosed, 81
  - eTransportConnIndicLocal, 81
  - eUnrecognizedPDUReturnReceived, 81
  - GetStateIndex, 83
- gdcmmodule::network::AAbortPDU, 87
  - AAbortPDU, 88
  - IsLastFragment, 88
  - Print, 88
  - Read, 88
  - SetReason, 89

SetSource, 89  
 Size, 89  
 Write, 89  
 gdcmm::network::AAssociateACPDU, 90  
   AAssociateACPDU, 91  
   AAssociateRQPDU, 94  
   AddPresentationContextAC, 92  
   GetNumberOfPresentationContextAC, 92  
   GetPresentationContextAC, 92  
   GetUserInformation, 92  
   InitFromRQ, 92  
   IsLastFragment, 92  
   Print, 93  
   Read, 93  
   SetCalledAETitle, 93  
   SetCallingAETitle, 93  
   Size, 93  
   SizeType, 91  
   Write, 93  
 gdcmm::network::AAssociateRJPDU, 94  
   AAssociateRJPDU, 95  
   IsLastFragment, 95  
   Print, 96  
   Read, 96  
   Size, 96  
   Write, 96  
 gdcmm::network::AAssociateRQPDU, 97  
   AAssociateACPDU, 103  
   AAssociateRQPDU, 99  
   AddPresentationContext, 99  
   GetCalledAETitle, 99  
   GetCallingAETitle, 99  
   GetNumberOfPresentationContext, 100  
   GetPresentationContext, 100  
   GetPresentationContextByAbstractSyntax, 100  
   GetPresentationContextByID, 100  
   GetPresentationContexts, 100  
   GetReserved43\_74, 100  
   GetUserInformation, 101  
   IsAETitleValid, 101  
   IsLastFragment, 101  
   PresentationContextArrayType, 98  
   Print, 101  
   Read, 101  
   SetCalledAETitle, 102  
   SetCallingAETitle, 102  
   SetUserInformation, 102  
   Size, 102  
   SizeType, 99  
   Write, 102  
 gdcmm::network::AbstractSyntax, 104  
   AbstractSyntax, 105  
   GetAsDataElement, 105  
   GetName, 105  
   operator==, 105  
   Print, 105  
   Read, 105  
   SetName, 106  
   SetNameFromUID, 106  
   Size, 106  
   Write, 106  
 gdcmm::network::ApplicationContext, 119  
   ApplicationContext, 120  
   GetName, 120  
   Print, 120  
   Read, 120  
   SetName, 120  
   Size, 120  
   Write, 121  
 gdcmm::network::AReleaseRPPDU, 124  
   AReleaseRPPDU, 125  
   IsLastFragment, 125  
   Print, 125  
   Read, 125  
   Size, 126  
   Write, 126  
 gdcmm::network::AReleaseRQPDU, 126  
   AReleaseRQPDU, 127  
   IsLastFragment, 127  
   Print, 128  
   Read, 128  
   Size, 128  
   Write, 128  
 gdcmm::network::ARTIMTimer, 129  
   ARTIMTimer, 129  
   GetElapsedTime, 129  
   GetHasExpired, 130  
   GetTimeout, 130  
   SetTimeout, 130  
   Start, 130  
   Stop, 130  
 gdcmm::network::AsynchronousOperationsWindowSub, 132  
   AsynchronousOperationsWindowSub, 133  
   Print, 133  
   Read, 133  
   Size, 133  
   Write, 134  
 gdcmm::network::BaseCompositeMessage, 171  
   ~BaseCompositeMessage, 172  
   ConstructPDV, 172  
 gdcmm::network::BaseNormalizedMessage, 173  
   ~BaseNormalizedMessage, 174  
   ConstructPDV, 174  
 gdcmm::network::BasePDU, 175  
   ~BasePDU, 176  
   IsLastFragment, 176  
   Print, 176

- Read, [176](#)
- Size, [177](#)
- Write, [177](#)
- gdcmm::network::CEchoRQ, [237](#)
  - AffectedSOPClassUID, [238](#)
  - ConstructPDV, [238](#)
  - MessageID, [238](#)
- gdcmm::network::CEchoRSP, [239](#)
  - ConstructPDVByDataSet, [240](#)
- gdcmm::network::CFind, [240](#)
- gdcmm::network::CFindCancelRQ, [240](#)
  - ConstructPDVByDataSet, [241](#)
- gdcmm::network::CFindRQ, [242](#)
  - ConstructPDV, [243](#)
- gdcmm::network::CFindRSP, [243](#)
  - ConstructPDVByDataSet, [244](#)
- gdcmm::network::CMoveCancelRq, [245](#)
  - ConstructPDVByDataSet, [245](#)
- gdcmm::network::CMoveRQ, [246](#)
  - ConstructPDV, [247](#)
- gdcmm::network::CMoveRSP, [247](#)
  - ConstructPDVByDataSet, [248](#)
- gdcmm::network::CompositeMessageFactory, [263](#)
  - ConstructCEchoRQ, [263](#)
  - ConstructCFindRQ, [263](#)
  - ConstructCMoveRQ, [263](#)
  - ConstructCStoreRQ, [264](#)
  - ConstructCStoreRSP, [264](#)
- gdcmm::network::CStoreRQ, [300](#)
  - ConstructPDV, [301](#)
- gdcmm::network::CStoreRSP, [301](#)
  - ConstructPDV, [302](#)
- gdcmm::network::DIMSE, [380](#)
  - C\_CANCEL\_RQ, [381](#)
  - C\_ECHO\_RQ, [381](#)
  - C\_ECHO\_RSP, [381](#)
  - C\_FIND\_RQ, [381](#)
  - C\_FIND\_RSP, [381](#)
  - C\_GET\_RQ, [381](#)
  - C\_GET\_RSP, [381](#)
  - C\_MOVE\_RQ, [381](#)
  - C\_MOVE\_RSP, [381](#)
  - C\_STORE\_RQ, [381](#)
  - C\_STORE\_RSP, [381](#)
  - CommandTypes, [381](#)
  - N\_ACTION\_RQ, [381](#)
  - N\_ACTION\_RSP, [381](#)
  - N\_CREATE\_RQ, [381](#)
  - N\_CREATE\_RSP, [381](#)
  - N\_DELETE\_RQ, [381](#)
  - N\_DELETE\_RSP, [381](#)
  - N\_EVENT\_REPORT\_RQ, [381](#)
  - N\_EVENT\_REPORT\_RSP, [381](#)
  - N\_GET\_RQ, [381](#)
  - N\_GET\_RSP, [381](#)
  - N\_SET\_RQ, [381](#)
  - N\_SET\_RSP, [381](#)
- gdcmm::network::ImplementationClassUIDSub, [586](#)
  - ImplementationClassUIDSub, [586](#)
  - Print, [586](#)
  - Read, [586](#)
  - Size, [587](#)
  - Write, [587](#)
- gdcmm::network::ImplementationUIDSub, [587](#)
  - ImplementationUIDSub, [587](#)
  - Write, [588](#)
- gdcmm::network::ImplementationVersionNameSub, [588](#)
  - ImplementationVersionNameSub, [588](#)
  - Print, [589](#)
  - Read, [589](#)
  - Size, [589](#)
  - Write, [589](#)
- gdcmm::network::MaximumLengthSub, [669](#)
  - GetMaximumLength, [670](#)
  - MaximumLengthSub, [670](#)
  - Print, [670](#)
  - Read, [670](#)
  - SetMaximumLength, [671](#)
  - Size, [671](#)
  - Write, [671](#)
- gdcmm::network::NActionRQ, [720](#)
  - ConstructPDV, [721](#)
- gdcmm::network::NActionRSP, [721](#)
  - ConstructPDVByDataSet, [722](#)
- gdcmm::network::NCreateRQ, [723](#)
  - ConstructPDV, [724](#)
- gdcmm::network::NCreateRSP, [724](#)
  - ConstructPDVByDataSet, [725](#)
- gdcmm::network::NDeleteRQ, [726](#)
  - ConstructPDV, [727](#)
- gdcmm::network::NDeleteRSP, [727](#)
  - ConstructPDVByDataSet, [728](#)
- gdcmm::network::NEventReportRQ, [732](#)
  - ConstructPDV, [733](#)
- gdcmm::network::NEventReportRSP, [733](#)
  - ConstructPDVByDataSet, [734](#)
- gdcmm::network::NGetRQ, [735](#)
  - ConstructPDV, [736](#)
- gdcmm::network::NGetRSP, [736](#)
  - ConstructPDVByDataSet, [737](#)
- gdcmm::network::NormalizedMessageFactory, [739](#)
  - ConstructNAction, [739](#)
  - ConstructNCreate, [739](#)
  - ConstructNDelete, [739](#)
  - ConstructNEventReport, [740](#)
  - ConstructNGet, [740](#)
  - ConstructNSet, [740](#)
- gdcmm::network::NSetRQ, [743](#)

- ConstructPDV, 744
- gdcmm::network::NSetRSP, 745
  - ConstructPDVByDataSet, 746
- gdcmm::network::PDataTFPDU, 781
  - AddPresentationDataValue, 782
  - GetNumberOfPresentationDataValues, 782
  - GetPresentationDataValue, 783
  - IsLastFragment, 783
  - PDataTFPDU, 782
  - Print, 783
  - Read, 783
  - ReadInto, 783
  - Size, 783
  - SizeType, 782
  - Write, 784
- gdcmm::network::PDUFactory, 793
  - ConstructAbortPDU, 794
  - ConstructPDU, 794
  - ConstructReleasePDU, 794
  - CreateCEchoPDU, 794
  - CreateCFindPDU, 794
  - CreateCMovePDU, 794
  - CreateCStoreRQPDU, 795
  - CreateCStoreRSPDU, 795
  - CreateNActionPDU, 795
  - CreateNCreatePDU, 795
  - CreateNDeletePDU, 795
  - CreateNEventReportPDU, 795
  - CreateNGetPDU, 796
  - CreateNSetPDU, 796
  - DetermineEventByPDU, 796
  - GetPDVs, 796
- gdcmm::network::PresentationContextAC, 845
  - GetPresentationContextID, 846
  - GetReason, 846
  - GetTransferSyntax, 846
  - PresentationContextAC, 845
  - Print, 846
  - Read, 846
  - SetPresentationContextID, 846
  - SetReason, 847
  - SetTransferSyntax, 847
  - Size, 847
  - Write, 847
- gdcmm::network::PresentationContextRQ, 851
  - AddTransferSyntax, 853
  - GetAbstractSyntax, 853
  - GetNumberOfTransferSyntaxes, 853
  - GetPresentationContextID, 853
  - GetTransferSyntax, 853
  - GetTransferSyntaxes, 854
  - operator==, 854
  - PresentationContextRQ, 852
  - Print, 854
  - Read, 854
  - SetAbstractSyntax, 854
  - SetPresentationContextID, 854
  - Size, 855
  - SizeType, 852
  - Write, 855
- gdcmm::network::PresentationDataValue, 855
  - ConcatenatePDVBlobs, 856
  - ConcatenatePDVBlobsAsExplicit, 856
  - GetBlob, 856
  - GetIsCommand, 857
  - GetIsLastFragment, 857
  - GetMessageHeader, 857
  - GetPresentationContextID, 857
  - PresentationDataValue, 856
  - Print, 857
  - Read, 857
  - ReadInto, 857
  - SetBlob, 858
  - SetCommand, 858
  - SetDataSet, 858
  - SetLastFragment, 858
  - SetMessageHeader, 858
  - SetPresentationContextID, 859
  - Size, 859
  - Write, 859
- gdcmm::network::RoleSelectionSub, 921
  - Print, 921
  - Read, 922
  - RoleSelectionSub, 921
  - SetTuple, 922
  - Size, 922
  - Write, 922
- gdcmm::network::ServiceClassApplicationInformation, 976
  - Print, 977
  - Read, 977
  - ServiceClassApplicationInformation, 976
  - SetTuple, 977
  - Size, 977
  - Write, 977
- gdcmm::network::SOPClassExtendedNegotiationSub, 1003
  - Print, 1004
  - Read, 1004
  - SetTuple, 1004
  - Size, 1004
  - SOPClassExtendedNegotiationSub, 1004
  - Write, 1004
- gdcmm::network::TableRow, 1100
  - ~TableRow, 1101
  - TableRow, 1101
  - transitions, 1101
- gdcmm::network::TransferSyntaxSub, 1135
  - GetName, 1135



- operator==, 1136
- Print, 1136
- Read, 1136
- SetName, 1136
- SetNameFromUID, 1136
- Size, 1136
- TransferSyntaxSub, 1135
- Write, 1137
- gdcmm::network::Transition, 1137
  - ~Transition, 1138
  - mAction, 1139
  - MakeNew, 1138
  - mEnd, 1139
  - Transition, 1138
- gdcmm::network::ULAction, 1183
  - ~ULAction, 1184
  - operator=, 1185
  - PerformAction, 1185
  - ULAction, 1184, 1185
- gdcmm::network::ULActionAA1, 1186
  - PerformAction, 1186
- gdcmm::network::ULActionAA2, 1187
  - PerformAction, 1188
- gdcmm::network::ULActionAA3, 1188
  - PerformAction, 1189
- gdcmm::network::ULActionAA4, 1190
  - PerformAction, 1190
- gdcmm::network::ULActionAA5, 1191
  - PerformAction, 1192
- gdcmm::network::ULActionAA6, 1192
  - PerformAction, 1193
- gdcmm::network::ULActionAA7, 1194
  - PerformAction, 1194
- gdcmm::network::ULActionAA8, 1195
  - PerformAction, 1196
- gdcmm::network::ULActionAE1, 1196
  - PerformAction, 1197
- gdcmm::network::ULActionAE2, 1198
  - PerformAction, 1198
- gdcmm::network::ULActionAE3, 1199
  - PerformAction, 1200
- gdcmm::network::ULActionAE4, 1200
  - PerformAction, 1201
- gdcmm::network::ULActionAE5, 1202
  - PerformAction, 1202
- gdcmm::network::ULActionAE6, 1203
  - PerformAction, 1204
- gdcmm::network::ULActionAE7, 1204
  - PerformAction, 1205
- gdcmm::network::ULActionAE8, 1206
  - PerformAction, 1206
- gdcmm::network::ULActionAR1, 1207
  - PerformAction, 1208
- gdcmm::network::ULActionAR10, 1208
  - PerformAction, 1209
- gdcmm::network::ULActionAR2, 1210
  - PerformAction, 1210
- gdcmm::network::ULActionAR3, 1211
  - PerformAction, 1212
- gdcmm::network::ULActionAR4, 1212
  - PerformAction, 1213
- gdcmm::network::ULActionAR5, 1214
  - PerformAction, 1214
- gdcmm::network::ULActionAR6, 1215
  - PerformAction, 1216
- gdcmm::network::ULActionAR7, 1216
  - PerformAction, 1217
- gdcmm::network::ULActionAR8, 1218
  - PerformAction, 1218
- gdcmm::network::ULActionAR9, 1219
  - PerformAction, 1220
- gdcmm::network::ULActionDT1, 1220
  - PerformAction, 1221
- gdcmm::network::ULActionDT2, 1222
  - PerformAction, 1222
- gdcmm::network::ULBasicCallback, 1223
  - ~ULBasicCallback, 1224
  - GetDataSets, 1224
  - GetResponses, 1224
  - HandleDataSet, 1225
  - HandleResponse, 1225
  - ULBasicCallback, 1224
- gdcmm::network::ULConnection, 1225
  - ~ULConnection, 1227
  - AddAcceptedPresentationContext, 1227
  - FindContext, 1227
  - GetAcceptedPresentationContexts, 1227, 1228
  - GetConnectionInfo, 1228
  - GetMaxPDUSize, 1228
  - GetPresentationContextACByID, 1228
  - GetPresentationContextIDFromPresentationContext, 1228
  - GetPresentationContextRQByID, 1228
  - GetPresentationContexts, 1229
  - GetProtocol, 1229
  - GetState, 1229
  - GetTimer, 1229
  - InitializeConnection, 1229
  - InitializeIncomingConnection, 1229
  - operator=, 1230
  - SetMaxPDUSize, 1230
  - SetPresentationContexts, 1230
  - SetState, 1230
  - StopProtocol, 1230
  - ULActionAE6, 1231
  - ULConnection, 1227
  - ULConnectionManager, 1231
- gdcmm::network::ULConnectionCallback, 1231

- ~ULConnectionCallback, 1232
- DataSetHandled, 1233
- DataSetHandles, 1233
- HandleDataSet, 1233
- HandleResponse, 1233
- mImplicit, 1234
- ResetHandledDataSet, 1233
- SetImplicitFlag, 1233
- ULConnectionCallback, 1232
- gdcmm::network::ULConnectionInfo, 1234
  - GetCalledAETitle, 1235
  - GetCalledComputerName, 1235
  - GetCalledIPAddress, 1235
  - GetCalledIPPort, 1235
  - GetCallingAETitle, 1235
  - GetMaxPDULength, 1235
  - Initialize, 1235
  - SetMaxPDULength, 1236
  - ULConnectionInfo, 1234
- gdcmm::network::ULConnectionManager, 1236
  - ~ULConnectionManager, 1238
  - BreakConnection, 1239
  - BreakConnectionNow, 1239
  - EstablishConnection, 1239
  - EstablishConnectionMove, 1239
  - mConnection, 1243
  - mSecondaryConnection, 1243
  - mTransitions, 1243
  - RunEventLoop, 1239
  - RunMoveEventLoop, 1240
  - SendEcho, 1240
  - SendFind, 1240
  - SendMove, 1240
  - SendNAction, 1241
  - SendNCreate, 1241
  - SendNDelete, 1241
  - SendNEventReport, 1242
  - SendNGet, 1242
  - SendNSet, 1242
  - SendStore, 1243
  - ULConnectionManager, 1238
- gdcmm::network::ULEvent, 1244
  - ~ULEvent, 1245
  - GetDataSetPos, 1245
  - GetEvent, 1245
  - GetIStream, 1245
  - GetPDUs, 1245
  - SetEvent, 1246
  - SetPDU, 1246
  - ULEvent, 1244, 1245
- gdcmm::network::ULTransitionTable, 1246
  - HandleEvent, 1247
  - PrintTable, 1247
  - ULTransitionTable, 1247
- gdcmm::network::ULWritingCallback, 1248
  - ~ULWritingCallback, 1249
  - HandleDataSet, 1249
  - HandleResponse, 1249
  - SetDirectory, 1249
  - ULWritingCallback, 1249
- gdcmm::network::UserInfo, 1260
  - ~UserInfo, 1260
  - AddRoleSelectionSub, 1261
  - AddSOPClassExtendedNegotiationSub, 1261
  - GetMaximumLengthSub, 1261
  - operator=, 1261
  - Print, 1261
  - Read, 1262
  - Size, 1262
  - UserInfo, 1260
  - Write, 1262
- gdcmm::NoEvent, 738
- gdcmm::NormalizedNetworkFunctions, 740
  - ConstructQuery, 741
  - NAction, 741
  - NCreate, 742
  - NDelete, 742
  - NEventReport, 742
  - NGet, 742
  - NSet, 743
- gdcmm::Object, 746
  - ~Object, 748
  - Object, 748
  - operator<<, 749
  - operator=, 749
  - Print, 749
  - Register, 749
  - SmartPointer, 750
  - UnRegister, 749
- gdcmm::OpenSSLCryptoFactory, 750
  - CreateCMSProvider, 751
  - InitOpenSSL, 751
  - OpenSSLCryptoFactory, 751
- gdcmm::OpenSSLCryptographicMessageSyntax, 752
  - ~OpenSSLCryptographicMessageSyntax, 753
  - Decrypt, 753
  - Encrypt, 753
  - GetCipherType, 754
  - OpenSSLCryptographicMessageSyntax, 753
  - ParseCertificateFile, 754
  - ParseKeyFile, 754
  - SetCipherType, 754
  - SetPassword, 754
- gdcmm::OpenSSL7CryptoFactory, 755
  - CreateCMSProvider, 756
  - OpenSSL7CryptoFactory, 756
- gdcmm::OpenSSL7CryptographicMessageSyntax, 757
  - ~OpenSSL7CryptographicMessageSyntax, 758



- Decrypt, [758](#)
- Encrypt, [758](#)
- GetCipherType, [759](#)
- OpenSSLP7CryptographicMessageSyntax, [758](#)
- ParseCertificateFile, [759](#)
- ParseKeyFile, [759](#)
- SetCipherType, [759](#)
- SetPassword, [759](#)
- gdcmm::Orientation, [760](#)
  - ~Orientation, [761](#)
  - AXIAL, [761](#)
  - CORONAL, [761](#)
  - GetLabel, [762](#)
  - GetMajorAxisFromPatientRelativeDirectionCosine, [762](#)
  - GetObliquityThresholdCosineValue, [762](#)
  - GetType, [762](#)
  - OBLIQUE, [761](#)
  - operator<=, [763](#)
  - Orientation, [761](#)
  - OrientationType, [761](#)
  - Print, [762](#)
  - SAGITTAL, [761](#)
  - SetObliquityThresholdCosineValue, [763](#)
  - UNKNOWN, [761](#)
- gdcmm::Overlay, [763](#)
  - ~Overlay, [767](#)
  - Decompress, [767](#)
  - GetBitPosition, [767](#)
  - GetBitsAllocated, [767](#)
  - GetColumns, [768](#)
  - GetDescription, [768](#)
  - GetGroup, [768](#)
  - GetOrigin, [768](#)
  - GetOverlayData, [768](#)
  - GetOverlayTypeAsString, [768](#)
  - GetOverlayTypeFromString, [769](#)
  - GetRows, [769](#)
  - GetType, [769](#)
  - GetTypeAsEnum, [769](#)
  - GetUnpackBuffer, [769](#)
  - GetUnpackBufferLength, [769](#)
  - GrabOverlayFromPixelData, [770](#)
  - Graphics, [766](#)
  - Invalid, [766](#)
  - IsEmpty, [770](#)
  - IsInPixelData, [770](#)
  - IsZero, [770](#)
  - operator=, [770](#)
  - Overlay, [767](#)
  - OverlayType, [766](#)
  - Print, [771](#)
  - ROI, [766](#)
  - SetBitPosition, [771](#)
  - SetBitsAllocated, [771](#)
  - SetColumns, [771](#)
  - SetDescription, [771](#)
  - SetFrameOrigin, [772](#)
  - SetGroup, [772](#)
  - SetNumberOfFrames, [772](#)
  - SetOrigin, [772](#)
  - SetOverlay, [772](#)
  - SetRows, [773](#)
  - SetType, [773](#)
  - Update, [773](#)
- gdcmm::ParseException, [774](#)
  - ~ParseException, [775](#)
  - GetLastElement, [775](#)
  - operator=, [775](#)
  - ParseException, [775](#)
  - SetLastElement, [775](#)
- gdcmm::Parser, [776](#)
  - ~Parser, [778](#)
  - DuplicateAttributeError, [778](#)
  - EndElementHandler, [777](#)
  - ErrorType, [777](#)
  - GetBuffer, [778](#)
  - GetCurrentByteIndex, [778](#)
  - GetErrorCode, [778](#)
  - GetErrorString, [779](#)
  - GetUserData, [779](#)
  - JunkAfterDocElementError, [778](#)
  - NoElementsError, [778](#)
  - NoError, [778](#)
  - NoMemoryError, [778](#)
  - Parse, [779](#)
  - ParseBuffer, [779](#)
  - Parser, [778](#)
  - Process, [779](#)
  - SetElementHandler, [779](#)
  - SetUserData, [780](#)
  - StartElementHandler, [777](#)
  - SyntaxError, [778](#)
  - TagMismatchError, [778](#)
  - UndefinedEntityError, [778](#)
  - UnexpectedStateError, [778](#)
- gdcmm::Patient, [780](#)
  - Patient, [780](#)
- gdcmm::PDBElement, [784](#)
  - GetName, [785](#)
  - GetValue, [786](#)
  - NameField, [787](#)
  - operator<=, [786](#)
  - operator==, [786](#)
  - PDBElement, [785](#)
  - SetName, [786](#)
  - SetValue, [786](#)
  - ValueField, [787](#)

gdcmm::PDBHeader, 787  
     ~PDBHeader, 788  
     FindPDBELEMENTByName, 789  
     GetPDBEEnd, 789  
     GetPDBELEMENTByName, 789  
     GetPDBInfoTag, 789  
     LoadFromDataElement, 789  
     operator<<, 790  
     PDBHeader, 788  
     Print, 790  
 gdcmm::PDFCodec, 790  
     ~PDFCodec, 792  
     CanCode, 792  
     CanDecode, 792  
     Decode, 792  
     PDFCodec, 792  
 gdcmm::PersonName, 796  
     Component, 798  
     GetMaxLength, 797  
     GetNumberOfComponents, 797  
     MaxLength, 798  
     MaxNumberOfComponents, 798  
     Padding, 799  
     Print, 797  
     Separator, 799  
     SetBlob, 798  
     SetComponents, 798  
 gdcmm::PGXCodec, 799  
     ~PGXCodec, 800  
     CanCode, 801  
     CanDecode, 801  
     Clone, 801  
     GetHeaderInfo, 801  
     PGXCodec, 800  
     Read, 801  
     Write, 802  
 gdcmm::PhotometricInterpretation, 802  
     ARGB, 804  
     CMYK, 804  
     GetPIString, 804  
     GetPIType, 804  
     GetSamplesPerPixel, 805  
     GetString, 805  
     GetType, 805  
     HSV, 804  
     IsLossless, 805  
     IsLossy, 805  
     IsRetired, 805  
     IsSameColorSpace, 805  
     MONOCHROME1, 804  
     MONOCHROME2, 804  
     operator PIType, 806  
     operator<<, 806  
     PALETTE\_COLOR, 804  
     PhotometricInterpretation, 804  
     PI\_END, 804  
     PIType, 803  
     RGB, 804  
     UNKNOWN, 804  
     YBR\_FULL, 804  
     YBR\_FULL\_422, 804  
     YBR\_ICT, 804  
     YBR\_PARTIAL\_420, 804  
     YBR\_PARTIAL\_422, 804  
     YBR\_RCT, 804  
 gdcmm::PixelFormat, 806  
     Bitmap, 814  
     FLOAT16, 808  
     FLOAT32, 808  
     FLOAT64, 808  
     GetBitsAllocated, 809  
     GetBitsStored, 809  
     GetHighBit, 810  
     GetMax, 810  
     GetMin, 810  
     GetPixelRepresentation, 810  
     GetPixelSize, 810  
     GetSamplesPerPixel, 811  
     GetScalarType, 811  
     GetScalarTypeAsString, 811  
     INT12, 808  
     INT16, 808  
     INT32, 808  
     INT64, 808  
     INT8, 808  
     IsCompatible, 811  
     IsValid, 812  
     operator ScalarType, 812  
     operator!=, 812  
     operator<<, 814  
     operator==, 812  
     PixelFormat, 809  
     Print, 813  
     ScalarType, 808  
     SetBitsAllocated, 813  
     SetBitsStored, 813  
     SetHighBit, 813  
     SetPixelRepresentation, 813  
     SetSamplesPerPixel, 813  
     SetScalarType, 814  
     SINGLEBIT, 808  
     UINT12, 808  
     UINT16, 808  
     UINT32, 808  
     UINT64, 808  
     UINT8, 808  
     UNKNOWN, 808  
     Validate, 814

- gdcm::Pixmap, 815
  - ~Pixmap, 817
  - AreOverlaysInPixelData, 817
  - Curves, 820
  - GetCurve, 817, 818
  - GetIconImage, 818
  - GetNumberOfCurves, 818
  - GetNumberOfOverlays, 818
  - GetOverlay, 818, 819
  - Icon, 820
  - Overlays, 820
  - Pixmap, 817
  - Print, 819
  - RemoveOverlay, 819
  - SetIconImage, 819
  - SetNumberOfCurves, 819
  - SetNumberOfOverlays, 819
  - UnusedBitsPresentInPixelData, 820
- gdcm::PixmapReader, 821
  - ~PixmapReader, 823
  - GetPixmap, 823
  - PixelData, 825
  - PixmapReader, 823
  - Read, 824
  - ReadACRNEMAIImage, 824
  - ReadImage, 824
  - ReadImageInternal, 824
- gdcm::PixmapToPixmapFilter, 825
  - ~PixmapToPixmapFilter, 827
  - GetInput, 827
  - GetOutput, 827
  - GetOutputAsPixmap, 827
  - PixmapToPixmapFilter, 827
- gdcm::PixmapWriter, 828
  - ~PixmapWriter, 830
  - DolconImage, 830
  - GetImage, 831
  - GetPixmap, 831
  - PixelData, 832
  - PixmapWriter, 830
  - PrepareWrite, 831
  - SetImage, 831
  - SetPixmap, 832
  - Write, 832
- gdcm::PNMCodec, 833
  - ~PNMCodec, 834
  - CanCode, 834
  - CanDecode, 835
  - Clone, 835
  - GetBufferLength, 835
  - GetHeaderInfo, 835
  - PNMCodec, 834
  - Read, 835
  - SetBufferLength, 836
  - Write, 836
- gdcm::Preamble, 836
  - ~Preamble, 838
  - Clear, 838
  - Create, 838
  - GetInternal, 838
  - GetLength, 838
  - IsEmpty, 839
  - IsValid, 839
  - operator<<, 840
  - operator=, 839
  - Preamble, 837, 838
  - Print, 839
  - Read, 839
  - Remove, 839
  - Valid, 840
  - Write, 840
- gdcm::PresentationContext, 841
  - AbstractSyntax, 844
  - AddTransferSyntax, 843
  - GetAbstractSyntax, 843
  - GetNumberOfTransferSyntaxes, 843
  - GetPresentationContextID, 843
  - GetTransferSyntax, 843
  - ID, 844
  - operator==, 843
  - PresentationContext, 842
  - Print, 844
  - SetAbstractSyntax, 844
  - SetPresentationContextID, 844
  - SizeType, 842
  - TransferSyntaxArrayType, 842
  - TransferSyntaxes, 844
- gdcm::PresentationContextGenerator, 847
  - AddFromFile, 849
  - AddPresentationContext, 849
  - GenerateFromFilenames, 849
  - GenerateFromUID, 850
  - GetDefaultTransferSyntax, 850
  - GetPresentationContexts, 850
  - PresentationContextArrayType, 849
  - PresentationContextGenerator, 849
  - SetDefaultTransferSyntax, 850
  - SetMergeModeToAbstractSyntax, 850
  - SetMergeModeToTransferSyntax, 851
  - SizeType, 849
- gdcm::Printer, 859
  - ~Printer, 861
  - CONDENSED\_STYLE, 861
  - CXX, 861
  - F, 864
  - GetPrintStyle, 862
  - MaxPrintLength, 864
  - Print, 862

- PrintDataElement, [862](#)
- PrintDataSet, [862](#)
- Printer, [861](#)
- PrintSQ, [863](#)
- PrintStyle, [864](#)
- PrintStyles, [861](#)
- SetColor, [863](#)
- SetFile, [863](#)
- SetStyle, [863](#)
- VERBOSE\_STYLE, [861](#)
- XML, [861](#)
- gdcmm::PrivateDict, [864](#)
  - ~PrivateDict, [865](#)
  - AddDictEntry, [865](#)
  - Dicts, [867](#)
  - FindDictEntry, [865](#)
  - GetDictEntry, [866](#)
  - IsEmpty, [866](#)
  - LoadDefault, [866](#)
  - operator<, [867](#)
  - PrintXML, [866](#)
  - PrivateDict, [865](#)
  - RemoveDictEntry, [866](#)
- gdcmm::PrivateTag, [867](#)
  - GetAsDataElement, [869](#)
  - GetOwner, [869](#)
  - operator<, [869](#)
  - operator<=, [870](#)
  - PrivateTag, [868](#), [869](#)
  - ReadFromCommaSeparatedString, [869](#)
  - SetOwner, [870](#)
- gdcmm::ProgressEvent, [870](#)
  - ~ProgressEvent, [872](#)
  - CheckEvent, [872](#)
  - GetEventName, [873](#)
  - GetProgress, [873](#)
  - MakeObject, [873](#)
  - operator=, [873](#)
  - ProgressEvent, [872](#)
  - Self, [872](#)
  - SetProgress, [873](#)
  - Superclass, [872](#)
- gdcmm::PVRGCodec, [874](#)
  - ~PVRGCodec, [875](#)
  - CanCode, [875](#)
  - CanDecode, [876](#)
  - Clone, [876](#)
  - Code, [876](#)
  - Decode, [876](#)
  - PVRGCodec, [875](#)
  - SetLossyFlag, [877](#)
- gdcmm::PythonFilter, [877](#)
  - ~PythonFilter, [878](#)
  - GetFile, [878](#)
  - PythonFilter, [878](#)
  - SetDicts, [878](#)
  - SetFile, [878](#)
  - ToPyObject, [878](#)
  - UseDictAlways, [879](#)
- gdcmm::QueryBase, [879](#)
  - ~QueryBase, [880](#)
  - GetAllRequiredTags, [880](#)
  - GetAllTags, [880](#)
  - GetHierarchicalSearchTags, [881](#)
  - GetName, [881](#)
  - GetOptionalTags, [881](#)
  - GetQueryLevel, [881](#)
  - GetRequiredTags, [881](#)
  - GetUniqueTags, [882](#)
- gdcmm::QueryFactory, [882](#)
  - GetCharacterFromCurrentLocale, [883](#)
  - ListCharSets, [883](#)
  - ProduceCharacterSetDataElement, [883](#)
  - ProduceQuery, [883](#)
- gdcmm::QueryImage, [884](#)
  - GetHierarchicalSearchTags, [885](#)
  - GetName, [885](#)
  - GetOptionalTags, [885](#)
  - GetQueryLevel, [885](#)
  - GetRequiredTags, [886](#)
  - GetUniqueTags, [886](#)
- gdcmm::QueryPatient, [886](#)
  - GetHierarchicalSearchTags, [887](#)
  - GetName, [887](#)
  - GetOptionalTags, [888](#)
  - GetQueryLevel, [888](#)
  - GetRequiredTags, [888](#)
  - GetUniqueTags, [888](#)
- gdcmm::QuerySeries, [889](#)
  - GetHierarchicalSearchTags, [890](#)
  - GetName, [890](#)
  - GetOptionalTags, [890](#)
  - GetQueryLevel, [890](#)
  - GetRequiredTags, [890](#)
  - GetUniqueTags, [891](#)
- gdcmm::QueryStudy, [891](#)
  - GetHierarchicalSearchTags, [892](#)
  - GetName, [892](#)
  - GetOptionalTags, [893](#)
  - GetQueryLevel, [893](#)
  - GetRequiredTags, [893](#)
  - GetUniqueTags, [893](#)
- gdcmm::RAWCodec, [894](#)
  - ~RAWCodec, [895](#)
  - CanCode, [895](#)
  - CanDecode, [896](#)
  - Clone, [896](#)
  - Code, [896](#)

- Decode, [896](#)
- DecodeByStreams, [897](#)
- DecodeBytes, [897](#)
- GetHeaderInfo, [897](#)
- RAWCodec, [895](#)
- gdcm::Reader, [898](#)
  - ~Reader, [901](#)
  - CanRead, [901](#)
  - F, [905](#)
  - GetFile, [901](#)
  - GetStreamCurrentPosition, [902](#)
  - GetStreamPtr, [902](#)
  - Read, [902](#)
  - ReadDataSet, [902](#)
  - Reader, [901](#)
  - ReadMetaInformation, [903](#)
  - ReadPreamble, [903](#)
  - ReadSelectedPrivateTags, [903](#)
  - ReadSelectedTags, [903](#)
  - ReadUpToTag, [903](#)
  - SetFile, [904](#)
  - SetFileName, [904](#)
  - SetStream, [904](#)
  - StreamImageReader, [905](#)
- gdcm::RealWorldValueMappingContent, [906](#)
  - CodeMeaning, [906](#)
  - CodeValue, [906](#)
  - RealWorldValueIntercept, [907](#)
  - RealWorldValueSlope, [907](#)
- gdcm::Region, [907](#)
  - ~Region, [908](#)
  - Area, [908](#)
  - Clone, [909](#)
  - ComputeBoundingBox, [909](#)
  - Empty, [909](#)
  - IsValid, [909](#)
  - Print, [909](#)
  - Region, [908](#)
- gdcm::Rescaler, [910](#)
  - ~Rescaler, [912](#)
  - ComputeInterceptSlopePixelType, [912](#)
  - ComputePixelTypeFromMinMax, [912](#)
  - GetIntercept, [912](#)
  - GetSlope, [912](#)
  - InverseRescale, [912](#)
  - InverseRescaleFunctionIntoBestFit, [913](#)
  - Rescale, [913](#)
  - RescaleFunctionIntoBestFit, [913](#)
  - Rescaler, [911](#)
  - SetIntercept, [913](#)
  - SetMinMaxForPixelType, [913](#)
  - SetPixelFormat, [914](#)
  - SetSlope, [914](#)
  - SetTargetPixelType, [914](#)
  - SetUseTargetPixelType, [914](#)
- gdcm::RLECodec, [915](#)
  - ~RLECodec, [917](#)
  - AppendFrameEncode, [917](#)
  - AppendRowEncode, [917](#)
  - CanCode, [917](#)
  - CanDecode, [917](#)
  - Clone, [918](#)
  - Code, [918](#)
  - Decode, [918](#)
  - DecodeByStreams, [918](#)
  - DecodeExtent, [919](#)
  - GetBufferLength, [919](#)
  - GetHeaderInfo, [919](#)
  - ImageRegionReader, [920](#)
  - IsFrameEncoder, [919](#)
  - IsRowEncoder, [919](#)
  - RLECodec, [916](#)
  - SetBufferLength, [920](#)
  - SetLength, [920](#)
  - StartEncode, [920](#)
  - StopEncode, [920](#)
- gdcm::Scanner, [923](#)
  - ~Scanner, [926](#)
  - AddPrivateTag, [926](#)
  - AddSkipTag, [927](#)
  - AddTag, [927](#)
  - Begin, [927](#)
  - ClearSkipTags, [927](#)
  - ClearTags, [927](#)
  - ConstIterator, [925](#)
  - End, [927](#)
  - GetAllFilenamesFromTagToValue, [928](#)
  - GetFilenameFromTagToValue, [928](#)
  - GetFilenames, [928](#)
  - GetKeys, [928](#)
  - GetMapping, [928](#)
  - GetMappingFromTagToValue, [929](#)
  - GetMapping, [929](#)
  - GetOrderedValues, [929](#)
  - GetValue, [929](#)
  - GetValues, [929](#), [930](#)
  - IsKey, [930](#)
  - MappingType, [925](#)
  - New, [930](#)
  - operator<=, [931](#)
  - Print, [930](#)
  - PrintTable, [931](#)
  - ProcessPublicTag, [931](#)
  - Scan, [931](#)
  - Scanner, [926](#)
  - TagToValue, [926](#)
  - TagToValueValueType, [926](#)
  - ValueType, [926](#)

gdcmm::Scanner::ltstr, 663  
     operator(), 663  
 gdcmm::Segment, 932  
     ~Segment, 935  
     AddSurface, 935  
     ALGOType, 934  
     ALGOType\_END, 935  
     AnatomicRegion, 940  
     AnatomicRegionModifiers, 940  
     AUTOMATIC, 935  
     BasicCodedEntryVector, 934  
     GetALGOType, 935  
     GetALGOTypeString, 935  
     GetAnatomicRegion, 936  
     GetAnatomicRegionModifiers, 936  
     GetPropertyCategory, 936  
     GetPropertyType, 936, 937  
     GetPropertyTypeModifiers, 937  
     GetSegmentAlgorithmName, 937  
     GetSegmentAlgorithmType, 937  
     GetSegmentDescription, 937  
     GetSegmentLabel, 937  
     GetSegmentNumber, 938  
     GetSurface, 938  
     GetSurfaceCount, 938  
     GetSurfaces, 938  
     MANUAL, 935  
     PropertyCategory, 941  
     PropertyType, 941  
     PropertyTypeModifiers, 941  
     Segment, 935  
     SegmentAlgorithmName, 941  
     SegmentAlgorithmType, 941  
     SegmentDescription, 941  
     SegmentLabel, 941  
     SegmentNumber, 942  
     SEMIAUTOMATIC, 935  
     SetAnatomicRegion, 938  
     SetAnatomicRegionModifiers, 938  
     SetPropertyCategory, 939  
     SetPropertyType, 939  
     SetPropertyTypeModifiers, 939  
     SetSegmentAlgorithmName, 939  
     SetSegmentAlgorithmType, 939  
     SetSegmentDescription, 940  
     SetSegmentLabel, 940  
     SetSegmentNumber, 940  
     SetSurfaceCount, 940  
     SurfaceCount, 942  
     Surfaces, 942  
     SurfaceVector, 934  
 gdcmm::SegmentedPaletteColorLookupTable, 942  
     ~SegmentedPaletteColorLookupTable, 943  
     Print, 944  
     SegmentedPaletteColorLookupTable, 943  
     SetLUT, 944  
 gdcmm::SegmentHelper, 84  
 gdcmm::SegmentHelper::BasicCodedEntry, 188  
     BasicCodedEntry, 189  
     CM, 190  
     CSD, 190  
     CSV, 190  
     CV, 191  
     IsEmpty, 190  
 gdcmm::SegmentReader, 945  
     ~SegmentReader, 947  
     GetSegments, 947  
     Read, 947  
     ReadSegment, 948  
     ReadSegments, 948  
     SegmentMap, 946  
     SegmentReader, 947  
     Segments, 948  
     SegmentVector, 947  
 gdcmm::SegmentWriter, 949  
     ~SegmentWriter, 950  
     AddSegment, 951  
     GetNumberOfSegments, 951  
     GetSegment, 951  
     GetSegments, 951  
     PrepareWrite, 951  
     Segments, 952  
     SegmentVector, 950  
     SegmentWriter, 950  
     SetNumberOfSegments, 951  
     SetSegments, 952  
     Write, 952  
 gdcmm::SequenceOfFragments, 953  
     AddFragment, 955  
     Begin, 956  
     Clear, 956  
     ComputeByteLength, 956  
     ComputeLength, 956  
     ConstIterator, 955  
     End, 956, 957  
     FragmentVector, 955  
     GetBuffer, 957  
     GetFragBuffer, 957  
     GetFragment, 957  
     GetLength, 957  
     GetNumberOfFragments, 958  
     GetTable, 958  
     Iterator, 955  
     New, 958  
     operator==, 958  
     Print, 958  
     Read, 959  
     ReadPreValue, 959

- ReadValue, [959](#)
- SequenceOfFragments, [955](#)
- SetLength, [959](#)
- SizeType, [955](#)
- Write, [959](#)
- WriteBuffer, [960](#)
- gdcmm::SequenceOfItems, [960](#)
  - AddItem, [964](#)
  - AddNewUndefinedLengthItem, [964](#)
  - Begin, [964](#)
  - Clear, [964](#)
  - ComputeLength, [965](#)
  - ConstIterator, [963](#)
  - End, [965](#)
  - FindDataElement, [965](#)
  - GetItem, [965](#)
  - GetLength, [966](#)
  - GetNumberOfItems, [966](#)
  - IsEmpty, [966](#)
  - IsUndefinedLength, [966](#)
  - Items, [968](#)
  - ItemVector, [963](#)
  - Iterator, [963](#)
  - New, [966](#)
  - operator=, [967](#)
  - operator==, [967](#)
  - Print, [967](#)
  - Read, [967](#)
  - RemoveItemByIndex, [967](#)
  - SequenceLengthField, [969](#)
  - SequenceOfItems, [963](#)
  - SetLength, [968](#)
  - SetLengthToUndefined, [968](#)
  - SetNumberOfItems, [968](#)
  - SizeType, [963](#)
  - Write, [968](#)
- gdcmm::SerieHelper, [969](#)
  - ~SerieHelper, [971](#)
  - AddFile, [971](#)
  - AddFileName, [972](#)
  - AddRestriction, [972](#)
  - Clear, [972](#)
  - CreateDefaultUniqueSeriesIdentifier, [972](#)
  - CreateUniqueSeriesIdentifier, [973](#)
  - elem, [974](#)
  - FileNameOrdering, [973](#)
  - GetFirstSingleSerieUIDFileSet, [973](#)
  - GetNextSingleSerieUIDFileSet, [973](#)
  - ImageNumberOrdering, [973](#)
  - ImagePositionPatientOrdering, [973](#)
  - ItFileSetHt, [974](#)
  - op, [975](#)
  - OrderFileList, [973](#)
  - Rule, [971](#)
  - SerieHelper, [971](#)
  - SerieRestrictions, [971](#)
  - SetDirectory, [974](#)
  - SetLoadMode, [974](#)
  - SetUseSeriesDetails, [974](#)
  - SingleSerieUIDFileSetHT, [975](#)
  - SingleSerieUIDFileSetmap, [971](#)
  - UserOrdering, [974](#)
  - value, [975](#)
- gdcmm::Series, [975](#)
  - Series, [976](#)
- gdcmm::ServiceClassUser, [978](#)
  - ~ServiceClassUser, [980](#)
  - GetAETitle, [981](#)
  - GetCalledAETitle, [981](#)
  - GetTimeout, [981](#)
  - InitializeConnection, [981](#)
  - IsPresentationContextAccepted, [981](#)
  - New, [981](#)
  - operator=, [982](#)
  - SendEcho, [982](#)
  - SendFind, [982](#)
  - SendMove, [982](#), [983](#)
  - SendStore, [983](#)
  - ServiceClassUser, [980](#)
  - SetAETitle, [983](#)
  - SetCalledAETitle, [984](#)
  - SetHostname, [984](#)
  - SetPort, [984](#)
  - SetPortSCP, [984](#)
  - SetPresentationContexts, [985](#)
  - SetTimeout, [985](#)
  - StartAssociation, [985](#)
  - StopAssociation, [985](#)
- gdcmm::SHA1, [986](#)
  - ~SHA1, [987](#)
  - Compute, [987](#)
  - ComputeFile, [987](#)
  - operator=, [987](#)
  - SHA1, [987](#)
- gdcmm::SimpleMemberCommand< T >, [988](#)
  - ~SimpleMemberCommand, [991](#)
  - Execute, [991](#)
  - m\_MemberFunction, [992](#)
  - m\_This, [992](#)
  - New, [991](#)
  - operator=, [992](#)
  - Self, [990](#)
  - SetCallbackFunction, [992](#)
  - SimpleMemberCommand, [990](#)
  - TMemberFunctionPointer, [990](#)
- gdcmm::SimpleSubjectWatcher, [993](#)
  - ~SimpleSubjectWatcher, [994](#)
  - EndFilter, [994](#)



- operator=, [994](#)
- ShowAbort, [994](#)
- ShowAnonymization, [995](#)
- ShowData, [995](#)
- ShowDataSet, [995](#)
- ShowFileName, [995](#)
- ShowIteration, [995](#)
- ShowProgress, [996](#)
- SimpleSubjectWatcher, [994](#)
- StartFilter, [996](#)
- TestAbortOff, [996](#)
- TestAbortOn, [996](#)
- gdcm::SmartPointer< ObjectType >, [999](#)
  - ~SmartPointer, [1001](#)
  - GetPointer, [1001](#)
  - operator ObjectType \*, [1002](#)
  - operator\*, [1002](#)
  - operator->, [1002](#)
  - operator=, [1002](#), [1003](#)
  - SmartPointer, [1000](#), [1001](#)
- gdcm::SOPClassUIDToIOD, [1005](#)
  - const, [1005](#)
  - GetIOD, [1006](#)
  - GetIODFromSOPClassUID, [1006](#)
  - GetNumberOfSOPClassToIOD, [1006](#)
  - GetSOPClassUIDFromIOD, [1006](#)
  - GetSOPClassUIDToIOD, [1006](#)
  - GetSOPClassUIDToIODs, [1006](#)
- gdcm::Sorter, [1007](#)
  - ~Sorter, [1009](#)
  - AddSelect, [1009](#)
  - FileNames, [1011](#)
  - GetFileNames, [1009](#)
  - operator<=, [1011](#)
  - Print, [1010](#)
  - Selection, [1011](#)
  - SelectionMap, [1009](#)
  - SetSortFunction, [1010](#)
  - SetTagsToRead, [1010](#)
  - Sort, [1010](#)
  - Sorter, [1009](#)
  - SortFunc, [1012](#)
  - SortFunction, [1009](#)
  - StableSort, [1011](#)
  - TagsToRead, [1012](#)
- gdcm::Spacing, [1012](#)
  - ~Spacing, [1014](#)
  - CALIBRATED, [1014](#)
  - ComputePixelAspectRatioFromPixelSpacing, [1014](#)
  - DETECTOR, [1014](#)
  - MAGNIFIED, [1014](#)
  - Spacing, [1014](#)
  - SpacingType, [1013](#)
  - UNKNOWN, [1014](#)
- gdcm::Spectroscopy, [1014](#)
  - Spectroscopy, [1015](#)
- gdcm::SplitMosaicFilter, [1015](#)
  - ~SplitMosaicFilter, [1016](#)
  - ComputeMOSAICDimensions, [1016](#)
  - ComputeMOSAICSliceNormal, [1016](#)
  - ComputeMOSAICSlicePosition, [1017](#)
  - GetAcquisitionSize, [1017](#)
  - GetFile, [1017](#)
  - GetImage, [1017](#)
  - GetNumberOfImagesInMosaic, [1018](#)
  - SetFile, [1018](#)
  - SetImage, [1018](#)
  - Split, [1018](#)
  - SplitMosaicFilter, [1016](#)
- gdcm::StartEvent, [1019](#)
- gdcm::static\_assert\_test< x >, [1020](#)
- gdcm::STATIC\_ASSERTION\_FAILURE< true >, [1020](#)
  - value, [1020](#)
- gdcm::STATIC\_ASSERTION\_FAILURE< x >, [1020](#)
- gdcm::StreamImageReader, [1021](#)
  - ~StreamImageReader, [1022](#)
  - CanReadImage, [1022](#)
  - DefinePixelExtent, [1022](#)
  - DefineProperBufferLength, [1022](#)
  - GetDimensionsValueForResolution, [1023](#)
  - GetFile, [1023](#)
  - Read, [1023](#)
  - ReadImageInformation, [1023](#)
  - SetFileName, [1024](#)
  - SetStream, [1024](#)
  - StreamImageReader, [1021](#)
- gdcm::StreamImageWriter, [1025](#)
  - ~StreamImageWriter, [1027](#)
  - CanWriteFile, [1027](#)
  - DefinePixelExtent, [1027](#)
  - DefineProperBufferLength, [1027](#)
  - mElementOffsets, [1030](#)
  - mElementOffsets1, [1030](#)
  - mSPFile, [1030](#)
  - mWriter, [1030](#)
  - mXMax, [1030](#)
  - mXMin, [1031](#)
  - mYMax, [1031](#)
  - mYMin, [1031](#)
  - mZMax, [1031](#)
  - mZMin, [1031](#)
  - SetFile, [1028](#)
  - SetFileName, [1028](#)
  - SetStream, [1028](#)
  - StreamImageWriter, [1027](#)
  - Write, [1028](#)
  - WriteImageInformation, [1029](#)
  - WriteImageSubregionRAW, [1029](#)



- WriteRawHeader, 1029
- gdcmm::StrictScanner, 1032
  - ~StrictScanner, 1035
  - AddPrivateTag, 1035
  - AddSkipTag, 1036
  - AddTag, 1036
  - Begin, 1036
  - ClearSkipTags, 1036
  - ClearTags, 1036
  - ConstIterator, 1034
  - End, 1036
  - GetAllFileNamesFromTagToValue, 1037
  - GetFilenameFromTagToValue, 1037
  - GetFileNames, 1037
  - GetKeys, 1037
  - GetMapping, 1037
  - GetMappingFromTagToValue, 1037
  - GetMappings, 1038
  - GetOrderedValues, 1038
  - GetValue, 1038
  - GetValues, 1038
  - IsKey, 1039
  - MappingType, 1034
  - New, 1039
  - operator<<, 1040
  - Print, 1039
  - PrintTable, 1039
  - ProcessPublicTag, 1039
  - Scan, 1040
  - StrictScanner, 1035
  - TagToValue, 1035
  - TagToValueValueType, 1035
  - ValueType, 1035
- gdcmm::StrictScanner::Itstr, 664
  - operator(), 664
- gdcmm::String< TDelimiter, TMaxLength, TPadChar >, 1041
  - const\_iterator, 1043
  - const\_reference, 1043
  - const\_reverse\_iterator, 1043
  - difference\_type, 1043
  - IsValid, 1045
  - iterator, 1043
  - operator const char \*, 1045
  - pointer, 1043
  - reference, 1044
  - reverse\_iterator, 1044
  - size\_type, 1044
  - String, 1044, 1045
  - Trim, 1045, 1046
  - Truncate, 1046
  - value\_type, 1044
- gdcmm::StringFilter, 1046
  - ~StringFilter, 1047
- ExecuteQuery, 1048
- FromString, 1048
- GetFile, 1048
- SetDicts, 1049
- SetFile, 1049
- StringFilter, 1047
- ToString, 1049
- ToStringPair, 1050
- UseDictAlways, 1050
- gdcmm::Study, 1051
  - Study, 1051
- gdcmm::Subject, 1051
  - ~Subject, 1053
  - AddObserver, 1053
  - GetCommand, 1054
  - HasObserver, 1054
  - InvokeEvent, 1054
  - RemoveAllObservers, 1054
  - RemoveObserver, 1054
  - Subject, 1053
- gdcmm::Surface, 1055
  - ~Surface, 1058
  - GetAlgorithmFamily, 1059
  - GetAlgorithmName, 1059
  - GetAlgorithmVersion, 1059
  - GetAxisOfRotation, 1059
  - GetCenterOfRotation, 1059
  - GetFiniteVolume, 1060
  - GetManifold, 1060
  - GetMaximumPointDistance, 1060
  - GetMeanPointDistance, 1060
  - GetMeshPrimitive, 1060
  - GetNumberOfSurfacePoints, 1061
  - GetNumberOfVectors, 1061
  - GetPointCoordinatesData, 1061
  - GetPointPositionAccuracy, 1061
  - GetPointsBoundingBoxCoordinates, 1061
  - GetProcessingAlgorithm, 1062
  - GetRecommendedDisplayCIELabValue, 1062
  - GetRecommendedDisplayGrayscaleValue, 1062
  - GetRecommendedPresentationOpacity, 1062
  - GetRecommendedPresentationType, 1063
  - GetSTATES, 1063
  - GetSTATESString, 1063
  - GetSurfaceComments, 1063
  - GetSurfaceNumber, 1063
  - GetSurfaceProcessing, 1063
  - GetSurfaceProcessingDescription, 1063
  - GetSurfaceProcessingRatio, 1064
  - GetVectorAccuracy, 1064
  - GetVectorCoordinateData, 1064
  - GetVectorDimensionality, 1064
  - GetVIEWType, 1064
  - GetVIEWTypeString, 1064

- NO, [1058](#)
- POINTS, [1058](#)
- SetAlgorithmFamily, [1065](#)
- SetAlgorithmName, [1065](#)
- SetAlgorithmVersion, [1065](#)
- SetAxisOfRotation, [1065](#)
- SetCenterOfRotation, [1065](#)
- SetFiniteVolume, [1065](#)
- SetManifold, [1066](#)
- SetMaximumPointDistance, [1066](#)
- SetMeanPointDistance, [1066](#)
- SetMeshPrimitive, [1066](#)
- SetNumberOfSurfacePoints, [1066](#)
- SetNumberOfVectors, [1066](#)
- SetPointCoordinatesData, [1067](#)
- SetPointPositionAccuracy, [1067](#)
- SetPointsBoundingBoxCoordinates, [1067](#)
- SetProcessingAlgorithm, [1067](#)
- SetRecommendedDisplayCIELabValue, [1067](#), [1068](#)
- SetRecommendedDisplayGrayscaleValue, [1068](#)
- SetRecommendedPresentationOpacity, [1068](#)
- SetRecommendedPresentationType, [1068](#)
- SetSurfaceComments, [1068](#)
- SetSurfaceNumber, [1068](#)
- SetSurfaceProcessing, [1069](#)
- SetSurfaceProcessingDescription, [1069](#)
- SetSurfaceProcessingRatio, [1069](#)
- SetVectorAccuracy, [1069](#)
- SetVectorCoordinateData, [1069](#)
- SetVectorDimensionality, [1069](#)
- STATES, [1058](#)
- STATES\_END, [1058](#)
- SURFACE, [1058](#)
- Surface, [1058](#)
- UNKNOWN, [1058](#)
- VIEWType, [1058](#)
- VIEWType\_END, [1058](#)
- WIREFRAME, [1058](#)
- YES, [1058](#)
- gdcmm::SurfaceHelper, [1070](#)
  - ColorArray, [1071](#)
  - RecommendedDisplayCIELabToRGB, [1071](#)
  - RGBToRecommendedDisplayCIELab, [1072](#)
  - RGBToRecommendedDisplayGrayscale, [1072](#)
- gdcmm::SurfaceReader, [1073](#)
  - ~SurfaceReader, [1075](#)
  - GetNumberOfSurfaces, [1075](#)
  - Read, [1076](#)
  - ReadPointMacro, [1076](#)
  - ReadSurface, [1076](#)
  - ReadSurfaces, [1076](#)
  - SurfaceReader, [1075](#)
- gdcmm::SurfaceWriter, [1077](#)
  - ~SurfaceWriter, [1078](#)
  - ComputeNumberOfSurfaces, [1078](#)
  - GetNumberOfSurfaces, [1078](#)
  - NumberOfSurfaces, [1079](#)
  - PrepareWrite, [1079](#)
  - PrepareWritePointMacro, [1079](#)
  - SetNumberOfSurfaces, [1079](#)
  - SurfaceWriter, [1078](#)
  - Write, [1079](#)
- gdcmm::SwapCode, [1080](#)
  - BadBigEndian, [1081](#)
  - BadLittleEndian, [1081](#)
  - BigEndian, [1081](#)
  - GetIndex, [1081](#)
  - GetSwapCodeString, [1081](#)
  - LittleEndian, [1081](#)
  - operator SwapCode::SwapCodeType, [1082](#)
  - operator<<, [1082](#)
  - SwapCode, [1081](#)
  - SwapCodeType, [1081](#)
  - Unknown, [1081](#)
- gdcmm::SwapperDoOp, [1082](#)
  - Swap, [1082](#)
  - SwapArray, [1083](#)
- gdcmm::SwapperNoOp, [1083](#)
  - Swap, [1083](#)
  - SwapArray, [1084](#)
- gdcmm::System, [1084](#)
  - ConvertToUNC, [1085](#)
  - DeleteDirectory, [1085](#)
  - EncodeBytes, [1086](#)
  - FileExists, [1086](#)
  - FileIsDirectory, [1086](#)
  - FileIsSymlink, [1086](#)
  - FileSize, [1087](#)
  - FileTime, [1087](#)
  - FormatDateTime, [1087](#)
  - GetCurrentDateTime, [1087](#)
  - GetCurrentModuleFileName, [1088](#)
  - GetCurrentProcessFileName, [1088](#)
  - GetCurrentResourcesDirectory, [1088](#)
  - GetCWD, [1088](#)
  - GetHostName, [1088](#)
  - GetLastSystemError, [1088](#)
  - GetLocaleCharset, [1089](#)
  - GetPermissions, [1089](#)
  - GetTimezoneOffsetFromUTC, [1089](#)
  - MakeDirectory, [1089](#)
  - ParseDateTime, [1089](#), [1090](#)
  - RemoveFile, [1090](#)
  - SetPermissions, [1090](#)
  - StrCaseCmp, [1090](#)
  - StrNCaseCmp, [1091](#)
  - StrSep, [1091](#)
  - StrTokR, [1091](#)

- gdcmm::Table, 1092
  - ~Table, 1093
  - GetTableEntry, 1093
  - InsertEntry, 1094
  - MapTableEntry, 1093
  - operator<<, 1094
  - operator=, 1094
  - Table, 1093
  - TableInternal, 1094
- gdcmm::TableEntry, 1095
  - ~TableEntry, 1095
  - TableEntry, 1095
- gdcmm::TableReader, 1096
  - ~TableReader, 1097
  - CharacterDataHandler, 1097
  - EndElement, 1097
  - GetDefs, 1097
  - GetFilename, 1097
  - HandleIOD, 1098
  - HandleIODEntry, 1098
  - HandleMacro, 1098
  - HandleMacroEntry, 1098
  - HandleMacroEntryDescription, 1098
  - HandleModule, 1098
  - HandleModuleEntry, 1099
  - HandleModuleEntryDescription, 1099
  - HandleModuleInclude, 1099
  - Read, 1099
  - SetFilename, 1099
  - StartElement, 1099
  - TableReader, 1097
- gdcmm::Tag, 1101
  - bytes, 1112
  - GetElement, 1104
  - GetElementTag, 1104
  - GetGroup, 1105
  - GetLength, 1105
  - GetPrivateCreator, 1105
  - IsGroupLength, 1105
  - IsGroupXX, 1105
  - IsIllegal, 1106
  - IsPrivate, 1106
  - IsPrivateCreator, 1106
  - IsPublic, 1106
  - operator!=, 1107
  - operator<, 1107
  - operator<<, 1111
  - operator<=, 1107
  - operator>>, 1111
  - operator=, 1107
  - operator==, 1107
  - operator[], 1108
  - PrintAsContinuousString, 1108
  - PrintAsContinuousUpperCaseString, 1108
  - PrintAsPipeSeparatedString, 1108
  - Read, 1109
  - ReadFromCommaSeparatedString, 1109
  - ReadFromContinuousString, 1109
  - ReadFromPipeSeparatedString, 1109
  - SetElement, 1109
  - SetElementTag, 1110
  - SetGroup, 1110
  - SetPrivateCreator, 1110
  - Tag, 1103, 1104
  - tag, 1112
  - tags, 1112
  - Write, 1111
- gdcmm::TagPath, 1112
  - ~TagPath, 1113
  - ConstructFromString, 1113
  - ConstructFromTagList, 1113
  - IsValid, 1114
  - Print, 1114
  - Push, 1114
  - TagPath, 1113
- gdcmm::terminal, 84
  - Attribute, 85
  - black, 85
  - blink, 85
  - blue, 85
  - bright, 85
  - Color, 85
  - CONSOLE, 86
  - cyan, 85
  - dim, 85
  - green, 85
  - hidden, 85
  - magenta, 85
  - Mode, 86
  - red, 85
  - reset, 85
  - reverse, 85
  - setattribute, 86
  - setbgcolor, 86
  - setfgcolor, 86
  - setmode, 86
  - underline, 85
  - VT100, 86
  - white, 85
  - yellow, 85
- gdcmm::Testing, 1115
  - ~Testing, 1116
  - ComputeFileMD5, 1117
  - ComputeMD5, 1117
  - GetDataExtraRoot, 1117
  - GetDataRoot, 1117
  - GetFileName, 1118
  - GetFileNames, 1118

- GetLossyFlagFromFile, 1118
- GetMD5DataImage, 1118
- GetMD5DataImages, 1119
- GetMD5FromBrokenFile, 1119
- GetMD5FromFile, 1119
- GetMediaStorageDataFile, 1119
- GetMediaStorageDataFiles, 1119
- GetMediaStorageFromFile, 1119
- GetNumberOfFileNames, 1120
- GetNumberOfMD5DataImages, 1120
- GetNumberOfMediaStorageDataFiles, 1120
- GetPixelSpacingDataRoot, 1120
- GetSelectedPrivateGroupOffsetFromFile, 1120
- GetSelectedTagsOffsetFromFile, 1121
- GetSourceDirectory, 1121
- GetStreamOffsetFromFile, 1121
- GetTempDirectory, 1121
- GetTempDirectoryW, 1121
- GetTempFilename, 1122
- GetTempFilenameW, 1122
- MD5DataImagesType, 1116
- MediaStorageDataFilesType, 1116
- Print, 1122
- Testing, 1116
- gdcmm::Trace, 1123
  - ~Trace, 1124
  - DebugOff, 1124
  - DebugOn, 1124
  - ErrorOff, 1125
  - ErrorOn, 1125
  - GetDebugFlag, 1125
  - GetDebugStream, 1125
  - GetErrorFlag, 1125
  - GetErrorStream, 1125
  - GetStream, 1126
  - GetWarningFlag, 1126
  - GetWarningStream, 1126
  - SetDebug, 1126
  - SetDebugStream, 1126
  - SetError, 1126
  - SetErrorStream, 1127
  - SetStream, 1127
  - SetStreamToFile, 1127
  - SetWarning, 1127
  - SetWarningStream, 1127
  - Trace, 1124
  - WarningOff, 1128
  - WarningOn, 1128
- gdcmm::TransferSyntax, 1128
  - CanStoreLossy, 1132
  - CT\_private\_ELE, 1131
  - DeflatedExplicitVRLittleEndian, 1131
  - Explicit, 1130
  - ExplicitVRBigEndian, 1131
  - ExplicitVRLittleEndian, 1131
  - GetNegociatedType, 1132
  - GetString, 1132
  - GetSwapCode, 1132
  - GetTSString, 1132
  - GetTSType, 1133
  - Implicit, 1130
  - ImplicitVRBigEndianACRNEMA, 1131
  - ImplicitVRBigEndianPrivateGE, 1131
  - ImplicitVRLittleEndian, 1131
  - IsEncapsulated, 1133
  - IsEncoded, 1133
  - IsExplicit, 1133
  - IsImplicit, 1133
  - IsLossless, 1134
  - IsLossy, 1134
  - IsValid, 1134
  - JPEG2000, 1131
  - JPEG2000Lossless, 1131
  - JPEG2000Part2, 1131
  - JPEG2000Part2Lossless, 1131
  - JPEGBaselineProcess1, 1131
  - JPEGExtendedProcess2\_4, 1131
  - JPEGExtendedProcess3\_5, 1131
  - JPEGFullProgressionProcess10\_12, 1131
  - JPEGLosslessProcess14, 1131
  - JPEGLosslessProcess14\_1, 1131
  - JPEGLSLossless, 1131
  - JPEGLSNearLossless, 1131
  - JPEGSpectralSelectionProcess6\_8, 1131
  - JPIPRreferenced, 1131
  - MPEG2MainProfile, 1131
  - MPEG2MainProfileHighLevel, 1131
  - MPEG4AVCH264BDcompatibleHighProfileLevel4\_1, 1131
  - MPEG4AVCH264HighProfileLevel4\_1, 1131
  - NegotiatedType, 1130
  - operator TSType, 1134
  - operator <<, 1134
  - RLELossless, 1131
  - TransferSyntax, 1131
  - TS\_END, 1131
  - TSType, 1131
  - Unknown, 1130
  - WeirdPapryus, 1131
- gdcmm::Type, 1139
  - GetTypeString, 1141
  - GetTypeType, 1141
  - operator TypeType, 1141
  - operator <<, 1141
  - T1, 1140
  - T1C, 1140
  - T2, 1140
  - T2C, 1140

- T3, [1140](#)
- Type, [1140](#)
- TypeType, [1140](#)
- UNKNOWN, [1140](#)
- gdcmm::UI, [1142](#)
  - Internal, [1142](#)
  - operator<<, [1142](#)
- gdcmm::UIDGenerator, [1142](#)
  - Generate, [1144](#)
  - GenerateUUID, [1144](#)
  - GetGDCMUID, [1144](#)
  - GetRoot, [1144](#)
  - IsValid, [1144](#)
  - SetRoot, [1145](#)
  - UIDGenerator, [1143](#)
- gdcmm::UIDs, [1145](#)
  - AbstractMultiDimensionalImageModel, [1171](#)
  - AcquisitionContextSRStorage, [1170](#)
  - AdultMouseAnatomyOntology, [1168](#)
  - AdvancedBlendingPresentationStateStorage, [1169](#)
  - AmbulatoryECGWaveformStorage, [1165](#)
  - ArterialPulseWaveformStorage, [1169](#)
  - AudioSRStorageTrialRetired, [1166](#)
  - AutorefractionMeasurementsStorage, [1169](#)
  - BasicAnnotationBoxSOPClass, [1164](#)
  - BasicColorImageBoxSOPClass, [1164](#)
  - BasicColorPrintManagementMetaSOPClass, [1164](#)
  - BasicFilmBoxSOPClass, [1164](#)
  - BasicFilmSessionSOPClass, [1164](#)
  - BasicGrayscaleImageBoxSOPClass, [1164](#)
  - BasicGrayscalePrintManagementMetaSOPClass, [1164](#)
  - BasicPrintImageOverlayBoxSOPClassRetired, [1164](#)
  - BasicStructuredDisplayStorage, [1170](#)
  - BasicStudyContentNotificationSOPClassRetired, [1163](#)
  - BasicTextSRStorage, [1166](#)
  - BasicVoiceAudioWaveformStorage, [1165](#)
  - BlendingSoftcopyPresentationStateStorageSOPClass, [1165](#)
  - BreastImagingRelevantPatientInformationQuery, [1167](#)
  - BreastProjectionXRayImageStorageForPresentation, [1169](#)
  - BreastProjectionXRayImageStorageForProcessing, [1169](#)
  - BreastTomosynthesisImageStorage, [1168](#)
  - CardiacElectrophysiologyWaveformStorage, [1165](#)
  - CardiacRelevantPatientInformationQuery, [1167](#)
  - ChestCADSRStorage, [1166](#)
  - ColonCADSRStorage, [1170](#)
  - ColorPaletteQueryRetrieveInformationModelFIND, [1171](#)
  - ColorPaletteQueryRetrieveInformationModelGET, [1171](#)
  - ColorPaletteQueryRetrieveInformationModelMOVE, [1171](#)
  - ColorPaletteStorage, [1171](#)
  - ColorSoftcopyPresentationStateStorageSOPClass, [1165](#)
  - CompositeInstanceRetrieveWithoutBulkDataGET, [1170](#)
  - CompositeInstanceRootRetrieveGET, [1170](#)
  - CompositeInstanceRootRetrieveMOVE, [1170](#)
  - CompositingPlanarMPRVolumetricPresentationStateStorage, [1169](#)
  - Comprehensive3DSRStorage, [1170](#)
  - ComprehensiveSRStorage, [1166](#)
  - ComprehensiveSRStorageTrialRetired, [1166](#)
  - ComputedRadiographylmageStorage, [1164](#)
  - ContentAssessmentResultsStorage, [1170](#)
  - CornealTopographyMapStorage, [1170](#)
  - CTDefinedProcedureProtocolStorage, [1170](#)
  - CTImageStorage, [1164](#)
  - CTPerformedProcedureProtocolStorage, [1170](#)
  - DefinedProcedureProtocolInformationModelFIND, [1170](#)
  - DefinedProcedureProtocolInformationModelGET, [1170](#)
  - DefinedProcedureProtocolInformationModelMOVE, [1170](#)
  - DeflatedExplicitVRLittleEndian, [1162](#)
  - DeformableSpatialRegistrationStorage, [1165](#)
  - DetachedInterpretationManagementSOPClassRetired, [1164](#)
  - DetachedPatientManagementMetaSOPClassRetired, [1163](#)
  - DetachedPatientManagementSOPClassRetired, [1163](#)
  - DetachedResultsManagementMetaSOPClassRetired, [1163](#)
  - DetachedResultsManagementSOPClassRetired, [1163](#)
  - DetachedStudyManagementMetaSOPClassRetired, [1163](#)
  - DetachedStudyManagementSOPClassRetired, [1163](#)
  - DetachedVisitManagementSOPClassRetired, [1163](#)
  - DetailSRStorageTrialRetired, [1166](#)
  - dicomAETitle, [1167](#)
  - dicomApplicationCluster, [1167](#)
  - DICOMApplicationContextName, [1163](#)
  - dicomAssociationAcceptor, [1167](#)
  - dicomAssociationInitiator, [1167](#)
  - dicomAuthorizedNodeCertificateReference, [1167](#)
  - dicomConfigurationRoot, [1168](#)
  - DICOMContentMappingResource, [1171](#)
  - DICOMControlledTerminology, [1163](#)

- dicomDescription, 1167
- dicomDevice, 1168
- dicomDeviceName, 1167
- dicomDeviceSerialNumber, 1168
- dicomDevicesRoot, 1168
- dicomHostname, 1167
- dicomInstalled, 1167
- dicomInstitutionAddress, 1168
- dicomInstitutionDepartmentName, 1168
- dicomInstitutionName, 1168
- dicomIssuerOfPatientID, 1168
- dicomManufacturer, 1167
- dicomManufacturerModelName, 1167
- dicomNetworkAE, 1168
- dicomNetworkConnection, 1168
- dicomNetworkConnectionReference, 1167
- dicomPort, 1167
- dicomPreferredCalledAETitle, 1167
- dicomPreferredCallingAETitle, 1168
- dicomPrimaryDeviceType, 1167
- dicomRelatedDeviceReference, 1167
- dicomSoftwareVersion, 1167
- dicomSOPClass, 1167
- dicomStationName, 1168
- dicomSupportedCharacterSet, 1168
- dicomThisNodeCertificateReference, 1167
- dicomTLSCyphersuite, 1167
- dicomTransferCapability, 1168
- dicomTransferRole, 1167
- dicomTransferSyntax, 1167
- DICOMUIDRegistry, 1163
- dicomUniqueAETitle, 1168
- dicomUniqueAETitlesRegistryRoot, 1168
- dicomVendorData, 1167
- DICOS2DAITStorage, 1170
- DICOS3DAITStorage, 1170
- DICOSCTImageStorage, 1170
- DICOSDigitalXRayImageStorageForPresentation, 1170
- DICOSDigitalXRayImageStorageForProcessing, 1170
- DICOSQuadrupoleResonanceQRStorage, 1170
- DICOSThreatDetectionReportStorage, 1170
- DigitalIntraoralXRayImageStorageForPresentation, 1164
- DigitalIntraoralXRayImageStorageForProcessing, 1164
- DigitalMammographyXRayImageStorageForPresentation, 1164
- DigitalMammographyXRayImageStorageForProcessing, 1164
- DigitalXRayImageStorageForPresentation, 1164
- DigitalXRayImageStorageForProcessing, 1164
- DisplaySystemSOPClass, 1169
- DisplaySystemSOPInstance, 1169
- ECG12leadWaveformStorage, 1165
- EddyCurrentImageStorage, 1170
- EddyCurrentMultiframeImageStorage, 1170
- EncapsulatedCDASStorage, 1166
- EncapsulatedPDFStorage, 1166
- EncapsulatedSTLStorage, 1170
- EnhancedCTImageStorage, 1164
- EnhancedMRColorImageStorage, 1171
- EnhancedMRIImageStorage, 1164
- EnhancedPETImageStorage, 1170
- EnhancedSRStorage, 1166
- EnhancedUSVolumeStorage, 1168
- EnhancedXAImageStorage, 1165
- EnhancedXRFImageStorage, 1165
- ExplicitVRBigEndian, 1162
- ExplicitVRLittleEndian, 1162
- ExtensibleSRStorage, 1170
- FallColorPaletteSOPInstance, 1168
- GeneralAudioWaveformStorage, 1169
- GeneralECGWaveformStorage, 1165
- GeneralPurposePerformedProcedureStepSOP-Class, 1167
- GeneralPurposeScheduledProcedureStepSOP-Class, 1167
- GeneralPurposeWorklistInformationModelFIND, 1166
- GeneralPurposeWorklistManagementMetaSOP-Class, 1167
- GeneralRelevantPatientInformationQuery, 1167
- GenericImplantTemplateInformationModelFIND, 1171
- GenericImplantTemplateInformationModelGET, 1171
- GenericImplantTemplateInformationModelMOVE, 1171
- GenericImplantTemplateStorage, 1171
- GetName, 1181
- GetNumberOfTransferSyntaxStrings, 1181
- GetString, 1181
- GetTransferSyntaxString, 1181
- GetTransferSyntaxStrings, 1181
- GetUIDName, 1182
- GetUIDString, 1182
- GrayscalePlanarMPRVolumetricPresentationStateStorage, 1169
- GrayscaleSoftcopyPresentationStateStorageSOP-Class, 1165
- HangingProtocolInformationModelFIND, 1167
- HangingProtocolInformationModelGET, 1171
- HangingProtocolInformationModelMOVE, 1167
- HangingProtocolStorage, 1167
- HardcopyColorImageStorageSOPClassRetired, 1164



- HardcopyGrayscaleImageStorageSOPClassRetired, 1164
- HemodynamicWaveformStorage, 1165
- HEVCH\_265Main10ProfileLevel5\_1, 1169
- HEVCH\_265MainProfileLevel5\_1, 1169
- HotIronColorPaletteSOPInstance, 1169
- HotMetalBlueColorPaletteSOPInstance, 1168
- ICBM452T1FrameofReference, 1163
- ICBMSingleSubjectMRIFrameofReference, 1163
- ICD11, 1168
- ImageBiomarkerStandardisationInitiative, 1168
- ImageOverlayBoxSOPClassRetired, 1164
- ImplantAssemblyTemplateInformationModelFIND, 1171
- ImplantAssemblyTemplateInformationModelGET, 1171
- ImplantAssemblyTemplateInformationModelMOVE, 1171
- ImplantAssemblyTemplateStorage, 1171
- ImplantationPlanSRStorage, 1170
- ImplantTemplateGroupInformationModelFIND, 1171
- ImplantTemplateGroupInformationModelGET, 1171
- ImplantTemplateGroupInformationModelMOVE, 1171
- ImplantTemplateGroupStorage, 1171
- ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM, 1162
- InstanceAvailabilityNotificationSOPClass, 1167
- IntegratedTaxonomicInformationSystemITISTaxonomicSerialNumberTSN, 1168
- IntraocularLensCalculationsStorage, 1169
- IntravascularOpticalCoherenceTomographyImageStorageForPresentation, 1169
- IntravascularOpticalCoherenceTomographyImageStorageForProcessing, 1169
- JPEG2000ImageCompression, 1162
- JPEG2000ImageCompressionLosslessOnly, 1162
- JPEG2000Part2MulticomponentImageCompression, 1162
- JPEG2000Part2MulticomponentImageCompressionLosslessOnly, Class, 1163
- JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImages, 1162
- JPEGExtendedHierarchicalProcess1618Retired, 1162
- JPEGExtendedHierarchicalProcess1719Retired, 1162
- JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG8BitImages, 1162
- JPEGExtendedProcess35Retired, 1162
- JPEGFullProgressionHierarchicalProcess2426Retired, 1162
- JPEGFullProgressionHierarchicalProcess2527Retired, 1162
- JPEGFullProgressionNonHierarchicalProcess1012Retired, 1162
- JPEGFullProgressionNonHierarchicalProcess1113Retired, 1162
- JPEGLosslessHierarchicalProcess28Retired, 1162
- JPEGLosslessHierarchicalProcess29Retired, 1162
- JPEGLosslessNonHierarchicalFirstOrderPrediction-Process14SelectionValue1DefaultTransferSyntaxforLosslessJPEG, 1162
- JPEGLosslessNonHierarchicalProcess14, 1162
- JPEGLosslessNonHierarchicalProcess15Retired, 1162
- JPEGLSLosslessImageCompression, 1162
- JPEGLSLossyNearLosslessImageCompression, 1162
- JPEGSpectralSelectionHierarchicalProcess2022Retired, 1162
- JPEGSpectralSelectionHierarchicalProcess2123Retired, 1162
- JPEGSpectralSelectionNonHierarchicalProcess68Retired, 1162
- JPEGSpectralSelectionNonHierarchicalProcess79Retired, 1162
- JPIPRReferenced, 1162
- JPIPRReferencedDeflate, 1162
- KeratometryMeasurementsStorage, 1169
- KeyObjectSelectionDocumentStorage, 1166
- LegacyConvertedEnhancedCTImageStorage, 1168
- LegacyConvertedEnhancedMRIImageStorage, 1168
- LegacyConvertedEnhancedPETImageStorage, 1168
- LensometryMeasurementsStorage, 1169
- MacularGridThicknessandVolumeReportStorage, 1169
- MammographyCADSRStorage, 1166
- MayoClinicNonradiologicalImagesSBSAnatomical-SurfaceRegionGuide, 1168
- MediaCreationManagementSOPClassUID, 1164
- MediaStorageDirectoryStorage, 1163
- ModalityPerformedProcedureStepNotificationSOP-ModalityPerformedProcedureStepRetrieveSOP-Class, 1163
- ModalityPerformedProcedureStepSOPClass, 1163
- ModalityWorklistInformationModelFIND, 1166
- MouseGenomeInitiativeMGI, 1168
- MPEG2MainProfileHighLevel, 1168
- MPEG2MainProfileMainLevel, 1162
- MPEG4AVCH\_264HighProfileLevel4\_1, 1168
- MPEG4AVCH\_264HighProfileLevel4\_2For2DVideo, 1169
- MPEG4AVCH\_264HighProfileLevel4\_2For3DVideo, 1169
- MPEG4AVCH\_264StereoHighProfileLevel4\_2, 1169

- MRImageStorage, [1164](#)
- MRSpectroscopyStorage, [1164](#)
- MultiframeGrayscaleByteSecondaryCaptureImageStorage, [1165](#)
- MultiframeGrayscaleWordSecondaryCaptureImageStorage, [1165](#)
- MultiframeSingleBitSecondaryCaptureImageStorage, [1165](#)
- MultiframeTrueColorSecondaryCaptureImageStorage, [1165](#)
- MultipleVolumeRenderingVolumetricPresentationStateStorage, [1169](#)
- NativeDICOMModel, [1171](#)
- NewYorkUniversityMelanomaClinicalCooperativeGroup, [1168](#)
- NuclearMedicineImageStorage, [1165](#)
- NuclearMedicineImageStorageRetired, [1165](#)
- Null0, [1169](#)
- Null1, [1169](#)
- operator TSType, [1182](#)
- OphthalmicAxialMeasurementsStorage, [1169](#)
- OphthalmicOpticalCoherenceTomographyBscanVolumeAnalysisStorage, [1169](#)
- OphthalmicOpticalCoherenceTomographyEnFaceImageStorage, [1169](#)
- OphthalmicPhotography16BitImageStorage, [1166](#)
- OphthalmicPhotography8BitImageStorage, [1166](#)
- OphthalmicThicknessMapStorage, [1169](#)
- OphthalmicTomographyImageStorage, [1166](#)
- OphthalmicVisualFieldStaticPerimetryMeasurementsStorage, [1169](#)
- Papyrus3ImplicitVRLittleEndian, [1168](#)
- ParametricMapStorage, [1169](#)
- PatientRadiationDoseSRStorage, [1170](#)
- PatientRootQueryRetrieveInformationModelFIND, [1166](#)
- PatientRootQueryRetrieveInformationModelGET, [1166](#)
- PatientRootQueryRetrieveInformationModelMOVE, [1166](#)
- PatientStudyOnlyQueryRetrieveInformationModelFINDRetired, [1166](#)
- PatientStudyOnlyQueryRetrieveInformationModelGETRetired, [1166](#)
- PatientStudyOnlyQueryRetrieveInformationModelMOVERetired, [1166](#)
- PerformedImagingAgentAdministrationSRStorage, [1170](#)
- PET20StepColorPaletteSOPInstance, [1168](#)
- PETColorPaletteSOPInstance, [1168](#)
- PlannedImagingAgentAdministrationSRStorage, [1170](#)
- PositronEmissionTomographyImageStorage, [1166](#)
- PresentationLUTSOPClass, [1164](#)
- PrinterConfigurationRetrievalSOPClass, [1164](#)
- PrinterConfigurationRetrievalSOPInstance, [1164](#)
- PrinterSOPClass, [1164](#)
- PrinterSOPInstance, [1164](#)
- PrintJobSOPClass, [1164](#)
- PrintQueueManagementSOPClassRetired, [1164](#)
- PrintQueueSOPInstanceRetired, [1164](#)
- ProceduralEventLoggingSOPClass, [1163](#)
- ProceduralEventLoggingSOPInstance, [1163](#)
- ProcedureLogStorage, [1166](#)
- ProductCharacteristicsQuerySOPClass, [1167](#)
- ProtocolApprovalInformationModelFIND, [1170](#)
- ProtocolApprovalInformationModelGET, [1170](#)
- ProtocolApprovalInformationModelMOVE, [1170](#)
- ProtocolApprovalStorage, [1170](#)
- PseudoColorSoftcopyPresentationStateStorageSOPClass, [1165](#)
- PubChemCompoundCID, [1168](#)
- PullPrintRequestSOPClassRetired, [1164](#)
- PullStoredPrintManagementMetaSOPClassRetired, [1164](#)
- RadiomicsOntology, [1168](#)
- RadiopharmaceuticalRadiationDoseSRStorage, [1170](#)
- RawDataStorage, [1165](#)
- RealWorldValueMappingStorage, [1165](#)
- ReferencedColorPrintManagementMetaSOPClassRetired, [1164](#)
- ReferencedGrayscalePrintManagementMetaSOPClassRetired, [1164](#)
- ReferencedImageBoxSOPClassRetired, [1164](#)
- RespiratoryWaveformStorage, [1169](#)
- RFC2557MIMEencapsulation, [1162](#)
- RLELossless, [1162](#)
- RTBeamsDeliveryInstructionStorage, [1171](#)
- RTBeamsDeliveryInstructionStorageSupplement74FrozenDraft, [1167](#)
- RTBeamsTreatmentRecordStorage, [1166](#)
- RTBrachyApplicationSetupDeliveryInstructionStorage, [1171](#)
- RTBrachyTreatmentRecordStorage, [1166](#)
- RTConventionalMachineVerification, [1171](#)
- RTConventionalMachineVerificationSupplement74FrozenDraft, [1167](#)
- RTDoseStorage, [1166](#)
- RTImageStorage, [1166](#)
- RTIonBeamsTreatmentRecordStorage, [1166](#)
- RTIonMachineVerification, [1171](#)
- RTIonMachineVerificationSupplement74FrozenDraft, [1167](#)
- RTIonPlanStorage, [1166](#)
- RTPhysicianIntentStorage, [1170](#)
- RTPlanStorage, [1166](#)
- RTSegmentAnnotationStorage, [1170](#)



- RTStructureSetStorage, [1166](#)
- RTTreatmentSummaryRecordStorage, [1166](#)
- SecondaryCaptureImageStorage, [1165](#)
- SegmentationStorage, [1165](#)
- SegmentedVolumeRenderingVolumetricPresentationStateStorage, [1169](#)
- SetFromUID, [1182](#)
- SimplifiedAdultEchoSRStorage, [1170](#)
- SpatialFiducialsStorage, [1165](#)
- SpatialRegistrationStorage, [1165](#)
- SpectaclePrescriptionReportStorage, [1169](#)
- SPM2AVG152PDFFrameofReference, [1163](#)
- SPM2AVG152T1FrameofReference, [1163](#)
- SPM2AVG152T2FrameofReference, [1163](#)
- SPM2AVG305T1FrameofReference, [1163](#)
- SPM2BRAINMASKFrameofReference, [1163](#)
- SPM2CSFFFrameofReference, [1163](#)
- SPM2EPIFrameofReference, [1163](#)
- SPM2FILT1FrameofReference, [1163](#)
- SPM2GRAYFrameofReference, [1163](#)
- SPM2PDFFrameofReference, [1163](#)
- SPM2PETFrameofReference, [1163](#)
- SPM2SINGLESUBJT1FrameofReference, [1163](#)
- SPM2SPECTFrameofReference, [1163](#)
- SPM2T1FrameofReference, [1163](#)
- SPM2T2FrameofReference, [1163](#)
- SPM2TRANSMFrameofReference, [1163](#)
- SPM2WHITEFrameofReference, [1163](#)
- SpringColorPaletteSOPInstance, [1168](#)
- StandaloneCurveStorageRetired, [1165](#)
- StandaloneModalityLUTStorageRetired, [1165](#)
- StandaloneOverlayStorageRetired, [1165](#)
- StandalonePETCurveStorageRetired, [1166](#)
- StandaloneVOILUTStorageRetired, [1165](#)
- StereometricRelationshipStorage, [1166](#)
- StorageCommitmentPullModelSOPClassRetired, [1163](#)
- StorageCommitmentPullModelSOPInstanceRetired, [1163](#)
- StorageCommitmentPushModelSOPClass, [1163](#)
- StorageCommitmentPushModelSOPInstance, [1163](#)
- StorageServiceClass, [1164](#)
- StoredPrintStorageSOPClassRetired, [1164](#)
- StudyComponentManagementSOPClassRetired, [1163](#)
- StudyRootQueryRetrieveInformationModelIFIND, [1166](#)
- StudyRootQueryRetrieveInformationModelIGET, [1166](#)
- StudyRootQueryRetrieveInformationModelIMOVE, [1166](#)
- SubjectiveRefractionMeasurementsStorage, [1169](#)
- SubstanceAdministrationLoggingSOPClass, [1163](#)
- SubstanceAdministrationLoggingSOPInstance, [1163](#)
- SubstanceApprovalQuerySOPClass, [1167](#)
- SummerColorPaletteSOPInstance, [1168](#)
- SurfaceScanMeshStorage, [1169](#)
- SurfaceScanPointCloudStorage, [1169](#)
- SurfaceSegmentationStorage, [1168](#)
- TalairachBrainAtlasFrameofReference, [1163](#)
- TextSRStorageTrialRetired, [1166](#)
- TractographyResultsStorage, [1169](#)
- TransferSyntaxStringsType, [1161](#)
- TSTName, [1162](#)
- TSType, [1171](#)
- UberonOntology, [1168](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_1, [1176](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_10, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_11, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_12, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_13, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_14, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_15, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_16, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_17, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_18, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_19, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_2, [1176](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_20, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_21, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_22, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_23, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_24, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_25, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_26, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_27, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_28, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_29, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_3, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_30, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_31, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_4, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_5, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_6, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_7, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_8, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_9, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_4\_1, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_4\_2, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_4\_3, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_4\_4, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_4\_5, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_4\_6, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_4\_7, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_4\_8, [1177](#)
- uid\_1\_2\_840\_10008\_15\_1\_1, [1180](#)
- uid\_1\_2\_840\_10008\_1\_1, [1171](#)
- uid\_1\_2\_840\_10008\_1\_2, [1171](#)

uid\_1\_2\_840\_10008\_1\_20, [1178](#)  
uid\_1\_2\_840\_10008\_1\_20\_1, [1173](#)  
uid\_1\_2\_840\_10008\_1\_20\_1\_1, [1173](#)  
uid\_1\_2\_840\_10008\_1\_20\_2, [1173](#)  
uid\_1\_2\_840\_10008\_1\_20\_2\_1, [1173](#)  
uid\_1\_2\_840\_10008\_1\_2\_1, [1171](#)  
uid\_1\_2\_840\_10008\_1\_2\_1\_99, [1171](#)  
uid\_1\_2\_840\_10008\_1\_2\_2, [1171](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_100, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_101, [1177](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_102, [1177](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_103, [1178](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_104, [1178](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_105, [1178](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_106, [1178](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_107, [1178](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_108, [1178](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_50, [1171](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_51, [1171](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_52, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_53, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_54, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_55, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_56, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_57, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_58, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_59, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_60, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_61, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_62, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_63, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_64, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_65, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_66, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_70, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_80, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_81, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_90, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_91, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_92, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_93, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_94, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_95, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_5, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_6\_1, [1172](#)  
uid\_1\_2\_840\_10008\_1\_2\_6\_2, [1172](#)  
uid\_1\_2\_840\_10008\_1\_3\_10, [1172](#)  
uid\_1\_2\_840\_10008\_1\_40, [1173](#)  
uid\_1\_2\_840\_10008\_1\_40\_1, [1173](#)  
uid\_1\_2\_840\_10008\_1\_42, [1173](#)  
uid\_1\_2\_840\_10008\_1\_42\_1, [1173](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_1, [1172](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_10, [1172](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_11, [1172](#)

uid\_1\_2\_840\_10008\_1\_4\_1\_12, [1172](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_13, [1172](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_14, [1172](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_15, [1172](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_16, [1172](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_17, [1172](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_18, [1173](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_2, [1172](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_3, [1172](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_4, [1172](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_5, [1172](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_6, [1172](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_7, [1172](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_8, [1172](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_9, [1172](#)  
uid\_1\_2\_840\_10008\_1\_4\_2\_1, [1173](#)  
uid\_1\_2\_840\_10008\_1\_4\_2\_2, [1173](#)  
uid\_1\_2\_840\_10008\_1\_5\_1, [1178](#)  
uid\_1\_2\_840\_10008\_1\_5\_2, [1178](#)  
uid\_1\_2\_840\_10008\_1\_5\_3, [1178](#)  
uid\_1\_2\_840\_10008\_1\_5\_4, [1178](#)  
uid\_1\_2\_840\_10008\_1\_5\_5, [1178](#)  
uid\_1\_2\_840\_10008\_1\_5\_6, [1178](#)  
uid\_1\_2\_840\_10008\_1\_5\_7, [1178](#)  
uid\_1\_2\_840\_10008\_1\_5\_8, [1178](#)  
uid\_1\_2\_840\_10008\_1\_9, [1173](#)  
uid\_1\_2\_840\_10008\_2\_16\_10, [1178](#)  
uid\_1\_2\_840\_10008\_2\_16\_11, [1178](#)  
uid\_1\_2\_840\_10008\_2\_16\_12, [1178](#)  
uid\_1\_2\_840\_10008\_2\_16\_13, [1178](#)  
uid\_1\_2\_840\_10008\_2\_16\_14, [1178](#)  
uid\_1\_2\_840\_10008\_2\_16\_4, [1173](#)  
uid\_1\_2\_840\_10008\_2\_16\_5, [1178](#)  
uid\_1\_2\_840\_10008\_2\_16\_6, [1178](#)  
uid\_1\_2\_840\_10008\_2\_16\_7, [1178](#)  
uid\_1\_2\_840\_10008\_2\_16\_8, [1178](#)  
uid\_1\_2\_840\_10008\_2\_16\_9, [1178](#)  
uid\_1\_2\_840\_10008\_2\_6\_1, [1173](#)  
uid\_1\_2\_840\_10008\_3\_1\_1\_1, [1173](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_1\_1, [1173](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_1\_4, [1173](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_2\_1, [1173](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_3\_1, [1173](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_3\_2, [1173](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_3\_3, [1173](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_3\_4, [1173](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_3\_5, [1173](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_5\_1, [1173](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_5\_4, [1173](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_5\_5, [1173](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_6\_1, [1173](#)  
uid\_1\_2\_840\_10008\_4\_2, [1173](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_1, [1173](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_14, [1173](#)

uid\_1\_2\_840\_10008\_5\_1\_1\_15, 1173  
uid\_1\_2\_840\_10008\_5\_1\_1\_16, 1173  
uid\_1\_2\_840\_10008\_5\_1\_1\_16\_376, 1173  
uid\_1\_2\_840\_10008\_5\_1\_1\_17, 1173  
uid\_1\_2\_840\_10008\_5\_1\_1\_17\_376, 1173  
uid\_1\_2\_840\_10008\_5\_1\_1\_18, 1173  
uid\_1\_2\_840\_10008\_5\_1\_1\_18\_1, 1173  
uid\_1\_2\_840\_10008\_5\_1\_1\_2, 1173  
uid\_1\_2\_840\_10008\_5\_1\_1\_22, 1173  
uid\_1\_2\_840\_10008\_5\_1\_1\_23, 1173  
uid\_1\_2\_840\_10008\_5\_1\_1\_24, 1173  
uid\_1\_2\_840\_10008\_5\_1\_1\_24\_1, 1174  
uid\_1\_2\_840\_10008\_5\_1\_1\_25, 1174  
uid\_1\_2\_840\_10008\_5\_1\_1\_26, 1174  
uid\_1\_2\_840\_10008\_5\_1\_1\_27, 1174  
uid\_1\_2\_840\_10008\_5\_1\_1\_29, 1174  
uid\_1\_2\_840\_10008\_5\_1\_1\_30, 1174  
uid\_1\_2\_840\_10008\_5\_1\_1\_31, 1174  
uid\_1\_2\_840\_10008\_5\_1\_1\_32, 1174  
uid\_1\_2\_840\_10008\_5\_1\_1\_33, 1174  
uid\_1\_2\_840\_10008\_5\_1\_1\_4, 1173  
uid\_1\_2\_840\_10008\_5\_1\_1\_40, 1178  
uid\_1\_2\_840\_10008\_5\_1\_1\_40\_1, 1178  
uid\_1\_2\_840\_10008\_5\_1\_1\_4\_1, 1173  
uid\_1\_2\_840\_10008\_5\_1\_1\_4\_2, 1173  
uid\_1\_2\_840\_10008\_5\_1\_1\_9, 1173  
uid\_1\_2\_840\_10008\_5\_1\_1\_9\_1, 1173  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1, 1174  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_10, 1174  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_104\_1, 1175  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_104\_2, 1176  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_104\_3, 1179  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11, 1174  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_1, 1174  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_10, 1178  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_11, 1178  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_2, 1174  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_3, 1174  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_4, 1174  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_5, 1178  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_6, 1178  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_7, 1178  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_8, 1178  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_9, 1178  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_128, 1176  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_128\_1, 1177  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_129, 1176  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_1, 1175  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_1\_1, 1175  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_2, 1175  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_2\_1, 1175  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_3, 1175  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_77, 1178  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_130, 1179  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_131, 1179  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_1, 1175  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_2, 1175  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_3, 1177  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_4, 1178  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_5, 1178  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_14\_1, 1178  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_14\_2, 1178  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_1, 1174  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_1\_1, 1174  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_2, 1174  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_2\_1, 1174  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_3, 1174  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_3\_1, 1174  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_2, 1174  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_20, 1175  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_1, 1179  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_2, 1179  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_3, 1179  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_4, 1179  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_5, 1179  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_6, 1179  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_2\_1, 1174  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_2\_2, 1177  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_3, 1174  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_30, 1178  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_3\_1, 1174  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4, 1174  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_40, 1178  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_1, 1176  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_10, 1179  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_11, 1179  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_2, 1176  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_3, 1176  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_4, 1176  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_5, 1176  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_6, 1176  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_7, 1176  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_8, 1176  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_9, 1176  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4\_1, 1174  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4\_2, 1174  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4\_3, 1180  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4\_4, 1177  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_5, 1174  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_1, 1179  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_2\_1, 1179  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_2\_2, 1179  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_3, 1179  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_4, 1180  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_5, 1180  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_6, 1180  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_6, 1174  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_601\_1, 1180

Generated by Doxygen

- uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4, [1176](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_1, [1176](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_2, [1176](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_3, [1176](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_4, [1176](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_34\_5, [1176](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_34\_5\_1, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_34\_6, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_34\_6\_1, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_34\_6\_2, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_34\_6\_3, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_34\_6\_4, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_34\_7, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_34\_8, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_34\_9, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_37\_1, [1176](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_37\_2, [1176](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_37\_3, [1176](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_38\_1, [1176](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_38\_2, [1176](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_38\_3, [1176](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_38\_4, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_39\_1, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_39\_2, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_39\_3, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_39\_4, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_41, [1176](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_42, [1176](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_43\_1, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_43\_2, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_43\_3, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_43\_4, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_44\_1, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_44\_2, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_44\_3, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_44\_4, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_45\_1, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_45\_2, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_45\_3, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_45\_4, [1180](#)
- uid\_1\_2\_840\_10008\_7\_1\_1, [1180](#)
- uid\_1\_2\_840\_10008\_7\_1\_2, [1180](#)
- uid\_1\_2\_840\_10008\_8\_1\_1, [1180](#)
- UltrasoundImageStorage, [1165](#)
- UltrasoundImageStorageRetired, [1165](#)
- UltrasoundMultiframeImageStorage, [1164](#)
- UltrasoundMultiframeImageStorageRetired, [1164](#)
- UnifiedProcedureStepEventSOPClass, [1167](#)
- UnifiedProcedureStepEventSOPClass1, [1171](#)
- UnifiedProcedureStepPullSOPClass, [1167](#)
- UnifiedProcedureStepPullSOPClass1, [1170](#)
- UnifiedProcedureStepPushSOPClass, [1167](#)
- UnifiedProcedureStepPushSOPClass1, [1170](#)
- UnifiedProcedureStepWatchSOPClass, [1167](#)
- UnifiedProcedureStepWatchSOPClass1, [1170](#)
- UnifiedWorklistandProcedureStepServiceClass, [1167](#)
- UnifiedWorklistandProcedureStepServiceClass1, [1170](#)
- UnifiedWorklistandProcedureStepSOPInstance, [1167](#)
- UniversalCoordinatedTime, [1171](#)
- UPSFilteredGlobalSubscriptionSOPInstance, [1170](#)
- VerificationSOPClass, [1162](#)
- VideoEndoscopicImageStorage, [1165](#)
- VideoMicroscopicImageStorage, [1165](#)
- VideoPhotographicImageStorage, [1166](#)
- VisualAcuityMeasurementsStorage, [1169](#)
- VLEndoscopicImageStorage, [1165](#)
- VLIImageStorageTrialRetired, [1165](#)
- VLMicroscopicImageStorage, [1165](#)
- VLMultiframeImageStorageTrialRetired, [1165](#)
- VLPotographicImageStorage, [1166](#)
- VLSlideCoordinatesMicroscopicImageStorage, [1166](#)
- VLWholeSlideMicroscopyImageStorage, [1168](#)
- VOILUTBoxSOPClass, [1164](#)
- VolumeRenderingVolumetricPresentationStateStorage, [1169](#)
- WaveformStorageTrialRetired, [1165](#)
- WideFieldOphthalmicPhotography3DCoordinatesImageStorage, [1169](#)
- WideFieldOphthalmicPhotographyStereographicProjectionImageStorage, [1169](#)
- WinterColorPaletteSOPInstance, [1168](#)
- XAXRFGrayscaleSoftcopyPresentationStateStorage, [1169](#)
- XMLEncoding, [1162](#)
- XRay3DAngiographicImageStorage, [1165](#)
- XRay3DCraniofacialImageStorage, [1165](#)
- XRayAngiographicBiPlaneImageStorageRetired, [1165](#)
- XRayAngiographicImageStorage, [1165](#)
- XRayRadiationDoseSRStorage, [1166](#)
- XRayRadiofluoroscopicImageStorage, [1165](#)
- gdcm::UNExplicitDataElement, [1250](#)
  - GetLength, [1251](#)
  - Read, [1252](#)
  - ReadPreValue, [1252](#)
  - ReadValue, [1252](#)
  - ReadWithLength, [1252](#)
- gdcm::UNExplicitImplicitDataElement, [1253](#)
  - GetLength, [1254](#)
  - Read, [1254](#)
  - ReadPreValue, [1254](#)
  - ReadValue, [1254](#)
- gdcm::Unpacker12Bits, [1255](#)
  - Pack, [1255](#)
  - Unpack, [1256](#)

- gdcmm::Usage, 1256
  - Conditional, 1257
  - GetUsageString, 1258
  - GetUsageType, 1258
  - Invalid, 1257
  - Mandatory, 1257
  - operator UsageType, 1258
  - operator<<, 1258
  - Usage, 1258
  - UsageType, 1257
  - UserOption, 1257
- gdcmm::UserEvent, 1259
- gdcmm::UUIDGenerator, 1262
  - Generate, 1263
  - IsValid, 1263
- gdcmm::Validate, 1263
  - ~Validate, 1264
  - F, 1265
  - GetValidatedFile, 1264
  - SetFile, 1264
  - V, 1265
  - Validate, 1264
  - Validation, 1265
- gdcmm::Value, 1266
  - ~Value, 1267
  - Clear, 1267
  - DataElement, 1268
  - GetLength, 1267
  - operator==, 1268
  - SetLength, 1268
  - SetLengthOnly, 1268
  - Value, 1267
- gdcmm::ValueIO< TDE, TSwap, TType >, 1269
  - Read, 1269
  - Write, 1269
- gdcmm::Version, 1270
  - ~Version, 1271
  - GetBuildVersion, 1271
  - GetMajorVersion, 1272
  - GetMinorVersion, 1272
  - GetVersion, 1272
  - operator<<, 1272
  - Print, 1272
  - Version, 1271
- gdcmm::VL, 1273
  - GetLength, 1274
  - GetVL16Max, 1274
  - GetVL32Max, 1275
  - IsOdd, 1275
  - IsUndefined, 1275
  - operator uint32\_t, 1275
  - operator<<, 1277
  - operator++, 1275
  - operator+=, 1275
  - Read, 1276
  - Read16, 1276
  - SetToUndefined, 1276
  - Type, 1274
  - VL, 1274
  - Write, 1276
  - Write16, 1276
- gdcmm::VM, 1277
  - Compatible, 1280
  - GetIndex, 1280
  - GetLength, 1280
  - GetNumberOfElementsFromArray, 1280
  - GetVMString, 1280
  - GetVMType, 1281
  - GetVMTypeFromLength, 1281
  - IsValid, 1281
  - operator VMType, 1281
  - operator<<, 1281
  - VM, 1280
  - VM0, 1279
  - VM1, 1279
  - VM10, 1279
  - VM12, 1279
  - VM16, 1279
  - VM18, 1279
  - VM1\_2, 1279
  - VM1\_3, 1279
  - VM1\_32, 1279
  - VM1\_4, 1279
  - VM1\_5, 1279
  - VM1\_8, 1279
  - VM1\_99, 1279
  - VM1\_n, 1279
  - VM2, 1279
  - VM24, 1279
  - VM256, 1279
  - VM28, 1279
  - VM2\_2n, 1279
  - VM2\_n, 1279
  - VM3, 1279
  - VM30\_30n, 1279
  - VM32, 1279
  - VM35, 1279
  - VM3\_3n, 1279
  - VM3\_4, 1279
  - VM3\_n, 1279
  - VM4, 1279
  - VM47\_47n, 1279
  - VM4\_4n, 1279
  - VM5, 1279
  - VM6, 1279
  - VM6\_6n, 1279
  - VM6\_n, 1279
  - VM7\_7n, 1279



- VM8, [1279](#)
- VM9, [1279](#)
- VM99, [1279](#)
- VM\_END, [1279](#)
- VMType, [1279](#)
- gdcmm::VMToLength< T >, [1282](#)
- gdcmm::VR, [1282](#)
  - AE, [1284](#)
  - AS, [1284](#)
  - AT, [1284](#)
  - CanDisplay, [1285](#)
  - Compatible, [1286](#)
  - CS, [1284](#)
  - DA, [1284](#)
  - DS, [1284](#)
  - DT, [1284](#)
  - FD, [1284](#)
  - FL, [1284](#)
  - GetLength, [1286](#)
  - GetSize, [1286](#)
  - GetSizeof, [1286](#)
  - GetVRString, [1286](#)
  - GetVRStringFromFile, [1287](#)
  - GetVRType, [1287](#)
  - GetVRTypeFromFile, [1287](#)
  - INVALID, [1284](#)
  - IS, [1284](#)
  - IsASCII, [1287](#)
  - IsASCII2, [1287](#)
  - IsBinary, [1287](#)
  - IsBinary2, [1288](#)
  - IsDual, [1288](#)
  - IsSwap, [1288](#)
  - IsValid, [1288](#)
  - IsVRFile, [1288](#)
  - LO, [1284](#)
  - LT, [1284](#)
  - OB, [1284](#)
  - OB\_OW, [1285](#)
  - OD, [1284](#)
  - OF, [1284](#)
  - OL, [1284](#)
  - operator VRTYPE, [1289](#)
  - operator<, [1289](#)
  - OV, [1284](#)
  - OW, [1284](#)
  - PN, [1284](#)
  - Read, [1289](#)
  - SH, [1284](#)
  - SL, [1285](#)
  - SQ, [1285](#)
  - SS, [1285](#)
  - ST, [1285](#)
  - SV, [1285](#)
  - TM, [1285](#)
  - UC, [1285](#)
  - UI, [1285](#)
  - UL, [1285](#)
  - UN, [1285](#)
  - UR, [1285](#)
  - US, [1285](#)
  - US\_OW, [1285](#)
  - US\_SS, [1285](#)
  - US\_SS\_OW, [1285](#)
  - UT, [1285](#)
  - UV, [1285](#)
  - VL16, [1285](#)
  - VL32, [1285](#)
  - VR, [1285](#)
  - VR\_END, [1285](#)
  - VR\_VM1, [1285](#)
  - VRALL, [1285](#)
  - VRASCII, [1285](#)
  - VRBINARY, [1285](#)
  - VRTYPE, [1284](#)
  - Write, [1289](#)
- gdcmm::VR16ExplicitDataElement, [1290](#)
  - GetLength, [1291](#)
  - Read, [1291](#)
  - ReadPreValue, [1291](#)
  - ReadValue, [1291](#)
  - ReadWithLength, [1292](#)
- gdcmm::VRToEncoding< T >, [1292](#)
- gdcmm::VRToType< T >, [1292](#)
- gdcmm::VRVLSIZE< 0 >, [1293](#)
  - Read, [1293](#)
  - Write, [1293](#)
- gdcmm::VRVLSIZE< 1 >, [1293](#)
  - Read, [1294](#)
  - Write, [1294](#)
- gdcmm::VRVLSIZE< T >, [1293](#)
- gdcmm::Waveform, [1410](#)
  - Waveform, [1411](#)
- gdcmm::WLMFindQuery, [1411](#)
  - GetAbstractSyntaxUID, [1413](#)
  - GetTagListByLevel, [1413](#)
  - GetValidDataSet, [1413](#)
  - InitializeDataSet, [1413](#)
  - QueryFactory, [1414](#)
  - ValidateQuery, [1413](#)
  - WLMFindQuery, [1412](#)
- gdcmm::Writer, [1414](#)
  - ~Writer, [1417](#)
  - CheckFileMetaInformationOff, [1417](#)
  - CheckFileMetaInformationOn, [1417](#)
  - GetCheckFileMetaInformation, [1417](#)
  - GetFile, [1417](#)
  - GetStreamPtr, [1418](#)

- Ofstream, [1420](#)
- SetCheckFileMetaInformation, [1418](#)
- SetFile, [1418](#)
- SetFileName, [1418](#)
- SetStream, [1419](#)
- SetWriteDataSetOnly, [1419](#)
- Stream, [1420](#)
- StreamImageWriter, [1420](#)
- Write, [1419](#)
- Writer, [1417](#)
- gdcM::XMLDictReader, [1420](#)
  - ~XMLDictReader, [1421](#)
  - CharacterDataHandler, [1422](#)
  - EndElement, [1422](#)
  - GetDict, [1422](#)
  - HandleDescription, [1422](#)
  - HandleEntry, [1422](#)
  - StartElement, [1423](#)
  - XMLDictReader, [1421](#)
- gdcM::XMLPrinter, [1423](#)
  - ~XMLPrinter, [1425](#)
  - F, [1426](#)
  - GetPrintStyle, [1425](#)
  - HandleBulkData, [1425](#)
  - LOADBULKDATA, [1424](#)
  - OnlyUUID, [1424](#)
  - Print, [1425](#)
  - PrintDataElement, [1425](#)
  - PrintDataSet, [1426](#)
  - PrintSQ, [1426](#)
  - PrintStyle, [1427](#)
  - PrintStyles, [1424](#)
  - SetFile, [1426](#)
  - SetStyle, [1426](#)
  - XMLPrinter, [1425](#)
- gdcM::XMLPrivateDictReader, [1427](#)
  - ~XMLPrivateDictReader, [1428](#)
  - CharacterDataHandler, [1428](#)
  - EndElement, [1429](#)
  - GetPrivateDict, [1429](#)
  - HandleDescription, [1429](#)
  - HandleEntry, [1429](#)
  - StartElement, [1429](#)
  - XMLPrivateDictReader, [1428](#)
- GDCM\_DIFFERENT
  - gdcM, [61](#)
- GDCM\_DO\_JOIN
  - gdcMStaticAssert.h, [1647](#)
- GDCM\_DO\_JOIN2
  - gdcMStaticAssert.h, [1648](#)
- GDCM\_EQUAL
  - gdcM, [61](#)
- GDCM\_EXPORT
  - gdcMWin32.h, [1705](#)
- GDCM\_FUNCTION
  - gdcMTrace.h, [1671](#)
- GDCM\_GREATER
  - gdcM, [61](#)
- GDCM\_GREATEROREQUAL
  - gdcM, [61](#)
- GDCM\_JOIN
  - gdcMStaticAssert.h, [1648](#)
- GDCM\_LEGACY
  - gdcMLegacyMacro.h, [1553](#)
- GDCM\_LEGACY\_BODY
  - gdcMLegacyMacro.h, [1554](#)
- GDCM\_LEGACY\_REPLACED\_BODY
  - gdcMLegacyMacro.h, [1554](#)
- GDCM\_LESS
  - gdcM, [61](#)
- GDCM\_LESOREQUAL
  - gdcM, [61](#)
- GDCM\_STATIC\_ASSERT
  - gdcM::Attribute< Group, Element, TVR, TVM >, [137](#)
  - gdcM::Attribute< Group, Element, TVR, VM::VM1 >, [146](#)
  - gdcM::Attribute< Group, Element, TVR, VM::VM1\_n >, [155](#)
  - gdcMStaticAssert.h, [1648](#)
- gdcMAAbortPDU.h, [1431](#)
- gdcMAAssociateACPDU.h, [1432](#)
- gdcMAAssociateRJPDU.h, [1432](#)
- gdcMAAssociateRQPDU.h, [1433](#)
- gdcMAbstractSyntax.h, [1434](#)
- gdcMAnonymizeEvent.h, [1435](#)
- gdcMAnonymizer.h, [1436](#)
- gdcMApplicationContext.h, [1437](#)
- gdcMApplicationEntity.h, [1438](#)
- gdcMReleaseRPPDU.h, [1438](#)
- gdcMReleaseRQPDU.h, [1439](#)
- gdcMARTIMTimer.h, [1440](#)
- gdcMASN1.h, [1441](#)
- gdcMAssertAlwaysMacro
  - gdcMTrace.h, [1671](#)
- gdcMAssertMacro
  - gdcMTrace.h, [1671](#)
- gdcMASynchronousOperationsWindowSub.h, [1442](#)
- gdcMAttribute.h, [1442](#)
- gdcMAudioCodec.h, [1444](#)
- gdcMBase64.h, [1445](#)
- gdcMBaseCompositeMessage.h, [1445](#)
- gdcMBaseNormalizedMessage.h, [1446](#)
- gdcMBasePDU.h, [1447](#)
- gdcMBaseQuery.h, [1448](#)
- gdcMBaseRootQuery.h, [1449](#)
- gdcMBasicOffsetTable.h, [1450](#)
- gdcMBitmap.h, [1451](#)
- gdcMBitmapToBitmapFilter.h, [1452](#)



gdcmBoxRegion.h, 1453  
gdcmByteBuffer.h, 1453  
gdcmByteSwap.h, 1455  
gdcmByteSwapFilter.h, 1455  
gdcmByteValue.h, 1456  
gdcmCAPICryptoFactory.h, 1457  
gdcmCAPICryptographicMessageSyntax.h, 1458  
gdcmCEchoMessages.h, 1458  
gdcmCFindMessages.h, 1459  
gdcmCMoveMessages.h, 1460  
gdcmCodec.h, 1461  
gdcmCoder.h, 1462  
gdcmCodeString.h, 1464  
gdcmCommand.h, 1465  
gdcmCommandDataSet.h, 1466  
gdcmCompositeMessageFactory.h, 1467  
gdcmCompositeNetworkFunctions.h, 1467  
gdcmConstCharWrapper.h, 1468  
gdcmCP246ExplicitDataElement.h, 1468  
gdcmCryptoFactory.h, 1469  
gdcmCryptographicMessageSyntax.h, 1470  
gdcmCSAElement.h, 1471  
gdcmCSAHeader.h, 1472  
gdcmCSAHeaderDict.h, 1473  
gdcmCSAHeaderDictEntry.h, 1474  
gdcmCStoreMessages.h, 1475  
gdcmCurve.h, 1476  
gdcmDataElement.h, 1477  
gdcmDataEvent.h, 1479  
gdcmDataSet.h, 1480  
gdcmDataSetEvent.h, 1481  
gdcmDataSetHelper.h, 1481  
gdcmDebugMacro  
    gdcmTrace.h, 1673  
gdcmDecoder.h, 1482  
gdcmDefinedTerms.h, 1484  
gdcmDeflateStream.h, 1484  
gdcmDefs.h, 1485  
gdcmDeltaEncodingCodec.h, 1486  
gdcmDICOMDIR.h, 1487  
gdcmDICOMDIRGenerator.h, 1488  
gdcmDict.h, 1488  
gdcmDictConverter.h, 1490  
gdcmDictEntry.h, 1490  
gdcmDictPrinter.h, 1492  
gdcmDicts.h, 1492  
gdcmDIMSE.h, 1494  
gdcmDirectionCosines.h, 1494  
gdcmDirectory.h, 1495  
gdcmDirectoryHelper.h, 1496  
gdcmDummyValueGenerator.h, 1497  
gdcmDumper.h, 1498  
gdcmElement.h, 1498  
gdcmEmptyMaskGenerator.h, 1500  
gdcmEncapsulatedDocument.h, 1501  
gdcmEnumeratedValues.h, 1501  
gdcmEquipmentManufacturer.h, 1502  
gdcmErrorMacro  
    gdcmTrace.h, 1673  
gdcmEvent.h, 1503  
    gdcmEventMacro, 1504  
gdcmEventMacro  
    gdcmEvent.h, 1504  
gdcmException.h, 1504  
gdcmExplicitDataElement.h, 1505  
gdcmExplicitImplicitDataElement.h, 1506  
gdcmFiducials.h, 1507  
gdcmFile.h, 1507  
gdcmFileAnonymizer.h, 1509  
gdcmFileChangeTransferSyntax.h, 1509  
gdcmFileDecompressLookupTable.h, 1510  
gdcmFileDerivation.h, 1511  
gdcmFileExplicitFilter.h, 1512  
gdcmFileMetaInformation.h, 1513  
gdcmFilename.h, 1514  
gdcmFileNameEvent.h, 1514  
gdcmFilenameGenerator.h, 1515  
gdcmFileSet.h, 1516  
gdcmFileStreamer.h, 1517  
gdcmFindPatientRootQuery.h, 1518  
gdcmFindStudyRootQuery.h, 1519  
gdcmFragment.h, 1519  
gdcmGlobal.h, 1521  
gdcmGroupDict.h, 1522  
gdcmIconImage.h, 1522  
gdcmIconImageFilter.h, 1523  
gdcmIconImageGenerator.h, 1524  
gdcmImage.h, 1525  
gdcmImageApplyLookupTable.h, 1526  
gdcmImageChangePhotometricInterpretation.h, 1527  
gdcmImageChangePlanarConfiguration.h, 1528  
gdcmImageChangeTransferSyntax.h, 1528  
gdcmImageCodec.h, 1529  
gdcmImageConverter.h, 1530  
gdcmImageFragmentSplitter.h, 1531  
gdcmImageHelper.h, 1532  
gdcmImageReader.h, 1533  
gdcmImageRegionReader.h, 1534  
gdcmImageToImageFilter.h, 1534  
gdcmImageWriter.h, 1535  
gdcmImplementationClassUIDSub.h, 1536  
gdcmImplementationUIDSub.h, 1537  
gdcmImplementationVersionNameSub.h, 1538  
gdcmImplicitDataElement.h, 1539  
gdcmIOD.h, 1540  
gdcmIODEntry.h, 1541  
gdcmIODs.h, 1543  
gdcmIPPSorter.h, 1544

gdcmItem.h, 1545  
gdcmJPEG12Codec.h, 1547  
gdcmJPEG16Codec.h, 1547  
gdcmJPEG2000Codec.h, 1548  
gdcmJPEG8Codec.h, 1549  
gdcmJPEGCodec.h, 1550  
gdcmJPEGLSCodec.h, 1551  
gdcmJSON.h, 1551  
gdcmKAKADUCodec.h, 1552  
gdcmLegacyMacro.h, 1553  
    GDCM\_LEGACY, 1553  
    GDCM\_LEGACY\_BODY, 1554  
    GDCM\_LEGACY\_REPLACED\_BODY, 1554  
gdcmLO.h, 1554  
gdcmLookupTable.h, 1555  
gdcmMacro.h, 1556  
gdcmMacroEntry.h, 1558  
    GDCMMACROENTRY\_H, 1559  
GDCMMACROENTRY\_H  
    gdcmMacroEntry.h, 1559  
gdcmMacros.h, 1560  
gdcmMaxLengthSub.h, 1561  
gdcmMD5.h, 1563  
gdcmMediaStorage.h, 1563  
gdcmMeshPrimitive.h, 1565  
gdcmModalityPerformedProcedureStepCreateQuery.h, 1566  
gdcmModalityPerformedProcedureStepSetQuery.h, 1567  
gdcmModule.h, 1567  
gdcmModuleEntry.h, 1569  
gdcmModules.h, 1571  
gdcmMovePatientRootQuery.h, 1572  
gdcmMoveStudyRootQuery.h, 1573  
gdcmMrProtocol.h, 1573  
gdcmNActionMessages.h, 1575  
gdcmNCreateMessages.h, 1575  
gdcmNDeleteMessages.h, 1576  
gdcmNestedModuleEntries.h, 1577  
gdcmNetworkEvents.h, 1578  
gdcmNetworkStateID.h, 1579  
gdcmNEventReportMessages.h, 1580  
gdcmNGetMessages.h, 1581  
gdcmNormalizedMessageFactory.h, 1581  
gdcmNormalizedNetworkFunctions.h, 1582  
gdcmNSetMessages.h, 1583  
gdcmObject.h, 1583  
gdcmOpenSSLCryptoFactory.h, 1585  
gdcmOpenSSLCryptographicMessageSyntax.h, 1585  
gdcmOpenSSLP7CryptoFactory.h, 1587  
gdcmOpenSSLP7CryptographicMessageSyntax.h, 1587  
gdcmOrientation.h, 1589  
gdcmOverlay.h, 1589  
gdcmParseException.h, 1590  
gdcmParser.h, 1592  
gdcmPatient.h, 1592  
gdcmPDataTFPDU.h, 1593  
gdcmPDBelement.h, 1594  
gdcmPDBHeader.h, 1595  
gdcmPDFCodec.h, 1596  
gdcmPDUFactory.h, 1597  
gdcmPersonName.h, 1597  
gdcmPGXCodec.h, 1598  
gdcmPhotometricInterpretation.h, 1599  
gdcmPixelFormat.h, 1600  
gdcmPixmap.h, 1601  
gdcmPixmapReader.h, 1602  
gdcmPixmapToPixmapFilter.h, 1603  
gdcmPixmapWriter.h, 1603  
gdcmPNMCodec.h, 1605  
gdcmPreamble.h, 1605  
gdcmPresentationContext.h, 1607  
gdcmPresentationContextAC.h, 1608  
gdcmPresentationContextGenerator.h, 1609  
gdcmPresentationContextRQ.h, 1609  
gdcmPresentationDataValue.h, 1610  
gdcmPrinter.h, 1611  
gdcmPrivateTag.h, 1613  
gdcmProgressEvent.h, 1614  
gdcmPVRGCodec.h, 1615  
gdcmPythonFilter.h, 1615  
gdcmQueryBase.h, 1616  
gdcmQueryFactory.h, 1617  
gdcmQueryImage.h, 1618  
gdcmQueryPatient.h, 1619  
gdcmQuerySeries.h, 1620  
gdcmQueryStudy.h, 1621  
gdcmRAWCodec.h, 1622  
gdcmReader.h, 1622  
gdcmRegion.h, 1624  
gdcmRescaler.h, 1625  
gdcmRLECodec.h, 1626  
gdcmRoleSelectionSub.h, 1626  
gdcmScanner.h, 1627  
gdcmSegment.h, 1628  
gdcmSegmentedPaletteColorLookupTable.h, 1629  
gdcmSegmentHelper.h, 1630  
gdcmSegmentReader.h, 1631  
gdcmSegmentWriter.h, 1632  
gdcmSequenceOfFragments.h, 1633  
gdcmSequenceOfItems.h, 1634  
gdcmSerieHelper.h, 1634  
gdcmSeries.h, 1636  
gdcmServiceClassApplicationInformation.h, 1637  
gdcmServiceClassUser.h, 1638  
gdcmSHA1.h, 1638  
gdcmSimpleSubjectWatcher.h, 1639  
gdcmSmartPointer.h, 1640  
gdcmSOPClassExtendedNegotiationSub.h, 1641

gdcmSOPClassUIDToIOD.h, 1642  
gdcmSorter.h, 1643  
gdcmSpacing.h, 1645  
gdcmSpectroscopy.h, 1645  
gdcmSplitMosaicFilter.h, 1646  
gdcmStaticAssert.h, 1647  
    GDCM\_DO\_JOIN, 1647  
    GDCM\_DO\_JOIN2, 1648  
    GDCM\_JOIN, 1648  
    GDCM\_STATIC\_ASSERT, 1648  
gdcmStreamImageReader.h, 1649  
gdcmStreamImageWriter.h, 1649  
gdcmStrictScanner.h, 1650  
gdcmString.h, 1651  
gdcmStringFilter.h, 1652  
gdcmStudy.h, 1653  
gdcmSubject.h, 1654  
gdcmSurface.h, 1655  
gdcmSurfaceHelper.h, 1656  
gdcmSurfaceReader.h, 1657  
gdcmSurfaceWriter.h, 1658  
gdcmSwapCode.h, 1658  
gdcmSwapper.h, 1660  
gdcmSystem.h, 1661  
gdcmTable.h, 1661  
gdcmTableEntry.h, 1662  
gdcmTableReader.h, 1664  
gdcmTag.h, 1665  
gdcmTagPath.h, 1666  
gdcmTagToVR.h, 1666  
gdcmTerminal.h, 1667  
gdcmTestDriver.h, 1668  
gdcmTesting.h, 1669  
gdcmTrace.h, 1670  
    GDCM\_FUNCTION, 1671  
    gdcmAssertAlwaysMacro, 1671  
    gdcmAssertMacro, 1671  
    gdcmDebugMacro, 1673  
    gdcmErrorMacro, 1673  
    gdcmWarningMacro, 1674  
gdcmTransferSyntax.h, 1674  
gdcmTransferSyntaxSub.h, 1675  
gdcmType.h, 1676  
gdcmTypes.h, 1677  
gdcmUIDGenerator.h, 1678  
gdcmUIDs.h, 1679  
gdcmULAction.h, 1680  
gdcmULActionAA.h, 1681  
gdcmULActionAE.h, 1681  
gdcmULActionAR.h, 1682  
gdcmULActionDT.h, 1683  
gdcmULBasicCallback.h, 1683  
gdcmULConnection.h, 1684  
gdcmULConnectionCallback.h, 1685  
gdcmULConnectionInfo.h, 1686  
gdcmULConnectionManager.h, 1687  
gdcmULEvent.h, 1688  
gdcmULTransitionTable.h, 1689  
gdcmULWritingCallback.h, 1690  
gdcmUNExplicitDataElement.h, 1691  
gdcmUNExplicitImplicitDataElement.h, 1691  
gdcmUnpacker12Bits.h, 1692  
gdcmUsage.h, 1693  
gdcmUserInformation.h, 1695  
gdcmUUIDGenerator.h, 1696  
gdcmValidate.h, 1696  
gdcmValue.h, 1697  
gdcmValueIO.h, 1698  
gdcmVersion.h, 1698  
gdcmVL.h, 1699  
gdcmVM.h, 1700  
    TYPETOLENGTH, 1701  
gdcmVR.h, 1701  
    TYPETOENCODING, 1703  
    VRTypeTemplateCase, 1703  
gdcmVR16ExplicitDataElement.h, 1703  
gdcmWarningMacro  
    gdcmTrace.h, 1674  
gdcmWaveform.h, 1704  
gdcmWin32.h, 1704  
    GDCM\_EXPORT, 1705  
gdcmWLMFindQuery.h, 1705  
gdcmWriter.h, 1706  
gdcmXMLDictReader.h, 1707  
gdcmXMLPrinter.h, 1707  
gdcmXMLPrivateDictReader.h, 1708  
GEMS  
    gdcm::Dicts, 377  
    gdcm::EquipmentManufacturer, 431  
GeneralAudioWaveformStorage  
    gdcm::UIDs, 1169  
GeneralECGWaveformStorage  
    gdcm::MediaStorage, 677  
    gdcm::UIDs, 1165  
GeneralElectricMagneticResonanceImageStorage  
    gdcm::MediaStorage, 678  
GeneralPurposePerformedProcedureStepSOPClass  
    gdcm::UIDs, 1167  
GeneralPurposeScheduledProcedureStepSOPClass  
    gdcm::UIDs, 1167  
GeneralPurposeWorklistInformationModelFIND  
    gdcm::UIDs, 1166  
GeneralPurposeWorklistManagementMetaSOPClass  
    gdcm::UIDs, 1167  
GeneralRelevantPatientInformationQuery  
    gdcm::UIDs, 1167  
Generate  
    gdcm::DICOMDIRGenerator, 357

- gdcm::DummyValueGenerator, 393
- gdcm::FilenameGenerator, 487
- gdcm::IconImageGenerator, 519
- gdcm::UIDGenerator, 1144
- gdcm::UUIDGenerator, 1263
- GenerateFromFileNames
  - gdcm::PresentationContextGenerator, 849
- GenerateFromUID
  - gdcm::PresentationContextGenerator, 850
- GenerateUUID
  - gdcm::UIDGenerator, 1144
- GenericImplantTemplateInformationModelFIND
  - gdcm::UIDs, 1171
- GenericImplantTemplateInformationModelGET
  - gdcm::UIDs, 1171
- GenericImplantTemplateInformationModelMOVE
  - gdcm::UIDs, 1171
- GenericImplantTemplateStorage
  - gdcm::UIDs, 1171
- GEPrivate3DModelStorage
  - gdcm::MediaStorage, 678
- Get
  - gdcm::ByteBuffer, 218
- GetAbbreviation
  - gdcm::GroupDict, 513
- GetAbstractSyntax
  - gdcm::network::PresentationContextRQ, 853
  - gdcm::PresentationContext, 843
- GetAbstractSyntaxUID
  - gdcm::BaseQuery, 180
  - gdcm::FindPatientRootQuery, 500
  - gdcm::FindStudyRootQuery, 503
  - gdcm::ModalityPerformedProcedureStepCreateQuery, 697
  - gdcm::ModalityPerformedProcedureStepSetQuery, 700
  - gdcm::MovePatientRootQuery, 713
  - gdcm::MoveStudyRootQuery, 716
  - gdcm::WLMFindQuery, 1413
- GetAcceptedPresentationContexts
  - gdcm::network::ULConnection, 1227, 1228
- GetAcquisitionSize
  - gdcm::SplitMosaicFilter, 1017
- GetAETitle
  - gdcm::ServiceClassUser, 981
- GetAlgorithmFamily
  - gdcm::Surface, 1059
- GetAlgorithmName
  - gdcm::Surface, 1059
- GetAlgorithmVersion
  - gdcm::Surface, 1059
- GetALGOType
  - gdcm::Segment, 935
- GetALGOTypeString
  - gdcm::Segment, 935
- GetAllFileNamesFromTagToValue
  - gdcm::Scanner, 928
  - gdcm::StrictScanner, 1037
- GetAllRequiredTags
  - gdcm::QueryBase, 880
- GetAllTags
  - gdcm::QueryBase, 880
- GetAnatomicRegion
  - gdcm::Segment, 936
- GetAnatomicRegionModifiers
  - gdcm::Segment, 936
- GetAsDataElement
  - gdcm::Attribute< Group, Element, TVR, TVM >, 137
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 146
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, 155
  - gdcm::Element< TVR, TVM >, 397
  - gdcm::Element< TVR, VM::VM1\_n >, 404
  - gdcm::network::AbstractSyntax, 105
  - gdcm::PrivateTag, 869
- GetAsPoints
  - gdcm::Curve, 305
- GetAsString
  - gdcm::CodeString, 255
- GetAxisOfRotation
  - gdcm::Surface, 1059
- GetBasicApplicationLevelConfidentialityProfileAttributes
  - gdcm::Anonymizer, 114
- GetBitPosition
  - gdcm::Overlay, 767
- GetBitsAllocated
  - gdcm::Overlay, 767
  - gdcm::PixelFormat, 809
- GetBitSample
  - gdcm::LookupTable, 659
- GetBitsStored
  - gdcm::PixelFormat, 809
- GetBlob
  - gdcm::network::PresentationDataValue, 856
- GetBuffer
  - gdcm::Bitmap, 198
  - gdcm::ByteValue, 226
  - gdcm::Parser, 778
  - gdcm::SequenceOfFragments, 957
- GetBuffer2
  - gdcm::Bitmap, 198
- GetBufferAsRGBA
  - gdcm::LookupTable, 659
- GetBufferLength
  - gdcm::Bitmap, 198
  - gdcm::JPEGLSCodec, 643
  - gdcm::PNMCodec, 835

- gdcmm::RLECodec, [919](#)
- GetBuildVersion
  - gdcmm::Version, [1271](#)
- GetByteValue
  - gdcmm::CSAElement, [281](#)
  - gdcmm::DataElement, [313](#)
- GetCalledAETitle
  - gdcmm::network::AAssociateRQPDU, [99](#)
  - gdcmm::network::ULConnectionInfo, [1235](#)
  - gdcmm::ServiceClassUser, [981](#)
- GetCalledComputerName
  - gdcmm::network::ULConnectionInfo, [1235](#)
- GetCalledIPAddress
  - gdcmm::network::ULConnectionInfo, [1235](#)
- GetCalledIPPort
  - gdcmm::network::ULConnectionInfo, [1235](#)
- GetCallingAETitle
  - gdcmm::network::AAssociateRQPDU, [99](#)
  - gdcmm::network::ULConnectionInfo, [1235](#)
- GetCenterOfRotation
  - gdcmm::Surface, [1059](#)
- GetCharacterFromCurrentLocale
  - gdcmm::QueryFactory, [883](#)
- GetCheckFileMetaInformation
  - gdcmm::Writer, [1417](#)
- GetCipherType
  - gdcmm::CAPICryptographicMessageSyntax, [235](#)
  - gdcmm::CryptographicMessageSyntax, [277](#)
  - gdcmm::OpenSSLCryptographicMessageSyntax, [754](#)
  - gdcmm::OpenSSL7CryptographicMessageSyntax, [759](#)
- GetCodec
  - gdcmm::FileChangeTransferSyntax, [456](#)
- GetColorLevel
  - vtkImageColorViewer, [1366](#)
- GetColorWindow
  - vtkImageColorViewer, [1367](#)
- GetColumns
  - gdcmm::Bitmap, [198](#)
  - gdcmm::Overlay, [768](#)
- GetCommand
  - gdcmm::Subject, [1054](#)
- GetConnectionInfo
  - gdcmm::network::ULConnection, [1228](#)
- GetConstructorString
  - gdcmm::Dicts, [378](#)
- GetContourReferencedFrameOfReferenceClassUID
  - vtkRTStructSetProperties, [1403](#)
- GetContourReferencedFrameOfReferenceInstanceUID
  - vtkRTStructSetProperties, [1403](#)
- GetCryptographicMessageSyntax
  - gdcmm::Anonymizer, [115](#)
- GetCSADataInfo
  - gdcmm::CSAHeader, [289](#)
- GetCSAEEnd
  - gdcmm::CSAHeader, [289](#)
- GetCSAElementByName
  - gdcmm::CSAHeader, [289](#)
- GetCSAHeaderDict
  - gdcmm::Dicts, [378](#)
- GetCSAHeaderDictEntry
  - gdcmm::CSAHeaderDict, [294](#)
- GetCSAImageHeaderInfoTag
  - gdcmm::CSAHeader, [290](#)
- GetCSASeriesHeaderInfoTag
  - gdcmm::CSAHeader, [290](#)
- GetCTImageSeriesUIDs
  - gdcmm::DirectoryHelper, [390](#)
- GetCurrentByteIndex
  - gdcmm::Parser, [778](#)
- GetCurrentDateTime
  - gdcmm::System, [1087](#)
- GetCurrentModuleFileName
  - gdcmm::System, [1088](#)
- GetCurrentProcessFileName
  - gdcmm::System, [1088](#)
- GetCurrentResourcesDirectory
  - gdcmm::System, [1088](#)
- GetCurve
  - gdcmm::Pixmap, [817](#), [818](#)
- GetCurveDataDescriptor
  - gdcmm::Curve, [305](#)
- GetCWD
  - gdcmm::System, [1088](#)
- GetData
  - gdcmm::DataEvent, [326](#)
- GetDataElement
  - gdcmm::Bitmap, [198](#), [199](#)
  - gdcmm::DataSet, [333](#)
  - gdcmm::Item, [610](#)
- GetDataExtraRoot
  - gdcmm::Testing, [1117](#)
- GetDataLength
  - gdcmm::DataEvent, [326](#)
- GetDataRoot
  - gdcmm::Testing, [1117](#)
- GetDataSet
  - gdcmm::CSAHeader, [290](#)
  - gdcmm::DataSetEvent, [343](#)
  - gdcmm::File, [448](#)
- GetDataSetPos
  - gdcmm::network::ULEvent, [1245](#)
- GetDataSets
  - gdcmm::network::ULBasicCallback, [1224](#)
- GetDataSetTransferSyntax
  - gdcmm::FileMetaInformation, [472](#)
- GetDataValueRepresentation
  - gdcmm::Curve, [305](#)

- GetDebugFlag
  - gdcm::Trace, [1125](#)
- GetDebugStream
  - gdcm::Trace, [1125](#)
- GetDecodeLength
  - gdcm::Base64, [170](#)
- GetDEEnd
  - gdcm::DataSet, [333](#)
- GetDefaultTransferSyntax
  - gdcm::PresentationContextGenerator, [850](#)
- GetDefs
  - gdcm::Global, [509](#)
  - gdcm::TableReader, [1097](#)
- GetDES
  - gdcm::DataSet, [334](#)
- GetDescription
  - gdcm::CSAHeaderDictEntry, [297](#)
  - gdcm::Exception, [437](#)
  - gdcm::ModuleEntry, [707](#)
  - gdcm::Overlay, [768](#)
- GetDescriptiveName
  - vtkGDCMImageReader, [1298](#)
  - vtkGDCMImageReader2, [1312](#)
  - vtkGDCMImageWriter, [1326](#)
- GetDict
  - gdcm::XMLDictReader, [1422](#)
- GetDictEntry
  - gdcm::Dict, [361](#)
  - gdcm::Dicts, [378](#)
  - gdcm::PrivateDict, [866](#)
- GetDictEntryByKeyword
  - gdcm::Dict, [362](#)
- GetDictEntryByName
  - gdcm::Dict, [362](#)
- GetDictName
  - gdcm::DictConverter, [366](#)
- GetDicts
  - gdcm::Global, [509](#), [510](#)
- GetDictVM
  - gdcm::Attribute< Group, Element, TVR, TVM >, [137](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [147](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [156](#)
- GetDictVR
  - gdcm::Attribute< Group, Element, TVR, TVM >, [138](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [147](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [156](#)
- GetDimension
  - gdcm::Bitmap, [199](#)
- GetDimensions
  - gdcm::Bitmap, [199](#)
  - gdcm::Curve, [306](#)
  - gdcm::ImageCodec, [552](#)
- GetDimensionsValue
  - gdcm::ImageHelper, [566](#)
- GetDimensionsValueForResolution
  - gdcm::StreamImageReader, [1023](#)
- GetDirectionCosines
  - gdcm::Image, [525](#)
- GetDirectionCosinesFromDataSet
  - gdcm::ImageHelper, [566](#)
- GetDirectionCosinesTolerance
  - gdcm::IPPSorter, [604](#)
- GetDirectionCosinesValue
  - gdcm::ImageHelper, [566](#)
- GetDirectories
  - gdcm::Directory, [388](#)
- GetElapsedTime
  - gdcm::network::ARTIMTimer, [129](#)
- GetElement
  - gdcm::Tag, [1104](#)
- GetElementTag
  - gdcm::Tag, [1104](#)
- GetEncodeLength
  - gdcm::Base64, [170](#)
- GetErrorCode
  - gdcm::Parser, [778](#)
- GetErrorFlag
  - gdcm::Trace, [1125](#)
- GetErrorStream
  - gdcm::Trace, [1125](#)
- GetErrorString
  - gdcm::Parser, [779](#)
- GetEvent
  - gdcm::network::ULEvent, [1245](#)
- GetEventName
  - gdcm::AnonymizeEvent, [109](#)
  - gdcm::DataEvent, [326](#)
  - gdcm::DataSetEvent, [343](#)
  - gdcm::Event, [434](#)
  - gdcm::FileNameEvent, [484](#)
  - gdcm::ProgressEvent, [873](#)
- GetExtension
  - gdcm::Filename, [479](#)
- GetFactoryInstance
  - gdcm::CryptoFactory, [274](#)
- GetFile
  - gdcm::Anonymizer, [115](#)
  - gdcm::DICOMDIRGenerator, [357](#)
  - gdcm::FileDecompressLookupTable, [460](#)
  - gdcm::FileDerivation, [463](#)
  - gdcm::FileExplicitFilter, [466](#)
  - gdcm::IconImageFilter, [516](#)
  - gdcm::PythonFilter, [878](#)
  - gdcm::Reader, [901](#)



- gdcm::SplitMosaicFilter, [1017](#)
- gdcm::StreamImageReader, [1023](#)
- gdcm::StringFilter, [1048](#)
- gdcm::Writer, [1417](#)
- vtkGDCMMedicalImageProperties, [1335](#)
- GetFileExtensions
  - vtkGDCMImageReader, [1298](#)
  - vtkGDCMImageReader2, [1312](#)
  - vtkGDCMImageWriter, [1327](#)
- GetFileMetaInformationVersion
  - gdcm::FileMetaInformation, [472](#)
- GetFileName
  - gdcm::Filename, [479](#)
  - gdcm::FileNameEvent, [484](#)
  - gdcm::Testing, [1118](#)
  - vtkGDCMImageWriter, [1327](#)
  - vtkGDCMThreadedImageReader2, [1356](#)
- GetFilename
  - gdcm::FilenameGenerator, [487](#)
  - gdcm::TableReader, [1097](#)
- GetFilenameFromTagToValue
  - gdcm::Scanner, [928](#)
  - gdcm::StrictScanner, [1037](#)
- GetFileNames
  - gdcm::Testing, [1118](#)
- GetFilenames
  - gdcm::Directory, [388](#)
  - gdcm::FilenameGenerator, [487](#)
  - gdcm::Scanner, [928](#)
  - gdcm::Sorter, [1009](#)
  - gdcm::StrictScanner, [1037](#)
- GetFilenamesFromSeriesUIDs
  - gdcm::DirectoryHelper, [390](#)
- GetFiles
  - gdcm::FileSet, [490](#)
- GetFiniteVolume
  - gdcm::Surface, [1060](#)
- GetFirstSingleSeriesUIDFileSet
  - gdcm::SerieHelper, [973](#)
- GetForcePixelSpacing
  - gdcm::ImageHelper, [566](#)
- GetForceRescaleInterceptSlope
  - gdcm::ImageHelper, [566](#)
- GetFormat
  - gdcm::CSAHeader, [291](#)
- GetFragBuffer
  - gdcm::SequenceOfFragments, [957](#)
- GetFragment
  - gdcm::SequenceOfFragments, [957](#)
- GetFragmentSizeMax
  - gdcm::ImageFragmentSplitter, [563](#)
- GetFrameOfReference
  - gdcm::DirectoryHelper, [391](#)
- GetFullLength
  - gdcm::FileMetaInformation, [472](#)
- GetGDCMDataRoot
  - vtkGDCMTesting, [1348](#)
- GetGDCMImplementationClassUID
  - gdcm::FileMetaInformation, [472](#)
- GetGDCMImplementationVersionName
  - gdcm::FileMetaInformation, [472](#)
- GetGDCMSourceApplicationEntityTitle
  - gdcm::FileMetaInformation, [473](#)
- GetGDCMUID
  - gdcm::UIDGenerator, [1144](#)
- GetGroup
  - gdcm::Curve, [306](#)
  - gdcm::Overlay, [768](#)
  - gdcm::Tag, [1105](#)
- GetHasExpired
  - gdcm::network::ARTIMTimer, [130](#)
- GetHeader
  - gdcm::File, [448](#)
- GetHeaderInfo
  - gdcm::ImageCodec, [552](#)
  - gdcm::JPEG12Codec, [615](#)
  - gdcm::JPEG16Codec, [618](#)
  - gdcm::JPEG2000Codec, [623](#)
  - gdcm::JPEG8Codec, [629](#)
  - gdcm::JPEGCodec, [635](#)
  - gdcm::JPEGLSCodec, [643](#)
  - gdcm::PGXCodec, [801](#)
  - gdcm::PNMCodec, [835](#)
  - gdcm::RAWCodec, [897](#)
  - gdcm::RLECodec, [919](#)
- GetHierarchicalSearchTags
  - gdcm::QueryBase, [881](#)
  - gdcm::QueryImage, [885](#)
  - gdcm::QueryPatient, [887](#)
  - gdcm::QuerySeries, [890](#)
  - gdcm::QueryStudy, [892](#)
- GetHighBit
  - gdcm::PixelFormat, [810](#)
- GetHostName
  - gdcm::System, [1088](#)
- GetIconImage
  - gdcm::IconImageFilter, [516](#)
  - gdcm::IconImageGenerator, [519](#)
  - gdcm::Pixmap, [818](#)
  - vtkGDCMImageReader, [1298](#)
  - vtkGDCMImageReader2, [1313](#)
- GetIconImagePort
  - vtkGDCMImageReader2, [1313](#)
- GetIE
  - gdcm::IODEntry, [597](#)
- GetImage
  - gdcm::ImageReader, [573](#)
  - gdcm::ImageWriter, [585](#)

- gdcm::PixmapWriter, [831](#)
- gdcm::SplitMosaicFilter, [1017](#)
- GetImplementationClassUID
  - gdcm::FileMetaInformation, [473](#)
- GetImplementationVersionName
  - gdcm::FileMetaInformation, [473](#)
- GetIndex
  - gdcm::SwapCode, [1081](#)
  - gdcm::VM, [1280](#)
- GetInitialized
  - gdcm::CAPICryptographicMessageSyntax, [235](#)
- GetInput
  - gdcm::ImageToImageFilter, [581](#)
  - gdcm::PixmapToPixmapFilter, [827](#)
  - vtkImageColorViewer, [1367](#)
- GetInputFilename
  - gdcm::DictConverter, [366](#)
- GetInstance
  - gdcm::Global, [510](#)
- GetIntercept
  - gdcm::Image, [525](#)
  - gdcm::Rescaler, [912](#)
- GetInterfile
  - gdcm::CSAHeader, [291](#)
- GetInternal
  - gdcm::Preamble, [838](#)
- GetIOD
  - gdcm::IODs, [602](#)
  - gdcm::SOPClassUIDToIOD, [1006](#)
- GetIODEntry
  - gdcm::IOD, [595](#)
- GetIODFromFile
  - gdcm::Defs, [349](#)
- GetIODFromSOPClassUID
  - gdcm::SOPClassUIDToIOD, [1006](#)
- GetIODNameFromMediaStorage
  - gdcm::Defs, [349](#)
- GetIODs
  - gdcm::Defs, [349](#)
- GetIsCommand
  - gdcm::network::PresentationDataValue, [857](#)
- GetIsLastFragment
  - gdcm::network::PresentationDataValue, [857](#)
- GetIStream
  - gdcm::network::ULEvent, [1245](#)
- GetItem
  - gdcm::SequenceOfItems, [965](#)
- GetKey
  - gdcm::CSAElement, [281](#)
- GetKeys
  - gdcm::Scanner, [928](#)
  - gdcm::StrictScanner, [1037](#)
- GetKeyword
  - gdcm::DictEntry, [369](#)
- GetKeywordFromTag
  - gdcm::Dict, [362](#)
- GetLabel
  - gdcm::Orientation, [762](#)
- GetLastElement
  - gdcm::ParseException, [775](#)
- GetLastSystemError
  - gdcm::System, [1088](#)
- GetLength
  - gdcm::ByteValue, [226](#)
  - gdcm::CP246ExplicitDataElement, [271](#)
  - gdcm::DataElement, [313](#)
  - gdcm::DataSet, [334](#)
  - gdcm::Element< TVR, TVM >, [397](#)
  - gdcm::Element< TVR, VM::VM1\_n >, [405](#)
  - gdcm::Element< VR::AS, VM::VM5 >, [416](#)
  - gdcm::ExplicitDataElement, [441](#)
  - gdcm::ExplicitImplicitDataElement, [444](#)
  - gdcm::Fragment, [506](#)
  - gdcm::ImplicitDataElement, [591](#)
  - gdcm::Item, [610](#)
  - gdcm::Preamble, [838](#)
  - gdcm::SequenceOfFragments, [957](#)
  - gdcm::SequenceOfItems, [966](#)
  - gdcm::Tag, [1105](#)
  - gdcm::UNExplicitDataElement, [1251](#)
  - gdcm::UNExplicitImplicitDataElement, [1254](#)
  - gdcm::Value, [1267](#)
  - gdcm::VL, [1274](#)
  - gdcm::VM, [1280](#)
  - gdcm::VR, [1286](#)
  - gdcm::VR16ExplicitDataElement, [1291](#)
- GetLocaleCharset
  - gdcm::System, [1089](#)
- GetLossless
  - gdcm::JPEGCodec, [635](#)
  - gdcm::JPEGLSCodec, [643](#)
- GetLossyFlag
  - gdcm::ImageCodec, [552](#)
- GetLossyFlagFromFile
  - gdcm::Testing, [1118](#)
- GetLUT
  - gdcm::Bitmap, [199](#)
  - gdcm::ImageCodec, [552](#)
  - gdcm::ImageHelper, [567](#)
  - gdcm::LookupTable, [659](#)
- GetLUTDescriptor
  - gdcm::LookupTable, [659](#)
- GetLUTLength
  - gdcm::LookupTable, [660](#)
- GetMacro
  - gdcm::Macros, [669](#)
- GetMacroEntry
  - gdcm::Macro, [666](#)



- GetMacros
  - gdcm::Defs, [350](#)
- GetMajorAxisFromPatientRelativeDirectionCosine
  - gdcm::Orientation, [762](#)
- GetMajorVersion
  - gdcm::Version, [1272](#)
- GetManifold
  - gdcm::Surface, [1060](#)
- GetMapping
  - gdcm::Scanner, [928](#)
  - gdcm::StrictScanner, [1037](#)
- GetMappingFromTagToValue
  - gdcm::Scanner, [929](#)
  - gdcm::StrictScanner, [1037](#)
- GetMappings
  - gdcm::Scanner, [929](#)
  - gdcm::StrictScanner, [1038](#)
- GetMax
  - gdcm::PixelFormat, [810](#)
- GetMaximumLength
  - gdcm::network::MaximumLengthSub, [670](#)
- GetMaximumLengthSub
  - gdcm::network::UserInformation, [1261](#)
- GetMaximumPointDistance
  - gdcm::Surface, [1060](#)
- GetMaxLength
  - gdcm::PersonName, [797](#)
- GetMaxPDULength
  - gdcm::network::ULConnectionInfo, [1235](#)
- GetMaxPDUSize
  - gdcm::network::ULConnection, [1228](#)
- GetMD5DataImage
  - gdcm::Testing, [1118](#)
- GetMD5DataImages
  - gdcm::Testing, [1119](#)
- GetMD5FromBrokenFile
  - gdcm::Testing, [1119](#)
- GetMD5FromFile
  - gdcm::Testing, [1119](#)
- GetMD5MetaImage
  - vtkGDCMTesting, [1348](#)
- GetMeanPointDistance
  - gdcm::Surface, [1060](#)
- GetMediaStorage
  - gdcm::DataSet, [334](#)
  - gdcm::FileMetaInformation, [473](#)
- GetMediaStorageAsString
  - gdcm::FileMetaInformation, [473](#)
- GetMediaStorageDataFile
  - gdcm::Testing, [1119](#)
- GetMediaStorageDataFiles
  - gdcm::Testing, [1119](#)
- GetMediaStorageFromFile
  - gdcm::Testing, [1119](#)
- GetMeshPrimitive
  - gdcm::Surface, [1060](#)
- GetMessageHeader
  - gdcm::network::PresentationDataValue, [857](#)
- GetMetaInformationTS
  - gdcm::FileMetaInformation, [473](#)
- GetMHDMD5FromFile
  - vtkGDCMTesting, [1348](#)
- GetMin
  - gdcm::PixelFormat, [810](#)
- GetMinorVersion
  - gdcm::Version, [1272](#)
- GetModality
  - gdcm::MediaStorage, [679](#)
- GetModalityDimension
  - gdcm::MediaStorage, [679](#)
- GetModule
  - gdcm::Modules, [711](#)
- GetModuleEntry
  - gdcm::NestedModuleEntries, [731](#)
- GetModuleEntryInMacros
  - gdcm::Module, [704](#)
- GetModules
  - gdcm::Defs, [350](#)
- GetMPTType
  - gdcm::MeshPrimitive, [692](#)
- GetMPTTypeString
  - gdcm::MeshPrimitive, [692](#)
- GetMRImageSeriesUIDs
  - gdcm::DirectoryHelper, [391](#)
- GetMrProtocol
  - gdcm::CSAHeader, [291](#)
- GetMrProtocolByName
  - gdcm::MrProtocol, [719](#)
- GetMSString
  - gdcm::MediaStorage, [680](#)
- GetMSType
  - gdcm::MediaStorage, [680](#)
- GetMTime
  - vtkImageMapToColors16, [1380](#)
- GetName
  - gdcm::CSAElement, [281](#)
  - gdcm::CSAHeaderDictEntry, [297](#)
  - gdcm::DictEntry, [370](#)
  - gdcm::Filename, [479](#)
  - gdcm::GroupDict, [513](#)
  - gdcm::IODEntry, [598](#)
  - gdcm::Macro, [666](#)
  - gdcm::Module, [704](#)
  - gdcm::ModuleEntry, [708](#)
  - gdcm::network::AbstractSyntax, [105](#)
  - gdcm::network::ApplicationContext, [120](#)
  - gdcm::network::TransferSyntaxSub, [1135](#)
  - gdcm::PDBelement, [785](#)

- gdcmm::QueryBase, 881
- gdcmm::QueryImage, 885
- gdcmm::QueryPatient, 887
- gdcmm::QuerySeries, 890
- gdcmm::QueryStudy, 892
- gdcmm::UIDs, 1181
- GetNeedByteSwap
  - gdcmm::Bitmap, 200
  - gdcmm::ImageCodec, 552
- GetNegotiatedType
  - gdcmm::TransferSyntax, 1132
- GetNestedDataSet
  - gdcmm::Item, 610
- GetNextSingleSeriesUIDFileSet
  - gdcmm::SerieHelper, 973
- GetNoOfItems
  - gdcmm::CSAElement, 281
- GetNumberOfComponents
  - gdcmm::PersonName, 797
- GetNumberOfContourReferencedFrameOfReferences
  - vtkRTStructSetProperties, 1403
- GetNumberOfCurves
  - gdcmm::Curve, 306
  - gdcmm::Pixmap, 818
- GetNumberOfDimensions
  - gdcmm::Bitmap, 200
  - gdcmm::ImageCodec, 552
- GetNumberOfElementsFromArray
  - gdcmm::VM, 1280
- GetNumberOfFileNames
  - gdcmm::Testing, 1120
- GetNumberOfFilenames
  - gdcmm::FilenameGenerator, 487
- GetNumberOfFragments
  - gdcmm::SequenceOfFragments, 958
- GetNumberOfIconImages
  - gdcmm::IconImageFilter, 517
- GetNumberOfImagesInMosaic
  - gdcmm::SplitMosaicFilter, 1018
- GetNumberOfIODs
  - gdcmm::IOD, 595
- GetNumberOfItems
  - gdcmm::SequenceOfItems, 966
- GetNumberOfMD5DataImages
  - gdcmm::Testing, 1120
- GetNumberOfMD5MetalImages
  - vtkGDCMTesting, 1348
- GetNumberOfMediaStorageDataFiles
  - gdcmm::Testing, 1120
- GetNumberOfModality
  - gdcmm::MediaStorage, 680
- GetNumberOfModuleEntries
  - gdcmm::NestedModuleEntries, 731
- GetNumberOfMSSString
  - gdcmm::MediaStorage, 680
- GetNumberOfMSType
  - gdcmm::MediaStorage, 680
- GetNumberOfOverlays
  - gdcmm::Pixmap, 818
- GetNumberOfPoints
  - gdcmm::Curve, 306
- GetNumberOfPresentationContext
  - gdcmm::network::AAssociateRQPDU, 100
- GetNumberOfPresentationContextAC
  - gdcmm::network::AAssociateACPDU, 92
- GetNumberOfPresentationDataValues
  - gdcmm::network::PDataTFPDU, 782
- GetNumberOfPrimitivesData
  - gdcmm::MeshPrimitive, 693
- GetNumberOfReferencedFrameOfReferences
  - vtkRTStructSetProperties, 1403
- GetNumberOfSegments
  - gdcmm::SegmentWriter, 951
- GetNumberOfSOPClassToIOD
  - gdcmm::SOPClassUIDToIOD, 1006
- GetNumberOfStructureSetROIs
  - vtkRTStructSetProperties, 1403
- GetNumberOfSurfacePoints
  - gdcmm::Surface, 1061
- GetNumberOfSurfaces
  - gdcmm::SurfaceReader, 1075
  - gdcmm::SurfaceWriter, 1078
- GetNumberOfTransferSyntaxes
  - gdcmm::network::PresentationContextRQ, 853
  - gdcmm::PresentationContext, 843
- GetNumberOfTransferSyntaxStrings
  - gdcmm::UIDs, 1181
- GetNumberOfValues
  - gdcmm::Attribute< Group, Element, TVR, TVM >, 138
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 147
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, 156
- GetNumberOfVectors
  - gdcmm::Surface, 1061
- GetObliquityThresholdCosineValue
  - gdcmm::Orientation, 762
- GetOffScreenRendering
  - vtkImageColorViewer, 1367
- GetOptionalTags
  - gdcmm::QueryBase, 881
  - gdcmm::QueryImage, 885
  - gdcmm::QueryPatient, 888
  - gdcmm::QuerySeries, 890
  - gdcmm::QueryStudy, 893
- GetOrderedValues
  - gdcmm::Scanner, 929
  - gdcmm::StrictScanner, 1038

- GetOrigin
  - gdcm::Image, [525](#)
  - gdcm::Overlay, [768](#)
- GetOriginValue
  - gdcm::ImageHelper, [567](#)
- GetOutput
  - gdcm::ImageConverter, [560](#)
- GetOutput
  - gdcm::BitmapToBitmapFilter, [211](#)
  - gdcm::ImageToImageFilter, [581](#)
  - gdcm::PixmapToPixmapFilter, [827](#)
- GetOutputAsBitmap
  - gdcm::BitmapToBitmapFilter, [211](#)
- GetOutputAsPixmap
  - gdcm::PixmapToPixmapFilter, [827](#)
- GetOutputFilename
  - gdcm::DictConverter, [366](#)
- GetOutputType
  - gdcm::DictConverter, [366](#)
- GetOverlay
  - gdcm::Pixmap, [818](#), [819](#)
  - vtkGDCMImageReader, [1298](#)
  - vtkGDCMImageReader2, [1313](#)
- GetOverlayData
  - gdcm::Overlay, [768](#)
- GetOverlayPort
  - vtkGDCMImageReader2, [1313](#)
- GetOverlayTypeAsString
  - gdcm::Overlay, [768](#)
- GetOverlayTypeFromString
  - gdcm::Overlay, [769](#)
- GetOverlayVisibility
  - vtkImageColorViewer, [1367](#)
- GetOwner
  - gdcm::PrivateTag, [869](#)
- GetPath
  - gdcm::Filename, [479](#)
- GetPattern
  - gdcm::FilenameGenerator, [488](#)
- GetPDPEnd
  - gdcm::PDBHeader, [789](#)
- GetPDBElementByName
  - gdcm::PDBHeader, [789](#)
- GetPDBInfoTag
  - gdcm::PDBHeader, [789](#)
- GetPDUs
  - gdcm::network::ULEvent, [1245](#)
- GetPDVs
  - gdcm::network::PDUFactory, [796](#)
- GetPermissions
  - gdcm::System, [1089](#)
- GetPhotometricInterpretation
  - gdcm::Bitmap, [200](#)
  - gdcm::ImageChangePhotometricInterpretation, [535](#)
  - gdcm::ImageCodec, [553](#)
- GetPhotometricInterpretationValue
  - gdcm::ImageHelper, [567](#)
- GetPIString
  - gdcm::PhotometricInterpretation, [804](#)
- GetPIType
  - gdcm::PhotometricInterpretation, [804](#)
- GetPixelFormat
  - gdcm::Bitmap, [200](#), [201](#)
  - gdcm::ImageCodec, [553](#)
- GetPixelFormatValue
  - gdcm::ImageHelper, [567](#)
- GetPixelRepresentation
  - gdcm::PixelFormat, [810](#)
- GetPixelSize
  - gdcm::PixelFormat, [810](#)
- GetPixelSpacingDataRoot
  - gdcm::Testing, [1120](#)
- GetPixmap
  - gdcm::FileDecompressLookupTable, [460](#)
  - gdcm::IconImageGenerator, [520](#)
  - gdcm::PixmapReader, [823](#)
  - gdcm::PixmapWriter, [831](#)
- GetPlanarConfiguration
  - gdcm::Bitmap, [201](#)
  - gdcm::ImageChangePlanarConfiguration, [538](#)
  - gdcm::ImageCodec, [553](#)
- GetPlanarConfigurationValue
  - gdcm::ImageHelper, [567](#)
- GetPMSRescaleInterceptSlope
  - gdcm::ImageHelper, [567](#)
- GetPointCoordinatesData
  - gdcm::Surface, [1061](#)
- GetPointer
  - gdcm::ByteValue, [227](#)
  - gdcm::LookupTable, [660](#)
  - gdcm::SmartPointer< ObjectType >, [1001](#)
  - vtkLookupTable16, [1397](#)
- GetPointerFromElement
  - gdcm::ImageHelper, [568](#)
- GetPointPositionAccuracy
  - gdcm::Surface, [1061](#)
- GetPointsBoundingBoxCoordinates
  - gdcm::Surface, [1061](#)
- GetPosition
  - vtkImageColorViewer, [1367](#)
- GetPreamble
  - gdcm::FileMetaInformation, [473](#), [474](#)
- GetPrefix
  - gdcm::FilenameGenerator, [488](#)
- GetPresentationContext
  - gdcm::network::AAssociateRQPDU, [100](#)
- GetPresentationContextAC
  - gdcm::network::AAssociateACPDU, [92](#)

- GetPresentationContextACByID
  - gdcm::network::ULConnection, [1228](#)
- GetPresentationContextByAbstractSyntax
  - gdcm::network::AAssociateRQPDU, [100](#)
- GetPresentationContextByID
  - gdcm::network::AAssociateRQPDU, [100](#)
- GetPresentationContextID
  - gdcm::network::PresentationContextAC, [846](#)
  - gdcm::network::PresentationContextRQ, [853](#)
  - gdcm::network::PresentationDataValue, [857](#)
  - gdcm::PresentationContext, [843](#)
- GetPresentationContextIDFromPresentationContext
  - gdcm::network::ULConnection, [1228](#)
- GetPresentationContextRQByID
  - gdcm::network::ULConnection, [1228](#)
- GetPresentationContexts
  - gdcm::network::AAssociateRQPDU, [100](#)
  - gdcm::network::ULConnection, [1229](#)
  - gdcm::PresentationContextGenerator, [850](#)
- GetPresentationDataValue
  - gdcm::network::PDataTFPDU, [783](#)
- GetPrettyPrint
  - gdcm::JSON, [646](#)
- GetPrimitiveData
  - gdcm::MeshPrimitive, [693](#)
- GetPrimitivesData
  - gdcm::MeshPrimitive, [693](#)
- GetPrimitiveType
  - gdcm::MeshPrimitive, [694](#)
- GetPrintStyle
  - gdcm::Printer, [862](#)
  - gdcm::XMLPrinter, [1425](#)
- GetPrivateCreator
  - gdcm::DataSet, [334](#)
  - gdcm::Tag, [1105](#)
- GetPrivateDict
  - gdcm::Dicts, [378](#), [379](#)
  - gdcm::XMLPrivateDictReader, [1429](#)
- GetProcessingAlgorithm
  - gdcm::Surface, [1062](#)
- GetProgress
  - gdcm::ProgressEvent, [873](#)
- GetPropertyCategory
  - gdcm::Segment, [936](#)
- GetPropertyType
  - gdcm::Segment, [936](#), [937](#)
- GetPropertyTypeModifiers
  - gdcm::Segment, [937](#)
- GetProtocol
  - gdcm::network::ULConnection, [1229](#)
- GetPublicDict
  - gdcm::Dicts, [379](#)
- GetQuality
  - gdcm::JPEG2000Codec, [623](#)
- gdcm::JPEGCodec, [635](#)
- GetQueryDataSet
  - gdcm::BaseQuery, [180](#)
- GetQueryLevel
  - gdcm::QueryBase, [881](#)
  - gdcm::QueryImage, [885](#)
  - gdcm::QueryPatient, [888](#)
  - gdcm::QuerySeries, [890](#)
  - gdcm::QueryStudy, [893](#)
- GetQueryLevelFromQueryRoot
  - gdcm::BaseRootQuery, [185](#)
- GetQueryLevelFromString
  - gdcm::BaseRootQuery, [185](#)
- GetQueryLevelString
  - gdcm::BaseRootQuery, [185](#)
- GetRate
  - gdcm::JPEG2000Codec, [624](#)
- GetRAWMD5FromFile
  - vtkGDCMTesting, [1349](#)
- GetRealWorldValueMappingContent
  - gdcm::ImageHelper, [568](#)
- GetReason
  - gdcm::network::PresentationContextAC, [846](#)
- GetRecommendedDisplayCIELabValue
  - gdcm::Surface, [1062](#)
- GetRecommendedDisplayGrayscaleValue
  - gdcm::Surface, [1062](#)
- GetRecommendedPresentationOpacity
  - gdcm::Surface, [1062](#)
- GetRecommendedPresentationType
  - gdcm::Surface, [1063](#)
- GetRef
  - gdcm::IODEntry, [598](#)
- GetReferencedFrameOfReferenceClassUID
  - vtkRTStructSetProperties, [1404](#)
- GetReferencedFrameOfReferenceInstanceUID
  - vtkRTStructSetProperties, [1404](#)
- GetRegion
  - gdcm::ImageRegionReader, [578](#)
- GetRequiredDataSet
  - gdcm::ModalityPerformedProcedureStepCreateQuery, [697](#)
  - gdcm::ModalityPerformedProcedureStepSetQuery, [700](#)
- GetRequiredTags
  - gdcm::QueryBase, [881](#)
  - gdcm::QueryImage, [886](#)
  - gdcm::QueryPatient, [888](#)
  - gdcm::QuerySeries, [890](#)
  - gdcm::QueryStudy, [893](#)
- GetRescaleInterceptSlopeValue
  - gdcm::ImageHelper, [568](#)
- GetReserved43\_74
  - gdcm::network::AAssociateRQPDU, [100](#)

- GetResponses
  - gdcm::network::ULBasicCallback, [1224](#)
- GetRetired
  - gdcm::DictEntry, [370](#)
- GetRoot
  - gdcm::UIDGenerator, [1144](#)
- GetRows
  - gdcm::Bitmap, [201](#)
  - gdcm::Overlay, [769](#)
- GetRTStructSeriesUIDs
  - gdcm::DirectoryHelper, [391](#)
- GetSamplesPerPixel
  - gdcm::PhotometricInterpretation, [805](#)
  - gdcm::PixelFormat, [811](#)
- GetScalarType
  - gdcm::PixelFormat, [811](#)
- GetScalarTypeAsString
  - gdcm::PixelFormat, [811](#)
- GetScanner
  - gdcm::DICOMDIRGenerator, [358](#)
- GetSegment
  - gdcm::SegmentWriter, [951](#)
- GetSegmentAlgorithmName
  - gdcm::Segment, [937](#)
- GetSegmentAlgorithmType
  - gdcm::Segment, [937](#)
- GetSegmentDescription
  - gdcm::Segment, [937](#)
- GetSegmentLabel
  - gdcm::Segment, [937](#)
- GetSegmentNumber
  - gdcm::Segment, [938](#)
- GetSegments
  - gdcm::SegmentReader, [947](#)
  - gdcm::SegmentWriter, [951](#)
- GetSelectedPrivateGroupOffsetFromFile
  - gdcm::Testing, [1120](#)
- GetSelectedTagsOffsetFromFile
  - gdcm::Testing, [1121](#)
- GetSequenceOfFragments
  - gdcm::DataElement, [314](#)
- GetSeriesUIDsBySOPClassUID
  - gdcm::DirectoryHelper, [391](#)
- GetSize
  - gdcm::VR, [1286](#)
  - vtkImageColorViewer, [1367](#)
- GetSizeof
  - gdcm::VR, [1286](#)
- GetSliceArray
  - gdcm::MrProtocol, [719](#)
- GetSliceMax
  - vtkImageColorViewer, [1367](#)
- GetSliceMin
  - vtkImageColorViewer, [1368](#)
- GetSliceRange
  - vtkImageColorViewer, [1368](#)
- GetSlope
  - gdcm::Image, [526](#)
  - gdcm::Rescaler, [912](#)
- GetSOPClassUID
  - gdcm::DirectoryHelper, [391](#)
- GetSOPClassUIDFromIOD
  - gdcm::SOPClassUIDToIOD, [1006](#)
- GetSOPClassUIDToIOD
  - gdcm::SOPClassUIDToIOD, [1006](#)
- GetSOPClassUIDToIODs
  - gdcm::SOPClassUIDToIOD, [1006](#)
- GetSOPInstanceUID
  - gdcm::BaseQuery, [180](#)
- GetSourceApplicationEntityTitle
  - gdcm::FileMetaInformation, [474](#)
- GetSourceDirectory
  - gdcm::Testing, [1121](#)
- GetSpacing
  - gdcm::Image, [526](#)
- GetSpacingTagFromMediaStorage
  - gdcm::ImageHelper, [568](#)
- GetSpacingValue
  - gdcm::ImageHelper, [568](#)
- GetStart
  - gdcm::ByteBuffer, [218](#)
- GetState
  - gdcm::network::ULConnection, [1229](#)
- GetStateIndex
  - gdcm::network, [83](#)
- GetSTATES
  - gdcm::Surface, [1063](#)
- GetSTATESString
  - gdcm::Surface, [1063](#)
- GetStream
  - gdcm::Trace, [1126](#)
- GetStreamCurrentPosition
  - gdcm::Reader, [902](#)
- GetStreamOffsetFromFile
  - gdcm::Testing, [1121](#)
- GetStreamPtr
  - gdcm::Reader, [902](#)
  - gdcm::Writer, [1418](#)
- GetString
  - gdcm::MediaStorage, [681](#)
  - gdcm::PhotometricInterpretation, [805](#)
  - gdcm::TransferSyntax, [1132](#)
  - gdcm::UIDs, [1181](#)
- GetStringValueFromTag
  - gdcm::DirectoryHelper, [392](#)
- GetStructureSetObservationNumber
  - vtkRTStructSetProperties, [1404](#)
- GetStructureSetROIDescription

- vtkRTStructSetProperties, [1404](#)
- GetStructureSetROIGenerationAlgorithm
  - vtkRTStructSetProperties, [1404](#)
- GetStructureSetROIName
  - vtkRTStructSetProperties, [1404](#)
- GetStructureSetROINumber
  - vtkRTStructSetProperties, [1405](#)
- GetStructureSetROIObservationLabel
  - vtkRTStructSetProperties, [1405](#)
- GetStructureSetROIRefFrameRefUID
  - vtkRTStructSetProperties, [1405](#)
- GetStructureSetRTROIInterpretedType
  - vtkRTStructSetProperties, [1405](#)
- GetSurface
  - gdcm::Segment, [938](#)
- GetSurfaceComments
  - gdcm::Surface, [1063](#)
- GetSurfaceCount
  - gdcm::Segment, [938](#)
- GetSurfaceNumber
  - gdcm::Surface, [1063](#)
- GetSurfaceProcessing
  - gdcm::Surface, [1063](#)
- GetSurfaceProcessingDescription
  - gdcm::Surface, [1063](#)
- GetSurfaceProcessingRatio
  - gdcm::Surface, [1064](#)
- GetSurfaces
  - gdcm::Segment, [938](#)
- GetSwapCode
  - gdcm::TransferSyntax, [1132](#)
- GetSwapCodeString
  - gdcm::SwapCode, [1081](#)
- GetSyngoDT
  - gdcm::CSAElement, [281](#)
- GetTable
  - gdcm::SequenceOfFragments, [958](#)
- GetTableEntry
  - gdcm::Table, [1093](#)
- GetTag
  - gdcm::AnonymizeEvent, [109](#)
  - gdcm::Attribute< Group, Element, TVR, TVM >, [138](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [147](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [156](#)
  - gdcm::DataElement, [314](#)
- GetTagListByLevel
  - gdcm::BaseRootQuery, [186](#)
  - gdcm::FindPatientRootQuery, [500](#)
  - gdcm::FindStudyRootQuery, [503](#)
  - gdcm::MovePatientRootQuery, [713](#)
  - gdcm::MoveStudyRootQuery, [716](#)
  - gdcm::WLMFindQuery, [1413](#)
- GetTempDirectory
  - gdcm::Testing, [1121](#)
- GetTempDirectoryW
  - gdcm::Testing, [1121](#)
- GetTempFilename
  - gdcm::Testing, [1122](#)
- GetTempFilenameW
  - gdcm::Testing, [1122](#)
- GetTimeout
  - gdcm::network::ARTIMTimer, [130](#)
  - gdcm::ServiceClassUser, [981](#)
- GetTimer
  - gdcm::network::ULConnection, [1229](#)
- GetTimezoneOffsetFromUTC
  - gdcm::System, [1089](#)
- GetToplevel
  - gdcm::Directory, [388](#)
- GetTransferSyntax
  - gdcm::Bitmap, [201](#)
  - gdcm::ImageChangeTransferSyntax, [543](#)
  - gdcm::network::PresentationContextAC, [846](#)
  - gdcm::network::PresentationContextRQ, [853](#)
  - gdcm::PresentationContext, [843](#)
- GetTransferSyntaxes
  - gdcm::network::PresentationContextRQ, [854](#)
- GetTransferSyntaxString
  - gdcm::UIDs, [1181](#)
- GetTransferSyntaxStrings
  - gdcm::UIDs, [1181](#)
- GetTSString
  - gdcm::TransferSyntax, [1132](#)
- GetTSType
  - gdcm::TransferSyntax, [1133](#)
- GetType
  - gdcm::ModuleEntry, [708](#)
  - gdcm::Orientation, [762](#)
  - gdcm::Overlay, [769](#)
  - gdcm::PhotometricInterpretation, [805](#)
- GetTypeAsEnum
  - gdcm::Overlay, [769](#)
- GetTypeFromTag
  - gdcm::Defs, [350](#)
  - gdcm::IOD, [596](#)
- GetTypeOfData
  - gdcm::Curve, [306](#)
- GetTypeOfDataDescription
  - gdcm::Curve, [306](#)
- GetTypeString
  - gdcm::Type, [1141](#)
- GetTypeType
  - gdcm::Type, [1141](#)
- GetUIDName
  - gdcm::UIDs, [1182](#)
- GetUIDString



- gdcM::UIDs, [1182](#)
- GetUniqueTags
  - gdcM::QueryBase, [882](#)
  - gdcM::QueryImage, [886](#)
  - gdcM::QueryPatient, [888](#)
  - gdcM::QuerySeries, [891](#)
  - gdcM::QueryStudy, [893](#)
- GetUnpackBuffer
  - gdcM::Overlay, [769](#)
- GetUnpackBufferLength
  - gdcM::Overlay, [769](#)
- GetUsage
  - gdcM::IODEntry, [598](#)
- GetUsageString
  - gdcM::Usage, [1258](#)
- GetUsageType
  - gdcM::IODEntry, [598](#)
  - gdcM::Usage, [1258](#)
- GetUserData
  - gdcM::Parser, [779](#)
- GetUserInformation
  - gdcM::network::AAssociateACPDU, [92](#)
  - gdcM::network::AAssociateRQPDU, [101](#)
- GetValidatedFile
  - gdcM::Validate, [1264](#)
- GetValidDataSet
  - gdcM::WLMFindQuery, [1413](#)
- GetValue
  - gdcM::Attribute< Group, Element, TVR, TVM >, [138](#), [139](#)
  - gdcM::Attribute< Group, Element, TVR, VM::VM1 >, [147](#)
  - gdcM::Attribute< Group, Element, TVR, VM::VM1\_n >, [156](#)
  - gdcM::CSAElement, [282](#)
  - gdcM::DataElement, [314](#), [315](#)
  - gdcM::Element< TVR, TVM >, [398](#)
  - gdcM::Element< TVR, VM::VM1\_n >, [405](#)
  - gdcM::PDSElement, [786](#)
  - gdcM::Scanner, [929](#)
  - gdcM::StrictScanner, [1038](#)
- GetValueAsSQ
  - gdcM::DataElement, [315](#)
- GetValues
  - gdcM::Attribute< Group, Element, TVR, TVM >, [139](#)
  - gdcM::Attribute< Group, Element, TVR, VM::VM1 >, [148](#)
  - gdcM::Attribute< Group, Element, TVR, VM::VM1\_n >, [157](#)
  - gdcM::Element< TVR, TVM >, [398](#)
  - gdcM::Scanner, [929](#), [930](#)
  - gdcM::StrictScanner, [1038](#)
- GetVectorAccuracy
  - gdcM::Surface, [1064](#)
- GetVectorCoordinateData
  - gdcM::Surface, [1064](#)
- GetVectorDimensionality
  - gdcM::Surface, [1064](#)
- GetVersion
  - gdcM::MrProtocol, [719](#)
  - gdcM::Version, [1272](#)
- GetVIEWType
  - gdcM::Surface, [1064](#)
- GetVIEWTypeString
  - gdcM::Surface, [1064](#)
- GetVL
  - gdcM::DataElement, [315](#), [316](#)
- GetVL16Max
  - gdcM::VL, [1274](#)
- GetVL32Max
  - gdcM::VL, [1275](#)
- GetVM
  - gdcM::Attribute< Group, Element, TVR, TVM >, [139](#)
  - gdcM::Attribute< Group, Element, TVR, VM::VM1 >, [148](#)
  - gdcM::Attribute< Group, Element, TVR, VM::VM1\_3 >, [151](#)
  - gdcM::Attribute< Group, Element, TVR, VM::VM1\_8 >, [153](#)
  - gdcM::Attribute< Group, Element, TVR, VM::VM1\_n >, [157](#)
  - gdcM::Attribute< Group, Element, TVR, VM::VM2\_2n >, [161](#)
  - gdcM::Attribute< Group, Element, TVR, VM::VM2\_n >, [162](#)
  - gdcM::Attribute< Group, Element, TVR, VM::VM3\_3n >, [164](#)
  - gdcM::Attribute< Group, Element, TVR, VM::VM3\_n >, [165](#)
  - gdcM::CSAElement, [282](#)
  - gdcM::CSAHeaderDictEntry, [297](#)
  - gdcM::DictEntry, [370](#)
  - gdcM::Element< TVR, TVM >, [398](#)
  - gdcM::Element< TVR, VM::VM1\_n >, [405](#)
- GetVMString
  - gdcM::VM, [1280](#)
- GetVMType
  - gdcM::VM, [1281](#)
- GetVMTypeFromLength
  - gdcM::VM, [1281](#)
- GetVoidPointer
  - gdcM::ByteValue, [227](#)
- GetVR
  - gdcM::Attribute< Group, Element, TVR, TVM >, [139](#)
  - gdcM::Attribute< Group, Element, TVR, VM::VM1 >, [148](#)
  - gdcM::Attribute< Group, Element, TVR, VM::VM1\_n >, [157](#)

- gdcm::CSAElement, [282](#)
- gdcm::CSAHeaderDictEntry, [297](#)
- gdcm::DataElement, [316](#)
- gdcm::DictEntry, [370](#)
- gdcm::Element< TVR, TVM >, [398](#)
- gdcm::Element< TVR, VM::VM1\_n >, [405](#)
- GetVRFromTag
  - gdcm, [64](#)
- GetVRString
  - gdcm::VR, [1286](#)
- GetVRStringFromFile
  - gdcm::VR, [1287](#)
- GetVRType
  - gdcm::VR, [1287](#)
- GetVRTypeFromFile
  - gdcm::VR, [1287](#)
- GetVTKDataRoot
  - vtkGDCMTesting, [1349](#)
- GetWarningFlag
  - gdcm::Trace, [1126](#)
- GetWarningStream
  - gdcm::Trace, [1126](#)
- GetWindowName
  - vtkImageColorViewer, [1368](#)
- GetXMax
  - gdcm::BoxRegion, [215](#)
- GetXMin
  - gdcm::BoxRegion, [215](#)
- GetYMax
  - gdcm::BoxRegion, [216](#)
- GetYMin
  - gdcm::BoxRegion, [216](#)
- GetZMax
  - gdcm::BoxRegion, [216](#)
- GetZMin
  - gdcm::BoxRegion, [216](#)
- GetZSpacing
  - gdcm::IPPSorter, [604](#)
- GetZSpacingTagFromMediaStorage
  - gdcm::ImageHelper, [569](#)
- GetZSpacingTolerance
  - gdcm::IPPSorter, [605](#)
- Global
  - gdcm::Defs, [351](#)
  - gdcm::Dicts, [379](#)
  - gdcm::Global, [508](#), [509](#)
- GlobalInstance
  - gdcm, [76](#)
- GrabOverlayFromPixelData
  - gdcm::Overlay, [770](#)
- Graphics
  - gdcm::Overlay, [766](#)
- GRAY
  - gdcm::LookupTable, [657](#)
- GrayscalePlanarMPRVolumetricPresentationStateStorage
  - gdcm::UIDs, [1169](#)
- GrayscaleSoftcopyPresentationStateStorageSOPClass
  - gdcm::MediaStorage, [677](#)
  - gdcm::UIDs, [1165](#)
- GREEN
  - gdcm::LookupTable, [657](#)
- green
  - gdcm::terminal, [85](#)
- GroupDict
  - gdcm::GroupDict, [512](#)
- GroupStringVector
  - gdcm::GroupDict, [512](#)
- GuessFromModality
  - gdcm::MediaStorage, [681](#)
- HandleBulkData
  - gdcm::XMLPrinter, [1425](#)
- HandleDataSet
  - gdcm::network::ULBasicCallback, [1225](#)
  - gdcm::network::ULConnectionCallback, [1233](#)
  - gdcm::network::ULWritingCallback, [1249](#)
- HandleDescription
  - gdcm::XMLDictReader, [1422](#)
  - gdcm::XMLPrivateDictReader, [1429](#)
- HandleEntry
  - gdcm::XMLDictReader, [1422](#)
  - gdcm::XMLPrivateDictReader, [1429](#)
- HandleEvent
  - gdcm::network::ULTransitionTable, [1247](#)
- HandleIOD
  - gdcm::TableReader, [1098](#)
- HandleIODEntry
  - gdcm::TableReader, [1098](#)
- HandleMacro
  - gdcm::TableReader, [1098](#)
- HandleMacroEntry
  - gdcm::TableReader, [1098](#)
- HandleMacroEntryDescription
  - gdcm::TableReader, [1098](#)
- HandleModule
  - gdcm::TableReader, [1098](#)
- HandleModuleEntry
  - gdcm::TableReader, [1099](#)
- HandleModuleEntryDescription
  - gdcm::TableReader, [1099](#)
- HandleModuleInclude
  - gdcm::TableReader, [1099](#)
- HandleResponse
  - gdcm::network::ULBasicCallback, [1225](#)
  - gdcm::network::ULConnectionCallback, [1233](#)
  - gdcm::network::ULWritingCallback, [1249](#)
- HangingProtocolInformationModelFIND
  - gdcm::UIDs, [1167](#)



HangingProtocolInformationModelGET  
gdcm::UIDs, [1171](#)

HangingProtocolInformationModelMOVE  
gdcm::UIDs, [1167](#)

HangingProtocolStorage  
gdcm::MediaStorage, [678](#)  
gdcm::UIDs, [1167](#)

HardcopyColorImageStorage  
gdcm::MediaStorage, [678](#)

HardcopyColorImageStorageSOPClassRetired  
gdcm::UIDs, [1164](#)

HardcopyGrayscaleImageStorage  
gdcm::MediaStorage, [677](#)

HardcopyGrayscaleImageStorageSOPClassRetired  
gdcm::UIDs, [1164](#)

HasObserver  
gdcm::Subject, [1054](#)

HemodynamicWaveformStorage  
gdcm::MediaStorage, [677](#)  
gdcm::UIDs, [1165](#)

HEVCH\_265Main10ProfileLevel5\_1  
gdcm::UIDs, [1169](#)

HEVCH\_265MainProfileLevel5\_1  
gdcm::UIDs, [1169](#)

hidden  
gdcm::terminal, [85](#)

HITACHI  
gdcm::EquipmentManufacturer, [431](#)

HotIronColorPaletteSOPInstance  
gdcm::UIDs, [1169](#)

HotMetalBlueColorPaletteSOPInstance  
gdcm::UIDs, [1168](#)

HSV  
gdcm::PhotometricInterpretation, [804](#)

ICBM452T1FrameofReference  
gdcm::UIDs, [1163](#)

ICBMSingleSubjectMRIFrameofReference  
gdcm::UIDs, [1163](#)

ICD11  
gdcm::UIDs, [1168](#)

Icon  
gdcm::Pixmap, [820](#)

IconDataScalarType  
vtkGDCMImageReader, [1307](#)  
vtkGDCMImageReader2, [1321](#)

IconImage  
gdcm, [59](#)

IconImageDataExtent  
vtkGDCMImageReader, [1307](#)  
vtkGDCMImageReader2, [1321](#)

IconImageFilter  
gdcm::IconImageFilter, [515](#)

IconImageGenerator  
gdcm::IconImageGenerator, [518](#)

IconNumberOfScalarComponents  
vtkGDCMImageReader, [1307](#)  
vtkGDCMImageReader2, [1322](#)

ID  
gdcm::PresentationContext, [844](#)

ignore\_char  
gdcm::ignore\_char, [521](#)

Image  
gdcm::Image, [524](#)

ImageActor  
vtkImageColorViewer, [1376](#)

ImageApplyLookupTable  
gdcm::ImageApplyLookupTable, [531](#)

ImageBiomarkerStandardisationInitiative  
gdcm::UIDs, [1168](#)

ImageChangePhotometricInterpretation  
gdcm::ImageChangePhotometricInterpretation, [534](#)  
gdcm::ImageCodec, [557](#)

ImageChangePlanarConfiguration  
gdcm::ImageChangePlanarConfiguration, [538](#)

ImageChangeTransferSyntax  
gdcm::Bitmap, [207](#)  
gdcm::ImageChangeTransferSyntax, [542](#)

ImageCodec  
gdcm::ImageCodec, [548](#)

ImageConverter  
gdcm::ImageConverter, [559](#)

ImageFormat  
vtkGDCMImageReader, [1307](#)  
vtkGDCMImageReader2, [1322](#)

ImageFragmentSplitter  
gdcm::ImageFragmentSplitter, [563](#)

ImageNumberOrdering  
gdcm::SerieHelper, [973](#)

ImageOrientationPatient  
vtkGDCMImageReader, [1307](#)  
vtkGDCMImageReader2, [1322](#)

ImageOverlayBoxSOPClassRetired  
gdcm::UIDs, [1164](#)

ImagePositionPatient  
vtkGDCMImageReader, [1307](#)  
vtkGDCMImageReader2, [1322](#)

ImagePositionPatientOrdering  
gdcm::SerieHelper, [973](#)

ImageReader  
gdcm::ImageReader, [573](#)

ImageRegionReader  
gdcm::ImageRegionReader, [577](#)  
gdcm::JPEG2000Codec, [626](#)  
gdcm::JPEGCodec, [638](#)  
gdcm::JPEGLSCCodec, [645](#)  
gdcm::RLECodec, [920](#)

ImageToImageFilter

- gdcm::ImageToImageFilter, 581
- ImageWriter
  - gdcm::ImageWriter, 584
- ImplantAssemblyTemplateInformationModelFIND
  - gdcm::UIDs, 1171
- ImplantAssemblyTemplateInformationModelGET
  - gdcm::UIDs, 1171
- ImplantAssemblyTemplateInformationModelMOVE
  - gdcm::UIDs, 1171
- ImplantAssemblyTemplateStorage
  - gdcm::UIDs, 1171
- ImplantationPlanSRStorage
  - gdcm::UIDs, 1170
- ImplantTemplateGroupInformationModelFIND
  - gdcm::UIDs, 1171
- ImplantTemplateGroupInformationModelGET
  - gdcm::UIDs, 1171
- ImplantTemplateGroupInformationModelMOVE
  - gdcm::UIDs, 1171
- ImplantTemplateGroupStorage
  - gdcm::UIDs, 1171
- ImplementationClassUIDSub
  - gdcm::network::ImplementationClassUIDSub, 586
- ImplementationUIDSub
  - gdcm::network::ImplementationUIDSub, 587
- ImplementationVersionNameSub
  - gdcm::network::ImplementationVersionNameSub, 588
- Implicit
  - gdcm::TransferSyntax, 1130
- ImplicitVRBigEndianACRNEMA
  - gdcm::TransferSyntax, 1131
- ImplicitVRBigEndianPrivateGE
  - gdcm::TransferSyntax, 1131
- ImplicitVRLittleEndian
  - gdcm::TransferSyntax, 1131
- ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM
  - gdcm::UIDs, 1162
- IncompleteLUT
  - gdcm::LookupTable, 663
- InitFromRQ
  - gdcm::network::AAssociateACPDU, 92
- Initialize
  - gdcm::network::ULConnectionInfo, 1235
- InitializeBlueLUT
  - gdcm::LookupTable, 660
- InitializeConnection
  - gdcm::network::ULConnection, 1229
  - gdcm::ServiceClassUser, 981
- Initialized
  - gdcm::LookupTable, 660
- InitializeDataSet
  - gdcm::BaseRootQuery, 186
  - gdcm::FindPatientRootQuery, 500
  - gdcm::FindStudyRootQuery, 503
  - gdcm::MovePatientRootQuery, 713
  - gdcm::MoveStudyRootQuery, 716
  - gdcm::WLMFindQuery, 1413
- InitializeGreenLUT
  - gdcm::LookupTable, 660
- InitializeIncomingConnection
  - gdcm::network::ULConnection, 1229
- InitializeLUT
  - gdcm::LookupTable, 661
- InitializeRedLUT
  - gdcm::LookupTable, 661
- InitializeRTStructSet
  - vtkGDCMPolyDataWriter, 1343
- InitOpenSSL
  - gdcm::OpenSSLCryptoFactory, 751
- Input
  - gdcm::BitmapToBitmapFilter, 212
- Insert
  - gdcm::CommandDataSet, 261
  - gdcm::DataSet, 334
  - gdcm::FileMetaInformation, 474
  - gdcm::GroupDict, 513
- InsertDataElement
  - gdcm::DataSet, 335
  - gdcm::Item, 610
- InsertEntry
  - gdcm::Table, 1094
- InstallPipeline
  - vtkImageColorViewer, 1368
- InstanceAvailabilityNotificationSOPClass
  - gdcm::UIDs, 1167
- INT12
  - gdcm::PixelFormat, 808
- INT16
  - gdcm::PixelFormat, 808
- INT32
  - gdcm::PixelFormat, 808
- INT64
  - gdcm::PixelFormat, 808
- INT8
  - gdcm::PixelFormat, 808
- IntegratedTaxonomicInformationSystemITISTaxonomicSerialNumberTSN
  - gdcm::UIDs, 1168
- Interactor
  - vtkImageColorViewer, 1376
- InteractorStyle
  - vtkImageColorViewer, 1376
- INTERFILE
  - gdcm::CSAHeader, 288
- Internal
  - gdcm::ApplicationEntity, 123
  - gdcm::Attribute< Group, Element, TVR, TVM >, 143

- gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 150
- gdcm::Element< TVR, TVM >, 400
- gdcm::Element< VR::AS, VM::VM5 >, 416
- gdcm::LookupTable, 663
- gdcm::UI, 1142
- InternalCode
  - gdcm::Coder, 251
  - gdcm::JPEG12Codec, 615
  - gdcm::JPEG16Codec, 618
  - gdcm::JPEG8Codec, 629
- Internals
  - vtkRTStructSetProperties, 1409
- IntraocularLensCalculationsStorage
  - gdcm::UIDs, 1169
- IntravascularOpticalCoherenceTomographyImageStorageForPresentation
  - gdcm::UIDs, 1169
- IntravascularOpticalCoherenceTomographyImageStorageForProcessing
  - gdcm::UIDs, 1169
- INVALID
  - gdcm::VR, 1284
- Invalid
  - gdcm::Overlay, 766
  - gdcm::Usage, 1257
- InverseRescale
  - gdcm::Rescaler, 912
- InverseRescaleFunctionIntoBestFit
  - gdcm::Rescaler, 913
- InvokeEvent
  - gdcm::Subject, 1054
- IOD
  - gdcm::IOD, 595
- IODEntry
  - gdcm::IODEntry, 597
- IODMapType
  - gdcm::IODs, 600
- IODMapTypeConstIterator
  - gdcm::IODs, 600
- IODName
  - gdcm::IODs, 600
- IODs
  - gdcm::IODs, 601
- IPPSorter
  - gdcm::IPPSorter, 604
- IS
  - gdcm::VR, 1284
- IsAETitleValid
  - gdcm::network::AAssociateRQPDU, 101
- IsASCII
  - gdcm::VR, 1287
- IsASCII2
  - gdcm::VR, 1287
- IsBinary
  - gdcm::VR, 1287
- IsBinary2
  - gdcm::VR, 1288
- IsCompatible
  - gdcm::PixelFormat, 811
- IsDual
  - gdcm::VR, 1288
- IsEmpty
  - gdcm::Bitmap, 201
  - gdcm::ByteValue, 228
  - gdcm::CSAElement, 282
  - gdcm::CSAHeaderDict, 294
  - gdcm::Curve, 306
  - gdcm::DataElement, 316
  - gdcm::DataSet, 335
  - gdcm::Defs, 350
  - gdcm::ElementDict, 362
  - gdcm::Dicts, 379
  - gdcm::Filename, 479
  - gdcm::Macros, 669
  - gdcm::Modules, 711
  - gdcm::Overlay, 770
  - gdcm::Preamble, 839
  - gdcm::PrivateDict, 866
  - gdcm::SegmentHelper::BasicCodedEntry, 190
  - gdcm::SequenceOfItems, 966
- IsEncapsulated
  - gdcm::TransferSyntax, 1133
- IsEncoded
  - gdcm::TransferSyntax, 1133
- IsExplicit
  - gdcm::TransferSyntax, 1133
- IsFrameEncoder
  - gdcm::ImageCodec, 553
  - gdcm::JPEG2000Codec, 624
  - gdcm::JPEGCodec, 636
  - gdcm::JPEGLSCCodec, 643
  - gdcm::RLECodec, 919
- IsGroupLength
  - gdcm::Tag, 1105
- IsGroupXX
  - gdcm::Tag, 1105
- IsIdentical
  - gdcm::Filename, 480
- IsIllegal
  - gdcm::Tag, 1106
- IsImage
  - gdcm::MediaStorage, 681
- IsImplicit
  - gdcm::TransferSyntax, 1133
- IsInPixelData
  - gdcm::Overlay, 770
- IsKey
  - gdcm::Scanner, 930
  - gdcm::StrictScanner, 1039

- IsLastFragment
  - gdcm::network::AAAbortPDU, [88](#)
  - gdcm::network::AAssociateACPDU, [92](#)
  - gdcm::network::AAssociateRJPDU, [95](#)
  - gdcm::network::AAssociateRQPDU, [101](#)
  - gdcm::network::AReleaseRPPDU, [125](#)
  - gdcm::network::AReleaseRQPDU, [127](#)
  - gdcm::network::BasePDU, [176](#)
  - gdcm::network::PDataTFPDU, [783](#)
- IsLossless
  - gdcm::PhotometricInterpretation, [805](#)
  - gdcm::TransferSyntax, [1134](#)
- IsLossy
  - gdcm::Bitmap, [202](#)
  - gdcm::ImageCodec, [553](#)
  - gdcm::PhotometricInterpretation, [805](#)
  - gdcm::TransferSyntax, [1134](#)
- IsOdd
  - gdcm::VL, [1275](#)
- IsPresentationContextAccepted
  - gdcm::ServiceClassUser, [981](#)
- IsPrintable
  - gdcm::ByteValue, [228](#)
- IsPrivate
  - gdcm::Tag, [1106](#)
- IsPrivateCreator
  - gdcm::Tag, [1106](#)
- IsPublic
  - gdcm::Tag, [1106](#)
- IsRetired
  - gdcm::PhotometricInterpretation, [805](#)
- IsRGB8
  - gdcm::LookupTable, [661](#)
- IsRowEncoder
  - gdcm::ImageCodec, [554](#)
  - gdcm::JPEG2000Codec, [624](#)
  - gdcm::JPEGCodec, [636](#)
  - gdcm::JPEGLSCCodec, [644](#)
  - gdcm::RLECodec, [919](#)
- IsSameColorSpace
  - gdcm::PhotometricInterpretation, [805](#)
- IsStateSuspension
  - gdcm::JPEG12Codec, [615](#)
  - gdcm::JPEG16Codec, [618](#)
  - gdcm::JPEG8Codec, [629](#)
  - gdcm::JPEGCodec, [636](#)
- IsSwap
  - gdcm::VR, [1288](#)
- IsTransferSyntaxCompatible
  - gdcm::Bitmap, [202](#)
- IsUndefined
  - gdcm::MediaStorage, [681](#)
  - gdcm::VL, [1275](#)
- IsUndefinedLength
  - gdcm::DataElement, [316](#)
  - gdcm::SequenceOfItems, [966](#)
- IsUnique
  - gdcm::DictEntry, [371](#)
- IsValid
  - gdcm::ApplicationEntity, [122](#)
  - gdcm::BoxRegion, [216](#)
  - gdcm::CodeString, [255](#)
  - gdcm::DirectionCosines, [384](#)
  - gdcm::FileMetaInformation, [474](#)
  - gdcm::ImageCodec, [554](#)
  - gdcm::JPEGCodec, [636](#)
  - gdcm::LO, [654](#)
  - gdcm::PixelFormat, [812](#)
  - gdcm::Preamble, [839](#)
  - gdcm::Region, [909](#)
  - gdcm::String< TDelimiter, TMaxLength, TPadChar >, [1045](#)
  - gdcm::TagPath, [1114](#)
  - gdcm::TransferSyntax, [1134](#)
  - gdcm::UIDGenerator, [1144](#)
  - gdcm::UUIDGenerator, [1263](#)
  - gdcm::VM, [1281](#)
  - gdcm::VR, [1288](#)
- IsVRFile
  - gdcm::VR, [1288](#)
- IsZero
  - gdcm::Overlay, [770](#)
- Item
  - gdcm::Item, [609](#)
- Items
  - gdcm::SequenceOfItems, [968](#)
- ItemVector
  - gdcm::SequenceOfItems, [963](#)
- Iterator
  - gdcm::CSAHeaderDict, [293](#)
  - gdcm::DataSet, [330](#)
  - gdcm::Dict, [360](#)
  - gdcm::SequenceOfFragments, [955](#)
  - gdcm::SequenceOfItems, [963](#)
- iterator
  - gdcm::CodeString, [253](#)
  - gdcm::LO, [652](#)
  - gdcm::String< TDelimiter, TMaxLength, TPadChar >, [1043](#)
- ItFileSetHt
  - gdcm::SerieHelper, [974](#)
- IVOCTForPresentation
  - gdcm::MediaStorage, [678](#)
- IVOCTForProcessing
  - gdcm::MediaStorage, [678](#)
- Join
  - gdcm::Filename, [480](#)

- JPEG12Codec
  - gdcm::JPEG12Codec, [614](#)
- JPEG16Codec
  - gdcm::JPEG16Codec, [617](#)
- JPEG2000
  - gdcm::TransferSyntax, [1131](#)
- JPEG2000\_COMPRESSION
  - vtkGDCMImageWriter, [1326](#)
- JPEG2000Codec
  - gdcm::JPEG2000Codec, [621](#)
- JPEG2000ImageCompression
  - gdcm::UIDs, [1162](#)
- JPEG2000ImageCompressionLosslessOnly
  - gdcm::UIDs, [1162](#)
- JPEG2000Lossless
  - gdcm::TransferSyntax, [1131](#)
- JPEG2000Part2
  - gdcm::TransferSyntax, [1131](#)
- JPEG2000Part2Lossless
  - gdcm::TransferSyntax, [1131](#)
- JPEG2000Part2MulticomponentImageCompression
  - gdcm::UIDs, [1162](#)
- JPEG2000Part2MulticomponentImageCompressionLosslessOnly
  - gdcm::UIDs, [1162](#)
- JPEG8Codec
  - gdcm::JPEG8Codec, [628](#)
- JPEG\_COMPRESSION
  - vtkGDCMImageWriter, [1326](#)
- JPEGBaselineProcess1
  - gdcm::TransferSyntax, [1131](#)
- JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BaselineProcess1
  - gdcm::UIDs, [1162](#)
- JPEGCodec
  - gdcm::JPEGCodec, [632](#)
- JPEGExtendedHierarchicalProcess1618Retired
  - gdcm::UIDs, [1162](#)
- JPEGExtendedHierarchicalProcess1719Retired
  - gdcm::UIDs, [1162](#)
- JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG2000
  - gdcm::UIDs, [1162](#)
- JPEGExtendedProcess2\_4
  - gdcm::TransferSyntax, [1131](#)
- JPEGExtendedProcess35Retired
  - gdcm::UIDs, [1162](#)
- JPEGExtendedProcess3\_5
  - gdcm::TransferSyntax, [1131](#)
- JPEGFullProgressionHierarchicalProcess2426Retired
  - gdcm::UIDs, [1162](#)
- JPEGFullProgressionHierarchicalProcess2527Retired
  - gdcm::UIDs, [1162](#)
- JPEGFullProgressionNonHierarchicalProcess1012Retired
  - gdcm::UIDs, [1162](#)
- JPEGFullProgressionNonHierarchicalProcess1113Retired
  - gdcm::UIDs, [1162](#)
- JPEGFullProgressionProcess10\_12
  - gdcm::TransferSyntax, [1131](#)
- JPEGLosslessHierarchicalProcess28Retired
  - gdcm::UIDs, [1162](#)
- JPEGLosslessHierarchicalProcess29Retired
  - gdcm::UIDs, [1162](#)
- JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue
  - gdcm::UIDs, [1162](#)
- JPEGLosslessNonHierarchicalProcess14
  - gdcm::UIDs, [1162](#)
- JPEGLosslessNonHierarchicalProcess15Retired
  - gdcm::UIDs, [1162](#)
- JPEGLosslessProcess14
  - gdcm::TransferSyntax, [1131](#)
- JPEGLosslessProcess14\_1
  - gdcm::TransferSyntax, [1131](#)
- JPEGLS\_COMPRESSION
  - vtkGDCMImageWriter, [1326](#)
- JPEGLSCodec
  - gdcm::JPEGLSCodec, [640](#)
- JPEGLSLossless
  - gdcm::TransferSyntax, [1131](#)
- JPEGLSLosslessImageCompression
  - gdcm::UIDs, [1162](#)
- JPEGLSLossyNearLosslessImageCompression
  - gdcm::UIDs, [1162](#)
- JPEGLSNearLossless
  - gdcm::TransferSyntax, [1131](#)
- JPEGSpectralSelectionHierarchicalProcess2022Retired
  - gdcm::UIDs, [1162](#)
- JPEGSpectralSelectionHierarchicalProcess2123Retired
  - gdcm::UIDs, [1162](#)
- JPEGSpectralSelectionNonHierarchicalProcess68Retired
  - gdcm::UIDs, [1162](#)
- JPEGSpectralSelectionNonHierarchicalProcess79Retired
  - gdcm::UIDs, [1162](#)
- JPEGSpectralSelectionProcess6\_8
  - gdcm::TransferSyntax, [1131](#)
- JPEGPredictiveImageCompressionProcess4only
  - gdcm::TransferSyntax, [1131](#)
  - gdcm::UIDs, [1162](#)
- JPIPReferencedDeflate
  - gdcm::UIDs, [1162](#)
- JSON
  - gdcm::JSON, [645](#)
- JunkAfterDocElementError
  - gdcm::Parser, [778](#)
- KAKADUCodec
  - gdcm::KAKADUCodec, [649](#)
- KeratometryMeasurementsStorage
  - gdcm::UIDs, [1169](#)
- KeyField
  - gdcm::CSAElement, [285](#)

- KeyObjectSelectionDocument
  - gdcm::MediaStorage, [678](#)
- KeyObjectSelectionDocumentStorage
  - gdcm::UIDs, [1166](#)
- KeyValuePairArrayType
  - gdcm::CompositeNetworkFunctions, [265](#)
- KeyValuePairType
  - gdcm::CompositeNetworkFunctions, [265](#)
- KODAK
  - gdcm::EquipmentManufacturer, [431](#)
- LD\_ALL
  - gdcm, [63](#)
- LD\_NOSEQ
  - gdcm, [63](#)
- LD\_NOSHADOW
  - gdcm, [63](#)
- LD\_NOSHADOWSEQ
  - gdcm, [63](#)
- LeadECGWaveformStorage
  - gdcm::MediaStorage, [677](#)
- LegacyConvertedEnhancedCTImageStorage
  - gdcm::MediaStorage, [678](#)
  - gdcm::UIDs, [1168](#)
- LegacyConvertedEnhancedMRImageStorage
  - gdcm::MediaStorage, [678](#)
  - gdcm::UIDs, [1168](#)
- LegacyConvertedEnhancedPETImageStorage
  - gdcm::MediaStorage, [678](#)
  - gdcm::UIDs, [1168](#)
- LensometryMeasurementsStorage
  - gdcm::UIDs, [1169](#)
- Level
  - vtkImageMapToWindowLevelColors2, [1387](#)
- LINE
  - gdcm::MeshPrimitive, [692](#)
- ListCharSets
  - gdcm::QueryFactory, [883](#)
- LittleEndian
  - gdcm::SwapCode, [1081](#)
- LO
  - gdcm::LO, [654](#)
  - gdcm::VR, [1284](#)
- Load
  - gdcm::Directory, [388](#)
  - gdcm::MrProtocol, [719](#)
- LOADBULKDATA
  - gdcm::XMLPrinter, [1424](#)
- LoadDefault
  - gdcm::CSAHeaderDict, [295](#)
  - gdcm::Dict, [363](#)
  - gdcm::PrivateDict, [866](#)
- LoadDefaults
  - gdcm::Defs, [350](#)
- gdcm::Dicts, [379](#)
- LoadFromDataElement
  - gdcm::CSAHeader, [291](#)
  - gdcm::PDBHeader, [789](#)
- LoadFromFile
  - gdcm::Defs, [351](#)
- LoadIconImage
  - vtkGDCMImageReader, [1308](#)
  - vtkGDCMImageReader2, [1322](#)
- LoadImageFromFiles
  - gdcm::DirectoryHelper, [392](#)
- LoadOverlays
  - vtkGDCMImageReader, [1308](#)
  - vtkGDCMImageReader2, [1322](#)
- LoadResourcesFiles
  - gdcm::Global, [510](#)
- LoadSingleFile
  - vtkGDCMImageReader, [1298](#)
  - vtkGDCMImageReader2, [1313](#)
- Locate
  - gdcm::Global, [510](#)
- LOComp
  - gdcm, [59](#)
- LodModeType
  - gdcm, [62](#)
- LookupTable
  - gdcm::LookupTable, [657](#)
  - vtkImageMapToColors16, [1383](#)
- LookupTableType
  - gdcm::LookupTable, [657](#)
- LossyFlag
  - gdcm::Bitmap, [208](#)
  - gdcm::ImageCodec, [557](#)
  - vtkGDCMImageReader, [1308](#)
  - vtkGDCMImageReader2, [1322](#)
- LT
  - gdcm::VR, [1284](#)
- LTComp
  - gdcm, [59](#)
- LUT
  - gdcm::Bitmap, [208](#)
  - gdcm::ImageCodec, [557](#)
- LUTPtr
  - gdcm::Bitmap, [197](#)
  - gdcm::ImageCodec, [548](#)
- m\_char
  - gdcm::ignore\_char, [522](#)
- m\_ConstMemberFunction
  - gdcm::MemberCommand< T >, [688](#)
- m\_DataSet
  - gdcm::DataSetEvent, [343](#)
- m\_MemberFunction
  - gdcm::MemberCommand< T >, [688](#)

- gdcmm::SimpleMemberCommand< T >, 992
- m\_This
  - gdcmm::MemberCommand< T >, 689
  - gdcmm::SimpleMemberCommand< T >, 992
- Macro
  - gdcmm::Macro, 665
- MacroEntry
  - gdcmm, 59
- Macros
  - gdcmm::Macros, 668
- mAction
  - gdcmm::network::Transition, 1139
- MacularGridThicknessandVolumeReportStorage
  - gdcmm::UIDs, 1169
- magenta
  - gdcmm::terminal, 85
- MAGNIFIED
  - gdcmm::Spacing, 1014
- MakeDirectory
  - gdcmm::System, 1089
- MakeNew
  - gdcmm::network::Transition, 1138
- MakeObject
  - gdcmm::AnonymizeEvent, 109
  - gdcmm::DataEvent, 326
  - gdcmm::DataSetEvent, 343
  - gdcmm::Event, 435
  - gdcmm::FileNameEvent, 484
  - gdcmm::ProgressEvent, 873
- MammographyCADSR
  - gdcmm::MediaStorage, 678
- MammographyCADSRStorage
  - gdcmm::UIDs, 1166
- Mandatory
  - gdcmm::Usage, 1257
- MANUAL
  - gdcmm::Segment, 935
- MapCSAHeaderDictEntry
  - gdcmm::CSAHeaderDict, 293
- MapDictEntry
  - gdcmm::Dict, 360
- MapIODEntry
  - gdcmm::IOD, 594
- MapModuleEntry
  - gdcmm::Macro, 665
  - gdcmm::Module, 703
- MappingType
  - gdcmm::Scanner, 925
  - gdcmm::StrictScanner, 1034
- MapScalarsThroughTable2
  - vtkLookupTable16, 1397
- MapTableEntry
  - gdcmm::Table, 1093
- MARCONI
  - gdcmm::EquipmentManufacturer, 431
- MaximumLengthSub
  - gdcmm::network::MaximumLengthSub, 670
- MaxLength
  - gdcmm::ApplicationEntity, 123
  - gdcmm::PersonName, 798
- MaxNumberOfComponents
  - gdcmm::ApplicationEntity, 123
  - gdcmm::PersonName, 798
- MaxPrintLength
  - gdcmm::Printer, 864
- MayoClinicNonradiologicalImagesSBSAnatomicalSurfaceRegionGuide
  - gdcmm::UIDs, 1168
- mConnection
  - gdcmm::network::ULConnectionManager, 1243
- MD5DataImagesType
  - gdcmm::Testing, 1116
- MD5MetaImagesType
  - vtkGDCMTesting, 1347
- mDataSet
  - gdcmm::BaseQuery, 182
- MediaCreationManagementSOPClassUID
  - gdcmm::UIDs, 1164
- MediaStorage
  - gdcmm::MediaStorage, 679
- MediaStorageDataFilesType
  - gdcmm::Testing, 1116
- MediaStorageDirectoryStorage
  - gdcmm::MediaStorage, 676
  - gdcmm::UIDs, 1163
- MedicalImageProperties
  - vtkGDCMImageReader, 1308
  - vtkGDCMPolyDataReader, 1341
  - vtkGDCMPolyDataWriter, 1345
- mElementOffsets
  - gdcmm::StreamImageWriter, 1030
- mElementOffsets1
  - gdcmm::StreamImageWriter, 1030
- MemberCommand
  - gdcmm::MemberCommand< T >, 686
- mEnd
  - gdcmm::network::Transition, 1139
- MeshPrimitive
  - gdcmm::MeshPrimitive, 692
- MessageID
  - gdcmm::network::CEchoRQ, 238
- MetaInformationTS
  - gdcmm::FileMetaInformation, 477
- mHelpDescription
  - gdcmm::BaseRootQuery, 187
- mImage
  - gdcmm::BaseRootQuery, 187
- mImplicit
  - gdcmm::network::ULConnectionCallback, 1234



- ModalityPerformedProcedureStepCreateQuery
  - gdcm::ModalityPerformedProcedureStepCreateQuery, 697
- ModalityPerformedProcedureStepNotificationSOPClass
  - gdcm::UIDs, 1163
- ModalityPerformedProcedureStepRetrieveSOPClass
  - gdcm::UIDs, 1163
- ModalityPerformedProcedureStepSetQuery
  - gdcm::ModalityPerformedProcedureStepSetQuery, 700
- ModalityPerformedProcedureStepSOPClass
  - gdcm::MediaStorage, 678
  - gdcm::UIDs, 1163
- ModalityWorklistInformationModelFIND
  - gdcm::UIDs, 1166
- Mode
  - gdcm::terminal, 86
- Module
  - gdcm::Module, 703
- ModuleEntry
  - gdcm::ModuleEntry, 707
- ModuleMapType
  - gdcm::Macros, 668
  - gdcm::Modules, 710
- Modules
  - gdcm::Modules, 710
- MONOCHROME1
  - gdcm::PhotometricInterpretation, 804
- MONOCHROME2
  - gdcm::PhotometricInterpretation, 804
- MouseGenomeInitiativeMGI
  - gdcm::UIDs, 1168
- MovePatientRootQuery
  - gdcm::MovePatientRootQuery, 713
- MoveStudyRootQuery
  - gdcm::MoveStudyRootQuery, 716
- mPatient
  - gdcm::BaseRootQuery, 187
- MPEG2MainProfile
  - gdcm::TransferSyntax, 1131
- MPEG2MainProfileHighLevel
  - gdcm::TransferSyntax, 1131
  - gdcm::UIDs, 1168
- MPEG2MainProfileMainLevel
  - gdcm::UIDs, 1162
- MPEG4AVCH264BDcompatibleHighProfileLevel4\_1
  - gdcm::TransferSyntax, 1131
- MPEG4AVCH264HighProfileLevel4\_1
  - gdcm::TransferSyntax, 1131
- MPEG4AVCH\_264BDcompatibleHighProfileLevel4\_1
  - gdcm::UIDs, 1168
- MPEG4AVCH\_264HighProfileLevel4\_1
  - gdcm::UIDs, 1168
- MPEG4AVCH\_264HighProfileLevel4\_2For2DVideo
  - gdcm::UIDs, 1169
- MPEG4AVCH\_264HighProfileLevel4\_2For3DVideo
  - gdcm::UIDs, 1169
- MPEG4AVCH\_264StereoHighProfileLevel4\_2
  - gdcm::UIDs, 1169
- MPTType
  - gdcm::MeshPrimitive, 691
- MPTType\_END
  - gdcm::MeshPrimitive, 692
- MRImageStorage
  - gdcm::MediaStorage, 676
  - gdcm::UIDs, 1164
- mRootType
  - gdcm::BaseRootQuery, 187
- MrProtocol
  - gdcm::MrProtocol, 718
- MRSpectroscopyStorage
  - gdcm::MediaStorage, 677
  - gdcm::UIDs, 1164
- MS\_END
  - gdcm::MediaStorage, 678
- mSecondaryConnection
  - gdcm::network::ULConnectionManager, 1243
- mSeries
  - gdcm::BaseRootQuery, 187
- mSopInstanceUID
  - gdcm::BaseQuery, 182
- mSPFile
  - gdcm::StreamImageWriter, 1030
- mStudy
  - gdcm::BaseRootQuery, 187
- MSType
  - gdcm::MediaStorage, 676
- mTransitions
  - gdcm::network::ULConnectionManager, 1243
- MultiframeGrayscaleByteSecondaryCaptureImageStorage
  - gdcm::MediaStorage, 677
  - gdcm::UIDs, 1165
- MultiframeGrayscaleWordSecondaryCaptureImageStorage
  - gdcm::MediaStorage, 677
  - gdcm::UIDs, 1165
- MultiframeSingleBitSecondaryCaptureImageStorage
  - gdcm::MediaStorage, 677
  - gdcm::UIDs, 1165
- MultiframeTrueColorSecondaryCaptureImageStorage
  - gdcm::MediaStorage, 677
  - gdcm::UIDs, 1165
- MultipleVolumeRenderingVolumetricPresentationStateStorage
  - gdcm::UIDs, 1169
- mWriter
  - gdcm::StreamImageWriter, 1030
- mXMax
  - gdcm::StreamImageWriter, 1030
- mXMin



- gdcm::StreamImageWriter, [1031](#)
- mYMax
  - gdcm::StreamImageWriter, [1031](#)
- mYMin
  - gdcm::StreamImageWriter, [1031](#)
- mZMax
  - gdcm::StreamImageWriter, [1031](#)
- mZMin
  - gdcm::StreamImageWriter, [1031](#)
- N\_ACTION\_RQ
  - gdcm::network::DIMSE, [381](#)
- N\_ACTION\_RSP
  - gdcm::network::DIMSE, [381](#)
- N\_CREATE\_RQ
  - gdcm::network::DIMSE, [381](#)
- N\_CREATE\_RSP
  - gdcm::network::DIMSE, [381](#)
- N\_DELETE\_RQ
  - gdcm::network::DIMSE, [381](#)
- N\_DELETE\_RSP
  - gdcm::network::DIMSE, [381](#)
- N\_EVENT\_REPORT\_RQ
  - gdcm::network::DIMSE, [381](#)
- N\_EVENT\_REPORT\_RSP
  - gdcm::network::DIMSE, [381](#)
- N\_GET\_RQ
  - gdcm::network::DIMSE, [381](#)
- N\_GET\_RSP
  - gdcm::network::DIMSE, [381](#)
- N\_SET\_RQ
  - gdcm::network::DIMSE, [381](#)
- N\_SET\_RSP
  - gdcm::network::DIMSE, [381](#)
- NAction
  - gdcm::NormalizedNetworkFunctions, [741](#)
- Name
  - gdcm::ModuleEntry, [709](#)
- NameField
  - gdcm::CSAElement, [285](#)
  - gdcm::PDBelement, [787](#)
- NativeDICOMModel
  - gdcm::UIDs, [1171](#)
- NCreate
  - gdcm::NormalizedNetworkFunctions, [742](#)
- NDelete
  - gdcm::NormalizedNetworkFunctions, [742](#)
- NeedByteSwap
  - gdcm::Bitmap, [208](#)
  - gdcm::ImageCodec, [557](#)
- NeedOverlayCleanup
  - gdcm::ImageCodec, [558](#)
- NegotiatedType
  - gdcm::TransferSyntax, [1130](#)
- NestedMacroEntries
  - gdcm, [59](#)
- NestedModuleEntries
  - gdcm::NestedModuleEntries, [730](#)
- NEventReport
  - gdcm::NormalizedNetworkFunctions, [742](#)
- New
  - gdcm::Anonymizer, [115](#)
  - gdcm::FileChangeTransferSyntax, [456](#)
  - gdcm::FileStreamer, [494](#)
  - gdcm::MemberCommand< T >, [687](#)
  - gdcm::Scanner, [930](#)
  - gdcm::SequenceOfFragments, [958](#)
  - gdcm::SequenceOfItems, [966](#)
  - gdcm::ServiceClassUser, [981](#)
  - gdcm::SimpleMemberCommand< T >, [991](#)
  - gdcm::StrictScanner, [1039](#)
  - vtkGDCMImageReader, [1299](#)
  - vtkGDCMImageReader2, [1313](#)
  - vtkGDCMImageWriter, [1327](#)
  - vtkGDCMMedicalImageProperties, [1335](#)
  - vtkGDCMPolyDataReader, [1338](#)
  - vtkGDCMPolyDataWriter, [1343](#)
  - vtkGDCMTesting, [1349](#)
  - vtkGDCMThreadedImageReader, [1352](#)
  - vtkGDCMThreadedImageReader2, [1356](#)
  - vtkImageColorViewer, [1368](#)
  - vtkImageMapToColors16, [1380](#)
  - vtkImageMapToWindowLevelColors2, [1386](#)
  - vtkImagePlanarComponentsToComponents, [1389](#)
  - vtkImageRGBToYBR, [1392](#)
  - vtkImageYBRToRGB, [1394](#)
  - vtkLookupTable16, [1397](#)
  - vtkRTStructSetProperties, [1405](#)
- NewYorkUniversityMelanomaClinicalCooperativeGroup
  - gdcm::UIDs, [1168](#)
- NGet
  - gdcm::NormalizedNetworkFunctions, [742](#)
- NO
  - gdcm::Surface, [1058](#)
- NO\_COMPRESSION
  - vtkGDCMImageWriter, [1326](#)
- NoElementsError
  - gdcm::Parser, [778](#)
- NoError
  - gdcm::Parser, [778](#)
- NOMAGIC
  - gdcm::CSAHeader, [288](#)
- NoMemoryError
  - gdcm::Parser, [778](#)
- NoObject
  - gdcm::MediaStorage, [679](#)
- NoOfItemsField
  - gdcm::CSAElement, [285](#)

- Normal
  - gdcm::MrProtocol::Slice, [997](#)
- Normalize
  - gdcm::DirectionCosines, [384](#)
- NSet
  - gdcm::NormalizedNetworkFunctions, [743](#)
- NuclearMedicineImageStorage
  - gdcm::MediaStorage, [677](#)
  - gdcm::UIDs, [1165](#)
- NuclearMedicineImageStorageRetired
  - gdcm::MediaStorage, [677](#)
  - gdcm::UIDs, [1165](#)
- Null0
  - gdcm::UIDs, [1169](#)
- Null1
  - gdcm::UIDs, [1169](#)
- NumberOfDimensions
  - gdcm::Bitmap, [208](#)
  - gdcm::ImageCodec, [558](#)
- NumberOfIconImages
  - vtkGDCMImageReader, [1308](#)
  - vtkGDCMImageReader2, [1323](#)
- NumberOfOverlays
  - vtkGDCMImageReader, [1308](#)
  - vtkGDCMImageReader2, [1323](#)
- NumberOfSurfaces
  - gdcm::SurfaceWriter, [1079](#)
- OB
  - gdcm::VR, [1284](#)
- OB\_OW
  - gdcm::VR, [1285](#)
- Object
  - gdcm::Object, [748](#)
- ObjectEnd
  - gdcm::MediaStorage, [679](#)
- ObjectType
  - gdcm::MediaStorage, [679](#)
- OBLIQUE
  - gdcm::Orientation, [761](#)
- OD
  - gdcm::VR, [1284](#)
- OF
  - gdcm::VR, [1284](#)
- Ofstream
  - gdcm::Writer, [1420](#)
- OL
  - gdcm::VR, [1284](#)
- OnlyUUID
  - gdcm::XMLPrinter, [1424](#)
- op
  - gdcm::SerieHelper, [975](#)
- OPENSSL
  - gdcm::CryptoFactory, [273](#)
- OpenSSLCryptoFactory
  - gdcm::OpenSSLCryptoFactory, [751](#)
- OpenSSLCryptographicMessageSyntax
  - gdcm::OpenSSLCryptographicMessageSyntax, [753](#)
- OPENSSLP7
  - gdcm::CryptoFactory, [273](#)
- OpenSSLP7CryptoFactory
  - gdcm::OpenSSLP7CryptoFactory, [756](#)
- OpenSSLP7CryptographicMessageSyntax
  - gdcm::OpenSSLP7CryptographicMessageSyntax, [758](#)
- operator const char \*
  - gdcm::ConstCharWrapper, [269](#)
  - gdcm::Filename, [480](#)
  - gdcm::String< TDelimiter, TMaxLength, TPadChar >, [1045](#)
- operator const double \*
  - gdcm::DirectionCosines, [385](#)
- operator const std::vector< char > &
  - gdcm::ByteValue, [228](#)
- operator MStype
  - gdcm::MediaStorage, [682](#)
- operator ObjectType \*
  - gdcm::SmartPointer< ObjectType >, [1002](#)
- operator PType
  - gdcm::PhotometricInterpretation, [806](#)
- operator ScalarType
  - gdcm::PixelFormat, [812](#)
- operator SwapCode::SwapCodeType
  - gdcm::SwapCode, [1082](#)
- operator TStype
  - gdcm::TransferSyntax, [1134](#)
  - gdcm::UIDs, [1182](#)
- operator TypeType
  - gdcm::Type, [1141](#)
- operator uint32\_t
  - gdcm::VL, [1275](#)
- operator UsageType
  - gdcm::Usage, [1258](#)
- operator VMType
  - gdcm::VM, [1281](#)
- operator VRType
  - gdcm::VR, [1289](#)
- operator!=
  - gdcm, [64](#)
  - gdcm::Attribute< Group, Element, TVR, TVM >, [140](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [148](#)
  - gdcm::CodeString, [256](#)
  - gdcm::PixelFormat, [812](#)
  - gdcm::Tag, [1107](#)
- operator<
  - gdcm::Attribute< Group, Element, TVR, TVM >, [140](#)

- gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 148
- gdcmm::CSAElement, 283
- gdcmm::CSAHeaderDictEntry, 297
- gdcmm::DataElement, 317
- gdcmm::PrivateTag, 869
- gdcmm::Tag, 1107
- operator<<
  - gdcmm, 64–74
  - gdcmm::BasicOffsetTable, 193
  - gdcmm::CodeString, 256
  - gdcmm::CommandDataSet, 262
  - gdcmm::CSAElement, 285
  - gdcmm::CSAHeader, 292
  - gdcmm::CSAHeaderDict, 295
  - gdcmm::CSAHeaderDictEntry, 298
  - gdcmm::DataElement, 321
  - gdcmm::DataSet, 339
  - gdcmm::Dict, 363
  - gdcmm::DictEntry, 372
  - gdcmm::Dicts, 380
  - gdcmm::Directory, 389
  - gdcmm::File, 450
  - gdcmm::FileMetaInformation, 476
  - gdcmm::FileSet, 491
  - gdcmm::Fragment, 507
  - gdcmm::Global, 511
  - gdcmm::GroupDict, 514
  - gdcmm::IOD, 596
  - gdcmm::IODEntry, 599
  - gdcmm::IODs, 602
  - gdcmm::Item, 611
  - gdcmm::Macro, 667
  - gdcmm::Macros, 669
  - gdcmm::MediaStorage, 683
  - gdcmm::Module, 704
  - gdcmm::ModuleEntry, 708
  - gdcmm::Modules, 711
  - gdcmm::MrProtocol, 720
  - gdcmm::NestedModuleEntries, 731
  - gdcmm::Object, 749
  - gdcmm::Orientation, 763
  - gdcmm::PDBelement, 786
  - gdcmm::PDBHeader, 790
  - gdcmm::PhotometricInterpretation, 806
  - gdcmm::PixelFormat, 814
  - gdcmm::Preamble, 840
  - gdcmm::PrivateDict, 867
  - gdcmm::PrivateTag, 870
  - gdcmm::Scanner, 931
  - gdcmm::Sorter, 1011
  - gdcmm::StrictScanner, 1040
  - gdcmm::SwapCode, 1082
  - gdcmm::Table, 1094
  - gdcmm::Tag, 1111
  - gdcmm::TransferSyntax, 1134
  - gdcmm::Type, 1141
  - gdcmm::UI, 1142
  - gdcmm::Usage, 1258
  - gdcmm::Version, 1272
  - gdcmm::VL, 1277
  - gdcmm::VM, 1281
  - gdcmm::VR, 1289
- operator<=
  - gdcmm::Tag, 1107
- operator>>
  - gdcmm, 74, 75
  - gdcmm::Tag, 1111
- operator\*
  - gdcmm::SmartPointer< ObjectType >, 1002
- operator()
  - gdcmm::DataSet, 335
  - gdcmm::Scanner::ltstr, 663
  - gdcmm::StrictScanner::ltstr, 664
- operator++
  - gdcmm::VL, 1275
- operator+=
  - gdcmm::VL, 1275
- operator->
  - gdcmm::SmartPointer< ObjectType >, 1002
- operator=
  - gdcmm::AnonymizeEvent, 110
  - gdcmm::ASN1, 132
  - gdcmm::Base64, 170
  - gdcmm::BoxRegion, 216
  - gdcmm::ByteSwapFilter, 222
  - gdcmm::ByteValue, 228
  - gdcmm::Command, 259
  - gdcmm::CryptographicMessageSyntax, 277
  - gdcmm::CSAElement, 283
  - gdcmm::CSAHeaderDict, 295
  - gdcmm::DataElement, 317
  - gdcmm::DataEvent, 327
  - gdcmm::DataSet, 335
  - gdcmm::DataSetEvent, 343
  - gdcmm::Defs, 351
  - gdcmm::Dict, 363
  - gdcmm::Dicts, 379
  - gdcmm::Element< TVR, VM::VM1\_n >, 405
  - gdcmm::Event, 435
  - gdcmm::FileNameEvent, 484
  - gdcmm::Global, 511
  - gdcmm::MemberCommand< T >, 687
  - gdcmm::network::ULAction, 1185
  - gdcmm::network::ULConnection, 1230
  - gdcmm::network::UserInformation, 1261
  - gdcmm::Object, 749
  - gdcmm::Overlay, 770

- gdcmm::ParseException, 775
- gdcmm::Preamble, 839
- gdcmm::ProgressEvent, 873
- gdcmm::SequenceOfItems, 967
- gdcmm::ServiceClassUser, 982
- gdcmm::SHA1, 987
- gdcmm::SimpleMemberCommand< T >, 992
- gdcmm::SimpleSubjectWatcher, 994
- gdcmm::SmartPointer< ObjectType >, 1002, 1003
- gdcmm::Table, 1094
- gdcmm::Tag, 1107
- operator==
  - gdcmm, 74
  - gdcmm::Attribute< Group, Element, TVR, TVM >, 140
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 148
  - gdcmm::ByteValue, 228, 229
  - gdcmm::CodeString, 256
  - gdcmm::CSAElement, 283
  - gdcmm::DataElement, 317
  - gdcmm::network::AbstractSyntax, 105
  - gdcmm::network::PresentationContextRQ, 854
  - gdcmm::network::TransferSyntaxSub, 1136
  - gdcmm::PDSElement, 786
  - gdcmm::PixelFormat, 812
  - gdcmm::PresentationContext, 843
  - gdcmm::SequenceOfFragments, 958
  - gdcmm::SequenceOfItems, 967
  - gdcmm::Tag, 1107
  - gdcmm::Value, 1268
- operator[]
  - gdcmm::Attribute< Group, Element, TVR, TVM >, 140
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, 157
  - gdcmm::DataSet, 336
  - gdcmm::Element< TVR, TVM >, 399
  - gdcmm::Element< TVR, VM::VM1\_n >, 406
  - gdcmm::Tag, 1108
- OphthalmicAxialMeasurementsStorage
  - gdcmm::UIDs, 1169
- OphthalmicOpticalCoherenceTomographyBscanVolumeAnalysisStorage
  - gdcmm::UIDs, 1169
- OphthalmicOpticalCoherenceTomographyEnFacelImageStorage
  - gdcmm::UIDs, 1169
- OphthalmicPhotography16BitImageStorage
  - gdcmm::MediaStorage, 678
  - gdcmm::UIDs, 1166
- OphthalmicPhotography8BitImageStorage
  - gdcmm::MediaStorage, 678
  - gdcmm::UIDs, 1166
- OphthalmicThicknessMapStorage
  - gdcmm::UIDs, 1169
- OphthalmicTomographyImageStorage
  - gdcmm::MediaStorage, 678
- gdcmm::UIDs, 1166
- OphthalmicVisualFieldStaticPerimetryMeasurementsStorage
  - gdcmm::UIDs, 1169
- OrderFileList
  - gdcmm::SerieHelper, 973
- Orientation
  - gdcmm::Orientation, 761
- OrientationType
  - gdcmm::Orientation, 761
- Output
  - gdcmm::BitmapToBitmapFilter, 212
- OutputFormat
  - vtkImageMapToColors16, 1383
- OutputTypes
  - gdcmm::DictConverter, 365
- OV
  - gdcmm::VR, 1284
- Overlay
  - gdcmm::Overlay, 767
- OverlayImageActor
  - vtkImageColorViewer, 1376
- Overlays
  - gdcmm::Pixmap, 820
- OverlayType
  - gdcmm::Overlay, 766
- OW
  - gdcmm::VR, 1284
- Pack
  - gdcmm::Unpacker12Bits, 1255
- Padding
  - gdcmm::ApplicationEntity, 123
  - gdcmm::PersonName, 799
- PALETTE\_COLOR
  - gdcmm::PhotometricInterpretation, 804
- Papyrus3ImplicitVRLittleEndian
  - gdcmm::UIDs, 1168
- ParametricMapStorage
  - gdcmm::UIDs, 1169
- Parent
  - gdcmm::Element< TVR, VM::VM1\_2 >, 402
  - gdcmm::Element< TVR, VM::VM2\_2n >, 409
  - gdcmm::Element< TVR, VM::VM2\_n >, 411
  - gdcmm::Element< TVR, VM::VM3\_3n >, 413
  - gdcmm::Element< TVR, VM::VM3\_n >, 415
- Parse
  - gdcmm::Parser, 779
- ParseBuffer
  - gdcmm::Parser, 779
- ParseCertificateFile
  - gdcmm::CAPICryptographicMessageSyntax, 236
  - gdcmm::CryptographicMessageSyntax, 277
  - gdcmm::OpenSSLCryptographicMessageSyntax, 754

- gdcm::OpenSSLP7CryptographicMessageSyntax, 759
- ParseDateTime
  - gdcm::System, 1089, 1090
- ParseDump
  - gdcm::ASN1, 132
- ParseDumpFile
  - gdcm::ASN1, 132
- ParseException
  - gdcm::ParseException, 775
- ParseKeyFile
  - gdcm::CAPICryptographicMessageSyntax, 236
  - gdcm::CryptographicMessageSyntax, 277
  - gdcm::OpenSSLCryptographicMessageSyntax, 754
  - gdcm::OpenSSLP7CryptographicMessageSyntax, 759
- Parser
  - gdcm::Parser, 778
- PassAlphaToOutput
  - vtkImageMapToColors16, 1384
- Patient
  - gdcm::Patient, 780
- PatientRadiationDoseSRStorage
  - gdcm::UIDs, 1170
- PatientRootQueryRetrieveInformationModelFIND
  - gdcm::UIDs, 1166
- PatientRootQueryRetrieveInformationModelGET
  - gdcm::UIDs, 1166
- PatientRootQueryRetrieveInformationModelMOVE
  - gdcm::UIDs, 1166
- PatientStudyOnlyQueryRetrieveInformationModelFINDRetired
  - gdcm::UIDs, 1166
- PatientStudyOnlyQueryRetrieveInformationModelGETRetired
  - gdcm::UIDs, 1166
- PatientStudyOnlyQueryRetrieveInformationModelMOVERetired
  - gdcm::UIDs, 1166
- PDataTFPDU
  - gdcm::network::PDataTFPDU, 782
- PDBElement
  - gdcm::PDBElement, 785
- PDBHeader
  - gdcm::PDBHeader, 788
- PDF
  - gdcm::MediaStorage, 679
- PDFCodec
  - gdcm::PDFCodec, 792
- PerformAction
  - gdcm::network::ULAction, 1185
  - gdcm::network::ULActionAA1, 1186
  - gdcm::network::ULActionAA2, 1188
  - gdcm::network::ULActionAA3, 1189
  - gdcm::network::ULActionAA4, 1190
  - gdcm::network::ULActionAA5, 1192
  - gdcm::network::ULActionAA6, 1193
  - gdcm::network::ULActionAA7, 1194
  - gdcm::network::ULActionAA8, 1196
  - gdcm::network::ULActionAE1, 1197
  - gdcm::network::ULActionAE2, 1198
  - gdcm::network::ULActionAE3, 1200
  - gdcm::network::ULActionAE4, 1201
  - gdcm::network::ULActionAE5, 1202
  - gdcm::network::ULActionAE6, 1204
  - gdcm::network::ULActionAE7, 1205
  - gdcm::network::ULActionAE8, 1206
  - gdcm::network::ULActionAR1, 1208
  - gdcm::network::ULActionAR10, 1209
  - gdcm::network::ULActionAR2, 1210
  - gdcm::network::ULActionAR3, 1212
  - gdcm::network::ULActionAR4, 1213
  - gdcm::network::ULActionAR5, 1214
  - gdcm::network::ULActionAR6, 1216
  - gdcm::network::ULActionAR7, 1217
  - gdcm::network::ULActionAR8, 1218
  - gdcm::network::ULActionAR9, 1220
  - gdcm::network::ULActionDT1, 1221
  - gdcm::network::ULActionDT2, 1222
- PerformedImagingAgentAdministrationSRStorage
  - gdcm::UIDs, 1170
- PET20StepColorPaletteSOPInstance
  - gdcm::UIDs, 1168
- PETColorPaletteSOPInstance
  - gdcm::UIDs, 1168
- PETImageStorage
  - gdcm::MediaStorage, 677
- gdcm::Bitmap, 208
- gdcm::ImageCodec, 558
- PGXCodec
  - gdcm::PGXCodec, 800
- PHILIPS
  - gdcm::Dicts, 377
- Philips3D
  - gdcm::MediaStorage, 677
- PhilipsPrivateMRSyntheticImageStorage
  - gdcm::MediaStorage, 678
- PhotometricInterpretation
  - gdcm::PhotometricInterpretation, 804
- PI
  - gdcm::Bitmap, 208
  - gdcm::ImageCodec, 558
- PI\_END
  - gdcm::PhotometricInterpretation, 804
- PIType
  - gdcm::PhotometricInterpretation, 803
- PixelData
  - gdcm::Bitmap, 209
  - gdcm::PixmapReader, 825
  - gdcm::PixmapWriter, 832

- PixelFormat
  - gdcm::PixelFormat, [809](#)
- Pixmap
  - gdcm::Pixmap, [817](#)
- PixmapReader
  - gdcm::Bitmap, [207](#)
  - gdcm::PixmapReader, [823](#)
- PixmapToPixmapFilter
  - gdcm::PixmapToPixmapFilter, [827](#)
- PixmapWriter
  - gdcm::PixmapWriter, [830](#)
- PlanarConfiguration
  - gdcm::Bitmap, [209](#)
  - gdcm::ImageCodec, [558](#)
  - vtkGDCMImageReader, [1308](#)
  - vtkGDCMImageReader2, [1323](#)
- PlannedImagingAgentAdministrationSRStorage
  - gdcm::UIDs, [1170](#)
- PMS
  - gdcm::EquipmentManufacturer, [431](#)
- PN
  - gdcm::VR, [1284](#)
- PNComp
  - gdcm, [59](#)
- PNMCodec
  - gdcm::PNMCodec, [834](#)
- pointer
  - gdcm::CodeString, [254](#)
  - gdcm::LO, [653](#)
  - gdcm::String< TDelimiter, TMaxLength, TPadChar >, [1043](#)
- POINTS
  - gdcm::Surface, [1058](#)
- Position
  - gdcm::MrProtocol::Slice, [997](#)
- PositronEmissionTomographyImageStorage
  - gdcm::UIDs, [1166](#)
- Preamble
  - gdcm::Preamble, [837](#), [838](#)
- PrepareWrite
  - gdcm::PixmapWriter, [831](#)
  - gdcm::SegmentWriter, [951](#)
  - gdcm::SurfaceWriter, [1079](#)
- PrepareWritePointMacro
  - gdcm::SurfaceWriter, [1079](#)
- Prepend
  - gdcm::Global, [511](#)
- PresentationContext
  - gdcm::PresentationContext, [842](#)
- PresentationContextAC
  - gdcm::network::PresentationContextAC, [845](#)
- PresentationContextArrayType
  - gdcm::network::AAssociateRQPDU, [98](#)
  - gdcm::PresentationContextGenerator, [849](#)
- PresentationContextGenerator
  - gdcm::PresentationContextGenerator, [849](#)
- PresentationContextRQ
  - gdcm::network::PresentationContextRQ, [852](#)
- PresentationDataValue
  - gdcm::network::PresentationDataValue, [856](#)
- PresentationLUTSOPClass
  - gdcm::UIDs, [1164](#)
- PrettyPrintOff
  - gdcm::JSON, [646](#)
- PrettyPrintOn
  - gdcm::JSON, [647](#)
- PrimitiveData
  - gdcm::MeshPrimitive, [694](#)
- PrimitivesData
  - gdcm::MeshPrimitive, [691](#)
- PrimitiveType
  - gdcm::MeshPrimitive, [695](#)
- Print
  - gdcm::ApplicationEntity, [122](#)
  - gdcm::Attribute< Group, Element, TVR, TVM >, [141](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [149](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [157](#)
  - gdcm::BaseQuery, [180](#)
  - gdcm::Bitmap, [202](#)
  - gdcm::BoxRegion, [217](#)
  - gdcm::ByteValue, [229](#)
  - gdcm::CSAHeader, [291](#)
  - gdcm::Curve, [307](#)
  - gdcm::DataSet, [336](#)
  - gdcm::DictPrinter, [375](#)
  - gdcm::DirectionCosines, [385](#)
  - gdcm::Directory, [389](#)
  - gdcm::Element< TVR, TVM >, [399](#)
  - gdcm::Element< TVR, VM::VM1\_n >, [406](#)
  - gdcm::Element< VR::AS, VM::VM5 >, [416](#)
  - gdcm::Event, [435](#)
  - gdcm::Image, [526](#)
  - gdcm::LookupTable, [661](#)
  - gdcm::MrProtocol, [719](#)
  - gdcm::network::AAbortPDU, [88](#)
  - gdcm::network::AAssociateACPDU, [93](#)
  - gdcm::network::AAssociateRJPDU, [96](#)
  - gdcm::network::AAssociateRQPDU, [101](#)
  - gdcm::network::AbstractSyntax, [105](#)
  - gdcm::network::ApplicationContext, [120](#)
  - gdcm::network::AReleaseRPPDU, [125](#)
  - gdcm::network::AReleaseRQPDU, [128](#)
  - gdcm::network::AsynchronousOperationsWindowSub, [133](#)
  - gdcm::network::BasePDU, [176](#)
  - gdcm::network::ImplementationClassUIDSub, [586](#)



- gdcm::network::ImplementationVersionNameSub, 589
- gdcm::network::MaximumLengthSub, 670
- gdcm::network::PDataTFPDU, 783
- gdcm::network::PresentationContextAC, 846
- gdcm::network::PresentationContextRQ, 854
- gdcm::network::PresentationDataValue, 857
- gdcm::network::RoleSelectionSub, 921
- gdcm::network::ServiceClassApplicationInformation, 977
- gdcm::network::SOPClassExtendedNegociationSub, 1004
- gdcm::network::TransferSyntaxSub, 1136
- gdcm::network::UserInformation, 1261
- gdcm::Object, 749
- gdcm::Orientation, 762
- gdcm::Overlay, 771
- gdcm::PDBHeader, 790
- gdcm::PersonName, 797
- gdcm::PixelFormat, 813
- gdcm::Pixmap, 819
- gdcm::Preamble, 839
- gdcm::PresentationContext, 844
- gdcm::Printer, 862
- gdcm::Region, 909
- gdcm::Scanner, 930
- gdcm::SegmentedPaletteColorLookupTable, 944
- gdcm::SequenceOfFragments, 958
- gdcm::SequenceOfItems, 967
- gdcm::Sorter, 1010
- gdcm::StrictScanner, 1039
- gdcm::TagPath, 1114
- gdcm::Testing, 1122
- gdcm::Version, 1272
- gdcm::XMLPrinter, 1425
- PrintASCII
  - gdcm::ByteValue, 229
- PrintASCIIXML
  - gdcm::ByteValue, 229
- PrintAsContinuousString
  - gdcm::Tag, 1108
- PrintAsContinuousUpperCaseString
  - gdcm::Tag, 1108
- PrintAsPipeSeparatedString
  - gdcm::Tag, 1108
- PrintDataElement
  - gdcm::Printer, 862
  - gdcm::XMLPrinter, 1425
- PrintDataElement2
  - gdcm::DictPrinter, 375
- PrintDataSet
  - gdcm::Printer, 862
  - gdcm::XMLPrinter, 1426
- PrintDataSet2
  - gdcm::DictPrinter, 375
- Printer
  - gdcm::Printer, 861
- PrinterConfigurationRetrievalSOPClass
  - gdcm::UIDs, 1164
- PrinterConfigurationRetrievalSOPInstance
  - gdcm::UIDs, 1164
- PrinterSOPClass
  - gdcm::UIDs, 1164
- PrinterSOPInstance
  - gdcm::UIDs, 1164
- PrintGroupLength
  - gdcm::ByteValue, 229
- PrintHex
  - gdcm::ByteValue, 229
- PrintHexXML
  - gdcm::ByteValue, 230
- PrintJobSOPClass
  - gdcm::UIDs, 1164
- PrintPNXML
  - gdcm::ByteValue, 230
- PrintQueueManagementSOPClassRetired
  - gdcm::UIDs, 1164
- PrintQueueSOPInstanceRetired
  - gdcm::UIDs, 1164
- PrintSelf
  - vtkGDCMImageReader, 1299
  - vtkGDCMImageReader2, 1314
  - vtkGDCMImageWriter, 1327
  - vtkGDCMMedicalImageProperties, 1335
  - vtkGDCMPolyDataReader, 1338
  - vtkGDCMPolyDataWriter, 1344
  - vtkGDCMTesting, 1349
  - vtkGDCMThreadedImageReader, 1352
  - vtkGDCMThreadedImageReader2, 1356
  - vtkImageColorViewer, 1369
  - vtkImageMapToColors16, 1380
  - vtkImageMapToWindowLevelColors2, 1386
  - vtkImagePlanarComponentsToComponents, 1389
  - vtkImageRGBToYBR, 1392
  - vtkImageYBRToRGB, 1394
  - vtkLookupTable16, 1398
  - vtkRTStructSetProperties, 1405
- PrintSQ
  - gdcm::Printer, 863
  - gdcm::XMLPrinter, 1426
- PrintStyle
  - gdcm::Printer, 864
  - gdcm::XMLPrinter, 1427
- PrintStyles
  - gdcm::Printer, 861
  - gdcm::XMLPrinter, 1424
- PrintTable
  - gdcm::network::ULTransitionTable, 1247

- gdcmm::Scanner, 931
- gdcmm::StrictScanner, 1039
- PrintXML
  - gdcmm::PrivateDict, 866
- PrivateDict
  - gdcmm::PrivateDict, 865
- PrivateTag
  - gdcmm::PrivateTag, 868, 869
- ProceduralEventLoggingSOPClass
  - gdcmm::UIDs, 1163
- ProceduralEventLoggingSOPInstance
  - gdcmm::UIDs, 1163
- ProcedureLogStorage
  - gdcmm::UIDs, 1166
- Process
  - gdcmm::Parser, 779
- ProcessDataSet
  - gdcmm::FileExplicitFilter, 466
- ProcessPublicTag
  - gdcmm::Scanner, 931
  - gdcmm::StrictScanner, 1039
- ProcessRequest
  - vtkGDCMImageReader2, 1314
- ProduceCharacterSetDataElement
  - gdcmm::QueryFactory, 883
- ProduceQuery
  - gdcmm::QueryFactory, 883
- ProductCharacteristicsQuerySOPClass
  - gdcmm::UIDs, 1167
- ProgressEvent
  - gdcmm::ProgressEvent, 872
- PropertyCategory
  - gdcmm::Segment, 941
- PropertyType
  - gdcmm::Segment, 941
- PropertyTypeModifiers
  - gdcmm::Segment, 941
- ProtocolApprovalInformationModelFIND
  - gdcmm::UIDs, 1170
- ProtocolApprovalInformationModelGET
  - gdcmm::UIDs, 1170
- ProtocolApprovalInformationModelMOVE
  - gdcmm::UIDs, 1170
- ProtocolApprovalStorage
  - gdcmm::UIDs, 1170
- PseudoColorSoftcopyPresentationStateStorageSOPClass
  - gdcmm::UIDs, 1165
- PubChemCompoundCID
  - gdcmm::UIDs, 1168
- PullPrintRequestSOPClassRetired
  - gdcmm::UIDs, 1164
- PullStoredPrintManagementMetaSOPClassRetired
  - gdcmm::UIDs, 1164
- Push
  - gdcmm::TagPath, 1114
- PushBackFile
  - vtkGDCMMedicalImageProperties, 1335
- PVRGCodec
  - gdcmm::PVRGCodec, 875
- PythonFilter
  - gdcmm::PythonFilter, 878
- Quality
  - gdcmm::JPEGCodec, 638
- QueryFactory
  - gdcmm::BaseQuery, 182
  - gdcmm::BaseRootQuery, 187
  - gdcmm::FindPatientRootQuery, 501
  - gdcmm::FindStudyRootQuery, 504
  - gdcmm::ModalityPerformedProcedureStepCreateQuery, 697
  - gdcmm::ModalityPerformedProcedureStepSetQuery, 700
  - gdcmm::MovePatientRootQuery, 714
  - gdcmm::MoveStudyRootQuery, 717
  - gdcmm::WLMFindQuery, 1414
- RadiomicsOntology
  - gdcmm::UIDs, 1168
- RadiopharmaceuticalRadiationDoseSRStorage
  - gdcmm::UIDs, 1170
- RAWCodec
  - gdcmm::RAWCodec, 895
- RawDataStorage
  - gdcmm::MediaStorage, 677
  - gdcmm::UIDs, 1165
- Read
  - gdcmm::BasicOffsetTable, 193
  - gdcmm::ByteValue, 230
  - gdcmm::CommandDataSet, 262
  - gdcmm::CP246ExplicitDataElement, 271
  - gdcmm::DataElement, 317
  - gdcmm::DataSet, 336
  - gdcmm::Element< TVR, TVM >, 399
  - gdcmm::Element< TVR, VM::VM1\_n >, 406
  - gdcmm::EncodingImplementation< VR::VRASCII >, 426
  - gdcmm::EncodingImplementation< VR::VRBINARY >, 428
  - gdcmm::ExplicitDataElement, 441
  - gdcmm::ExplicitImplicitDataElement, 444
  - gdcmm::File, 449
  - gdcmm::FileMetaInformation, 474
  - gdcmm::Fragment, 506
  - gdcmm::ImageReader, 574
  - gdcmm::ImageRegionReader, 578
  - gdcmm::ImplicitDataElement, 591
  - gdcmm::Item, 611
  - gdcmm::network::AAbortPDU, 88



- gdcm::network::AAssociateACPDU, 93
- gdcm::network::AAssociateRJPDU, 96
- gdcm::network::AAssociateRQPDU, 101
- gdcm::network::AbstractSyntax, 105
- gdcm::network::ApplicationContext, 120
- gdcm::network::AReleaseRPPDU, 125
- gdcm::network::AReleaseRQPDU, 128
- gdcm::network::AsynchronousOperationsWindowSub, 133
- gdcm::network::BasePDU, 176
- gdcm::network::ImplementationClassUIDSub, 586
- gdcm::network::ImplementationVersionNameSub, 589
- gdcm::network::MaximumLengthSub, 670
- gdcm::network::PDataTFPDU, 783
- gdcm::network::PresentationContextAC, 846
- gdcm::network::PresentationContextRQ, 854
- gdcm::network::PresentationDataValue, 857
- gdcm::network::RoleSelectionSub, 922
- gdcm::network::ServiceClassApplicationInformation, 977
- gdcm::network::SOPClassExtendedNegociationSub, 1004
- gdcm::network::TransferSyntaxSub, 1136
- gdcm::network::UserInformation, 1262
- gdcm::PGXCodec, 801
- gdcm::PixmapReader, 824
- gdcm::PNMCodec, 835
- gdcm::Preamble, 839
- gdcm::Reader, 902
- gdcm::SegmentReader, 947
- gdcm::SequenceOfFragments, 959
- gdcm::SequenceOfItems, 967
- gdcm::StreamImageReader, 1023
- gdcm::SurfaceReader, 1076
- gdcm::TableReader, 1099
- gdcm::Tag, 1109
- gdcm::UNExplicitDataElement, 1252
- gdcm::UNExplicitImplicitDataElement, 1254
- gdcm::ValueIO< TDE, TSwap, TType >, 1269
- gdcm::VL, 1276
- gdcm::VR, 1289
- gdcm::VR16ExplicitDataElement, 1291
- gdcm::VRVLSIZE< 0 >, 1293
- gdcm::VRVLSIZE< 1 >, 1294
- Read16
  - gdcm::VL, 1276
- ReadACRNEMAIImage
  - gdcm::ImageReader, 574
  - gdcm::PixmapReader, 824
- ReadBacktrack
  - gdcm::Fragment, 506
- ReadCompat
  - gdcm::FileMetaInformation, 474
- ReadCompatInternal
  - gdcm::FileMetaInformation, 475
- ReadComputeLength
  - gdcm::EncodingImplementation< VR::VRASCII >, 426
  - gdcm::EncodingImplementation< VR::VRBINARY >, 428
- ReadDataSet
  - gdcm::Reader, 902
- Reader
  - gdcm::Reader, 901
- ReadFiles
  - vtkGDCMThreadedImageReader, 1352
- ReadFromCommaSeparatedString
  - gdcm::PrivateTag, 869
  - gdcm::Tag, 1109
- ReadFromContinuousString
  - gdcm::Tag, 1109
- ReadFromPipeSeparatedString
  - gdcm::Tag, 1109
- ReadImage
  - gdcm::ImageReader, 574
  - gdcm::PixmapReader, 824
- ReadImageInformation
  - gdcm::StreamImageReader, 1023
- ReadImageInternal
  - gdcm::PixmapReader, 824
- ReadInformation
  - gdcm::ImageRegionReader, 578
- ReadInto
  - gdcm::network::PDataTFPDU, 783
  - gdcm::network::PresentationDataValue, 857
- ReadIntoBuffer
  - gdcm::ImageRegionReader, 578
- README.txt, 1709
- ReadMetaInformation
  - gdcm::Reader, 903
- ReadNested
  - gdcm::DataSet, 336
- ReadNoSwap
  - gdcm::EncodingImplementation< VR::VRASCII >, 426
  - gdcm::EncodingImplementation< VR::VRBINARY >, 428
- ReadOrSkip
  - gdcm::DataElement, 317
- ReadPointMacro
  - gdcm::SurfaceReader, 1076
- ReadPreamble
  - gdcm::Reader, 903
- ReadPreValue
  - gdcm::CP246ExplicitDataElement, 271
  - gdcm::DataElement, 318
  - gdcm::ExplicitDataElement, 441

- gdcmm::ExplicitImplicitDataElement, 444
- gdcmm::Fragment, 506
- gdcmm::ImplicitDataElement, 591
- gdcmm::SequenceOfFragments, 959
- gdcmm::UNExplicitDataElement, 1252
- gdcmm::UNExplicitImplicitDataElement, 1254
- gdcmm::VR16ExplicitDataElement, 1291
- ReadSegment
  - gdcmm::SegmentReader, 948
- ReadSegments
  - gdcmm::SegmentReader, 948
- ReadSelectedPrivateTags
  - gdcmm::DataSet, 336
  - gdcmm::Reader, 903
- ReadSelectedPrivateTagsWithLength
  - gdcmm::DataSet, 337
- ReadSelectedTags
  - gdcmm::DataSet, 337
  - gdcmm::Reader, 903
- ReadSelectedTagsWithLength
  - gdcmm::DataSet, 337
- ReadSurface
  - gdcmm::SurfaceReader, 1076
- ReadSurfaces
  - gdcmm::SurfaceReader, 1076
- Readuint16
  - gdcmm::DictConverter, 366
- ReadUpToTag
  - gdcmm::DataSet, 337
  - gdcmm::Reader, 903
- ReadUpToTagWithLength
  - gdcmm::DataSet, 337
- ReadValue
  - gdcmm::CP246ExplicitDataElement, 271
  - gdcmm::DataElement, 318
  - gdcmm::ExplicitDataElement, 441
  - gdcmm::ExplicitImplicitDataElement, 444
  - gdcmm::Fragment, 507
  - gdcmm::ImplicitDataElement, 591
  - gdcmm::SequenceOfFragments, 959
  - gdcmm::UNExplicitDataElement, 1252
  - gdcmm::UNExplicitImplicitDataElement, 1254
  - gdcmm::VR16ExplicitDataElement, 1291
- ReadValueWithLength
  - gdcmm::DataElement, 318
  - gdcmm::ImplicitDataElement, 592
- ReadVM
  - gdcmm::DictConverter, 367
- ReadVR
  - gdcmm::DictConverter, 367
- ReadWithLength
  - gdcmm::CP246ExplicitDataElement, 272
  - gdcmm::DataElement, 318
  - gdcmm::DataSet, 338
- gdcmm::ExplicitDataElement, 441
- gdcmm::ExplicitImplicitDataElement, 444
- gdcmm::ImplicitDataElement, 592
- gdcmm::UNExplicitDataElement, 1252
- gdcmm::VR16ExplicitDataElement, 1292
- RealWorldValueIntercept
  - gdcmm::RealWorldValueMappingContent, 907
- RealWorldValueMappingStorage
  - gdcmm::UIDs, 1165
- RealWorldValueSlope
  - gdcmm::RealWorldValueMappingContent, 907
- RecommendedDisplayCIELabToRGB
  - gdcmm::SurfaceHelper, 1071
- RecurseDataSet
  - gdcmm::Anonymizer, 115
- RED
  - gdcmm::LookupTable, 657
- red
  - gdcmm::terminal, 85
- reference
  - gdcmm::CodeString, 254
  - gdcmm::LO, 653
  - gdcmm::String< TDelimiter, TMaxLength, TPadChar >, 1044
- ReferencedColorPrintManagementMetaSOPClassRetired
  - gdcmm::UIDs, 1164
- ReferencedGrayscalePrintManagementMetaSOPClassRetired
  - gdcmm::UIDs, 1164
- ReferencedImageBoxSOPClassRetired
  - gdcmm::UIDs, 1164
- ReferenceFrameOfReferenceUID
  - vtkRTStructSetProperties, 1409
- ReferenceSeriesInstanceUID
  - vtkRTStructSetProperties, 1409
- Region
  - gdcmm::Region, 908
- Register
  - gdcmm::Object, 749
- Remove
  - gdcmm::Anonymizer, 115
  - gdcmm::DataSet, 338
  - gdcmm::FileAnonymizer, 452
  - gdcmm::Preamble, 839
- RemoveAllObservers
  - gdcmm::Subject, 1054
- RemoveDictEntry
  - gdcmm::PrivateDict, 866
- RemoveFile
  - gdcmm::System, 1090
- RemoveGroupLength
  - gdcmm::Anonymizer, 115
- RemoveItemByIndex
  - gdcmm::SequenceOfItems, 967
- RemoveObserver

- gdcm::Subject, [1054](#)
- RemoveOverlay
  - gdcm::Pixmap, [819](#)
- RemovePrivateTags
  - gdcm::Anonymizer, [116](#)
- RemoveRetired
  - gdcm::Anonymizer, [116](#)
- Render
  - vtkImageColorViewer, [1369](#)
- Renderer
  - vtkImageColorViewer, [1376](#)
- RenderWindow
  - vtkImageColorViewer, [1377](#)
- Replace
  - gdcm::Anonymizer, [116](#)
  - gdcm::CommandDataSet, [262](#)
  - gdcm::DataSet, [338](#)
  - gdcm::FileAnonymizer, [453](#)
  - gdcm::FileMetaInformation, [475](#)
- ReplaceEmpty
  - gdcm::DataSet, [338](#)
- RequestData
  - vtkGDCMImageReader2, [1314](#)
  - vtkGDCMPolyDataReader, [1339](#)
  - vtkImageMapToColors16, [1380](#)
  - vtkImageMapToWindowLevelColors2, [1386](#)
  - vtkImagePlanarComponentsToComponents, [1390](#)
- RequestData\_HemodynamicWaveformStorage
  - vtkGDCMPolyDataReader, [1339](#)
- RequestData\_RTStructureSetStorage
  - vtkGDCMPolyDataReader, [1339](#)
- RequestDataCompat
  - vtkGDCMImageReader, [1299](#)
  - vtkGDCMImageReader2, [1314](#)
  - vtkGDCMThreadedImageReader, [1353](#)
- RequestInformation
  - vtkGDCMImageReader2, [1314](#)
  - vtkGDCMPolyDataReader, [1339](#)
  - vtkGDCMThreadedImageReader2, [1357](#)
  - vtkImageMapToColors16, [1380](#)
  - vtkImageMapToWindowLevelColors2, [1386](#)
- RequestInformation\_HemodynamicWaveformStorage
  - vtkGDCMPolyDataReader, [1339](#)
- RequestInformation\_RTStructureSetStorage
  - vtkGDCMPolyDataReader, [1339](#)
- RequestInformationCompat
  - vtkGDCMImageReader, [1299](#)
  - vtkGDCMImageReader2, [1315](#)
- RequestPaddedCompositePixelCode
  - gdcm::ImageCodec, [558](#)
- RequestPlanarConfiguration
  - gdcm::ImageCodec, [558](#)
- Rescale
  - gdcm::Rescaler, [913](#)
- RescaleFunctionIntoBestFit
  - gdcm::Rescaler, [913](#)
- Rescaler
  - gdcm::Rescaler, [911](#)
- ReserveDataElement
  - gdcm::FileStreamer, [494](#)
- ReserveGroupDataElement
  - gdcm::FileStreamer, [494](#)
- reset
  - gdcm::terminal, [85](#)
- ResetHandledDataSet
  - gdcm::network::ULConnectionCallback, [1233](#)
- RespiratoryWaveformStorage
  - gdcm::UIDs, [1169](#)
- RetrieveSOPInstanceUIDFromIndex
  - gdcm::DirectoryHelper, [392](#)
- RetrieveSOPInstanceUIDFromZPosition
  - gdcm::DirectoryHelper, [392](#)
- reverse
  - gdcm::terminal, [85](#)
- reverse\_iterator
  - gdcm::CodeString, [254](#)
  - gdcm::LO, [653](#)
  - gdcm::String< TDelimiter, TMaxLength, TPadChar >, [1044](#)
- RFC2557MIMEencapsulation
  - gdcm::UIDs, [1162](#)
- RGB
  - gdcm::PhotometricInterpretation, [804](#)
- RGB2YBR
  - gdcm::ImageChangePhotometricInterpretation, [535](#)
- RGBPixelsToRGBPlanes
  - gdcm::ImageChangePlanarConfiguration, [539](#)
- RGBPlanesToRGBPixels
  - gdcm::ImageChangePlanarConfiguration, [539](#)
- RGBToRecommendedDisplayCIELab
  - gdcm::SurfaceHelper, [1072](#)
- RGBToRecommendedDisplayGrayscale
  - gdcm::SurfaceHelper, [1072](#)
- RLE\_COMPRESSION
  - vtkGDCMImageWriter, [1326](#)
- RLECodec
  - gdcm::RLECodec, [916](#)
- RLELossless
  - gdcm::TransferSyntax, [1131](#)
  - gdcm::UIDs, [1162](#)
- ROI
  - gdcm::Overlay, [766](#)
- RoleSelectionSub
  - gdcm::network::RoleSelectionSub, [921](#)
- Round
  - gdcm, [75](#)
- roundat
  - gdcm, [75](#)

- RTBeamsDeliveryInstructionStorage
  - gdcm::UIDs, [1171](#)
- RTBeamsDeliveryInstructionStorageSupplement74FrozenDraft
  - gdcm::UIDs, [1167](#)
- RTBeamsTreatmentRecordStorage
  - gdcm::UIDs, [1166](#)
- RTBrachyApplicationSetupDeliveryInstructionStorage
  - gdcm::UIDs, [1171](#)
- RTBrachyTreatmentRecordStorage
  - gdcm::UIDs, [1166](#)
- RTConventionalMachineVerification
  - gdcm::UIDs, [1171](#)
- RTConventionalMachineVerificationSupplement74FrozenDraft
  - gdcm::UIDs, [1167](#)
- RTDoseStorage
  - gdcm::MediaStorage, [677](#)
  - gdcm::UIDs, [1166](#)
- RTImageStorage
  - gdcm::MediaStorage, [677](#)
  - gdcm::UIDs, [1166](#)
- RTIonBeamsTreatmentRecordStorage
  - gdcm::MediaStorage, [678](#)
  - gdcm::UIDs, [1166](#)
- RTIonMachineVerification
  - gdcm::UIDs, [1171](#)
- RTIonMachineVerificationSupplement74FrozenDraft
  - gdcm::UIDs, [1167](#)
- RTIonPlanStorage
  - gdcm::MediaStorage, [678](#)
  - gdcm::UIDs, [1166](#)
- RTPhysicianIntentStorage
  - gdcm::UIDs, [1170](#)
- RTPlanStorage
  - gdcm::MediaStorage, [677](#)
  - gdcm::UIDs, [1166](#)
- RTSegmentAnnotationStorage
  - gdcm::UIDs, [1170](#)
- RTStructSetProperties
  - vtkGDCMPolyDataReader, [1341](#)
  - vtkGDCMPolyDataWriter, [1346](#)
- RTStructureSetStorage
  - gdcm::MediaStorage, [677](#)
  - gdcm::UIDs, [1166](#)
- RTTreatmentSummaryRecordStorage
  - gdcm::MediaStorage, [678](#)
  - gdcm::UIDs, [1166](#)
- Rule
  - gdcm::SerieHelper, [971](#)
- RunEventLoop
  - gdcm::network::ULConnectionManager, [1239](#)
- RunMoveEventLoop
  - gdcm::network::ULConnectionManager, [1240](#)
- SAGITTAL
  - gdcm::Orientation, [761](#)
- ScalarType
  - gdcm::PixelFormat, [808](#)
- Scale
  - vtkGDCMImageReader, [1309](#)
  - vtkGDCMImageReader2, [1323](#)
- Scan
  - gdcm::Scanner, [931](#)
  - gdcm::StrictScanner, [1040](#)
- Scanner
  - gdcm::Scanner, [926](#)
- SecondaryCaptureImageStorage
  - gdcm::MediaStorage, [677](#)
  - gdcm::UIDs, [1165](#)
- Segment
  - gdcm::Segment, [935](#)
- SegmentAlgorithmName
  - gdcm::Segment, [941](#)
- SegmentAlgorithmType
  - gdcm::Segment, [941](#)
- Segmentation
  - gdcm::MediaStorage, [679](#)
- SegmentationStorage
  - gdcm::MediaStorage, [678](#)
  - gdcm::UIDs, [1165](#)
- SegmentDescription
  - gdcm::Segment, [941](#)
- SegmentedPaletteColorLookupTable
  - gdcm::SegmentedPaletteColorLookupTable, [943](#)
- SegmentedVolumeRenderingVolumetricPresentationStateStorage
  - gdcm::UIDs, [1169](#)
- SegmentLabel
  - gdcm::Segment, [941](#)
- SegmentMap
  - gdcm::SegmentReader, [946](#)
- SegmentNumber
  - gdcm::Segment, [942](#)
- SegmentReader
  - gdcm::SegmentReader, [947](#)
- Segments
  - gdcm::SegmentReader, [948](#)
  - gdcm::SegmentWriter, [952](#)
- SegmentVector
  - gdcm::SegmentReader, [947](#)
  - gdcm::SegmentWriter, [950](#)
- SegmentWriter
  - gdcm::SegmentWriter, [950](#)
- Selection
  - gdcm::Sorter, [1011](#)
- SelectionMap
  - gdcm::Sorter, [1009](#)
- Self
  - gdcm::AnonymizeEvent, [108](#)
  - gdcm::DataEvent, [325](#)

- gdcmm::DataSetEvent, [342](#)
- gdcmm::FileNameEvent, [483](#)
- gdcmm::MemberCommand< T >, [685](#)
- gdcmm::ProgressEvent, [872](#)
- gdcmm::SimpleMemberCommand< T >, [990](#)
- SEMIAUTOMATIC
  - gdcmm::Segment, [935](#)
- SendEcho
  - gdcmm::network::ULConnectionManager, [1240](#)
  - gdcmm::ServiceClassUser, [982](#)
- SendFind
  - gdcmm::network::ULConnectionManager, [1240](#)
  - gdcmm::ServiceClassUser, [982](#)
- SendMove
  - gdcmm::network::ULConnectionManager, [1240](#)
  - gdcmm::ServiceClassUser, [982](#), [983](#)
- SendNAction
  - gdcmm::network::ULConnectionManager, [1241](#)
- SendNCreate
  - gdcmm::network::ULConnectionManager, [1241](#)
- SendNDelete
  - gdcmm::network::ULConnectionManager, [1241](#)
- SendNEventReport
  - gdcmm::network::ULConnectionManager, [1242](#)
- SendNGet
  - gdcmm::network::ULConnectionManager, [1242](#)
- SendNSet
  - gdcmm::network::ULConnectionManager, [1242](#)
- SendStore
  - gdcmm::network::ULConnectionManager, [1243](#)
  - gdcmm::ServiceClassUser, [983](#)
- Separator
  - gdcmm::ApplicationEntity, [123](#)
  - gdcmm::PersonName, [799](#)
- SequenceLengthField
  - gdcmm::SequenceOfItems, [969](#)
- SequenceOfFragments
  - gdcmm::SequenceOfFragments, [955](#)
- SequenceOfItems
  - gdcmm::SequenceOfItems, [963](#)
- SerieHelper
  - gdcmm::SerieHelper, [971](#)
- SerieRestrictions
  - gdcmm::SerieHelper, [971](#)
- Series
  - gdcmm::Series, [976](#)
- SeriesInstanceUID
  - vtkRTStructSetProperties, [1409](#)
- ServiceClassApplicationInformation
  - gdcmm::network::ServiceClassApplicationInformation, [976](#)
- ServiceClassUser
  - gdcmm::ServiceClassUser, [980](#)
- Set
  - gdcmm::Attribute< Group, Element, TVR, TVM >, [141](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [149](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, [158](#)
  - gdcmm::Element< TVR, TVM >, [399](#)
  - gdcmm::Element< TVR, VM::VM1\_n >, [406](#)
  - SetAbstractSyntax
    - gdcmm::network::PresentationContextRQ, [854](#)
    - gdcmm::PresentationContext, [844](#)
  - SetAETitle
    - gdcmm::ServiceClassUser, [983](#)
  - SetAlgorithmFamily
    - gdcmm::Surface, [1065](#)
  - SetAlgorithmName
    - gdcmm::Surface, [1065](#)
  - SetAlgorithmVersion
    - gdcmm::Surface, [1065](#)
  - SetAnatomicRegion
    - gdcmm::Segment, [938](#)
  - SetAnatomicRegionModifiers
    - gdcmm::Segment, [938](#)
  - SetAppendDerivationHistory
    - gdcmm::FileDerivation, [463](#)
  - SetArray
    - gdcmm::Element< TVR, VM::VM1\_n >, [406](#)
  - setattribute
    - gdcmm::terminal, [86](#)
  - SetAxisOfRotation
    - gdcmm::Surface, [1065](#)
  - setbgcolor
    - gdcmm::terminal, [86](#)
  - SetBitPosition
    - gdcmm::Overlay, [771](#)
  - SetBitsAllocated
    - gdcmm::Overlay, [771](#)
    - gdcmm::PixelFormat, [813](#)
  - SetBitSample
    - gdcmm::JPEGCodec, [636](#)
  - SetBitsStored
    - gdcmm::PixelFormat, [813](#)
  - SetBlob
    - gdcmm::ApplicationEntity, [122](#)
    - gdcmm::network::PresentationDataValue, [858](#)
    - gdcmm::PersonName, [798](#)
  - SetBlueLUT
    - gdcmm::LookupTable, [661](#)
  - SetBufferLength
    - gdcmm::JPEGLSCCodec, [644](#)
    - gdcmm::PNMCodec, [836](#)
    - gdcmm::RLECodec, [920](#)
  - SetByteSwapTag
    - gdcmm::ByteSwapFilter, [222](#)
  - SetByteValue

- gdcmm::Attribute< Group, Element, TVR, TVM >, 141
- gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 149
- gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, 158
- gdcmm::CSAElement, 283
- gdcmm::DataElement, 318
- SetByteValueNoSwap
  - gdcmm::Attribute< Group, Element, TVR, TVM >, 141
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 149
- SetCallbackFunction
  - gdcmm::MemberCommand< T >, 688
  - gdcmm::SimpleMemberCommand< T >, 992
- SetCalledAETitle
  - gdcmm::network::AAssociateACPDU, 93
  - gdcmm::network::AAssociateRQPDU, 102
  - gdcmm::ServiceClassUser, 984
- SetCallingAETitle
  - gdcmm::network::AAssociateACPDU, 93
  - gdcmm::network::AAssociateRQPDU, 102
- SetCenterOfRotation
  - gdcmm::Surface, 1065
- SetChangePrivateTags
  - gdcmm::FileExplicitFilter, 467
- SetCheckFileMetaInformation
  - gdcmm::Writer, 1418
- SetCipherType
  - gdcmm::CAPICryptographicMessageSyntax, 236
  - gdcmm::CryptographicMessageSyntax, 277
  - gdcmm::OpenSSLCryptographicMessageSyntax, 754
  - gdcmm::OpenSSLP7CryptographicMessageSyntax, 759
- SetColor
  - gdcmm::Printer, 863
- SetColorLevel
  - vtkImageColorViewer, 1369
- SetColorWindow
  - vtkImageColorViewer, 1369
- SetColumns
  - gdcmm::Bitmap, 202
  - gdcmm::Overlay, 771
- SetCommand
  - gdcmm::network::PresentationDataValue, 858
- SetComponents
  - gdcmm::PersonName, 798
- SetCompressIconImage
  - gdcmm::ImageChangeTransferSyntax, 543
- SetComputeZSpacing
  - gdcmm::IPPSorter, 605
- SetCoordinateStartValue
  - gdcmm::Curve, 307
- SetCoordinateStepValue
  - gdcmm::Curve, 307
- SetCryptographicMessageSyntax
  - gdcmm::Anonymizer, 117
- SetCurve
  - gdcmm::Curve, 307
  - vtkGDCMImageReader, 1299
  - vtkGDCMImageReader2, 1315
- SetCurveDataDescriptor
  - gdcmm::Curve, 307
- SetCurveDescription
  - gdcmm::Curve, 307
- SetData
  - gdcmm::DataEvent, 327
- SetDataElement
  - gdcmm::Bitmap, 202
- SetDataSet
  - gdcmm::File, 449
  - gdcmm::network::PresentationDataValue, 858
- SetDataSetTransferSyntax
  - gdcmm::FileMetaInformation, 475
- SetDataValueRepresentation
  - gdcmm::Curve, 308
- SetDebug
  - gdcmm::Trace, 1126
- SetDebugStream
  - gdcmm::Trace, 1126
- SetDefaultTransferSyntax
  - gdcmm::PresentationContextGenerator, 850
- SetDerivationCodeSequenceCodeValue
  - gdcmm::FileDerivation, 463
- SetDerivationDescription
  - gdcmm::FileDerivation, 464
- SetDescription
  - gdcmm::CSAHeaderDictEntry, 298
  - gdcmm::ModuleEntry, 708
  - gdcmm::Overlay, 771
- SetDescriptor
  - gdcmm::DICOMDIRGenerator, 358
- SetDictName
  - gdcmm::DictConverter, 367
- SetDicts
  - gdcmm::PythonFilter, 878
  - gdcmm::StringFilter, 1049
- SetDimension
  - gdcmm::Bitmap, 203
- SetDimensions
  - gdcmm::Bitmap, 203
  - gdcmm::Curve, 308
  - gdcmm::ImageCodec, 554
- SetDimensionsValue
  - gdcmm::ImageHelper, 569
- SetDirectionCosines
  - gdcmm::Image, 526, 527
  - vtkGDCMImageWriter, 1327
- SetDirectionCosinesFromImageOrientationPatient



- vtkGDCMImageWriter, [1327](#)
- SetDirectionCosinesTolerance
  - gdcm::IPPSorter, [605](#)
- SetDirectionCosinesValue
  - gdcm::ImageHelper, [569](#)
- SetDirectory
  - gdcm::network::ULWritingCallback, [1249](#)
  - gdcm::SerieHelper, [974](#)
- SetDisplayId
  - vtkImageColorViewer, [1369](#)
- SetDomain
  - gdcm::BoxRegion, [217](#)
- SetDropDuplicatePositions
  - gdcm::IPPSorter, [605](#)
- SetElement
  - gdcm::Tag, [1109](#)
- SetElementHandler
  - gdcm::Parser, [779](#)
- SetElementTag
  - gdcm::Tag, [1110](#)
- SetElementXX
  - gdcm::DictEntry, [371](#)
- SetError
  - gdcm::Trace, [1126](#)
- SetErrorStream
  - gdcm::Trace, [1127](#)
- SetEvent
  - gdcm::network::ULEvent, [1246](#)
- setfgcolor
  - gdcm::terminal, [86](#)
- SetFile
  - gdcm::Anonymizer, [117](#)
  - gdcm::DICOMDIRGenerator, [358](#)
  - gdcm::FileDecompressLookupTable, [460](#)
  - gdcm::FileDerivation, [464](#)
  - gdcm::FileExplicitFilter, [467](#)
  - gdcm::IconImageFilter, [517](#)
  - gdcm::Printer, [863](#)
  - gdcm::PythonFilter, [878](#)
  - gdcm::Reader, [904](#)
  - gdcm::SplitMosaicFilter, [1018](#)
  - gdcm::StreamImageWriter, [1028](#)
  - gdcm::StringFilter, [1049](#)
  - gdcm::Validate, [1264](#)
  - gdcm::Writer, [1418](#)
  - gdcm::XMLPrinter, [1426](#)
- SetFileName
  - gdcm::FileNameEvent, [484](#)
  - gdcm::Reader, [904](#)
  - gdcm::StreamImageReader, [1024](#)
  - gdcm::StreamImageWriter, [1028](#)
  - gdcm::Writer, [1418](#)
  - vtkGDCMThreadedImageReader2, [1357](#)
- SetFilename
  - gdcm::TableReader, [1099](#)
- SetFileNames
  - vtkGDCMImageReader, [1300](#)
  - vtkGDCMImageWriter, [1328](#)
  - vtkGDCMThreadedImageReader2, [1357](#)
- SetFilenames
  - gdcm::DICOMDIRGenerator, [358](#)
- SetFilePattern
  - vtkGDCMImageReader, [1300](#)
  - vtkGDCMImageReader2, [1315](#)
- SetFilePrefix
  - vtkGDCMImageReader, [1300](#)
  - vtkGDCMImageReader2, [1315](#)
- SetFiles
  - gdcm::FileSet, [491](#)
- SetFiniteVolume
  - gdcm::Surface, [1065](#)
- SetForce
  - gdcm::ImageChangeTransferSyntax, [543](#)
  - gdcm::ImageFragmentSplitter, [563](#)
- SetForcePixelSpacing
  - gdcm::ImageHelper, [569](#)
- SetForceRescaleInterceptSlope
  - gdcm::ImageHelper, [569](#)
- SetFragmentSizeMax
  - gdcm::ImageFragmentSplitter, [563](#)
- SetFrameOrigin
  - gdcm::Overlay, [772](#)
- SetFromDataElement
  - gdcm::Attribute< Group, Element, TVR, TVM >, [142](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [149](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [158](#)
  - gdcm::Element< TVR, TVM >, [399](#)
  - gdcm::Element< TVR, VM::VM1\_n >, [407](#)
- SetFromDataSet
  - gdcm::Attribute< Group, Element, TVR, TVM >, [142](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [150](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [158](#)
  - gdcm::MediaStorage, [682](#)
- SetFromFile
  - gdcm::MediaStorage, [682](#)
- SetFromHeader
  - gdcm::MediaStorage, [682](#)
- SetFromModality
  - gdcm::MediaStorage, [682](#)
- SetFromSourceImageSequence
  - gdcm::MediaStorage, [683](#)
- SetFromString
  - gdcm::DirectionCosines, [385](#)
- SetFromUID

- gdcM::UIDs, [1182](#)
- SetGreenLUT
  - gdcM::LookupTable, [662](#)
- SetGroup
  - gdcM::Curve, [308](#)
  - gdcM::Overlay, [772](#)
  - gdcM::Tag, [1110](#)
- SetGroupXX
  - gdcM::DictEntry, [371](#)
- SetHeader
  - gdcM::File, [449](#)
- SetHighBit
  - gdcM::PixelFormat, [813](#)
- SetHostname
  - gdcM::ServiceClassUser, [984](#)
- SetIconImage
  - gdcM::Pixmap, [819](#)
- SetIE
  - gdcM::IODEntry, [598](#)
- SetImage
  - gdcM::PixmapWriter, [831](#)
  - gdcM::SplitMosaicFilter, [1018](#)
- SetImplementationClassUID
  - gdcM::FileMetaInformation, [475](#)
- SetImplementationVersionName
  - gdcM::FileMetaInformation, [476](#)
- SetImplicitFlag
  - gdcM::network::ULConnectionCallback, [1233](#)
- SetInput
  - gdcM::BitmapToBitmapFilter, [211](#)
  - gdcM::ImageConverter, [560](#)
  - vtkImageColorViewer, [1370](#)
- SetInputConnection
  - vtkImageColorViewer, [1370](#)
- SetInputDirectory
  - gdcM::EmptyMaskGenerator, [423](#)
- SetInputFileName
  - gdcM::DictConverter, [367](#)
  - gdcM::FileAnonymizer, [453](#)
  - gdcM::FileChangeTransferSyntax, [457](#)
- SetIntercept
  - gdcM::Image, [527](#)
  - gdcM::Rescaler, [913](#)
- SetKey
  - gdcM::CSAElement, [283](#)
- SetKeyword
  - gdcM::DictEntry, [371](#)
- SetLastElement
  - gdcM::ParseException, [775](#)
- SetLastFragment
  - gdcM::network::PresentationDataValue, [858](#)
- SetLength
  - gdcM::ByteValue, [230](#)
  - gdcM::Element< TVR, VM::VM1\_2 >, [402](#)
  - gdcM::Element< TVR, VM::VM1\_n >, [407](#)
  - gdcM::Element< TVR, VM::VM2\_2n >, [409](#)
  - gdcM::Element< TVR, VM::VM2\_n >, [411](#)
  - gdcM::Element< TVR, VM::VM3\_3n >, [413](#)
  - gdcM::Element< TVR, VM::VM3\_n >, [415](#)
  - gdcM::RLECodec, [920](#)
  - gdcM::SequenceOfFragments, [959](#)
  - gdcM::SequenceOfItems, [968](#)
  - gdcM::Value, [1268](#)
- SetLengthOnly
  - gdcM::ByteValue, [230](#)
  - gdcM::Value, [1268](#)
- SetLengthToUndefined
  - gdcM::SequenceOfItems, [968](#)
- SetLoadMode
  - gdcM::SerieHelper, [974](#)
- SetLookupTable
  - vtkImageMapToColors16, [1380](#)
- SetLossless
  - gdcM::JPEGCodec, [636](#)
  - gdcM::JPEGLSCodec, [644](#)
- SetLossyError
  - gdcM::JPEGLSCodec, [644](#)
- SetLossyFlag
  - gdcM::Bitmap, [203](#)
  - gdcM::ImageCodec, [554](#)
  - gdcM::PVRGCodec, [877](#)
- SetLUT
  - gdcM::Bitmap, [203](#)
  - gdcM::ImageCodec, [555](#)
  - gdcM::LookupTable, [662](#)
  - gdcM::SegmentedPaletteColorLookupTable, [944](#)
- SetManifold
  - gdcM::Surface, [1066](#)
- SetMaximumLength
  - gdcM::network::MaximumLengthSub, [671](#)
- SetMaximumPointDistance
  - gdcM::Surface, [1066](#)
- SetMaxPDULength
  - gdcM::network::ULConnectionInfo, [1236](#)
- SetMaxPDUSize
  - gdcM::network::ULConnection, [1230](#)
- SetMeanPointDistance
  - gdcM::Surface, [1066](#)
- SetMedicalImageProperties
  - vtkGDCMImageReader, [1300](#)
  - vtkGDCMImageReader2, [1315](#)
  - vtkGDCMImageWriter, [1328](#)
  - vtkGDCMPolyDataWriter, [1344](#)
- SetMergeModeToAbstractSyntax
  - gdcM::PresentationContextGenerator, [850](#)
- SetMergeModeToTransferSyntax
  - gdcM::PresentationContextGenerator, [851](#)
- SetMeshPrimitive



- gdcm::Surface, [1066](#)
- SetMessageHeader
  - gdcm::network::PresentationDataValue, [858](#)
- SetMinMaxForPixelType
  - gdcm::Rescaler, [913](#)
- setmode
  - gdcm::terminal, [86](#)
- SetName
  - gdcm::CSAElement, [284](#)
  - gdcm::CSAHeaderDictEntry, [298](#)
  - gdcm::DictEntry, [371](#)
  - gdcm::IODEntry, [598](#)
  - gdcm::Macro, [666](#)
  - gdcm::Module, [704](#)
  - gdcm::ModuleEntry, [708](#)
  - gdcm::network::AbstractSyntax, [106](#)
  - gdcm::network::ApplicationContext, [120](#)
  - gdcm::network::TransferSyntaxSub, [1136](#)
  - gdcm::PDBElement, [786](#)
- SetNameFromUID
  - gdcm::network::AbstractSyntax, [106](#)
  - gdcm::network::TransferSyntaxSub, [1136](#)
- SetNeedByteSwap
  - gdcm::Bitmap, [204](#)
  - gdcm::ImageCodec, [555](#)
- SetNeedOverlayCleanup
  - gdcm::ImageCodec, [555](#)
- SetNestedDataSet
  - gdcm::Item, [611](#)
- SetNoOfItems
  - gdcm::CSAElement, [284](#)
- SetNoSwap
  - gdcm::Element< TVR, TVM >, [400](#)
  - gdcm::Element< TVR, VM::VM1\_n >, [407](#)
- SetNumberOfCurves
  - gdcm::Pixmap, [819](#)
- SetNumberOfDimensions
  - gdcm::Bitmap, [204](#)
  - gdcm::ImageCodec, [555](#)
- SetNumberOfFilenames
  - gdcm::FilenameGenerator, [488](#)
- SetNumberOfFrames
  - gdcm::Overlay, [772](#)
- SetNumberOfInputPorts
  - vtkGDCMPolyDataWriter, [1344](#)
- SetNumberOfItems
  - gdcm::SequenceOfItems, [968](#)
- SetNumberOfOverlays
  - gdcm::Pixmap, [819](#)
- SetNumberOfPoints
  - gdcm::Curve, [308](#)
- SetNumberOfResolutions
  - gdcm::JPEG2000Codec, [624](#)
- SetNumberOfSegments
  - gdcm::SegmentWriter, [951](#)
- SetNumberOfSurfacePoints
  - gdcm::Surface, [1066](#)
- SetNumberOfSurfaces
  - gdcm::SurfaceWriter, [1079](#)
- SetNumberOfTableValues
  - vtkLookupTable16, [1398](#)
- SetNumberOfThreadsForDecompression
  - gdcm::JPEG2000Codec, [624](#)
- SetNumberOfValues
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [158](#)
- SetNumberOfVectors
  - gdcm::Surface, [1066](#)
- SetObliquityThresholdCosineValue
  - gdcm::Orientation, [763](#)
- SetOffScreenRendering
  - vtkImageColorViewer, [1370](#)
- SetOrigin
  - gdcm::Image, [527](#), [528](#)
  - gdcm::Overlay, [772](#)
- SetOriginValue
  - gdcm::ImageHelper, [570](#)
- SetOutputDimensions
  - gdcm::IconImageGenerator, [520](#)
- SetOutputDirectory
  - gdcm::EmptyMaskGenerator, [424](#)
- SetOutputFileName
  - gdcm::DictConverter, [367](#)
  - gdcm::FileAnonymizer, [453](#)
  - gdcm::FileChangeTransferSyntax, [457](#)
  - gdcm::FileStreamer, [495](#)
- SetOutputFormatToLuminance
  - vtkImageMapToColors16, [1381](#)
- SetOutputFormatToLuminanceAlpha
  - vtkImageMapToColors16, [1381](#)
- SetOutputFormatToRGB
  - vtkImageMapToColors16, [1381](#)
- SetOutputFormatToRGBA
  - vtkImageMapToColors16, [1381](#)
- SetOutputType
  - gdcm::DictConverter, [367](#)
- SetOutsideValuePixel
  - gdcm::IconImageGenerator, [520](#)
- SetOverlay
  - gdcm::Overlay, [772](#)
- SetOverlayVisibility
  - vtkImageColorViewer, [1370](#)
- SetOwner
  - gdcm::PrivateTag, [870](#)
- SetParentId
  - vtkImageColorViewer, [1370](#)
- SetPassword
  - gdcm::CAPICryptographicMessageSyntax, [236](#)

- gdcm::CryptographicMessageSyntax, 278
- gdcm::OpenSSLCryptographicMessageSyntax, 754
- gdcm::OpenSSL7CryptographicMessageSyntax, 759
- SetPattern
  - gdcm::FilenameGenerator, 488
- SetPDU
  - gdcm::network::ULEvent, 1246
- SetPermissions
  - gdcm::System, 1090
- SetPhotometricInterpretation
  - gdcm::Bitmap, 204
  - gdcm::ImageChangePhotometricInterpretation, 535
  - gdcm::ImageCodec, 555
- SetPixelFormat
  - gdcm::Bitmap, 204
  - gdcm::ImageCodec, 556
  - gdcm::JPEGCodec, 637
  - gdcm::Rescaler, 914
- SetPixelMinMax
  - gdcm::IconImageGenerator, 520
- SetPixelRepresentation
  - gdcm::PixelFormat, 813
- SetPixmap
  - gdcm::FileDecompressLookupTable, 460
  - gdcm::IconImageGenerator, 521
  - gdcm::PixmapWriter, 832
- SetPlanarConfiguration
  - gdcm::Bitmap, 205
  - gdcm::ImageChangePlanarConfiguration, 539
  - gdcm::ImageCodec, 556
- SetPMSRescaleInterceptSlope
  - gdcm::ImageHelper, 570
- SetPointCoordinatesData
  - gdcm::Surface, 1067
- SetPointPositionAccuracy
  - gdcm::Surface, 1067
- SetPointsBoundingBoxCoordinates
  - gdcm::Surface, 1067
- SetPort
  - gdcm::ServiceClassUser, 984
- SetPortSCP
  - gdcm::ServiceClassUser, 984
- SetPosition
  - vtkImageColorViewer, 1370, 1371
- SetPreamble
  - gdcm::FileMetaInformation, 476
- SetPrefix
  - gdcm::FilenameGenerator, 488
- SetPresentationContextID
  - gdcm::network::PresentationContextAC, 846
  - gdcm::network::PresentationContextRQ, 854
  - gdcm::network::PresentationDataValue, 859
  - gdcm::PresentationContext, 844
- SetPresentationContexts
  - gdcm::network::ULConnection, 1230
  - gdcm::ServiceClassUser, 985
- SetPrettyPrint
  - gdcm::JSON, 647
- SetPrimitiveData
  - gdcm::MeshPrimitive, 694
- SetPrimitivesData
  - gdcm::MeshPrimitive, 694
- SetPrimitiveType
  - gdcm::MeshPrimitive, 694
- SetPrivateCreator
  - gdcm::Tag, 1110
- SetProcessingAlgorithm
  - gdcm::Surface, 1067
- SetProgress
  - gdcm::ProgressEvent, 873
- SetPropertyCategory
  - gdcm::Segment, 939
- SetPropertyType
  - gdcm::Segment, 939
- SetPropertyTypeModifiers
  - gdcm::Segment, 939
- SetPurposeOfReferenceCodeSequenceCodeValue
  - gdcm::FileDerivation, 464
- SetQuality
  - gdcm::JPEG2000Codec, 625
  - gdcm::JPEGCodec, 637
- SetRate
  - gdcm::JPEG2000Codec, 625
- SetReason
  - gdcm::network::AAabortPDU, 89
  - gdcm::network::PresentationContextAC, 847
- SetRecommendedDisplayCIELabValue
  - gdcm::Surface, 1067, 1068
- SetRecommendedDisplayGrayscaleValue
  - gdcm::Surface, 1068
- SetRecommendedPresentationOpacity
  - gdcm::Surface, 1068
- SetRecommendedPresentationType
  - gdcm::Surface, 1068
- SetRecomputeItemLength
  - gdcm::FileExplicitFilter, 467
- SetRecomputeSequenceLength
  - gdcm::FileExplicitFilter, 467
- SetRedLUT
  - gdcm::LookupTable, 662
- SetRef
  - gdcm::IODEntry, 598
- SetRegion
  - gdcm::ImageRegionReader, 579
- SetRenderer
  - vtkImageColorViewer, 1371
- SetRenderWindow

- vtkImageColorViewer, 1371
- SetRescaleInterceptSlopeValue
  - gdcm::ImageHelper, 570
- SetRetired
  - gdcm::DictEntry, 372
- SetReversible
  - gdcm::JPEG2000Codec, 625
- SetRGB8
  - gdcm::ImageApplyLookupTable, 531
- SetRoot
  - gdcm::UIDGenerator, 1145
- SetRootDirectory
  - gdcm::DICOMDIRGenerator, 358
- SetRows
  - gdcm::Bitmap, 205
  - gdcm::Overlay, 773
- SetRTStructSetProperties
  - vtkGDCMPolyDataWriter, 1344
- SetSamplesPerPixel
  - gdcm::PixelFormat, 813
- SetScalarType
  - gdcm::PixelFormat, 814
- SetSearchParameter
  - gdcm::BaseQuery, 180, 181
- SetSegmentAlgorithmName
  - gdcm::Segment, 939
- SetSegmentAlgorithmType
  - gdcm::Segment, 939
- SetSegmentDescription
  - gdcm::Segment, 940
- SetSegmentLabel
  - gdcm::Segment, 940
- SetSegmentNumber
  - gdcm::Segment, 940
- SetSegments
  - gdcm::SegmentWriter, 952
- SetSize
  - vtkImageColorViewer, 1371
- SetSlice
  - vtkImageColorViewer, 1372
- SetSliceOrientation
  - vtkImageColorViewer, 1372
- SetSliceOrientationToXY
  - vtkImageColorViewer, 1372
- SetSliceOrientationToXZ
  - vtkImageColorViewer, 1372
- SetSliceOrientationToYZ
  - vtkImageColorViewer, 1372
- SetSlope
  - gdcm::Image, 528
  - gdcm::Rescaler, 914
- SetSOPClassUIDMode
  - gdcm::EmptyMaskGenerator, 424
- SetSOPInstanceUID
  - gdcm::BaseQuery, 181
- SetSortFunction
  - gdcm::Sorter, 1010
- SetSource
  - gdcm::network::AAAbortPDU, 89
- SetSourceApplicationEntityTitle
  - gdcm::FileMetaInformation, 476
- SetSpacing
  - gdcm::Image, 528
- SetSpacingValue
  - gdcm::ImageHelper, 570
- SetState
  - gdcm::network::ULConnection, 1230
- SetStream
  - gdcm::Reader, 904
  - gdcm::StreamImageReader, 1024
  - gdcm::StreamImageWriter, 1028
  - gdcm::Trace, 1127
  - gdcm::Writer, 1419
- SetStreamToFile
  - gdcm::Trace, 1127
- SetStyle
  - gdcm::Printer, 863
  - gdcm::XMLPrinter, 1426
- SetSurfaceComments
  - gdcm::Surface, 1068
- SetSurfaceCount
  - gdcm::Segment, 940
- SetSurfaceNumber
  - gdcm::Surface, 1068
- SetSurfaceProcessing
  - gdcm::Surface, 1069
- SetSurfaceProcessingDescription
  - gdcm::Surface, 1069
- SetSurfaceProcessingRatio
  - gdcm::Surface, 1069
- SetSyngoDT
  - gdcm::CSAElement, 284
- SetTag
  - gdcm::AnonymizeEvent, 110
  - gdcm::DataElement, 319
- SetTagsToRead
  - gdcm::Sorter, 1010
- SetTargetPixelType
  - gdcm::Rescaler, 914
- SetTemplateFileName
  - gdcm::FileStreamer, 495
- SetTileSize
  - gdcm::JPEG2000Codec, 625
- SetTimeout
  - gdcm::network::ARTIMTimer, 130
  - gdcm::ServiceClassUser, 985
- SetToUndefined
  - gdcm::VL, 1276

- SetTransferSyntax
  - gdcm::Bitmap, [205](#)
  - gdcm::FileChangeTransferSyntax, [457](#)
  - gdcm::ImageChangeTransferSyntax, [543](#)
  - gdcm::network::PresentationContextAC, [847](#)
- SetTuple
  - gdcm::network::RoleSelectionSub, [922](#)
  - gdcm::network::ServiceClassApplicationInformation, [977](#)
  - gdcm::network::SOPClassExtendedNegociationSub, [1004](#)
- SetType
  - gdcm::ModuleEntry, [708](#)
  - gdcm::Overlay, [773](#)
- SetTypeOfData
  - gdcm::Curve, [308](#)
- SetupInteractor
  - vtkImageColorViewer, [1372](#)
- SetUsage
  - gdcm::IODEntry, [599](#)
- SetUserCodec
  - gdcm::ImageChangeTransferSyntax, [544](#)
- SetUserData
  - gdcm::Parser, [780](#)
- SetUserInformation
  - gdcm::network::AAssociateRQPDU, [102](#)
- SetUseSeriesDetails
  - gdcm::SerieHelper, [974](#)
- SetUseTargetPixelType
  - gdcm::Rescaler, [914](#)
- SetUseVRUN
  - gdcm::FileExplicitFilter, [467](#)
- SetValue
  - gdcm::Attribute< Group, Element, TVR, TVM >, [142](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [150](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [159](#)
  - gdcm::CSAElement, [284](#)
  - gdcm::DataElement, [319](#)
  - gdcm::Element< TVR, TVM >, [400](#)
  - gdcm::Element< TVR, VM::VM1\_n >, [407](#)
  - gdcm::PDBelement, [786](#)
- SetValueFieldLength
  - gdcm::DataElement, [320](#)
- SetValues
  - gdcm::Attribute< Group, Element, TVR, TVM >, [143](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [159](#)
- SetVectorAccuracy
  - gdcm::Surface, [1069](#)
- SetVectorCoordinateData
  - gdcm::Surface, [1069](#)
- SetVectorDimensionality
  - gdcm::Surface, [1069](#)
- SetVL
  - gdcm::DataElement, [320](#)
- SetVLToUndefined
  - gdcm::DataElement, [320](#)
- SetVM
  - gdcm::CSAElement, [284](#)
  - gdcm::CSAHeaderDictEntry, [298](#)
  - gdcm::DictEntry, [372](#)
- SetVR
  - gdcm::CSAElement, [284](#)
  - gdcm::CSAHeaderDictEntry, [298](#)
  - gdcm::DataElement, [320](#)
  - gdcm::DictEntry, [372](#)
- SetWarning
  - gdcm::Trace, [1127](#)
- SetWarningStream
  - gdcm::Trace, [1127](#)
- SetWindowId
  - vtkImageColorViewer, [1373](#)
- SetWriteDataSetOnly
  - gdcm::Writer, [1419](#)
- SetZSpacingTolerance
  - gdcm::IPPSorter, [606](#)
- SH
  - gdcm::VR, [1284](#)
- SHA1
  - gdcm::SHA1, [987](#)
- SHComp
  - gdcm, [59](#)
- Shift
  - vtkGDCMImageReader, [1309](#)
  - vtkGDCMImageReader2, [1323](#)
- ShiftEnd
  - gdcm::ByteBuffer, [218](#)
- ShowAbort
  - gdcm::SimpleSubjectWatcher, [994](#)
- ShowAnonymization
  - gdcm::SimpleSubjectWatcher, [995](#)
- ShowData
  - gdcm::SimpleSubjectWatcher, [995](#)
- ShowDataSet
  - gdcm::SimpleSubjectWatcher, [995](#)
- ShowFileName
  - gdcm::SimpleSubjectWatcher, [995](#)
- ShowIteration
  - gdcm::SimpleSubjectWatcher, [995](#)
- ShowProgress
  - gdcm::SimpleSubjectWatcher, [996](#)
- SIEMENS
  - gdcm::Dicts, [377](#)
  - gdcm::EquipmentManufacturer, [431](#)
- SimpleMemberCommand
  - gdcm::SimpleMemberCommand< T >, [990](#)

- SimpleSubjectWatcher
  - gdcm::SimpleSubjectWatcher, [994](#)
- SimplifiedAdultEchoSRStorage
  - gdcm::UIDs, [1170](#)
- SINGLEBIT
  - gdcm::PixelFormat, [808](#)
- SingleSerieUIDFileSetHT
  - gdcm::SerieHelper, [975](#)
- SingleSerieUIDFileSetmap
  - gdcm::SerieHelper, [971](#)
- Size
  - gdcm::CodeString, [256](#)
  - gdcm::DataSet, [339](#)
  - gdcm::GroupDict, [513](#)
  - gdcm::network::AAAbortPDU, [89](#)
  - gdcm::network::AAssociateACPDU, [93](#)
  - gdcm::network::AAssociateRJPDU, [96](#)
  - gdcm::network::AAssociateRQPDU, [102](#)
  - gdcm::network::AbstractSyntax, [106](#)
  - gdcm::network::ApplicationContext, [120](#)
  - gdcm::network::AReleaseRPPDU, [126](#)
  - gdcm::network::AReleaseRQPDU, [128](#)
  - gdcm::network::AsynchronousOperationsWindowSub, [133](#)
  - gdcm::network::BasePDU, [177](#)
  - gdcm::network::ImplementationClassUIDSub, [587](#)
  - gdcm::network::ImplementationVersionNameSub, [589](#)
  - gdcm::network::MaximumLengthSub, [671](#)
  - gdcm::network::PDataTFPDU, [783](#)
  - gdcm::network::PresentationContextAC, [847](#)
  - gdcm::network::PresentationContextRQ, [855](#)
  - gdcm::network::PresentationDataValue, [859](#)
  - gdcm::network::RoleSelectionSub, [922](#)
  - gdcm::network::ServiceClassApplicationInformation, [977](#)
  - gdcm::network::SOPClassExtendedNegociationSub, [1004](#)
  - gdcm::network::TransferSyntaxSub, [1136](#)
  - gdcm::network::UserInformation, [1262](#)
- size\_type
  - gdcm::CodeString, [254](#)
  - gdcm::LO, [653](#)
  - gdcm::String< TDelimiter, TMaxLength, TPadChar >, [1044](#)
- SizeType
  - gdcm::DataSet, [330](#)
  - gdcm::FilenameGenerator, [486](#)
  - gdcm::IOD, [595](#)
  - gdcm::NestedModuleEntries, [730](#)
  - gdcm::network::AAssociateACPDU, [91](#)
  - gdcm::network::AAssociateRQPDU, [99](#)
  - gdcm::network::PDataTFPDU, [782](#)
  - gdcm::network::PresentationContextRQ, [852](#)
  - gdcm::PresentationContext, [842](#)
  - gdcm::PresentationContextGenerator, [849](#)
  - gdcm::SequenceOfFragments, [955](#)
  - gdcm::SequenceOfItems, [963](#)
- SL
  - gdcm::VR, [1285](#)
- Slice
  - vtkImageColorViewer, [1377](#)
- SLICE\_ORIENTATION\_XY
  - vtkImageColorViewer, [1366](#)
- SLICE\_ORIENTATION\_XZ
  - vtkImageColorViewer, [1366](#)
- SLICE\_ORIENTATION\_YZ
  - vtkImageColorViewer, [1366](#)
- SliceOrientation
  - vtkImageColorViewer, [1377](#)
- Slices
  - gdcm::MrProtocol::SliceArray, [998](#)
- SmartPointer
  - gdcm::Object, [750](#)
  - gdcm::SmartPointer< ObjectType >, [1000](#), [1001](#)
- SOPClassExtendedNegociationSub
  - gdcm::network::SOPClassExtendedNegociationSub, [1004](#)
- SOPClassUIDMode
  - gdcm::EmptyMaskGenerator, [423](#)
- SOPInstanceUID
  - vtkRTStructSetProperties, [1409](#)
- Sort
  - gdcm::IPPSorter, [606](#)
  - gdcm::Sorter, [1010](#)
- Sorter
  - gdcm::Sorter, [1009](#)
- SortFunc
  - gdcm::Sorter, [1012](#)
- SortFunction
  - gdcm::Sorter, [1009](#)
- SpacialFiducialsStorage
  - gdcm::MediaStorage, [677](#)
- SpacialRegistrationStorage
  - gdcm::MediaStorage, [677](#)
- Spacing
  - gdcm::Spacing, [1014](#)
- SpacingType
  - gdcm::Spacing, [1013](#)
- SpatialFiducialsStorage
  - gdcm::UIDs, [1165](#)
- SpatialRegistrationStorage
  - gdcm::UIDs, [1165](#)
- SpectaclePrescriptionReportStorage
  - gdcm::UIDs, [1169](#)
- Spectroscopy
  - gdcm::Spectroscopy, [1015](#)
- Split

- gdcm::ImageFragmentSplitter, [564](#)
- gdcm::SplitMosaicFilter, [1018](#)
- SplitExtent
  - vtkGDCMThreadedImageReader2, [1357](#)
- SplitMosaicFilter
  - gdcm::SplitMosaicFilter, [1016](#)
- SPM2AVG152PDFrameofReference
  - gdcm::UIDs, [1163](#)
- SPM2AVG152T1FrameofReference
  - gdcm::UIDs, [1163](#)
- SPM2AVG152T2FrameofReference
  - gdcm::UIDs, [1163](#)
- SPM2AVG305T1FrameofReference
  - gdcm::UIDs, [1163](#)
- SPM2BRAINMASKFrameofReference
  - gdcm::UIDs, [1163](#)
- SPM2CSFFrameofReference
  - gdcm::UIDs, [1163](#)
- SPM2EPIFrameofReference
  - gdcm::UIDs, [1163](#)
- SPM2FILT1FrameofReference
  - gdcm::UIDs, [1163](#)
- SPM2GRAYFrameofReference
  - gdcm::UIDs, [1163](#)
- SPM2PDFrameofReference
  - gdcm::UIDs, [1163](#)
- SPM2PETFrameofReference
  - gdcm::UIDs, [1163](#)
- SPM2SINGLESUBJT1FrameofReference
  - gdcm::UIDs, [1163](#)
- SPM2SPECTFrameofReference
  - gdcm::UIDs, [1163](#)
- SPM2T1FrameofReference
  - gdcm::UIDs, [1163](#)
- SPM2T2FrameofReference
  - gdcm::UIDs, [1163](#)
- SPM2TRANSMFrameofReference
  - gdcm::UIDs, [1163](#)
- SPM2WHITEFrameofReference
  - gdcm::UIDs, [1163](#)
- SpringColorPaletteSOPInstance
  - gdcm::UIDs, [1168](#)
- SQ
  - gdcm::VR, [1285](#)
- Squeeze
  - gdcm::ApplicationEntity, [123](#)
- SS
  - gdcm::VR, [1285](#)
- ST
  - gdcm::VR, [1285](#)
- StableSort
  - gdcm::Sorter, [1011](#)
- StandaloneCurveStorage
  - gdcm::MediaStorage, [677](#)
- StandaloneCurveStorageRetired
  - gdcm::UIDs, [1165](#)
- StandaloneModalityLUTStorage
  - gdcm::MediaStorage, [677](#)
- StandaloneModalityLUTStorageRetired
  - gdcm::UIDs, [1165](#)
- StandaloneOverlayStorage
  - gdcm::MediaStorage, [677](#)
- StandaloneOverlayStorageRetired
  - gdcm::UIDs, [1165](#)
- StandalonePETCurveStorageRetired
  - gdcm::UIDs, [1166](#)
- StandaloneVOILUTStorage
  - gdcm::MediaStorage, [677](#)
- StandaloneVOILUTStorageRetired
  - gdcm::UIDs, [1165](#)
- Start
  - gdcm::network::ARTIMTimer, [130](#)
- StartAssociation
  - gdcm::ServiceClassUser, [985](#)
- StartDataElement
  - gdcm::FileStreamer, [495](#)
- StartElement
  - gdcm::TableReader, [1099](#)
  - gdcm::XMLDictReader, [1423](#)
  - gdcm::XMLPrivateDictReader, [1429](#)
- StartElementHandler
  - gdcm::Parser, [777](#)
- StartEncode
  - gdcm::ImageCodec, [556](#)
  - gdcm::JPEG2000Codec, [625](#)
  - gdcm::JPEGCodec, [637](#)
  - gdcm::JPEGLSCodec, [644](#)
  - gdcm::RLECodec, [920](#)
- StartFilter
  - gdcm::SimpleSubjectWatcher, [996](#)
- StartGroupDataElement
  - gdcm::FileStreamer, [495](#)
- STATES
  - gdcm::Surface, [1058](#)
- STATES\_END
  - gdcm::Surface, [1058](#)
- STComp
  - gdcm, [60](#)
- StereometricRelationshipStorage
  - gdcm::UIDs, [1166](#)
- Stop
  - gdcm::network::ARTIMTimer, [130](#)
- StopAssociation
  - gdcm::ServiceClassUser, [985](#)
- StopDataElement
  - gdcm::FileStreamer, [496](#)
- StopEncode
  - gdcm::ImageCodec, [556](#)



- gdcm::JPEG2000Codec, [625](#)
- gdcm::JPEGCodec, [637](#)
- gdcm::JPEGLSCCodec, [644](#)
- gdcm::RLECodec, [920](#)
- StopGroupDataElement
  - gdcm::FileStreamer, [496](#)
- StopProtocol
  - gdcm::network::ULConnection, [1230](#)
- StorageCommitmentPullModelSOPClassRetired
  - gdcm::UIDs, [1163](#)
- StorageCommitmentPullModelSOPInstanceRetired
  - gdcm::UIDs, [1163](#)
- StorageCommitmentPushModelSOPClass
  - gdcm::UIDs, [1163](#)
- StorageCommitmentPushModelSOPInstance
  - gdcm::UIDs, [1163](#)
- StorageServiceClass
  - gdcm::UIDs, [1164](#)
- StoredPrintStorageSOPClassRetired
  - gdcm::UIDs, [1164](#)
- StrCaseCmp
  - gdcm::System, [1090](#)
- Stream
  - gdcm::Writer, [1420](#)
- StreamImageReader
  - gdcm::Reader, [905](#)
  - gdcm::StreamImageReader, [1021](#)
- StreamImageWriter
  - gdcm::StreamImageWriter, [1027](#)
  - gdcm::Writer, [1420](#)
- StrictScanner
  - gdcm::StrictScanner, [1035](#)
- String
  - gdcm::String< TDelimiter, TMaxLength, TPadChar >, [1044](#), [1045](#)
- StringFilter
  - gdcm::StringFilter, [1047](#)
- StrNCaseCmp
  - gdcm::System, [1091](#)
- StrSep
  - gdcm::System, [1091](#)
- StrTokR
  - gdcm::System, [1091](#)
- StructureSetDate
  - vtkRTStructSetProperties, [1410](#)
- StructureSetLabel
  - vtkRTStructSetProperties, [1410](#)
- StructureSetName
  - vtkRTStructSetProperties, [1410](#)
- StructureSetTime
  - vtkRTStructSetProperties, [1410](#)
- Study
  - gdcm::Study, [1051](#)
- StudyComponentManagementSOPClass
  - gdcm::MediaStorage, [677](#)
- StudyComponentManagementSOPClassRetired
  - gdcm::UIDs, [1163](#)
- StudyInstanceUID
  - vtkRTStructSetProperties, [1410](#)
- StudyRootQueryRetrieveInformationModelFIND
  - gdcm::UIDs, [1166](#)
- StudyRootQueryRetrieveInformationModelGET
  - gdcm::UIDs, [1166](#)
- StudyRootQueryRetrieveInformationModelMOVE
  - gdcm::UIDs, [1166](#)
- Subject
  - gdcm::Subject, [1053](#)
- SubjectiveRefractionMeasurementsStorage
  - gdcm::UIDs, [1169](#)
- SubstanceAdministrationLoggingSOPClass
  - gdcm::UIDs, [1163](#)
- SubstanceAdministrationLoggingSOPInstance
  - gdcm::UIDs, [1163](#)
- SubstanceApprovalQuerySOPClass
  - gdcm::UIDs, [1167](#)
- SummerColorPaletteSOPInstance
  - gdcm::UIDs, [1168](#)
- Superclass
  - gdcm::AnonymizeEvent, [108](#)
  - gdcm::DataEvent, [325](#)
  - gdcm::DataSetEvent, [342](#)
  - gdcm::FileNameEvent, [483](#)
  - gdcm::LO, [653](#)
  - gdcm::ProgressEvent, [872](#)
- SURFACE
  - gdcm::Surface, [1058](#)
- Surface
  - gdcm::Surface, [1058](#)
- SurfaceCount
  - gdcm::Segment, [942](#)
- SurfaceReader
  - gdcm::SurfaceReader, [1075](#)
- Surfaces
  - gdcm::Segment, [942](#)
- SurfaceScanMeshStorage
  - gdcm::UIDs, [1169](#)
- SurfaceScanPointCloudStorage
  - gdcm::UIDs, [1169](#)
- SurfaceSegmentationStorage
  - gdcm::MediaStorage, [678](#)
  - gdcm::UIDs, [1168](#)
- SurfaceVector
  - gdcm::Segment, [934](#)
- SurfaceWriter
  - gdcm::SurfaceWriter, [1078](#)
- SV
  - gdcm::VR, [1285](#)
- SV10

- gdcm::CSAHeader, [288](#)
- Swap
  - gdcm::ByteSwap< T >, [219](#)
  - gdcm::SwapperDoOp, [1082](#)
  - gdcm::SwapperNoOp, [1083](#)
- SwapArray
  - gdcm::SwapperDoOp, [1083](#)
  - gdcm::SwapperNoOp, [1084](#)
- SwapCode
  - gdcm::SwapCode, [1081](#)
- SwapCodeType
  - gdcm::SwapCode, [1081](#)
- SwapFromSwapCodeIntoSystem
  - gdcm::ByteSwap< T >, [220](#)
- SwapRange
  - gdcm::ByteSwap< T >, [220](#)
- SwapRangeFromSwapCodeIntoSystem
  - gdcm::ByteSwap< T >, [220](#)
- SyngoDTField
  - gdcm::CSAElement, [286](#)
- SyntaxError
  - gdcm::Parser, [778](#)
- SystemIsBigEndian
  - gdcm::ByteSwap< T >, [220](#)
- SystemIsLittleEndian
  - gdcm::ByteSwap< T >, [221](#)
- T1
  - gdcm::Type, [1140](#)
- T1C
  - gdcm::Type, [1140](#)
- T2
  - gdcm::Type, [1140](#)
- T2C
  - gdcm::Type, [1140](#)
- T3
  - gdcm::Type, [1140](#)
- Table
  - gdcm::Table, [1093](#)
- Table16
  - vtkLookupTable16, [1398](#)
- TableEntry
  - gdcm::TableEntry, [1095](#)
- TableInternal
  - gdcm::Table, [1094](#)
- TableReader
  - gdcm::TableReader, [1097](#)
- TableRow
  - gdcm::network::TableRow, [1101](#)
- Tag
  - gdcm::Tag, [1103](#), [1104](#)
- tag
  - gdcm::Tag, [1112](#)
- TagField
  - gdcm::DataElement, [321](#)
- TagMismatchError
  - gdcm::Parser, [778](#)
- TagPath
  - gdcm::TagPath, [1113](#)
- tags
  - gdcm::Tag, [1112](#)
- TagsToRead
  - gdcm::Sorter, [1012](#)
- TagToValue
  - gdcm::Scanner, [926](#)
  - gdcm::StrictScanner, [1035](#)
- TagToValueValueType
  - gdcm::Scanner, [926](#)
  - gdcm::StrictScanner, [1035](#)
- TalairachBrainAtlasFrameofReference
  - gdcm::UIDs, [1163](#)
- TConstMemberFunctionPointer
  - gdcm::MemberCommand< T >, [686](#)
- TestAbortOff
  - gdcm::SimpleSubjectWatcher, [996](#)
- TestAbortOn
  - gdcm::SimpleSubjectWatcher, [996](#)
- Testing
  - gdcm::Testing, [1116](#)
- TestPBKDF2
  - gdcm::ASN1, [132](#)
- TestsList.txt, [1709](#)
- TextSRStorageTrialRetired
  - gdcm::UIDs, [1166](#)
- ThreadedExecute
  - vtkImageRGBToYBR, [1392](#)
  - vtkImageYBRToRGB, [1394](#)
- ThreadedRequestData
  - vtkGDCMThreadedImageReader2, [1357](#)
  - vtkImageMapToColors16, [1381](#)
  - vtkImageMapToWindowLevelColors2, [1386](#)
- TM
  - gdcm::VR, [1285](#)
- TMComp
  - gdcm, [60](#)
- TMemberFunctionPointer
  - gdcm::MemberCommand< T >, [686](#)
  - gdcm::SimpleMemberCommand< T >, [990](#)
- ToPyObject
  - gdcm::PythonFilter, [878](#)
- TOSHIBA
  - gdcm::EquipmentManufacturer, [431](#)
- ToshibaPrivateDataStorage
  - gdcm::MediaStorage, [678](#)
- ToString
  - gdcm::StringFilter, [1049](#)
- ToStringPair
  - gdcm::StringFilter, [1050](#)



- ToUnixSlashes
  - gdcm::Filename, [480](#)
- ToWindowsSlashes
  - gdcm::Filename, [480](#)
- Trace
  - gdcm::Trace, [1124](#)
- TractographyResultsStorage
  - gdcm::UIDs, [1169](#)
- TransferSyntax
  - gdcm::TransferSyntax, [1131](#)
- TransferSyntaxArrayType
  - gdcm::PresentationContext, [842](#)
- TransferSyntaxes
  - gdcm::PresentationContext, [844](#)
- TransferSyntaxStringsType
  - gdcm::UIDs, [1161](#)
- TransferSyntaxSub
  - gdcm::network::TransferSyntaxSub, [1135](#)
- Transition
  - gdcm::network::Transition, [1138](#)
- transitions
  - gdcm::network::TableRow, [1101](#)
- TRIANGLE
  - gdcm::MeshPrimitive, [692](#)
- TRIANGLE\_FAN
  - gdcm::MeshPrimitive, [692](#)
- TRIANGLE\_STRIP
  - gdcm::MeshPrimitive, [692](#)
- Trim
  - gdcm::String< TDelimiter, TMaxLength, TPadChar >, [1045](#), [1046](#)
- TrimInternal
  - gdcm::CodeString, [256](#)
- Truncate
  - gdcm::String< TDelimiter, TMaxLength, TPadChar >, [1046](#)
- TryJPEG2000Codec
  - gdcm::Bitmap, [205](#)
  - gdcm::ImageChangeTransferSyntax, [544](#)
- TryJPEG2000Codec2
  - gdcm::Bitmap, [206](#)
- TryJPEGCodec
  - gdcm::Bitmap, [206](#)
  - gdcm::ImageChangeTransferSyntax, [544](#)
- TryJPEGCodec2
  - gdcm::Bitmap, [206](#)
- TryJPEGLSCodec
  - gdcm::Bitmap, [206](#)
  - gdcm::ImageChangeTransferSyntax, [544](#)
- TryKAKADUCodec
  - gdcm::Bitmap, [206](#)
- TryPVRGCodec
  - gdcm::Bitmap, [206](#)
- TryRAWCodec
  - gdcm::Bitmap, [207](#)
  - gdcm::ImageChangeTransferSyntax, [545](#)
- TryRLECodec
  - gdcm::Bitmap, [207](#)
  - gdcm::ImageChangeTransferSyntax, [545](#)
- TS
  - gdcm::Bitmap, [209](#)
- TS\_END
  - gdcm::TransferSyntax, [1131](#)
- TSName
  - gdcm::UIDs, [1162](#)
- TSType
  - gdcm::TransferSyntax, [1131](#)
  - gdcm::UIDs, [1171](#)
- Type
  - gdcm::Element< TVR, TVM >, [397](#)
  - gdcm::Element< TVR, VM::VM1\_n >, [404](#)
  - gdcm::EquipmentManufacturer, [431](#)
  - gdcm::Type, [1140](#)
  - gdcm::VL, [1274](#)
- TYPETOENCODING
  - gdcm, [75](#)
  - gdcmVR.h, [1703](#)
- TYPETOLENGTH
  - gdcmVM.h, [1701](#)
- TypeType
  - gdcm::Type, [1140](#)
- UberonOntology
  - gdcm::UIDs, [1168](#)
- UC
  - gdcm::VR, [1285](#)
- UCComp
  - gdcm, [60](#)
- UI
  - gdcm::VR, [1285](#)
- UIComp
  - gdcm, [60](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_1
  - gdcm::UIDs, [1176](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_10
  - gdcm::UIDs, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_11
  - gdcm::UIDs, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_12
  - gdcm::UIDs, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_13
  - gdcm::UIDs, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_14
  - gdcm::UIDs, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_15
  - gdcm::UIDs, [1177](#)
- uid\_1\_2\_840\_10008\_15\_0\_3\_16
  - gdcm::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_3\_17  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_3\_18  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_3\_19  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_3\_2  
gdcn::UIDs, [1176](#)

uid\_1\_2\_840\_10008\_15\_0\_3\_20  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_3\_21  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_3\_22  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_3\_23  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_3\_24  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_3\_25  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_3\_26  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_3\_27  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_3\_28  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_3\_29  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_3\_3  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_3\_30  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_3\_31  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_3\_4  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_3\_5  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_3\_6  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_3\_7  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_3\_8  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_3\_9  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_4\_1  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_4\_2  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_4\_3  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_4\_4  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_4\_5  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_4\_6  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_4\_7  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_0\_4\_8  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_15\_1\_1  
gdcn::UIDs, [1180](#)

uid\_1\_2\_840\_10008\_1\_1  
gdcn::UIDs, [1171](#)

uid\_1\_2\_840\_10008\_1\_2  
gdcn::UIDs, [1171](#)

uid\_1\_2\_840\_10008\_1\_20  
gdcn::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_1\_20\_1  
gdcn::UIDs, [1173](#)

uid\_1\_2\_840\_10008\_1\_20\_1\_1  
gdcn::UIDs, [1173](#)

uid\_1\_2\_840\_10008\_1\_20\_2  
gdcn::UIDs, [1173](#)

uid\_1\_2\_840\_10008\_1\_20\_2\_1  
gdcn::UIDs, [1173](#)

uid\_1\_2\_840\_10008\_1\_2\_1  
gdcn::UIDs, [1171](#)

uid\_1\_2\_840\_10008\_1\_2\_1\_99  
gdcn::UIDs, [1171](#)

uid\_1\_2\_840\_10008\_1\_2\_2  
gdcn::UIDs, [1171](#)

uid\_1\_2\_840\_10008\_1\_2\_4\_100  
gdcn::UIDs, [1172](#)

uid\_1\_2\_840\_10008\_1\_2\_4\_101  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_1\_2\_4\_102  
gdcn::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_1\_2\_4\_103  
gdcn::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_1\_2\_4\_104  
gdcn::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_1\_2\_4\_105  
gdcn::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_1\_2\_4\_106  
gdcn::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_1\_2\_4\_107  
gdcn::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_1\_2\_4\_108  
gdcn::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_1\_2\_4\_50  
gdcn::UIDs, [1171](#)

uid\_1\_2\_840\_10008\_1\_2\_4\_51  
gdcn::UIDs, [1171](#)

uid\_1\_2\_840\_10008\_1\_2\_4\_52  
gdcn::UIDs, [1172](#)

uid\_1\_2\_840\_10008\_1\_2\_4\_53  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_4\_54  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_4\_55  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_4\_56  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_4\_57  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_4\_58  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_4\_59  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_4\_60  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_4\_61  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_4\_62  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_4\_63  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_4\_64  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_4\_65  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_4\_66  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_4\_70  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_4\_80  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_4\_81  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_4\_90  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_4\_91  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_4\_92  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_4\_93  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_4\_94  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_4\_95  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_5  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_6\_1  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_2\_6\_2  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_3\_10  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_40  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_1\_40\_1  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_1\_42  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_1\_42\_1  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_1\_4\_1\_1  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_4\_1\_10  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_4\_1\_11  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_4\_1\_12  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_4\_1\_13  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_4\_1\_14  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_4\_1\_15  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_4\_1\_16  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_4\_1\_17  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_4\_1\_18  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_1\_4\_1\_2  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_4\_1\_3  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_4\_1\_4  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_4\_1\_5  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_4\_1\_6  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_4\_1\_7  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_4\_1\_8  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_4\_1\_9  
gdcml:UIDs, 1172

uid\_1\_2\_840\_10008\_1\_4\_2\_1  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_1\_4\_2\_2  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_1\_5\_1  
gdcml:UIDs, 1178

uid\_1\_2\_840\_10008\_1\_5\_2  
gdcml:UIDs, 1178

uid\_1\_2\_840\_10008\_1\_5\_3  
gdcml:UIDs, 1178

uid\_1\_2\_840\_10008\_1\_5\_4  
gdcml:UIDs, 1178

uid\_1\_2\_840\_10008\_1\_5\_5  
gdcml:UIDs, 1178

uid\_1\_2\_840\_10008\_1\_5\_6  
gdcml:UIDs, 1178

uid\_1\_2\_840\_10008\_1\_5\_7  
gdcml:UIDs, 1178

uid\_1\_2\_840\_10008\_1\_5\_8  
gdcml:UIDs, 1178

uid\_1\_2\_840\_10008\_1\_9  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_2\_16\_10  
gdcml:UIDs, 1178

uid\_1\_2\_840\_10008\_2\_16\_11  
gdcml:UIDs, 1178

uid\_1\_2\_840\_10008\_2\_16\_12  
gdcml:UIDs, 1178

uid\_1\_2\_840\_10008\_2\_16\_13  
gdcml:UIDs, 1178

uid\_1\_2\_840\_10008\_2\_16\_14  
gdcml:UIDs, 1178

uid\_1\_2\_840\_10008\_2\_16\_4  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_2\_16\_5  
gdcml:UIDs, 1178

uid\_1\_2\_840\_10008\_2\_16\_6  
gdcml:UIDs, 1178

uid\_1\_2\_840\_10008\_2\_16\_7  
gdcml:UIDs, 1178

uid\_1\_2\_840\_10008\_2\_16\_8  
gdcml:UIDs, 1178

uid\_1\_2\_840\_10008\_2\_16\_9  
gdcml:UIDs, 1178

uid\_1\_2\_840\_10008\_2\_6\_1  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_3\_1\_1\_1  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_3\_1\_2\_1\_1  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_3\_1\_2\_1\_4  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_3\_1\_2\_2\_1  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_3\_1\_2\_3\_1  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_3\_1\_2\_3\_2  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_3\_1\_2\_3\_3  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_3\_1\_2\_3\_4  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_3\_1\_2\_3\_5  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_3\_1\_2\_5\_1  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_3\_1\_2\_5\_4  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_3\_1\_2\_5\_5  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_3\_1\_2\_6\_1  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_4\_2  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_5\_1\_1\_1  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_5\_1\_1\_14  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_5\_1\_1\_15  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_5\_1\_1\_16  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_5\_1\_1\_16\_376  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_5\_1\_1\_17  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_5\_1\_1\_17\_376  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_5\_1\_1\_18  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_5\_1\_1\_18\_1  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_5\_1\_1\_2  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_5\_1\_1\_22  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_5\_1\_1\_23  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_5\_1\_1\_24  
gdcml:UIDs, 1173

uid\_1\_2\_840\_10008\_5\_1\_1\_24\_1  
gdcml:UIDs, 1174

uid\_1\_2\_840\_10008\_5\_1\_1\_25  
gdcml:UIDs, 1174

uid\_1\_2\_840\_10008\_5\_1\_1\_26  
gdcml:UIDs, 1174

uid\_1\_2\_840\_10008\_5\_1\_1\_27  
gdcml:UIDs, 1174

uid\_1\_2\_840\_10008\_5\_1\_1\_29  
gdcml:UIDs, 1174

uid\_1\_2\_840\_10008\_5\_1\_1\_30  
gdcml:UIDs, 1174

uid\_1\_2\_840\_10008\_5\_1\_1\_31  
gdcml:UIDs, 1174

uid\_1\_2\_840\_10008\_5\_1\_1\_32  
gdcml:UIDs, 1174

uid\_1\_2\_840\_10008\_5\_1\_1\_33  
gdcml:UIDs, 1174

uid\_1\_2\_840\_10008\_5\_1\_1\_4  
gdcm::UIDs, [1173](#)

uid\_1\_2\_840\_10008\_5\_1\_1\_40  
gdcm::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_5\_1\_1\_40\_1  
gdcm::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_5\_1\_1\_4\_1  
gdcm::UIDs, [1173](#)

uid\_1\_2\_840\_10008\_5\_1\_1\_4\_2  
gdcm::UIDs, [1173](#)

uid\_1\_2\_840\_10008\_5\_1\_1\_9  
gdcm::UIDs, [1173](#)

uid\_1\_2\_840\_10008\_5\_1\_1\_9\_1  
gdcm::UIDs, [1173](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_10  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_104\_1  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_104\_2  
gdcm::UIDs, [1176](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_104\_3  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_1  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_10  
gdcm::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_11  
gdcm::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_2  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_3  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_4  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_5  
gdcm::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_6  
gdcm::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_7  
gdcm::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_8  
gdcm::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_9  
gdcm::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_128  
gdcm::UIDs, [1176](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_128\_1  
gdcm::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_129  
gdcm::UIDs, [1176](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_1  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_1\_1  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_2  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_2\_1  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_3  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_77  
gdcm::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_130  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_131  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_1  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_2  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_3  
gdcm::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_4  
gdcm::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_5  
gdcm::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_14\_1  
gdcm::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_14\_2  
gdcm::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_1  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_1\_1  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_2  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_2\_1  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_3  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_3\_1  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_2  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_20  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_1  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_2  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_3  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_4  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_5  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_200\_6  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_2\_1  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_2\_2  
gdcm::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_3  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_30  
gdcm::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_3\_1  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_40  
gdcm::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_1  
gdcm::UIDs, [1176](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_10  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_11  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_2  
gdcm::UIDs, [1176](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_3  
gdcm::UIDs, [1176](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_4  
gdcm::UIDs, [1176](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_5  
gdcm::UIDs, [1176](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_6  
gdcm::UIDs, [1176](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_7  
gdcm::UIDs, [1176](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_8  
gdcm::UIDs, [1176](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_9  
gdcm::UIDs, [1176](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4\_1  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4\_2  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4\_3  
gdcm::UIDs, [1180](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4\_4  
gdcm::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_5  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_1  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_2\_1  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_2\_2  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_3  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_4  
gdcm::UIDs, [1180](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_5  
gdcm::UIDs, [1180](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_501\_6  
gdcm::UIDs, [1180](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_6  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_601\_1  
gdcm::UIDs, [1180](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_601\_2  
gdcm::UIDs, [1180](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_1  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_2  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_3  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_4  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_5  
gdcm::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_6  
gdcm::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_67  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_68\_1  
gdcm::UIDs, [1178](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_68\_2  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_6\_1  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_6\_2  
gdcm::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_7  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_1  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_2  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_2\_1  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_3  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_4  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_4\_1  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_1  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_2  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_3  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_4  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_5  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_6  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_7  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_8  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_6  
gdcm::UIDs, [1177](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_2  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_78\_1  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_78\_2  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_78\_3  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_78\_4  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_78\_5  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_78\_6  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_78\_7  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_78\_8  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_79\_1  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_7\_1  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_7\_2  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_7\_3  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_7\_4  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_8  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_80\_1  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_81\_1  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_82\_1  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_1  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_11  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_2  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_22  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_3  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_33  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_34  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_35  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_4  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_40  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_50  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_59  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_65  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_67  
gdcm::UIDs, [1175](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_68  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_69  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_70  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_71  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_72  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_73  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_74  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_75  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_90\_1  
gdcm::UIDs, [1179](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_1  
gdcm::UIDs, [1174](#)

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_1\_1  
gdcmm::UIDs, 1174

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_1\_2  
gdcmm::UIDs, 1174

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_1\_3  
gdcmm::UIDs, 1174

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_2\_1  
gdcmm::UIDs, 1174

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_3\_1  
gdcmm::UIDs, 1174

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_4\_1  
gdcmm::UIDs, 1174

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_4\_2  
gdcmm::UIDs, 1178

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_5\_1  
gdcmm::UIDs, 1178

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_6\_1  
gdcmm::UIDs, 1178

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_1\_1  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_1\_2  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_1\_3  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_2\_1  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_2\_2  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_2\_3  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_3\_1  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_3\_2  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_3\_3  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_4\_2  
gdcmm::UIDs, 1180

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_4\_3  
gdcmm::UIDs, 1180

uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_5\_3  
gdcmm::UIDs, 1180

uid\_1\_2\_840\_10008\_5\_1\_4\_20\_1  
gdcmm::UIDs, 1180

uid\_1\_2\_840\_10008\_5\_1\_4\_20\_2  
gdcmm::UIDs, 1180

uid\_1\_2\_840\_10008\_5\_1\_4\_20\_3  
gdcmm::UIDs, 1180

uid\_1\_2\_840\_10008\_5\_1\_4\_31  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_32  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_32\_1  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_32\_2  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_32\_3  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_33  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_34\_1  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_34\_10  
gdcmm::UIDs, 1180

uid\_1\_2\_840\_10008\_5\_1\_4\_34\_2  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_34\_3  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_1  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_2  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_3  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_4  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_34\_5  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_34\_5\_1  
gdcmm::UIDs, 1180

uid\_1\_2\_840\_10008\_5\_1\_4\_34\_6  
gdcmm::UIDs, 1180

uid\_1\_2\_840\_10008\_5\_1\_4\_34\_6\_1  
gdcmm::UIDs, 1180

uid\_1\_2\_840\_10008\_5\_1\_4\_34\_6\_2  
gdcmm::UIDs, 1180

uid\_1\_2\_840\_10008\_5\_1\_4\_34\_6\_3  
gdcmm::UIDs, 1180

uid\_1\_2\_840\_10008\_5\_1\_4\_34\_6\_4  
gdcmm::UIDs, 1180

uid\_1\_2\_840\_10008\_5\_1\_4\_34\_7  
gdcmm::UIDs, 1180

uid\_1\_2\_840\_10008\_5\_1\_4\_34\_8  
gdcmm::UIDs, 1180

uid\_1\_2\_840\_10008\_5\_1\_4\_34\_9  
gdcmm::UIDs, 1180

uid\_1\_2\_840\_10008\_5\_1\_4\_37\_1  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_37\_2  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_37\_3  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_38\_1  
gdcmm::UIDs, 1176

uid\_1\_2\_840\_10008\_5\_1\_4\_38\_2  
gdcmm::UIDs, 1176



- uid\_1\_2\_840\_10008\_5\_1\_4\_38\_3
  - gdcm::UIDs, [1176](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_38\_4
  - gdcm::UIDs, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_39\_1
  - gdcm::UIDs, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_39\_2
  - gdcm::UIDs, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_39\_3
  - gdcm::UIDs, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_39\_4
  - gdcm::UIDs, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_41
  - gdcm::UIDs, [1176](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_42
  - gdcm::UIDs, [1176](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_43\_1
  - gdcm::UIDs, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_43\_2
  - gdcm::UIDs, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_43\_3
  - gdcm::UIDs, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_43\_4
  - gdcm::UIDs, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_44\_1
  - gdcm::UIDs, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_44\_2
  - gdcm::UIDs, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_44\_3
  - gdcm::UIDs, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_44\_4
  - gdcm::UIDs, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_45\_1
  - gdcm::UIDs, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_45\_2
  - gdcm::UIDs, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_45\_3
  - gdcm::UIDs, [1180](#)
- uid\_1\_2\_840\_10008\_5\_1\_4\_45\_4
  - gdcm::UIDs, [1180](#)
- uid\_1\_2\_840\_10008\_7\_1\_1
  - gdcm::UIDs, [1180](#)
- uid\_1\_2\_840\_10008\_7\_1\_2
  - gdcm::UIDs, [1180](#)
- uid\_1\_2\_840\_10008\_8\_1\_1
  - gdcm::UIDs, [1180](#)
- UIDGenerator
  - gdcm::UIDGenerator, [1143](#)
- UINT12
  - gdcm::PixelFormat, [808](#)
- UINT16
  - gdcm::PixelFormat, [808](#)
- UINT32
  - gdcm::PixelFormat, [808](#)
- UINT64
  - gdcm::PixelFormat, [808](#)
- UINT8
  - gdcm::PixelFormat, [808](#)
- UL
  - gdcm::VR, [1285](#)
- ULAction
  - gdcm::network::ULAction, [1184](#), [1185](#)
- ULActionAE6
  - gdcm::network::ULConnection, [1231](#)
- ULBasicCallback
  - gdcm::network::ULBasicCallback, [1224](#)
- ULConnection
  - gdcm::network::ULConnection, [1227](#)
- ULConnectionCallback
  - gdcm::network::ULConnectionCallback, [1232](#)
- ULConnectionInfo
  - gdcm::network::ULConnectionInfo, [1234](#)
- ULConnectionManager
  - gdcm::network::ULConnection, [1231](#)
  - gdcm::network::ULConnectionManager, [1238](#)
- ULEvent
  - gdcm::network::ULEvent, [1244](#), [1245](#)
- ULTransitionTable
  - gdcm::network::ULTransitionTable, [1247](#)
- UltrasoundImageStorage
  - gdcm::MediaStorage, [676](#)
  - gdcm::UIDs, [1165](#)
- UltrasoundImageStorageRetired
  - gdcm::MediaStorage, [676](#)
  - gdcm::UIDs, [1165](#)
- UltrasoundMultiFrameImageStorage
  - gdcm::MediaStorage, [676](#)
- UltrasoundMultiframeImageStorage
  - gdcm::UIDs, [1164](#)
- UltrasoundMultiFrameImageStorageRetired
  - gdcm::MediaStorage, [676](#)
- UltrasoundMultiframeImageStorageRetired
  - gdcm::UIDs, [1164](#)
- ULWritingCallback
  - gdcm::network::ULWritingCallback, [1249](#)
- UN
  - gdcm::VR, [1285](#)
- UndefinedEntityError
  - gdcm::Parser, [778](#)
- underline
  - gdcm::terminal, [85](#)
- UnexpectedStateError
  - gdcm::Parser, [778](#)
- UnifiedProcedureStepEventSOPClass
  - gdcm::UIDs, [1167](#)
- UnifiedProcedureStepEventSOPClass1
  - gdcm::UIDs, [1171](#)
- UnifiedProcedureStepPullSOPClass

- gdcm::UIDs, [1167](#)
- UnifiedProcedureStepPullSOPClass1
  - gdcm::UIDs, [1170](#)
- UnifiedProcedureStepPushSOPClass
  - gdcm::UIDs, [1167](#)
- UnifiedProcedureStepPushSOPClass1
  - gdcm::UIDs, [1170](#)
- UnifiedProcedureStepWatchSOPClass
  - gdcm::UIDs, [1167](#)
- UnifiedProcedureStepWatchSOPClass1
  - gdcm::UIDs, [1170](#)
- UnifiedWorklistandProcedureStepServiceClass
  - gdcm::UIDs, [1167](#)
- UnifiedWorklistandProcedureStepServiceClass1
  - gdcm::UIDs, [1170](#)
- UnifiedWorklistandProcedureStepSOPInstance
  - gdcm::UIDs, [1167](#)
- UnInstallPipeline
  - vtkImageColorViewer, [1373](#)
- UniversalCoordinatedTime
  - gdcm::UIDs, [1171](#)
- UNKNOWN
  - gdcm::CSAHeader, [288](#)
  - gdcm::EquipmentManufacturer, [431](#)
  - gdcm::LookupTable, [657](#)
  - gdcm::Orientation, [761](#)
  - gdcm::PhotometricInterpretation, [804](#)
  - gdcm::PixelFormat, [808](#)
  - gdcm::Spacing, [1014](#)
  - gdcm::Surface, [1058](#)
  - gdcm::Type, [1140](#)
- Unknown
  - gdcm::SwapCode, [1081](#)
  - gdcm::TransferSyntax, [1130](#)
- Unpack
  - gdcm::Unpacker12Bits, [1256](#)
- UnRegister
  - gdcm::Object, [749](#)
- UnusedBitsPresentInPixelData
  - gdcm::Bitmap, [207](#)
  - gdcm::Pixmap, [820](#)
- Update
  - gdcm::Curve, [308](#)
  - gdcm::Overlay, [773](#)
- UpdateDisplayExtent
  - vtkImageColorViewer, [1373](#)
- UpdateOrientation
  - vtkImageColorViewer, [1373](#)
- UpdatePosition
  - gdcm::ByteBuffer, [218](#)
- UPSFilteredGlobalSubscriptionSOPInstance
  - gdcm::UIDs, [1170](#)
- UR
  - gdcm::VR, [1285](#)
- URComp
  - gdcm, [60](#)
- URI
  - gdcm::MediaStorage, [679](#)
- US
  - gdcm::VR, [1285](#)
- US\_OW
  - gdcm::VR, [1285](#)
- US\_SS
  - gdcm::VR, [1285](#)
- US\_SS\_OW
  - gdcm::VR, [1285](#)
- Usage
  - gdcm::Usage, [1258](#)
- UsageType
  - gdcm::Usage, [1257](#)
- UseDictAlways
  - gdcm::PythonFilter, [879](#)
  - gdcm::StringFilter, [1050](#)
- UseGrayscaleSecondaryImageStorage
  - gdcm::EmptyMaskGenerator, [423](#)
- UseOriginalSOPClassUID
  - gdcm::EmptyMaskGenerator, [423](#)
- UserInformation
  - gdcm::network::UserInformation, [1260](#)
- UserOption
  - gdcm::Usage, [1257](#)
- UserOrdering
  - gdcm::SerieHelper, [974](#)
- UT
  - gdcm::VR, [1285](#)
- UTComp
  - gdcm, [60](#)
- UV
  - gdcm::VR, [1285](#)
- V
  - gdcm::Validate, [1265](#)
- Valid
  - gdcm::Preamble, [840](#)
- Validate
  - gdcm::PixelFormat, [814](#)
  - gdcm::Validate, [1264](#)
- ValidateQuery
  - gdcm::BaseQuery, [181](#)
  - gdcm::BaseRootQuery, [186](#)
  - gdcm::FindPatientRootQuery, [500](#)
  - gdcm::FindStudyRootQuery, [503](#)
  - gdcm::ModalityPerformedProcedureStepCreateQuery, [697](#)
  - gdcm::ModalityPerformedProcedureStepSetQuery, [700](#)
  - gdcm::MovePatientRootQuery, [714](#)
  - gdcm::MoveStudyRootQuery, [717](#)

- gdcm::WLMFindQuery, [1413](#)
- Validation
  - gdcm::Validate, [1265](#)
- ValidDataSet
  - gdcm::BaseQuery, [181](#)
- Value
  - gdcm::Value, [1267](#)
- value
  - gdcm::SerieHelper, [975](#)
  - gdcm::STATIC\_ASSERTION\_FAILURE< true >, [1020](#)
- value\_type
  - gdcm::CodeString, [254](#)
  - gdcm::LO, [653](#)
  - gdcm::String< TDelimiter, TMaxLength, TPadChar >, [1044](#)
- ValueField
  - gdcm::DataElement, [322](#)
  - gdcm::PDBelement, [787](#)
- ValueLengthField
  - gdcm::DataElement, [322](#)
- ValueMultiplicityField
  - gdcm::CSAElement, [286](#)
- ValuePtr
  - gdcm::DataElement, [312](#)
- ValueType
  - gdcm::Scanner, [926](#)
  - gdcm::StrictScanner, [1035](#)
- VERBOSE\_STYLE
  - gdcm::Printer, [861](#)
- VerificationSOPClass
  - gdcm::UIDs, [1162](#)
- Verify
  - gdcm::Defs, [351](#)
  - gdcm::Macro, [666](#)
  - gdcm::Module, [704](#)
- Version
  - gdcm::Version, [1271](#)
- VERTEX
  - gdcm::MeshPrimitive, [692](#)
- Video
  - gdcm::MediaStorage, [679](#)
- VideoEndoscopicImageStorage
  - gdcm::MediaStorage, [678](#)
  - gdcm::UIDs, [1165](#)
- VideoMicroscopicImageStorage
  - gdcm::UIDs, [1165](#)
- VideoPhotographicImageStorage
  - gdcm::MediaStorage, [678](#)
  - gdcm::UIDs, [1166](#)
- VIEWType
  - gdcm::Surface, [1058](#)
- VIEWType\_END
  - gdcm::Surface, [1058](#)
- VisualAcuityMeasurementsStorage
  - gdcm::UIDs, [1169](#)
- VL
  - gdcm::VL, [1274](#)
- VL16
  - gdcm::VR, [1285](#)
- VL32
  - gdcm::VR, [1285](#)
- VLEndoscopicImageStorage
  - gdcm::MediaStorage, [678](#)
  - gdcm::UIDs, [1165](#)
- VLIImageStorageTrialRetired
  - gdcm::UIDs, [1165](#)
- VLMicroscopicImageStorage
  - gdcm::MediaStorage, [678](#)
  - gdcm::UIDs, [1165](#)
- VLMultiframeImageStorageTrialRetired
  - gdcm::UIDs, [1165](#)
- VLPhotographicImageStorage
  - gdcm::MediaStorage, [678](#)
  - gdcm::UIDs, [1166](#)
- VLSlideCoordinatesMicroscopicImageStorage
  - gdcm::UIDs, [1166](#)
- VLWholeSlideMicroscopyImageStorage
  - gdcm::MediaStorage, [678](#)
  - gdcm::UIDs, [1168](#)
- VM
  - gdcm::VM, [1280](#)
- VM0
  - gdcm::VM, [1279](#)
- VM1
  - gdcm::VM, [1279](#)
- VM10
  - gdcm::VM, [1279](#)
- VM12
  - gdcm::VM, [1279](#)
- VM16
  - gdcm::VM, [1279](#)
- VM18
  - gdcm::VM, [1279](#)
- VM1\_2
  - gdcm::VM, [1279](#)
- VM1\_3
  - gdcm::VM, [1279](#)
- VM1\_32
  - gdcm::VM, [1279](#)
- VM1\_4
  - gdcm::VM, [1279](#)
- VM1\_5
  - gdcm::VM, [1279](#)
- VM1\_8
  - gdcm::VM, [1279](#)
- VM1\_99
  - gdcm::VM, [1279](#)

- VM1\_n
  - gdcm::VM, [1279](#)
- VM2
  - gdcm::VM, [1279](#)
- VM24
  - gdcm::VM, [1279](#)
- VM256
  - gdcm::VM, [1279](#)
- VM28
  - gdcm::VM, [1279](#)
- VM2\_2n
  - gdcm::VM, [1279](#)
- VM2\_n
  - gdcm::VM, [1279](#)
- VM3
  - gdcm::VM, [1279](#)
- VM30\_30n
  - gdcm::VM, [1279](#)
- VM32
  - gdcm::VM, [1279](#)
- VM35
  - gdcm::VM, [1279](#)
- VM3\_3n
  - gdcm::VM, [1279](#)
- VM3\_4
  - gdcm::VM, [1279](#)
- VM3\_n
  - gdcm::VM, [1279](#)
- VM4
  - gdcm::VM, [1279](#)
- VM47\_47n
  - gdcm::VM, [1279](#)
- VM4\_4n
  - gdcm::VM, [1279](#)
- VM5
  - gdcm::VM, [1279](#)
- VM6
  - gdcm::VM, [1279](#)
- VM6\_6n
  - gdcm::VM, [1279](#)
- VM6\_n
  - gdcm::VM, [1279](#)
- VM7\_7n
  - gdcm::VM, [1279](#)
- VM8
  - gdcm::VM, [1279](#)
- VM9
  - gdcm::VM, [1279](#)
- VM99
  - gdcm::VM, [1279](#)
- VM\_END
  - gdcm::VM, [1279](#)
- VMType
  - gdcm::Attribute< Group, Element, TVR, TVM >, [137](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [146](#)
  - gdcm::VM, [1279](#)
- VOILUTBoxSOPClass
  - gdcm::UIDs, [1164](#)
- VolumeRenderingVolumetricPresentationStateStorage
  - gdcm::UIDs, [1169](#)
- VR
  - gdcm::VR, [1285](#)
- VR\_END
  - gdcm::VR, [1285](#)
- VR\_VM1
  - gdcm::VR, [1285](#)
- VRALL
  - gdcm::VR, [1285](#)
- VRASCI
  - gdcm::VR, [1285](#)
- VRBINARY
  - gdcm, [76](#)
  - gdcm::VR, [1285](#)
- VRField
  - gdcm::CSAElement, [286](#)
  - gdcm::DataElement, [322](#)
- VRType
  - gdcm::VR, [1284](#)
- VRTypeTemplateCase
  - gdcmVR.h, [1703](#)
- VT100
  - gdcm::terminal, [86](#)
- VTK\_CMYK
  - vtkGDCMImageReader.h, [1710](#)
  - vtkGDCMImageReader2.h, [1711](#)
- VTK\_INVERSE\_LUMINANCE
  - vtkGDCMImageReader.h, [1710](#)
  - vtkGDCMImageReader2.h, [1711](#)
- VTK\_LEGACY
  - vtkImageColorViewer, [1373](#), [1374](#)
- VTK\_LOOKUP\_TABLE
  - vtkGDCMImageReader.h, [1710](#)
  - vtkGDCMImageReader2.h, [1712](#)
- VTK\_YBR
  - vtkGDCMImageReader.h, [1710](#)
  - vtkGDCMImageReader2.h, [1712](#)
- vtkBooleanMacro
  - vtkGDCMImageReader, [1300](#), [1301](#)
  - vtkGDCMImageReader2, [1315](#), [1316](#)
  - vtkGDCMImageWriter, [1328](#)
  - vtkGDCMThreadedImageReader, [1353](#)
  - vtkGDCMThreadedImageReader2, [1358](#)
  - vtkImageColorViewer, [1374](#)
  - vtkImageMapToColors16, [1381](#)
- vtkGDCMImageReader, [1294](#)
- ~vtkGDCMImageReader, [1297](#)
- ApplyInverseVideo, [1305](#)

- ApplyLookupTable, [1306](#)
- ApplyPlanarConfiguration, [1306](#)
- ApplyShiftScale, [1306](#)
- ApplyYBRToRGB, [1306](#)
- CanReadFile, [1297](#)
- Curve, [1306](#)
- DirectionCosines, [1306](#)
- ExecuteData, [1297](#)
- ExecuteInformation, [1298](#)
- FileNames, [1306](#)
- FillMedicalImageInformation, [1298](#)
- ForceRescale, [1307](#)
- GetDescriptiveName, [1298](#)
- GetFileExtensions, [1298](#)
- GetIconImage, [1298](#)
- GetOverlay, [1298](#)
- IconDataScalarType, [1307](#)
- IconImageDataExtent, [1307](#)
- IconNumberOfScalarComponents, [1307](#)
- ImageFormat, [1307](#)
- ImageOrientationPatient, [1307](#)
- ImagePositionPatient, [1307](#)
- LoadIconImage, [1308](#)
- LoadOverlays, [1308](#)
- LoadSingleFile, [1298](#)
- LossyFlag, [1308](#)
- MedicalImageProperties, [1308](#)
- New, [1299](#)
- NumberOfIconImages, [1308](#)
- NumberOfOverlays, [1308](#)
- PlanarConfiguration, [1308](#)
- PrintSelf, [1299](#)
- RequestDataCompat, [1299](#)
- RequestInformationCompat, [1299](#)
- Scale, [1309](#)
- SetCurve, [1299](#)
- SetFileNames, [1300](#)
- SetFilePattern, [1300](#)
- SetFilePrefix, [1300](#)
- SetMedicalImageProperties, [1300](#)
- Shift, [1309](#)
- vtkBooleanMacro, [1300](#), [1301](#)
- vtkGDCMImageReader, [1297](#)
- vtkGDCMMedicalImageProperties, [1335](#)
- vtkGetMacro, [1301–1303](#)
- vtkGetObjectMacro, [1303](#)
- vtkGetStringMacro, [1304](#)
- vtkGetVector3Macro, [1304](#)
- vtkGetVector6Macro, [1304](#)
- vtkSetMacro, [1304](#), [1305](#)
- vtkSetVector6Macro, [1305](#)
- vtkTypeMacro, [1305](#)
- vtkGDCMImageReader.h, [1709](#)
- VTK\_CMYK, [1710](#)
- VTK\_INVERSE\_LUMINANCE, [1710](#)
- VTK\_LOOKUP\_TABLE, [1710](#)
- VTK\_YBR, [1710](#)
- vtkGDCMImageReader2, [1309](#)
- ~vtkGDCMImageReader2, [1312](#)
- ApplyInverseVideo, [1320](#)
- ApplyLookupTable, [1320](#)
- ApplyPlanarConfiguration, [1320](#)
- ApplyShiftScale, [1321](#)
- ApplyYBRToRGB, [1321](#)
- CanReadFile, [1312](#)
- Curve, [1321](#)
- DirectionCosines, [1321](#)
- FillMedicalImageInformation, [1312](#)
- ForceRescale, [1321](#)
- GetDescriptiveName, [1312](#)
- GetFileExtensions, [1312](#)
- GetIconImage, [1313](#)
- GetIconImagePort, [1313](#)
- GetOverlay, [1313](#)
- GetOverlayPort, [1313](#)
- IconDataScalarType, [1321](#)
- IconImageDataExtent, [1321](#)
- IconNumberOfScalarComponents, [1322](#)
- ImageFormat, [1322](#)
- ImageOrientationPatient, [1322](#)
- ImagePositionPatient, [1322](#)
- LoadIconImage, [1322](#)
- LoadOverlays, [1322](#)
- LoadSingleFile, [1313](#)
- LossyFlag, [1322](#)
- New, [1313](#)
- NumberOfIconImages, [1323](#)
- NumberOfOverlays, [1323](#)
- PlanarConfiguration, [1323](#)
- PrintSelf, [1314](#)
- ProcessRequest, [1314](#)
- RequestData, [1314](#)
- RequestDataCompat, [1314](#)
- RequestInformation, [1314](#)
- RequestInformationCompat, [1315](#)
- Scale, [1323](#)
- SetCurve, [1315](#)
- SetFilePattern, [1315](#)
- SetFilePrefix, [1315](#)
- SetMedicalImageProperties, [1315](#)
- Shift, [1323](#)
- vtkBooleanMacro, [1315](#), [1316](#)
- vtkGDCMImageReader2, [1312](#)
- vtkGDCMMedicalImageProperties, [1336](#)
- vtkGetMacro, [1316–1318](#)
- vtkGetObjectMacro, [1318](#)
- vtkGetStringMacro, [1318](#), [1319](#)
- vtkGetVector3Macro, [1319](#)

- vtkGetVector6Macro, 1319
  - vtkSetMacro, 1319, 1320
  - vtkSetVector6Macro, 1320
  - vtkTypeMacro, 1320
- vtkGDCMImageReader2.h, 1711
  - VTK\_CMYK, 1711
  - VTK\_INVERSE\_LUMINANCE, 1711
  - VTK\_LOOKUP\_TABLE, 1712
  - VTK\_YBR, 1712
- vtkGDCMImageWriter, 1324
  - ~vtkGDCMImageWriter, 1326
  - CompressionTypes, 1326
  - GetDescriptiveName, 1326
  - GetFileExtensions, 1327
  - GetFileName, 1327
  - JPEG2000\_COMPRESSION, 1326
  - JPEG\_COMPRESSION, 1326
  - JPEGLS\_COMPRESSION, 1326
  - New, 1327
  - NO\_COMPRESSION, 1326
  - PrintSelf, 1327
  - RLE\_COMPRESSION, 1326
  - SetDirectionCosines, 1327
  - SetDirectionCosinesFromImageOrientationPatient, 1327
  - SetFileNames, 1328
  - SetMedicalImageProperties, 1328
  - vtkBooleanMacro, 1328
  - vtkGDCMImageWriter, 1326
  - vtkGDCMMedicalImageProperties, 1336
  - vtkGetMacro, 1328, 1329
  - vtkGetObjectMacro, 1330
  - vtkGetStringMacro, 1330
  - vtkSetMacro, 1330, 1331
  - vtkSetStringMacro, 1332
  - vtkTypeMacro, 1332
  - Write, 1332
  - WriteGDCMData, 1332
  - WriteSlice, 1333
- vtkGDCMImageWriter.h, 1712
- vtkGDCMMedicalImageProperties, 1333
  - ~vtkGDCMMedicalImageProperties, 1334
  - Clear, 1334
  - GetFile, 1335
  - New, 1335
  - PrintSelf, 1335
  - PushBackFile, 1335
  - vtkGDCMImageReader, 1335
  - vtkGDCMImageReader2, 1336
  - vtkGDCMImageWriter, 1336
  - vtkGDCMMedicalImageProperties, 1334
  - vtkTypeMacro, 1335
- vtkGDCMMedicalImageProperties.h, 1713
- vtkGDCMPolyDataReader, 1336
  - ~vtkGDCMPolyDataReader, 1338
  - FileName, 1341
  - FillMedicalImageInformation, 1338
  - MedicalImageProperties, 1341
  - New, 1338
  - PrintSelf, 1338
  - RequestData, 1339
  - RequestData\_HemodynamicWaveformStorage, 1339
  - RequestData\_RTStructureSetStorage, 1339
  - RequestInformation, 1339
  - RequestInformation\_HemodynamicWaveformStorage, 1339
  - RequestInformation\_RTStructureSetStorage, 1339
  - RTStructSetProperties, 1341
  - vtkGDCMPolyDataReader, 1338
  - vtkGetObjectMacro, 1340
  - vtkGetStringMacro, 1340
  - vtkSetStringMacro, 1340
  - vtkTypeMacro, 1340
- vtkGDCMPolyDataReader.h, 1713
- vtkGDCMPolyDataWriter, 1341
  - ~vtkGDCMPolyDataWriter, 1343
  - InitializeRTStructSet, 1343
  - MedicalImageProperties, 1345
  - New, 1343
  - PrintSelf, 1344
  - RTStructSetProperties, 1346
  - SetMedicalImageProperties, 1344
  - SetNumberOfInputPorts, 1344
  - SetRTStructSetProperties, 1344
  - vtkGDCMPolyDataWriter, 1343
  - vtkTypeMacro, 1345
  - WriteData, 1345
  - WriteRTSTRUCTData, 1345
  - WriteRTSTRUCTInfo, 1345
- vtkGDCMPolyDataWriter.h, 1714
- vtkGDCMTesting, 1346
  - ~vtkGDCMTesting, 1348
  - GetGDCMDataRoot, 1348
  - GetMD5MetaImage, 1348
  - GetMHDMD5FromFile, 1348
  - GetNumberOfMD5MetaImages, 1348
  - GetRAWMD5FromFile, 1349
  - GetVTKDataRoot, 1349
  - MD5MetaImagesType, 1347
  - New, 1349
  - PrintSelf, 1349
  - vtkGDCMTesting, 1347
  - vtkTypeMacro, 1349
- vtkGDCMTesting.h, 1715
- vtkGDCMThreadedImageReader, 1350
  - ~vtkGDCMThreadedImageReader, 1352
  - ExecuteData, 1352

- ExecuteInformation, [1352](#)
- New, [1352](#)
- PrintSelf, [1352](#)
- ReadFiles, [1352](#)
- RequestDataCompat, [1353](#)
- vtkBooleanMacro, [1353](#)
- vtkGDCMThreadedImageReader, [1351](#)
- vtkGetMacro, [1353](#)
- vtkSetMacro, [1353](#)
- vtkTypeMacro, [1354](#)
- vtkGDCMThreadedImageReader.h, [1715](#)
- vtkGDCMThreadedImageReader2, [1354](#)
  - ~vtkGDCMThreadedImageReader2, [1356](#)
  - GetFileName, [1356](#)
  - New, [1356](#)
  - PrintSelf, [1356](#)
  - RequestInformation, [1357](#)
  - SetFileName, [1357](#)
  - SetFileNames, [1357](#)
  - SplitExtent, [1357](#)
  - ThreadedRequestData, [1357](#)
  - vtkBooleanMacro, [1358](#)
  - vtkGDCMThreadedImageReader2, [1356](#)
  - vtkGetMacro, [1358](#), [1359](#)
  - vtkGetObjectMacro, [1360](#)
  - vtkGetVector3Macro, [1360](#)
  - vtkGetVector6Macro, [1360](#)
  - vtkSetMacro, [1360](#), [1361](#)
  - vtkSetVector3Macro, [1361](#), [1362](#)
  - vtkSetVector6Macro, [1362](#)
  - vtkTypeMacro, [1362](#)
- vtkGDCMThreadedImageReader2.h, [1716](#)
- vtkGetMacro
  - vtkGDCMImageReader, [1301–1303](#)
  - vtkGDCMImageReader2, [1316–1318](#)
  - vtkGDCMImageWriter, [1328](#), [1329](#)
  - vtkGDCMThreadedImageReader, [1353](#)
  - vtkGDCMThreadedImageReader2, [1358](#), [1359](#)
  - vtkImageColorViewer, [1374](#)
  - vtkImageMapToColors16, [1382](#)
  - vtkImageMapToWindowLevelColors2, [1386](#), [1387](#)
- vtkGetObjectMacro
  - vtkGDCMImageReader, [1303](#)
  - vtkGDCMImageReader2, [1318](#)
  - vtkGDCMImageWriter, [1330](#)
  - vtkGDCMPolyDataReader, [1340](#)
  - vtkGDCMThreadedImageReader2, [1360](#)
  - vtkImageColorViewer, [1374](#), [1375](#)
  - vtkImageMapToColors16, [1382](#)
- vtkGetStringMacro
  - vtkGDCMImageReader, [1304](#)
  - vtkGDCMImageReader2, [1318](#), [1319](#)
  - vtkGDCMImageWriter, [1330](#)
  - vtkGDCMPolyDataReader, [1340](#)
  - vtkRTStructSetProperties, [1406](#), [1407](#)
- vtkGetVector3Macro
  - vtkGDCMImageReader, [1304](#)
  - vtkGDCMImageReader2, [1319](#)
  - vtkGDCMThreadedImageReader2, [1360](#)
- vtkGetVector6Macro
  - vtkGDCMImageReader, [1304](#)
  - vtkGDCMImageReader2, [1319](#)
  - vtkGDCMThreadedImageReader2, [1360](#)
- vtkImageColorViewer, [1363](#)
  - ~vtkImageColorViewer, [1366](#)
  - AddInput, [1366](#)
  - AddInputConnection, [1366](#)
  - FirstRender, [1376](#)
  - GetColorLevel, [1366](#)
  - GetColorWindow, [1367](#)
  - GetInput, [1367](#)
  - GetOffScreenRendering, [1367](#)
  - GetOverlayVisibility, [1367](#)
  - GetPosition, [1367](#)
  - GetSize, [1367](#)
  - GetSliceMax, [1367](#)
  - GetSliceMin, [1368](#)
  - GetSliceRange, [1368](#)
  - GetWindowName, [1368](#)
  - ImageActor, [1376](#)
  - InstallPipeline, [1368](#)
  - Interactor, [1376](#)
  - InteractorStyle, [1376](#)
  - New, [1368](#)
  - OverlayImageActor, [1376](#)
  - PrintSelf, [1369](#)
  - Render, [1369](#)
  - Renderer, [1376](#)
  - RenderWindow, [1377](#)
  - SetColorLevel, [1369](#)
  - SetColorWindow, [1369](#)
  - SetDisplayId, [1369](#)
  - SetInput, [1370](#)
  - SetInputConnection, [1370](#)
  - SetOffScreenRendering, [1370](#)
  - SetOverlayVisibility, [1370](#)
  - SetParentId, [1370](#)
  - SetPosition, [1370](#), [1371](#)
  - SetRenderer, [1371](#)
  - SetRenderWindow, [1371](#)
  - SetSize, [1371](#)
  - SetSlice, [1372](#)
  - SetSliceOrientation, [1372](#)
  - SetSliceOrientationToXY, [1372](#)
  - SetSliceOrientationToXZ, [1372](#)
  - SetSliceOrientationToYZ, [1372](#)
  - SetupInteractor, [1372](#)
  - SetWindowId, [1373](#)



- Slice, [1377](#)
- SLICE\_ORIENTATION\_XY, [1366](#)
- SLICE\_ORIENTATION\_XZ, [1366](#)
- SLICE\_ORIENTATION\_YZ, [1366](#)
- SliceOrientation, [1377](#)
- UnInstallPipeline, [1373](#)
- UpdateDisplayExtent, [1373](#)
- UpdateOrientation, [1373](#)
- VTK\_LEGACY, [1373](#), [1374](#)
- vtkBooleanMacro, [1374](#)
- vtkGetMacro, [1374](#)
- vtkGetObjectMacro, [1374](#), [1375](#)
- vtkImageColorViewer, [1366](#)
- vtkImageColorViewerCallback, [1376](#)
- vtkTypeMacro, [1375](#)
- WindowLevel, [1377](#)
- vtkImageColorViewer.h, [1716](#)
- vtkImageColorViewerCallback
  - vtkImageColorViewer, [1376](#)
- vtkImageMapToColors16, [1378](#)
  - ~vtkImageMapToColors16, [1379](#)
  - ActiveComponent, [1383](#)
  - DataWasPassed, [1383](#)
  - GetMTime, [1380](#)
  - LookupTable, [1383](#)
  - New, [1380](#)
  - OutputFormat, [1383](#)
  - PassAlphaToOutput, [1384](#)
  - PrintSelf, [1380](#)
  - RequestData, [1380](#)
  - RequestInformation, [1380](#)
  - SetLookupTable, [1380](#)
  - SetOutputFormatToLuminance, [1381](#)
  - SetOutputFormatToLuminanceAlpha, [1381](#)
  - SetOutputFormatToRGB, [1381](#)
  - SetOutputFormatToRGBA, [1381](#)
  - ThreadedRequestData, [1381](#)
  - vtkBooleanMacro, [1381](#)
  - vtkGetMacro, [1382](#)
  - vtkGetObjectMacro, [1382](#)
  - vtkImageMapToColors16, [1379](#)
  - vtkSetMacro, [1382](#), [1383](#)
  - vtkTypeMacro, [1383](#)
- vtkImageMapToColors16.h, [1717](#)
- vtkImageMapToWindowLevelColors2, [1384](#)
  - ~vtkImageMapToWindowLevelColors2, [1385](#)
  - Level, [1387](#)
  - New, [1386](#)
  - PrintSelf, [1386](#)
  - RequestData, [1386](#)
  - RequestInformation, [1386](#)
  - ThreadedRequestData, [1386](#)
  - vtkGetMacro, [1386](#), [1387](#)
  - vtkImageMapToWindowLevelColors2, [1385](#)
- vtkSetMacro, [1387](#)
- vtkTypeMacro, [1387](#)
- Window, [1388](#)
- vtkImageMapToWindowLevelColors2.h, [1717](#)
- vtkImagePlanarComponentsToComponents, [1388](#)
  - ~vtkImagePlanarComponentsToComponents, [1389](#)
  - New, [1389](#)
  - PrintSelf, [1389](#)
  - RequestData, [1390](#)
  - vtkImagePlanarComponentsToComponents, [1389](#)
  - vtkTypeMacro, [1390](#)
- vtkImagePlanarComponentsToComponents.h, [1718](#)
- vtkImageRGBToYBR, [1390](#)
  - ~vtkImageRGBToYBR, [1391](#)
  - New, [1392](#)
  - PrintSelf, [1392](#)
  - ThreadedExecute, [1392](#)
  - vtkImageRGBToYBR, [1391](#)
  - vtkTypeMacro, [1392](#)
- vtkImageRGBToYBR.h, [1718](#)
- vtkImageYBRToRGB, [1393](#)
  - ~vtkImageYBRToRGB, [1394](#)
  - New, [1394](#)
  - PrintSelf, [1394](#)
  - ThreadedExecute, [1394](#)
  - vtkImageYBRToRGB, [1394](#)
  - vtkTypeMacro, [1395](#)
- vtkImageYBRToRGB.h, [1719](#)
- vtkLookupTable16, [1395](#)
  - ~vtkLookupTable16, [1397](#)
  - Build, [1397](#)
  - GetPointer, [1397](#)
  - MapScalarsThroughTable2, [1397](#)
  - New, [1397](#)
  - PrintSelf, [1398](#)
  - SetNumberOfTableValues, [1398](#)
  - Table16, [1398](#)
  - vtkLookupTable16, [1396](#)
  - vtkTypeMacro, [1398](#)
  - WritePointer, [1398](#)
- vtkLookupTable16.h, [1719](#)
- vtkRTStructSetProperties, [1399](#)
  - ~vtkRTStructSetProperties, [1401](#)
  - AddContourReferencedFrameOfReference, [1401](#)
  - AddReferencedFrameOfReference, [1402](#)
  - AddStructureSetROI, [1402](#)
  - AddStructureSetROIObservation, [1402](#)
  - Clear, [1402](#)
  - DeepCopy, [1402](#)
  - GetContourReferencedFrameOfReferenceClassUID, [1403](#)
  - GetContourReferencedFrameOfReferenceInstanceUID, [1403](#)



- GetNumberOfContourReferencedFrameOfReferences, [1403](#)
- GetNumberOfReferencedFrameOfReferences, [1403](#)
- GetNumberOfStructureSetROIs, [1403](#)
- GetReferencedFrameOfReferenceClassUID, [1404](#)
- GetReferencedFrameOfReferenceInstanceUID, [1404](#)
- GetStructureSetObservationNumber, [1404](#)
- GetStructureSetROIDescription, [1404](#)
- GetStructureSetROIGenerationAlgorithm, [1404](#)
- GetStructureSetROIName, [1404](#)
- GetStructureSetROINumber, [1405](#)
- GetStructureSetROIObservationLabel, [1405](#)
- GetStructureSetROIRefFrameRefUID, [1405](#)
- GetStructureSetRTROIInterpretedType, [1405](#)
- Internals, [1409](#)
- New, [1405](#)
- PrintSelf, [1405](#)
- ReferenceFrameOfReferenceUID, [1409](#)
- ReferenceSeriesInstanceUID, [1409](#)
- SeriesInstanceUID, [1409](#)
- SOPInstanceUID, [1409](#)
- StructureSetDate, [1410](#)
- StructureSetLabel, [1410](#)
- StructureSetName, [1410](#)
- StructureSetTime, [1410](#)
- StudyInstanceUID, [1410](#)
- vtkGetStringMacro, [1406](#), [1407](#)
- vtkRTStructSetProperties, [1401](#)
- vtkSetStringMacro, [1407](#), [1408](#)
- vtkTypeMacro, [1409](#)
- vtkRTStructSetProperties.h, [1720](#)
- vtkSetMacro
  - vtkGDCMImageReader, [1304](#), [1305](#)
  - vtkGDCMImageReader2, [1319](#), [1320](#)
  - vtkGDCMImageWriter, [1330](#), [1331](#)
  - vtkGDCMThreadedImageReader, [1353](#)
  - vtkGDCMThreadedImageReader2, [1360](#), [1361](#)
  - vtkImageMapToColors16, [1382](#), [1383](#)
  - vtkImageMapToWindowLevelColors2, [1387](#)
- vtkSetStringMacro
  - vtkGDCMImageWriter, [1332](#)
  - vtkGDCMPolyDataReader, [1340](#)
  - vtkRTStructSetProperties, [1407](#), [1408](#)
- vtkSetVector3Macro
  - vtkGDCMThreadedImageReader2, [1361](#), [1362](#)
- vtkSetVector6Macro
  - vtkGDCMImageReader, [1305](#)
  - vtkGDCMImageReader2, [1320](#)
  - vtkGDCMThreadedImageReader2, [1362](#)
- vtkTypeMacro
  - vtkGDCMImageReader, [1305](#)
  - vtkGDCMImageReader2, [1320](#)
  - vtkGDCMImageWriter, [1332](#)
  - vtkGDCMMedicalImageProperties, [1335](#)
  - vtkGDCMPolyDataReader, [1340](#)
  - vtkGDCMPolyDataWriter, [1345](#)
  - vtkGDCMTesting, [1349](#)
  - vtkGDCMThreadedImageReader, [1354](#)
  - vtkGDCMThreadedImageReader2, [1362](#)
  - vtkImageColorViewer, [1375](#)
  - vtkImageMapToColors16, [1383](#)
  - vtkImageMapToWindowLevelColors2, [1387](#)
  - vtkImagePlanarComponentsToComponents, [1390](#)
  - vtkImageRGBToYBR, [1392](#)
  - vtkImageYBRToRGB, [1395](#)
  - vtkLookupTable16, [1398](#)
  - vtkRTStructSetProperties, [1409](#)
- WarningOff
  - gdcm::Trace, [1128](#)
- WarningOn
  - gdcm::Trace, [1128](#)
- Waveform
  - gdcm::MediaStorage, [679](#)
  - gdcm::Waveform, [1411](#)
- WaveformStorageTrialRetired
  - gdcm::UIDs, [1165](#)
- WeirdPapryus
  - gdcm::TransferSyntax, [1131](#)
- what
  - gdcm::Exception, [437](#)
- white
  - gdcm::terminal, [85](#)
- WideFieldOphthalmicPhotography3DCoordinatesImageStorage
  - gdcm::UIDs, [1169](#)
- WideFieldOphthalmicPhotographyStereographicProjectionImageStorage
  - gdcm::UIDs, [1169](#)
- Window
  - vtkImageMapToWindowLevelColors2, [1388](#)
- WindowLevel
  - vtkImageColorViewer, [1377](#)
- WinterColorPaletteSOPInstance
  - gdcm::UIDs, [1168](#)
- WIREFRAME
  - gdcm::Surface, [1058](#)
- WLMFindQuery
  - gdcm::WLMFindQuery, [1412](#)
- Write
  - gdcm::ByteValue, [231](#)
  - gdcm::CommandDataSet, [262](#)
  - gdcm::DataElement, [321](#)
  - gdcm::DataSet, [339](#)
  - gdcm::Element< TVR, TVM >, [400](#)
  - gdcm::Element< TVR, VM::VM1\_n >, [407](#)
  - gdcm::EncodingImplementation< VR::VRASCII >, [427](#)
  - gdcm::EncodingImplementation< VR::VRBINARY >, [428](#)

- gdcm::ExplicitDataElement, [441](#)
- gdcm::File, [449](#)
- gdcm::FileAnonymizer, [454](#)
- gdcm::FileMetaInformation, [476](#)
- gdcm::Fragment, [507](#)
- gdcm::ImageWriter, [585](#)
- gdcm::ImplicitDataElement, [592](#)
- gdcm::Item, [611](#)
- gdcm::network::AAAbortPDU, [89](#)
- gdcm::network::AAAssociateACPDU, [93](#)
- gdcm::network::AAAssociateRJPDU, [96](#)
- gdcm::network::AAAssociateRQPDU, [102](#)
- gdcm::network::AbstractSyntax, [106](#)
- gdcm::network::ApplicationContext, [121](#)
- gdcm::network::AReleaseRPPDU, [126](#)
- gdcm::network::AReleaseRQPDU, [128](#)
- gdcm::network::AsynchronousOperationsWindowSub, [134](#)
- gdcm::network::BasePDU, [177](#)
- gdcm::network::ImplementationClassUIDSub, [587](#)
- gdcm::network::ImplementationUIDSub, [588](#)
- gdcm::network::ImplementationVersionNameSub, [589](#)
- gdcm::network::MaximumLengthSub, [671](#)
- gdcm::network::PDataTFPDU, [784](#)
- gdcm::network::PresentationContextAC, [847](#)
- gdcm::network::PresentationContextRQ, [855](#)
- gdcm::network::PresentationDataValue, [859](#)
- gdcm::network::RoleSelectionSub, [922](#)
- gdcm::network::ServiceClassApplicationInformation, [977](#)
- gdcm::network::SOPClassExtendedNegociationSub, [1004](#)
- gdcm::network::TransferSyntaxSub, [1137](#)
- gdcm::network::UserInformation, [1262](#)
- gdcm::PGXCodec, [802](#)
- gdcm::PixmapWriter, [832](#)
- gdcm::PNMCodec, [836](#)
- gdcm::Preamble, [840](#)
- gdcm::SegmentWriter, [952](#)
- gdcm::SequenceOfFragments, [959](#)
- gdcm::SequenceOfItems, [968](#)
- gdcm::StreamImageWriter, [1028](#)
- gdcm::SurfaceWriter, [1079](#)
- gdcm::Tag, [1111](#)
- gdcm::ValueIO< TDE, TSwap, TType >, [1269](#)
- gdcm::VL, [1276](#)
- gdcm::VR, [1289](#)
- gdcm::VRVLSIZE< 0 >, [1293](#)
- gdcm::VRVLSIZE< 1 >, [1294](#)
- gdcm::Writer, [1419](#)
- vtkGDCMImageWriter, [1332](#)
- Write16
  - gdcm::VL, [1276](#)
- WriteASCII
  - gdcm::Element< TVR, VM::VM1\_n >, [408](#)
- WriteBuffer
  - gdcm::ByteValue, [231](#)
  - gdcm::SequenceOfFragments, [960](#)
- WriteBufferAsRGBA
  - gdcm::LookupTable, [662](#)
- WriteData
  - vtkGDCMPolyDataWriter, [1345](#)
- WriteFooter
  - gdcm::DictConverter, [368](#)
- WriteGDCMData
  - vtkGDCMImageWriter, [1332](#)
- WriteHeader
  - gdcm::DictConverter, [368](#)
- WriteHelpFile
  - gdcm::BaseQuery, [182](#)
- WriteImageInformation
  - gdcm::StreamImageWriter, [1029](#)
- WriteImageSubregionRAW
  - gdcm::StreamImageWriter, [1029](#)
- WritePointer
  - vtkLookupTable16, [1398](#)
- WriteQuery
  - gdcm::BaseQuery, [182](#)
- Writer
  - gdcm::Writer, [1417](#)
- WriteRawHeader
  - gdcm::StreamImageWriter, [1029](#)
- WriteRTSTRUCTData
  - vtkGDCMPolyDataWriter, [1345](#)
- WriteRTSTRUCTInfo
  - vtkGDCMPolyDataWriter, [1345](#)
- WriteSlice
  - vtkGDCMImageWriter, [1333](#)
- x16printf
  - gdcm, [76](#)
- XAXRFGreyscaleSoftcopyPresentationStateStorage
  - gdcm::UIDs, [1169](#)
- XML
  - gdcm::Printer, [861](#)
- XMLDictReader
  - gdcm::XMLDictReader, [1421](#)
- XMLEncoding
  - gdcm::UIDs, [1162](#)
- XMLPrinter
  - gdcm::XMLPrinter, [1425](#)
- XMLPrivateDictReader
  - gdcm::XMLPrivateDictReader, [1428](#)
- XRay3DAngiographicImageStorage
  - gdcm::MediaStorage, [678](#)
  - gdcm::UIDs, [1165](#)
- XRay3DCraniofacialImageStorage

- gdcm::MediaStorage, [678](#)
- gdcm::UIDs, [1165](#)
- XRayAngiographicBiPlaneImageStorageRetired
  - gdcm::MediaStorage, [677](#)
  - gdcm::UIDs, [1165](#)
- XRayAngiographicImageStorage
  - gdcm::MediaStorage, [677](#)
  - gdcm::UIDs, [1165](#)
- XRayRadiationDoseSR
  - gdcm::MediaStorage, [678](#)
- XRayRadiationDoseSRStorage
  - gdcm::UIDs, [1166](#)
- XRayRadiofluoroscopicImageStorage
  - gdcm::UIDs, [1165](#)
- XRayRadiofluoroscopicImageStorage
  - gdcm::MediaStorage, [677](#)
- YBR2RGB
  - gdcm::ImageChangePhotometricInterpretation, [536](#)
- YBR\_FULL
  - gdcm::PhotometricInterpretation, [804](#)
- YBR\_FULL\_422
  - gdcm::PhotometricInterpretation, [804](#)
- YBR\_ICT
  - gdcm::PhotometricInterpretation, [804](#)
- YBR\_PARTIAL\_420
  - gdcm::PhotometricInterpretation, [804](#)
- YBR\_PARTIAL\_422
  - gdcm::PhotometricInterpretation, [804](#)
- YBR\_RCT
  - gdcm::PhotometricInterpretation, [804](#)
- yellow
  - gdcm::terminal, [85](#)
- YES
  - gdcm::Surface, [1058](#)
- ZEROED\_OUT
  - gdcm::CSAHeader, [288](#)
- ZSpacing
  - gdcm::IPPSorter, [607](#)
- ZTolerance
  - gdcm::IPPSorter, [607](#)