

GDCM

2.6.1

Generated by Doxygen 1.8.9.1

Mon Dec 21 2015 23:27:54

Contents

1	GDCM Documentation	1
2	off-screen rendering of DICOM images	3
2.1	SYNOPSIS	3
2.2	DESCRIPTION	3
2.3	PARAMETERS	3
2.4	OPTIONS	3
2.4.1	OPTIONS	3
2.4.2	general options	3
2.5	Simple usage	4
2.6	SEE ALSO	4
2.7	COPYRIGHT	4
3	Convert a file supported by VTK into DICOM.	5
3.1	SYNOPSIS	5
3.2	DESCRIPTION	5
3.3	PARAMETERS	5
3.4	OPTIONS	5
3.4.1	OPTIONS	5
3.4.2	compression options	6
3.4.3	general options	6
3.4.4	environment variable	6
3.5	DESCRIPTION	6
3.5.1	CONVERT Metalmage (mhd, mha)	6
3.5.2	CONVERT MHA/MHD	7
3.5.3	CONVERT VTI	7
3.5.4	CONVERT VTK	7
3.6	CONVERT DICOM	7
3.7	RoundTrip DICOM to MHD to DICOM	7

3.8	gdcm2vtk notes	7
3.9	SEE ALSO	8
3.10	COPYRIGHT	8
4	Tool to anonymize a DICOM file.	9
4.1	SYNOPSIS	9
4.2	DESCRIPTION	9
4.3	PARAMETERS	9
4.4	OPTIONS	10
4.4.1	Required parameters	10
4.4.2	OPTIONS	10
4.4.3	encryption options	10
4.4.4	dumb mode options	10
4.4.5	general options	10
4.4.6	environment variable	11
4.5	Typical usage	11
4.5.1	De-identification (anonymization, encrypt)	11
4.5.2	Re-identification (de-anonymization,decrypt)	11
4.5.3	Multiple files caveat	11
4.5.4	Dumb mode	11
4.5.4.1	Irreversible Anonymization	12
4.6	OpenSSL	12
4.6.1	Generating a Private Key	12
4.6.2	Generating a Certificate	13
4.7	DICOM Standard:	13
4.8	Warnings	13
4.9	SEE ALSO	13
4.10	COPYRIGHT	13
5	Tool to convert DICOM to DICOM.	15
5.1	SYNOPSIS	15
5.2	DESCRIPTION	15
5.3	PARAMETERS	15
5.4	OPTIONS	15
5.4.1	PARAMETERS	15
5.4.2	OPTIONS	15
5.4.3	image options	16
5.4.4	JPEG options	16

5.4.5	JPEG-LS options	16
5.4.6	J2K options	16
5.4.7	general options	16
5.4.8	special options	16
5.4.9	environment variable	17
5.5	Simple usage	17
5.6	Typical usage	17
5.6.1	File Meta Header	17
5.6.2	Conversion to Explicit Transfer Syntax	18
5.6.3	Compressing to lossless JPEG	18
5.6.4	Compressing to lossy JPEG	18
5.6.5	Compressing to lossless JPEG-LS	18
5.6.6	Compressing to lossy JPEG-LS	18
5.6.7	Compressing to lossless J2K	18
5.6.8	Compressing to lossy J2K	18
5.6.9	Compressing to lossless RLE	19
5.6.10	Split encapsulated DICOM:	19
5.6.11	Forcing (re)compression	19
5.6.12	Decompressing a Compressed DICOM	19
5.6.13	Compressing an uncompressed Icon	19
5.6.14	Generating an Icon	20
5.6.15	Changing the planar Configuration	20
5.7	Lossless Conversion	20
5.8	Quality Control	20
5.8.1	DCMTK / dicom3tools	20
5.8.2	VIM: vimdiff	21
5.8.3	vbindiff	21
5.9	SEE ALSO	21
5.10	COPYRIGHT	21
6	dumps differences of two DICOM files	23
6.1	SYNOPSIS	23
6.2	DESCRIPTION	23
6.3	PARAMETERS	23
6.4	OPTIONS	23
6.4.1	OPTIONS	23
6.4.2	general options	23

6.5	Simple usage	24
6.6	SEE ALSO	24
6.7	COPYRIGHT	24
7	dumps a DICOM file, it will display the structure and values contained in the specified DICOM file.	25
7.1	SYNOPSIS	25
7.2	DESCRIPTION	25
7.3	PARAMETERS	25
7.4	OPTIONS	25
7.4.1	OPTIONS	25
7.4.2	general options	26
7.4.3	special options	26
7.5	Typical usage	26
7.5.1	Printing Implicit Transfer Syntax	26
7.5.2	Print Private Attributes	27
7.5.3	SIEMENS CSA Header	27
7.5.4	GEMS Protocol Data Block	27
7.5.5	ELSCINT Protocol Information	28
7.5.6	VEPRO Protocol Information	28
7.5.7	Philips Private MR Series Data Storage (1.3.46.670589.11.0.0.12.2)	29
7.5.8	Encapsulated ASN1 Structure	30
7.6	SEE ALSO	31
7.7	COPYRIGHT	31
8	Tool to generate a DICOMDIR file from a File-Set.	33
8.1	SYNOPSIS	33
8.2	DESCRIPTION	33
8.3	PARAMETERS	33
8.4	OPTIONS	33
8.4.1	Parameters	33
8.4.2	OPTIONS	33
8.4.3	general options	33
8.4.4	environment variable	34
8.5	Typical usage	34
8.6	NOTE	34
8.7	SEE ALSO	34
8.8	COPYRIGHT	34

9	Manipulate DICOM image file.	35
9.1	SYNOPSIS	35
9.2	DESCRIPTION	35
9.3	PARAMETERS	35
9.4	OPTIONS	35
9.4.1	PARAMETERS	35
9.4.2	OPTIONS	36
9.4.3	fill options	36
9.4.4	general options	36
9.4.5	environment variable	36
9.5	Supported File Format (appropriate file extension) <code>gdcmimg</code>	36
9.6	Typical usage	37
9.6.1	Remove a rectangular part of the image	37
9.6.2	Convert RAW to DICOM	37
9.6.3	Convert PGM/PNM/PPM to DICOM	38
9.6.4	Convert RLE to DICOM	38
9.6.5	Convert JPEG to DICOM	38
9.6.6	Convert J2K to DICOM	38
9.6.7	Specifying a SOP Class UID	38
9.7	Multiple Files	38
9.8	Start Offset	39
9.9	Warning	39
9.10	SEE ALSO	39
9.11	COPYRIGHT	39
10	Display meta info about the input DICOM file.	41
10.1	SYNOPSIS	41
10.2	DESCRIPTION	41
10.3	PARAMETERS	41
10.4	OPTIONS	41
10.4.1	OPTIONS	41
10.4.2	general options	41
10.4.3	environment variable	42
10.5	Simple usage	42
10.5.1	<code>gdcmData</code>	42
10.5.2	Davie Clunie datasets:	42
10.5.3	Checking the md5sum of the Pixel Data	43

10.5.4 Checking if Pixel Data is lossless	43
10.6 SEE ALSO	43
10.7 COPYRIGHT	43
11 Tool to convert PYPYRUS 3.0 to DICOM.	45
11.1 SYNOPSIS	45
11.2 DESCRIPTION	45
11.3 PARAMETERS	45
11.4 OPTIONS	45
11.4.1 PARAMETERS	45
11.4.2 OPTIONS	45
11.4.3 general options	45
11.4.4 environment variable	46
11.5 Simple usage	46
11.6 SEE ALSO	46
11.7 COPYRIGHT	46
12 Tool to convert PDF to PDF/DICOM.	47
12.1 SYNOPSIS	47
12.2 DESCRIPTION	47
12.3 PARAMETERS	47
12.4 OPTIONS	47
12.4.1 general options	47
12.5 Usage Example	48
12.6 PDF Info Mapping	48
12.7 SEE ALSO	49
12.8 COPYRIGHT	49
13 Extract Data Element Value Field.	51
13.1 SYNOPSIS	51
13.2 DESCRIPTION	51
13.3 PARAMETERS	51
13.4 OPTIONS	51
13.4.1 PARAMETERS	51
13.4.2 OPTIONS	51
13.4.3 general options	51
13.5 Typical usage	52
13.5.1 Copy Attribute Value to file	52

13.5.2 Extract Pixel Data	52
13.5.3 Encapsulated Syntax	52
13.5.4 Extract fragments as single file	53
13.6 Footnote about JPEG files	54
13.7 SEE ALSO	54
13.8 COPYRIGHT	54
14 Scan a directory containing DICOM files.	55
14.1 SYNOPSIS	55
14.2 DESCRIPTION	55
14.2.1 PARAMETERS	55
14.2.2 OPTIONS	55
14.2.3 general options	55
14.3 Typical usage	56
14.4 Simple usage	56
14.5 Complex usage	56
14.6 SEE ALSO	56
14.7 COPYRIGHT	56
15 Tool to execute a DICOM Query/Retrieve operation	57
15.1 SYNOPSIS	57
15.2 DESCRIPTION	57
15.3 PARAMETERS	57
15.4 OPTIONS	57
15.4.1 OPTIONS	57
15.4.2 mode options	57
15.4.3 C-STORE options	58
15.4.4 C-FIND/C-MOVE options	58
15.4.5 C-MOVE options	58
15.4.6 general options	58
15.4.7 environment variable	58
15.5 C-ECHO usage	59
15.6 C-STORE usage	59
15.7 C-FIND usage	59
15.8 C-MOVE usage	60
15.9 patientroot notes	60
15.10 Debugging	60
15.11 Port Warning	60

15.12C-STORE Warnings	61
15.13C-MOVE Warnings	61
15.14C-FIND IMAGE level (Composite Object Instance)	61
15.15Storing the Query	61
15.16DICOM Public Servers	62
15.17SEE ALSO	62
15.18COPYRIGHT	62
16 Concatenate/Extract DICOM files.	63
16.1 SYNOPSIS	63
16.2 DESCRIPTION	63
16.3 PARAMETERS	63
16.4 OPTIONS	63
16.4.1 OPTIONS	63
16.4.2 general options	63
16.4.3 environment variable	64
16.5 Typical usage	64
16.5.1 SIEMENS Mosaic	64
16.6 SEE ALSO	65
16.7 COPYRIGHT	65
17 Simple DICOM viewer.	67
17.1 SYNOPSIS	67
17.2 DESCRIPTION	67
17.3 PARAMETERS	67
17.4 OPTIONS	67
17.4.1 OPTIONS	67
17.4.2 general options	67
17.5 Typical usage	68
17.6 Simple usage	68
17.7 Wiki Link	68
17.8 SEE ALSO	68
17.9 COPYRIGHT	68
18 provides a tool to convert a DICOM file into a XML infoset and vice-versa.	69
18.1 SYNOPSIS	69
18.2 DESCRIPTION	69
18.3 PARAMETERS	69

18.4 OPTIONS	69
18.4.1 PARAMETERS	69
18.4.2 Options for DICOM to XML:	69
18.4.3 Options for XML to DICOM:	70
18.4.4 general options	70
18.5 SEE ALSO	70
18.6 COPYRIGHT	70
19 Todo List	71
20 Deprecated List	73
21 Bug List	75
22 Namespace Index	77
22.1 Namespace List	77
23 Hierarchical Index	79
23.1 Class Hierarchy	79
24 Class Index	89
24.1 Class List	89
25 File Index	105
25.1 File List	105
26 Namespace Documentation	113
26.1 gdcm Namespace Reference	113
26.1.1 Detailed Description	128
26.1.2 Typedef Documentation	128
26.1.2.1 AEComp	128
26.1.2.2 ASComp	128
26.1.2.3 BOOL_FUNCTION_PFILE_PFILE_POINTER	128
26.1.2.4 CSComp	128
26.1.2.5 DComp	129
26.1.2.6 DTComp	129
26.1.2.7 FileList	129
26.1.2.8 IconImage	129
26.1.2.9 LOComp	129
26.1.2.10 LTComp	129

26.1.2.11 MacroEntry	129
26.1.2.12 NestedMacroEntries	129
26.1.2.13 PNComp	129
26.1.2.14 SHComp	129
26.1.2.15 STComp	129
26.1.2.16 TMComp	129
26.1.2.17 UIComp	129
26.1.2.18 UTComp	129
26.1.3 Enumeration Type Documentation	129
26.1.3.1 CompOperators	129
26.1.3.2 ECharSet	129
26.1.3.3 ENQueryType	130
26.1.3.4 EQueryLevel	130
26.1.3.5 EQueryType	130
26.1.3.6 ERootType	130
26.1.3.7 LodModeType	131
26.1.4 Function Documentation	131
26.1.4.1 backslash	131
26.1.4.2 GetVRFromTag	131
26.1.4.3 operator"! =	131
26.1.4.4 operator"! =	131
26.1.4.5 operator<<	131
26.1.4.6 operator<<	131
26.1.4.7 operator<<	131
26.1.4.8 operator<<	131
26.1.4.9 operator<<	131
26.1.4.10 operator<<	131
26.1.4.11 operator<<	131
26.1.4.12 operator<<	132
26.1.4.13 operator<<	132
26.1.4.14 operator<<	132
26.1.4.15 operator<<	132
26.1.4.16 operator<<	132
26.1.4.17 operator<<	132
26.1.4.18 operator<<	132
26.1.4.19 operator<<	132
26.1.4.20 operator<<	132

26.1.4.21 operator<<	132
26.1.4.22 operator<<	132
26.1.4.23 operator<<	132
26.1.4.24 operator<<	132
26.1.4.25 operator<<	132
26.1.4.26 operator<<	132
26.1.4.27 operator<<	133
26.1.4.28 operator<<	133
26.1.4.29 operator<<	133
26.1.4.30 operator<<	133
26.1.4.31 operator<<	133
26.1.4.32 operator<<	133
26.1.4.33 operator<<	133
26.1.4.34 operator<<	133
26.1.4.35 operator<<	133
26.1.4.36 operator<<	133
26.1.4.37 operator<<	133
26.1.4.38 operator<<	133
26.1.4.39 operator<<	133
26.1.4.40 operator<<	133
26.1.4.41 operator<<	133
26.1.4.42 operator<<	133
26.1.4.43 operator<<	134
26.1.4.44 operator<<	134
26.1.4.45 operator<<	134
26.1.4.46 operator<<	134
26.1.4.47 operator<<	134
26.1.4.48 operator<<	134
26.1.4.49 operator<<	134
26.1.4.50 operator<<	134
26.1.4.51 operator<<	134
26.1.4.52 operator<<	134
26.1.4.53 operator<<	134
26.1.4.54 operator<<	134
26.1.4.55 operator<<	135
26.1.4.56 operator<<	135
26.1.4.57 operator<<	135

26.1.4.58 operator<<	135
26.1.4.59 operator<<	135
26.1.4.60 operator==	135
26.1.4.61 operator>>	135
26.1.4.62 operator>>	135
26.1.4.63 operator>>	135
26.1.4.64 to_string	135
26.1.4.65 TYPETOENCODING	135
26.1.5 Variable Documentation	135
26.1.5.1 GlobalInstance	135
26.1.5.2 VRINARY	135
26.2 gdcm::network Namespace Reference	136
26.2.1 Enumeration Type Documentation	140
26.2.1.1 EEventID	140
26.2.1.2 EStateID	141
26.2.2 Function Documentation	141
26.2.2.1 GetStateIndex	141
26.2.3 Variable Documentation	141
26.2.3.1 cMaxEventID	141
26.2.3.2 cMaxStateID	141
26.3 gdcm::SegmentHelper Namespace Reference	142
26.4 gdcm::terminal Namespace Reference	142
26.4.1 Detailed Description	142
26.4.2 Enumeration Type Documentation	143
26.4.2.1 Attribute	143
26.4.2.2 Color	143
26.4.2.3 Mode	143
26.4.3 Function Documentation	143
26.4.3.1 setattribute	143
26.4.3.2 setbgcolor	143
26.4.3.3 setfgcolor	143
26.4.3.4 setmode	143
27 Class Documentation	145
27.1 gdcm::network::AAbortPDU Class Reference	145
27.1.1 Detailed Description	146
27.1.2 Constructor & Destructor Documentation	146

27.1.2.1	AAbortPDU	146
27.1.3	Member Function Documentation	146
27.1.3.1	IsLastFragment	146
27.1.3.2	Print	146
27.1.3.3	Read	146
27.1.3.4	SetReason	147
27.1.3.5	SetSource	147
27.1.3.6	Size	147
27.1.3.7	Write	147
27.2	gdcmm::network::AAssociateACPDU Class Reference	147
27.2.1	Detailed Description	148
27.2.2	Member Typedef Documentation	149
27.2.2.1	SizeType	149
27.2.3	Constructor & Destructor Documentation	149
27.2.3.1	AAssociateACPDU	149
27.2.4	Member Function Documentation	149
27.2.4.1	AddPresentationContextAC	149
27.2.4.2	GetNumberOfPresentationContextAC	149
27.2.4.3	GetPresentationContextAC	149
27.2.4.4	GetUserInformation	149
27.2.4.5	InitFromRQ	149
27.2.4.6	IsLastFragment	149
27.2.4.7	Print	149
27.2.4.8	Read	149
27.2.4.9	SetCalledAETitle	149
27.2.4.10	SetCallingAETitle	149
27.2.4.11	Size	149
27.2.4.12	Write	149
27.2.5	Friends And Related Function Documentation	149
27.2.5.1	AAssociateRQPDU	150
27.3	gdcmm::network::AAssociateRJPDU Class Reference	150
27.3.1	Detailed Description	151
27.3.2	Constructor & Destructor Documentation	151
27.3.2.1	AAssociateRJPDU	151
27.3.3	Member Function Documentation	151
27.3.3.1	IsLastFragment	151
27.3.3.2	Print	151

27.3.3.3	Read	151
27.3.3.4	Size	151
27.3.3.5	Write	151
27.4	gdcm::network::AAssociateRQPDU Class Reference	151
27.4.1	Detailed Description	153
27.4.2	Member Typedef Documentation	153
27.4.2.1	PresentationContextArrayType	153
27.4.2.2	SizeType	153
27.4.3	Constructor & Destructor Documentation	153
27.4.3.1	AAssociateRQPDU	153
27.4.3.2	AAssociateRQPDU	153
27.4.4	Member Function Documentation	153
27.4.4.1	AddPresentationContext	153
27.4.4.2	GetCalledAETitle	154
27.4.4.3	GetCallingAETitle	154
27.4.4.4	GetNumberOfPresentationContext	154
27.4.4.5	GetPresentationContext	154
27.4.4.6	GetPresentationContextByAbstractSyntax	154
27.4.4.7	GetPresentationContextByID	154
27.4.4.8	GetPresentationContexts	154
27.4.4.9	GetReserved43_74	154
27.4.4.10	GetUserInformation	154
27.4.4.11	IsAETitleValid	154
27.4.4.12	IsLastFragment	154
27.4.4.13	Print	154
27.4.4.14	Read	154
27.4.4.15	SetCalledAETitle	154
27.4.4.16	SetCallingAETitle	154
27.4.4.17	SetUserInformation	155
27.4.4.18	Size	155
27.4.4.19	Write	155
27.4.5	Friends And Related Function Documentation	155
27.4.5.1	AAssociateACPDU	155
27.5	gdcm::AbortEvent Class Reference	155
27.6	gdcm::network::AbstractSyntax Class Reference	156
27.6.1	Detailed Description	156
27.6.2	Constructor & Destructor Documentation	157

27.6.2.1	AbstractSyntax	157
27.6.3	Member Function Documentation	157
27.6.3.1	GetAsDataElement	157
27.6.3.2	GetName	157
27.6.3.3	operator==	157
27.6.3.4	Print	157
27.6.3.5	Read	157
27.6.3.6	SetName	157
27.6.3.7	SetNameFromUID	157
27.6.3.8	Size	157
27.6.3.9	Write	157
27.7	gdcm::AnonymizeEvent Class Reference	157
27.7.1	Detailed Description	159
27.7.2	Member Typedef Documentation	159
27.7.2.1	Self	159
27.7.2.2	Superclass	159
27.7.3	Constructor & Destructor Documentation	159
27.7.3.1	AnonymizeEvent	159
27.7.3.2	~AnonymizeEvent	159
27.7.3.3	AnonymizeEvent	159
27.7.4	Member Function Documentation	159
27.7.4.1	CheckEvent	159
27.7.4.2	GetEventName	159
27.7.4.3	GetTag	159
27.7.4.4	MakeObject	159
27.7.4.5	SetTag	159
27.8	gdcm::Anonymizer Class Reference	160
27.8.1	Detailed Description	161
27.8.2	Constructor & Destructor Documentation	162
27.8.2.1	Anonymizer	162
27.8.2.2	~Anonymizer	162
27.8.3	Member Function Documentation	162
27.8.3.1	BALCPPProtect	162
27.8.3.2	BasicApplicationLevelConfidentialityProfile	162
27.8.3.3	CanEmptyTag	163
27.8.3.4	ClearInternalUIDs	163
27.8.3.5	Empty	163

27.8.3.6	GetBasicApplicationLevelConfidentialityProfileAttributes	163
27.8.3.7	GetCryptographicMessageSyntax	163
27.8.3.8	GetFile	163
27.8.3.9	New	163
27.8.3.10	RecurseDataSet	163
27.8.3.11	Remove	163
27.8.3.12	RemoveGroupLength	163
27.8.3.13	RemovePrivateTags	164
27.8.3.14	RemoveRetired	164
27.8.3.15	Replace	164
27.8.3.16	Replace	164
27.8.3.17	SetCryptographicMessageSyntax	164
27.8.3.18	SetFile	164
27.9	gdcm::AnyEvent Class Reference	164
27.10	gdcm::network::ApplicationContext Class Reference	166
27.10.1	Detailed Description	166
27.10.2	Constructor & Destructor Documentation	166
27.10.2.1	ApplicationContext	166
27.10.3	Member Function Documentation	167
27.10.3.1	GetName	167
27.10.3.2	Print	167
27.10.3.3	Read	167
27.10.3.4	SetName	167
27.10.3.5	Size	167
27.10.3.6	Write	167
27.11	gdcm::ApplicationEntity Class Reference	167
27.11.1	Detailed Description	168
27.11.2	Member Function Documentation	168
27.11.2.1	IsValid	168
27.11.2.2	Print	168
27.11.2.3	SetBlob	168
27.11.2.4	Squeeze	168
27.11.3	Member Data Documentation	168
27.11.3.1	Internal	168
27.11.3.2	MaxLength	168
27.11.3.3	MaxNumberOfComponents	168
27.11.3.4	Padding	168

27.11.3.5 Separator	168
27.12gdcmm::network::AReleaseRPPDU Class Reference	169
27.12.1 Detailed Description	170
27.12.2 Constructor & Destructor Documentation	170
27.12.2.1 AReleaseRPPDU	170
27.12.3 Member Function Documentation	170
27.12.3.1 IsLastFragment	170
27.12.3.2 Print	170
27.12.3.3 Read	170
27.12.3.4 Size	170
27.12.3.5 Write	170
27.13gdcmm::network::AReleaseRQPDU Class Reference	170
27.13.1 Detailed Description	171
27.13.2 Constructor & Destructor Documentation	171
27.13.2.1 AReleaseRQPDU	171
27.13.3 Member Function Documentation	172
27.13.3.1 IsLastFragment	172
27.13.3.2 Print	172
27.13.3.3 Read	172
27.13.3.4 Size	172
27.13.3.5 Write	172
27.14gdcmm::network::ARTIMTimer Class Reference	172
27.14.1 Detailed Description	172
27.14.2 Constructor & Destructor Documentation	173
27.14.2.1 ARTIMTimer	173
27.14.3 Member Function Documentation	173
27.14.3.1 GetElapsedTime	173
27.14.3.2 GetHasExpired	173
27.14.3.3 GetTimeout	173
27.14.3.4 SetTimeout	173
27.14.3.5 Start	173
27.14.3.6 Stop	173
27.15gdcmm::ASN1 Class Reference	173
27.15.1 Detailed Description	174
27.15.2 Constructor & Destructor Documentation	174
27.15.2.1 ASN1	174
27.15.2.2 ~ASN1	174

27.15.3 Member Function Documentation	174
27.15.3.1 ParseDump	174
27.15.3.2 ParseDumpFile	174
27.15.3.3 TestPBKDF2	174
27.16gdcm::network::AsynchronousOperationsWindowSub Class Reference	174
27.16.1 Detailed Description	174
27.16.2 Constructor & Destructor Documentation	174
27.16.2.1 AsynchronousOperationsWindowSub	174
27.16.3 Member Function Documentation	174
27.16.3.1 Print	175
27.16.3.2 Read	175
27.16.3.3 Size	175
27.16.3.4 Write	175
27.17gdcm::Attribute< Group, Element, TVR, TVM > Class Template Reference	175
27.17.1 Detailed Description	176
27.17.2 Member Typedef Documentation	177
27.17.2.1 ArrayType	177
27.17.3 Member Enumeration Documentation	177
27.17.3.1 anonymous enum	177
27.17.4 Member Function Documentation	177
27.17.4.1 GDCM_STATIC_ASSERT	177
27.17.4.2 GDCM_STATIC_ASSERT	177
27.17.4.3 GDCM_STATIC_ASSERT	177
27.17.4.4 GetAsDataElement	177
27.17.4.5 GetDictVM	177
27.17.4.6 GetDictVR	177
27.17.4.7 GetNumberOfValues	178
27.17.4.8 GetTag	178
27.17.4.9 GetValue	178
27.17.4.10GetValue	178
27.17.4.11GetValues	178
27.17.4.12GetVM	178
27.17.4.13GetVR	178
27.17.4.14operator!=	178
27.17.4.15operator<	178
27.17.4.16operator==	179
27.17.4.17operator[]	179

27.17.4.18operator[]	179
27.17.4.19Print	179
27.17.4.20Set	179
27.17.4.21SetByteValue	179
27.17.4.22SetByteValueNoSwap	179
27.17.4.23SetFromDataElement	179
27.17.4.24SetFromDataSet	179
27.17.4.25SetValue	180
27.17.4.26SetValues	180
27.17.5 Member Data Documentation	180
27.17.5.1 Internal	180
27.18gdcmm::Attribute< Group, Element, TVR, VM::VM1 > Class Template Reference	180
27.18.1 Member Typedef Documentation	182
27.18.1.1 ArrayType	182
27.18.2 Member Enumeration Documentation	182
27.18.2.1 anonymous enum	182
27.18.3 Member Function Documentation	182
27.18.3.1 GDCM_STATIC_ASSERT	182
27.18.3.2 GDCM_STATIC_ASSERT	182
27.18.3.3 GDCM_STATIC_ASSERT	182
27.18.3.4 GDCM_STATIC_ASSERT	182
27.18.3.5 GetAsDataElement	182
27.18.3.6 GetDictVM	182
27.18.3.7 GetDictVR	182
27.18.3.8 GetNumberOfValues	182
27.18.3.9 GetTag	182
27.18.3.10GetValue	182
27.18.3.11GetValue	182
27.18.3.12GetValues	183
27.18.3.13GetVM	183
27.18.3.14GetVR	183
27.18.3.15operator"!="	183
27.18.3.16operator<	183
27.18.3.17operator==	183
27.18.3.18Print	183
27.18.3.19Set	183
27.18.3.20SetByteValue	183

27.18.3.21SetByteValueNoSwap	183
27.18.3.22SetFromDataElement	183
27.18.3.23SetFromDataSet	184
27.18.3.24SetValue	184
27.18.4 Member Data Documentation	184
27.18.4.1 Internal	184
27.19gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 > Class Template Reference	184
27.19.1 Member Function Documentation	185
27.19.1.1 GetVM	185
27.20gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 > Class Template Reference	185
27.20.1 Member Function Documentation	186
27.20.1.1 GetVM	186
27.21gdcmm::Attribute< Group, Element, TVR, VM::VM1_n > Class Template Reference	187
27.21.1 Member Typedef Documentation	188
27.21.1.1 ArrayType	188
27.21.2 Constructor & Destructor Documentation	188
27.21.2.1 Attribute	188
27.21.2.2 ~Attribute	188
27.21.3 Member Function Documentation	188
27.21.3.1 GDCM_STATIC_ASSERT	188
27.21.3.2 GDCM_STATIC_ASSERT	188
27.21.3.3 GDCM_STATIC_ASSERT	188
27.21.3.4 GetAsDataElement	188
27.21.3.5 GetDictVM	188
27.21.3.6 GetDictVR	188
27.21.3.7 GetNumberOfValues	188
27.21.3.8 GetTag	189
27.21.3.9 GetValue	189
27.21.3.10GetValue	189
27.21.3.11GetValues	189
27.21.3.12GetVM	189
27.21.3.13GetVR	189
27.21.3.14operator[]	189
27.21.3.15operator[]	189
27.21.3.16Print	189
27.21.3.17Set	189
27.21.3.18SetByteValue	189

27.21.3.19SetFromDataElement	189
27.21.3.20SetFromDataSet	189
27.21.3.21SetNumberOfValues	190
27.21.3.22SetValue	190
27.21.3.23SetValue	190
27.21.3.24SetValues	190
27.22gdcmm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference	190
27.22.1 Member Function Documentation	191
27.22.1.1 GetVM	191
27.23gdcmm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference	191
27.23.1 Member Function Documentation	192
27.23.1.1 GetVM	193
27.24gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n > Class Template Reference	193
27.24.1 Member Function Documentation	194
27.24.1.1 GetVM	194
27.25gdcmm::Attribute< Group, Element, TVR, VM::VM3_n > Class Template Reference	194
27.25.1 Member Function Documentation	195
27.25.1.1 GetVM	196
27.26gdcmm::AudioCodec Class Reference	196
27.26.1 Detailed Description	197
27.26.2 Constructor & Destructor Documentation	197
27.26.2.1 AudioCodec	197
27.26.2.2 ~AudioCodec	197
27.26.3 Member Function Documentation	197
27.26.3.1 CanCode	197
27.26.3.2 CanDecode	198
27.26.3.3 Decode	198
27.27gdcmm::Base64 Class Reference	198
27.27.1 Detailed Description	198
27.27.2 Member Function Documentation	198
27.27.2.1 Decode	198
27.27.2.2 Encode	199
27.27.2.3 GetDecodeLength	199
27.27.2.4 GetEncodeLength	199
27.28gdcmm::network::BaseCompositeMessage Class Reference	199
27.28.1 Detailed Description	200
27.28.2 Constructor & Destructor Documentation	201

27.28.2.1 ~BaseCompositeMessage	201
27.28.3 Member Function Documentation	201
27.28.3.1 ConstructPDV	201
27.29gdcmm::network::BaseNormalizedMessage Class Reference	201
27.29.1 Detailed Description	202
27.29.2 Constructor & Destructor Documentation	203
27.29.2.1 ~BaseNormalizedMessage	203
27.29.3 Member Function Documentation	203
27.29.3.1 ConstructPDV	203
27.30gdcmm::network::BasePDU Class Reference	203
27.30.1 Detailed Description	204
27.30.2 Constructor & Destructor Documentation	205
27.30.2.1 ~BasePDU	205
27.30.3 Member Function Documentation	205
27.30.3.1 IsLastFragment	205
27.30.3.2 Print	205
27.30.3.3 Read	205
27.30.3.4 Size	205
27.30.3.5 Write	205
27.31gdcmm::BaseQuery Class Reference	205
27.31.1 Detailed Description	207
27.31.2 Constructor & Destructor Documentation	207
27.31.2.1 BaseQuery	207
27.31.2.2 ~BaseQuery	207
27.31.3 Member Function Documentation	207
27.31.3.1 AddQueryDataSet	207
27.31.3.2 GetAbstractSyntaxUID	207
27.31.3.3 GetQueryDataSet	207
27.31.3.4 GetQueryDataSet	208
27.31.3.5 GetSOPInstanceUID	208
27.31.3.6 Print	208
27.31.3.7 SetSearchParameter	208
27.31.3.8 SetSearchParameter	208
27.31.3.9 SetSearchParameter	208
27.31.3.10SetSOPInstanceUID	208
27.31.3.11ValidateQuery	208
27.31.3.12ValidDataSet	208

27.31.3.13WriteHelpFile	208
27.31.3.14WriteQuery	208
27.31.4 Friends And Related Function Documentation	208
27.31.4.1 QueryFactory	208
27.31.5 Member Data Documentation	208
27.31.5.1 mDataSet	208
27.31.5.2 mHelpDescription	208
27.31.5.3 mSopInstanceUID	208
27.32gdcmm::BaseRootQuery Class Reference	208
27.32.1 Detailed Description	210
27.32.2 Constructor & Destructor Documentation	210
27.32.2.1 BaseRootQuery	210
27.32.2.2 ~BaseRootQuery	210
27.32.3 Member Function Documentation	210
27.32.3.1 Construct	210
27.32.3.2 GetQueryLevelFromQueryRoot	210
27.32.3.3 GetQueryLevelFromString	210
27.32.3.4 GetQueryLevelString	210
27.32.3.5 GetTagListByLevel	210
27.32.3.6 InitializeDataSet	211
27.32.3.7 ValidateQuery	211
27.32.4 Friends And Related Function Documentation	211
27.32.4.1 QueryFactory	211
27.32.5 Member Data Documentation	211
27.32.5.1 mHelpDescription	211
27.32.5.2 mImage	211
27.32.5.3 mPatient	211
27.32.5.4 mRootType	211
27.32.5.5 mSeries	211
27.32.5.6 mStudy	211
27.33gdcmm::SegmentHelper::BasicCodedEntry Struct Reference	212
27.33.1 Detailed Description	213
27.33.2 Constructor & Destructor Documentation	213
27.33.2.1 BasicCodedEntry	213
27.33.2.2 BasicCodedEntry	213
27.33.2.3 BasicCodedEntry	213
27.33.3 Member Function Documentation	213

27.33.3.1 IsEmpty	213
27.33.4 Member Data Documentation	213
27.33.4.1 CM	213
27.33.4.2 CSD	213
27.33.4.3 CSV	214
27.33.4.4 CV	214
27.34gdcmm::BasicOffsetTable Class Reference	214
27.34.1 Detailed Description	215
27.34.2 Constructor & Destructor Documentation	215
27.34.2.1 BasicOffsetTable	215
27.34.3 Member Function Documentation	215
27.34.3.1 Read	216
27.34.4 Friends And Related Function Documentation	216
27.34.4.1 operator<<	216
27.35gdcmm::Bitmap Class Reference	216
27.35.1 Detailed Description	219
27.35.2 Member Typedef Documentation	219
27.35.2.1 LUTPtr	219
27.35.3 Constructor & Destructor Documentation	219
27.35.3.1 Bitmap	219
27.35.3.2 ~Bitmap	219
27.35.4 Member Function Documentation	219
27.35.4.1 AreOverlaysInPixelData	219
27.35.4.2 Clear	219
27.35.4.3 ComputeLossyFlag	219
27.35.4.4 GetBuffer	219
27.35.4.5 GetBuffer2	219
27.35.4.6 GetBufferLength	219
27.35.4.7 GetColumns	220
27.35.4.8 GetDataElement	220
27.35.4.9 GetDataElement	220
27.35.4.10GetDimension	220
27.35.4.11GetDimensions	220
27.35.4.12GetLUT	220
27.35.4.13GetLUT	220
27.35.4.14GetNeedByteSwap	220
27.35.4.15GetNumberOfDimensions	220

27.35.4.16GetPhotometricInterpretation	220
27.35.4.17GetPixelFormat	221
27.35.4.18GetPixelFormat	221
27.35.4.19GetPlanarConfiguration	221
27.35.4.20GetRows	221
27.35.4.21GetTransferSyntax	221
27.35.4.22IsEmpty	221
27.35.4.23IsLossy	221
27.35.4.24IsTransferSyntaxCompatible	221
27.35.4.25Print	221
27.35.4.26SetColumns	221
27.35.4.27SetDataElement	221
27.35.4.28SetDimension	222
27.35.4.29SetDimensions	222
27.35.4.30SetLossyFlag	222
27.35.4.31SetLUT	222
27.35.4.32SetNeedByteSwap	222
27.35.4.33SetNumberOfDimensions	222
27.35.4.34SetPhotometricInterpretation	222
27.35.4.35SetPixelFormat	222
27.35.4.36SetPlanarConfiguration	222
27.35.4.37SetRows	223
27.35.4.38SetTransferSyntax	223
27.35.4.39TryJPEG2000Codec	223
27.35.4.40TryJPEG2000Codec2	223
27.35.4.41TryJPEGCodec	223
27.35.4.42TryJPEGCodec2	223
27.35.4.43TryJPEGLSCodec	223
27.35.4.44TryKAKADUCodec	223
27.35.4.45TryPVRGCodec	223
27.35.4.46TryRAWCodec	223
27.35.4.47TryRLECodec	223
27.35.5 Friends And Related Function Documentation	223
27.35.5.1 ImageChangeTransferSyntax	223
27.35.5.2 PixmapReader	223
27.35.6 Member Data Documentation	223
27.35.6.1 Dimensions	223

27.35.6.2 LossyFlag	223
27.35.6.3 LUT	223
27.35.6.4 NeedByteSwap	223
27.35.6.5 NumberOfDimensions	223
27.35.6.6 PF	223
27.35.6.7 PI	223
27.35.6.8 PixelData	223
27.35.6.9 PlanarConfiguration	224
27.35.6.10TS	224
27.36gdcmm::BitmapToBitmapFilter Class Reference	224
27.36.1 Detailed Description	225
27.36.2 Constructor & Destructor Documentation	225
27.36.2.1 BitmapToBitmapFilter	225
27.36.2.2 ~BitmapToBitmapFilter	225
27.36.3 Member Function Documentation	225
27.36.3.1 GetOutput	225
27.36.3.2 GetOutputAsBitmap	225
27.36.3.3 SetInput	225
27.36.4 Member Data Documentation	225
27.36.4.1 Input	225
27.36.4.2 Output	225
27.37gdcmm::BoxRegion Class Reference	226
27.37.1 Detailed Description	227
27.37.2 Constructor & Destructor Documentation	227
27.37.2.1 BoxRegion	227
27.37.2.2 ~BoxRegion	227
27.37.2.3 BoxRegion	227
27.37.3 Member Function Documentation	227
27.37.3.1 Area	227
27.37.3.2 BoundingBox	228
27.37.3.3 Clone	228
27.37.3.4 ComputeBoundingBox	228
27.37.3.5 Empty	228
27.37.3.6 GetXMax	228
27.37.3.7 GetXMin	228
27.37.3.8 GetYMax	228
27.37.3.9 GetYMin	228

27.37.3.10GetZMax	228
27.37.3.11GetZMin	228
27.37.3.12IsValid	228
27.37.3.13operator=	228
27.37.3.14Print	228
27.37.3.15SetDomain	229
27.38gdcm::ByteBuffer Class Reference	229
27.38.1 Detailed Description	229
27.38.2 Constructor & Destructor Documentation	229
27.38.2.1 ByteBuffer	229
27.38.3 Member Function Documentation	229
27.38.3.1 Get	229
27.38.3.2 GetStart	229
27.38.3.3 ShiftEnd	229
27.38.3.4 UpdatePosition	229
27.39gdcm::ByteSwap< T > Class Template Reference	230
27.39.1 Detailed Description	230
27.39.2 Member Function Documentation	230
27.39.2.1 Swap	230
27.39.2.2 SwapFromSwapCodeIntoSystem	230
27.39.2.3 SwapRange	230
27.39.2.4 SwapRangeFromSwapCodeIntoSystem	230
27.39.2.5 SystemIsBigEndian	230
27.39.2.6 SystemIsLittleEndian	231
27.40gdcm::ByteSwapFilter Class Reference	231
27.40.1 Detailed Description	231
27.40.2 Constructor & Destructor Documentation	231
27.40.2.1 ByteSwapFilter	231
27.40.2.2 ~ByteSwapFilter	231
27.40.3 Member Function Documentation	231
27.40.3.1 ByteSwap	231
27.40.3.2 SetByteSwapTag	231
27.41gdcm::ByteValue Class Reference	231
27.41.1 Detailed Description	233
27.41.2 Constructor & Destructor Documentation	233
27.41.2.1 ByteValue	233
27.41.2.2 ByteValue	234

27.41.2.3 ~ByteValue	234
27.41.3 Member Function Documentation	234
27.41.3.1 Append	234
27.41.3.2 Clear	234
27.41.3.3 ComputeLength	234
27.41.3.4 Fill	234
27.41.3.5 GetBuffer	234
27.41.3.6 GetLength	234
27.41.3.7 GetPointer	235
27.41.3.8 IsEmpty	235
27.41.3.9 IsPrintable	235
27.41.3.10operator const std::vector< char > &	235
27.41.3.11operator=	235
27.41.3.12operator==	235
27.41.3.13operator==	235
27.41.3.14Print	235
27.41.3.15PrintASCII	235
27.41.3.16PrintASCIIXML	235
27.41.3.17PrintGroupLength	235
27.41.3.18PrintHex	235
27.41.3.19PrintHexXML	235
27.41.3.20PrintPNXML	235
27.41.3.21Read	236
27.41.3.22Read	236
27.41.3.23SetLength	236
27.41.3.24SetLengthOnly	236
27.41.3.25Write	236
27.41.3.26Write	236
27.41.3.27WriteBuffer	236
27.42gdcmm::CAPICryptoFactory Class Reference	236
27.42.1 Constructor & Destructor Documentation	237
27.42.1.1 CAPICryptoFactory	237
27.42.2 Member Function Documentation	237
27.42.2.1 CreateCMSProvider	237
27.43gdcmm::CAPICryptographicMessageSyntax Class Reference	237
27.43.1 Constructor & Destructor Documentation	239
27.43.1.1 CAPICryptographicMessageSyntax	239

27.43.1.2 ~CAPICryptographicMessageSyntax	239
27.43.2 Member Function Documentation	239
27.43.2.1 Decrypt	239
27.43.2.2 Encrypt	239
27.43.2.3 GetCipherType	239
27.43.2.4 GetInitialized	239
27.43.2.5 ParseCertificateFile	239
27.43.2.6 ParseKeyFile	239
27.43.2.7 SetCipherType	239
27.43.2.8 SetPassword	239
27.44gdcn::network::CEchoRQ Class Reference	240
27.44.1 Detailed Description	241
27.44.2 Member Function Documentation	241
27.44.2.1 ConstructPDV	241
27.44.3 Member Data Documentation	241
27.44.3.1 AffectedSOPClassUID	241
27.44.3.2 MessageID	241
27.45gdcn::network::CEchoRSP Class Reference	241
27.45.1 Detailed Description	242
27.45.2 Member Function Documentation	242
27.45.2.1 ConstructPDVByDataSet	242
27.46gdcn::network::CFind Class Reference	242
27.46.1 Detailed Description	242
27.47gdcn::network::CFindCancelRQ Class Reference	243
27.47.1 Detailed Description	243
27.47.2 Member Function Documentation	243
27.47.2.1 ConstructPDVByDataSet	244
27.48gdcn::network::CFindRQ Class Reference	244
27.48.1 Detailed Description	245
27.48.2 Member Function Documentation	245
27.48.2.1 ConstructPDV	245
27.49gdcn::network::CFindRSP Class Reference	245
27.49.1 Detailed Description	246
27.49.2 Member Function Documentation	246
27.49.2.1 ConstructPDVByDataSet	246
27.50gdcn::network::CMoveCancelRq Class Reference	246
27.50.1 Member Function Documentation	247

27.50.1.1 ConstructPDVByDataSet	247
27.51gdcmm::network::CMoveRQ Class Reference	248
27.51.1 Detailed Description	248
27.51.2 Member Function Documentation	248
27.51.2.1 ConstructPDV	249
27.52gdcmm::network::CMoveRSP Class Reference	249
27.52.1 Detailed Description	250
27.52.2 Member Function Documentation	250
27.52.2.1 ConstructPDVByDataSet	250
27.53gdcmm::Codec Class Reference	250
27.53.1 Detailed Description	251
27.54gdcmm::Coder Class Reference	251
27.54.1 Detailed Description	252
27.54.2 Constructor & Destructor Documentation	252
27.54.2.1 ~Coder	252
27.54.3 Member Function Documentation	252
27.54.3.1 CanCode	252
27.54.3.2 Code	253
27.54.3.3 InternalCode	253
27.55gdcmm::CodeString Class Reference	253
27.55.1 Detailed Description	254
27.55.2 Member Typedef Documentation	254
27.55.2.1 const_iterator	254
27.55.2.2 const_reference	254
27.55.2.3 const_reverse_iterator	254
27.55.2.4 difference_type	254
27.55.2.5 iterator	254
27.55.2.6 pointer	254
27.55.2.7 reference	254
27.55.2.8 reverse_iterator	254
27.55.2.9 size_type	254
27.55.2.10value_type	254
27.55.3 Constructor & Destructor Documentation	254
27.55.3.1 CodeString	255
27.55.3.2 CodeString	255
27.55.3.3 CodeString	255
27.55.3.4 CodeString	255

27.55.4 Member Function Documentation	255
27.55.4.1 GetAsString	255
27.55.4.2 IsValid	255
27.55.4.3 Size	255
27.55.4.4 TrimInternal	255
27.55.5 Friends And Related Function Documentation	255
27.55.5.1 operator"!="	255
27.55.5.2 operator<<	255
27.55.5.3 operator==	255
27.56gdcm::Command Class Reference	255
27.56.1 Detailed Description	257
27.56.2 Constructor & Destructor Documentation	257
27.56.2.1 Command	257
27.56.2.2 ~Command	257
27.56.3 Member Function Documentation	257
27.56.3.1 Execute	257
27.56.3.2 Execute	257
27.57gdcm::CommandDataSet Class Reference	257
27.57.1 Detailed Description	259
27.57.2 Constructor & Destructor Documentation	259
27.57.2.1 CommandDataSet	259
27.57.2.2 ~CommandDataSet	259
27.57.3 Member Function Documentation	259
27.57.3.1 Insert	259
27.57.3.2 Read	259
27.57.3.3 Replace	259
27.57.3.4 Write	259
27.57.4 Friends And Related Function Documentation	259
27.57.4.1 operator<<	259
27.58gdcm::network::CompositeMessageFactory Class Reference	259
27.58.1 Detailed Description	260
27.58.2 Member Function Documentation	260
27.58.2.1 ConstructCEchoRQ	260
27.58.2.2 ConstructCFindRQ	260
27.58.2.3 ConstructCMoveRQ	260
27.58.2.4 ConstructCStoreRQ	260
27.58.2.5 ConstructCStoreRSP	260

27.59gdcmm::CompositeNetworkFunctions Class Reference	260
27.59.1 Detailed Description	261
27.59.2 Member Typedef Documentation	261
27.59.2.1 KeyValuePairArrayType	261
27.59.2.2 KeyValuePairType	261
27.59.3 Member Function Documentation	261
27.59.3.1 CEcho	262
27.59.3.2 CFind	263
27.59.3.3 CMove	263
27.59.3.4 ConstructQuery	264
27.59.3.5 ConstructQuery	264
27.59.3.6 CStore	264
27.60gdcmm::ConstCharWrapper Class Reference	265
27.60.1 Detailed Description	265
27.60.2 Constructor & Destructor Documentation	265
27.60.2.1 ConstCharWrapper	265
27.60.3 Member Function Documentation	265
27.60.3.1 operator const char *	265
27.61gdcmm::CP246ExplicitDataElement Class Reference	265
27.61.1 Detailed Description	266
27.61.2 Member Function Documentation	266
27.61.2.1 GetLength	266
27.61.2.2 Read	267
27.61.2.3 ReadPreValue	267
27.61.2.4 ReadValue	267
27.61.2.5 ReadWithLength	267
27.62gdcmm::CryptoFactory Class Reference	267
27.62.1 Detailed Description	268
27.62.2 Member Enumeration Documentation	268
27.62.2.1 CryptoLib	268
27.62.3 Constructor & Destructor Documentation	268
27.62.3.1 CryptoFactory	268
27.62.3.2 CryptoFactory	268
27.62.3.3 ~CryptoFactory	268
27.62.4 Member Function Documentation	268
27.62.4.1 CreateCMSProvider	268
27.62.4.2 GetFactoryInstance	268

27.63gdcmm::CryptographicMessageSyntax Class Reference	269
27.63.1 Member Enumeration Documentation	269
27.63.1.1 CipherTypes	269
27.63.2 Constructor & Destructor Documentation	270
27.63.2.1 CryptographicMessageSyntax	270
27.63.2.2 ~CryptographicMessageSyntax	270
27.63.3 Member Function Documentation	270
27.63.3.1 Decrypt	270
27.63.3.2 Encrypt	270
27.63.3.3 GetCipherType	270
27.63.3.4 ParseCertificateFile	270
27.63.3.5 ParseKeyFile	270
27.63.3.6 SetCipherType	270
27.63.3.7 SetPassword	271
27.64gdcmm::CSAElement Class Reference	271
27.64.1 Detailed Description	272
27.64.2 Member Typedef Documentation	273
27.64.2.1 DataPtr	273
27.64.3 Constructor & Destructor Documentation	273
27.64.3.1 CSAElement	273
27.64.3.2 CSAElement	273
27.64.4 Member Function Documentation	273
27.64.4.1 GetByteValue	273
27.64.4.2 GetKey	273
27.64.4.3 GetName	273
27.64.4.4 GetNumberOfItems	273
27.64.4.5 GetSyngoDT	273
27.64.4.6 GetValue	273
27.64.4.7 GetValue	274
27.64.4.8 GetVM	274
27.64.4.9 GetVR	274
27.64.4.10IsEmpty	274
27.64.4.11operator<	274
27.64.4.12operator=	274
27.64.4.13operator==	274
27.64.4.14SetByteValue	274
27.64.4.15SetKey	274

27.64.4.16SetName	274
27.64.4.17SetNoOfItems	274
27.64.4.18SetSyngoDT	274
27.64.4.19SetValue	274
27.64.4.20SetVM	274
27.64.4.21SetVR	274
27.64.5 Friends And Related Function Documentation	275
27.64.5.1 operator<<	275
27.64.6 Member Data Documentation	275
27.64.6.1 DataField	275
27.64.6.2 KeyField	275
27.64.6.3 NameField	275
27.64.6.4 NoOfItemsField	275
27.64.6.5 SyngoDTField	275
27.64.6.6 ValueMultiplicityField	275
27.64.6.7 VRField	275
27.65gdcm::CSAHeader Class Reference	275
27.65.1 Detailed Description	277
27.65.2 Member Enumeration Documentation	277
27.65.2.1 CSAHeaderType	277
27.65.3 Constructor & Destructor Documentation	277
27.65.3.1 CSAHeader	277
27.65.3.2 ~CSAHeader	277
27.65.4 Member Function Documentation	277
27.65.4.1 FindCSAElementByName	278
27.65.4.2 GetCSADataInfo	278
27.65.4.3 GetCSAEEnd	278
27.65.4.4 GetCSAElementByName	278
27.65.4.5 GetCSAImageHeaderInfoTag	278
27.65.4.6 GetCSASeriesHeaderInfoTag	278
27.65.4.7 GetDataSet	279
27.65.4.8 GetFormat	279
27.65.4.9 GetInterfile	279
27.65.4.10LoadFromDataElement	279
27.65.4.11Print	279
27.65.4.12Read	279
27.65.4.13Write	279

27.65.5 Friends And Related Function Documentation	279
27.65.5.1 operator<<	279
27.66gdcmm::CSAHeaderDict Class Reference	279
27.66.1 Detailed Description	280
27.66.2 Member Typedef Documentation	280
27.66.2.1 ConstIterator	280
27.66.2.2 Iterator	280
27.66.2.3 MapCSAHeaderDictEntry	280
27.66.3 Constructor & Destructor Documentation	280
27.66.3.1 CSAHeaderDict	280
27.66.4 Member Function Documentation	280
27.66.4.1 AddCSAHeaderDictEntry	280
27.66.4.2 Begin	281
27.66.4.3 End	281
27.66.4.4 GetCSAHeaderDictEntry	281
27.66.4.5 IsEmpty	281
27.66.4.6 LoadDefault	281
27.66.5 Friends And Related Function Documentation	281
27.66.5.1 Dicts	281
27.66.5.2 operator<<	281
27.67gdcmm::CSAHeaderDictEntry Class Reference	281
27.67.1 Detailed Description	282
27.67.2 Constructor & Destructor Documentation	282
27.67.2.1 CSAHeaderDictEntry	282
27.67.3 Member Function Documentation	282
27.67.3.1 GetDescription	282
27.67.3.2 GetName	282
27.67.3.3 GetVM	282
27.67.3.4 GetVR	282
27.67.3.5 operator<	282
27.67.3.6 SetDescription	283
27.67.3.7 SetName	283
27.67.3.8 SetVM	283
27.67.3.9 SetVR	283
27.67.4 Friends And Related Function Documentation	283
27.67.4.1 operator<<	283
27.68gdcmm::CSAHeaderDictException Class Reference	283

27.69gdcm::network::CStoreRQ Class Reference	284
27.69.1 Detailed Description	285
27.69.2 Member Function Documentation	285
27.69.2.1 ConstructPDV	285
27.70gdcm::network::CStoreRSP Class Reference	285
27.70.1 Detailed Description	286
27.70.2 Member Function Documentation	286
27.70.2.1 ConstructPDV	286
27.71gdcm::Curve Class Reference	287
27.71.1 Detailed Description	288
27.71.2 Constructor & Destructor Documentation	288
27.71.2.1 Curve	288
27.71.2.2 ~Curve	288
27.71.2.3 Curve	288
27.71.3 Member Function Documentation	288
27.71.3.1 Decode	288
27.71.3.2 GetAsPoints	288
27.71.3.3 GetCurveDataDescriptor	289
27.71.3.4 GetDataValueRepresentation	289
27.71.3.5 GetDimensions	289
27.71.3.6 GetGroup	289
27.71.3.7 GetNumberOfCurves	289
27.71.3.8 GetNumberOfPoints	289
27.71.3.9 GetTypeInfoData	289
27.71.3.10GetTypeInfoDataDescription	289
27.71.3.11IsEmpty	289
27.71.3.12Print	289
27.71.3.13SetCoordinateStartValue	289
27.71.3.14SetCoordinateStepValue	289
27.71.3.15SetCurve	289
27.71.3.16SetCurveDataDescriptor	289
27.71.3.17SetCurveDescription	289
27.71.3.18SetDataValueRepresentation	289
27.71.3.19SetDimensions	289
27.71.3.20SetGroup	289
27.71.3.21SetNumberOfPoints	289
27.71.3.22SetTypeInfoData	289

27.71.3.23Update	289
27.72gdcmm::DataElement Class Reference	289
27.72.1 Detailed Description	292
27.72.2 Member Typedef Documentation	292
27.72.2.1 ValuePtr	292
27.72.3 Constructor & Destructor Documentation	293
27.72.3.1 DataElement	293
27.72.3.2 DataElement	293
27.72.4 Member Function Documentation	293
27.72.4.1 Clear	293
27.72.4.2 Empty	293
27.72.4.3 GetByteValue	293
27.72.4.4 GetLength	293
27.72.4.5 GetSequenceOfFragments	293
27.72.4.6 GetSequenceOfFragments	294
27.72.4.7 GetTag	294
27.72.4.8 GetTag	294
27.72.4.9 GetValue	294
27.72.4.10GetValue	294
27.72.4.11GetValueAsSQ	294
27.72.4.12GetVL	294
27.72.4.13GetVL	295
27.72.4.14GetVR	295
27.72.4.15IsEmpty	295
27.72.4.16IsUndefinedLength	295
27.72.4.17operator<	295
27.72.4.18operator=	295
27.72.4.19operator==	295
27.72.4.20Read	296
27.72.4.21ReadOrSkip	296
27.72.4.22ReadPreValue	296
27.72.4.23ReadValue	296
27.72.4.24ReadValueWithLength	296
27.72.4.25ReadWithLength	296
27.72.4.26SetByteValue	296
27.72.4.27SetTag	296
27.72.4.28SetValue	296

27.72.4.29 SetValueFieldLength	297
27.72.4.30 SetVL	297
27.72.4.31 SetVLToUndefined	297
27.72.4.32 SetVR	297
27.72.4.33 Write	298
27.72.5 Friends And Related Function Documentation	298
27.72.5.1 operator<<	298
27.72.6 Member Data Documentation	298
27.72.6.1 TagField	298
27.72.6.2 ValueField	298
27.72.6.3 ValueLengthField	298
27.72.6.4 VRField	298
27.73 gdcmm::DataElementException Class Reference	298
27.74 gdcmm::DataEvent Class Reference	299
27.74.1 Detailed Description	300
27.74.2 Member Typedef Documentation	300
27.74.2.1 Self	300
27.74.2.2 Superclass	300
27.74.3 Constructor & Destructor Documentation	300
27.74.3.1 DataEvent	301
27.74.3.2 ~DataEvent	301
27.74.3.3 DataEvent	301
27.74.4 Member Function Documentation	301
27.74.4.1 CheckEvent	301
27.74.4.2 GetData	301
27.74.4.3 GetDataLength	301
27.74.4.4 GetEventName	301
27.74.4.5 MakeObject	301
27.74.4.6 SetData	301
27.75 gdcmm::DataSet Class Reference	301
27.75.1 Detailed Description	304
27.75.2 Member Typedef Documentation	304
27.75.2.1 ConstIterator	304
27.75.2.2 DataElementSet	304
27.75.2.3 Iterator	304
27.75.2.4 SizeType	304
27.75.3 Member Function Documentation	304

27.75.3.1 Begin	304
27.75.3.2 Begin	304
27.75.3.3 Clear	304
27.75.3.4 ComputeDataElement	305
27.75.3.5 ComputeGroupLength	305
27.75.3.6 End	305
27.75.3.7 End	305
27.75.3.8 FindDataElement	305
27.75.3.9 FindDataElement	305
27.75.3.10FindNextDataElement	305
27.75.3.11GetDataElement	305
27.75.3.12GetDataElement	306
27.75.3.13GetDEEnd	306
27.75.3.14GetDES	306
27.75.3.15GetDES	306
27.75.3.16GetLength	306
27.75.3.17GetMediaStorage	306
27.75.3.18GetPrivateCreator	306
27.75.3.19Insert	306
27.75.3.20InsertDataElement	306
27.75.3.21IsEmpty	307
27.75.3.22operator()	307
27.75.3.23operator=	307
27.75.3.24operator[]	307
27.75.3.25Print	307
27.75.3.26Read	307
27.75.3.27ReadNested	307
27.75.3.28ReadSelectedPrivateTags	307
27.75.3.29ReadSelectedPrivateTagsWithLength	307
27.75.3.30ReadSelectedTags	307
27.75.3.31ReadSelectedTagsWithLength	307
27.75.3.32ReadUpToTag	307
27.75.3.33ReadUpToTagWithLength	307
27.75.3.34ReadWithLength	307
27.75.3.35Remove	307
27.75.3.36Replace	308
27.75.3.37ReplaceEmpty	308

27.75.3.38Size	308
27.75.3.39Write	308
27.75.4 Friends And Related Function Documentation	308
27.75.4.1 CSAHeader	308
27.75.4.2 operator<<	308
27.76gdcm::DataSetEvent Class Reference	308
27.76.1 Detailed Description	310
27.76.2 Member Typedef Documentation	310
27.76.2.1 Self	310
27.76.2.2 Superclass	310
27.76.3 Constructor & Destructor Documentation	310
27.76.3.1 DataSetEvent	310
27.76.3.2 ~DataSetEvent	310
27.76.3.3 DataSetEvent	310
27.76.4 Member Function Documentation	310
27.76.4.1 CheckEvent	310
27.76.4.2 GetDataSet	310
27.76.4.3 GetEventName	310
27.76.4.4 MakeObject	310
27.77gdcm::DataSetHelper Class Reference	311
27.77.1 Detailed Description	311
27.77.2 Member Function Documentation	311
27.77.2.1 ComputeVR	311
27.78gdcm::Decoder Class Reference	311
27.78.1 Detailed Description	312
27.78.2 Constructor & Destructor Documentation	312
27.78.2.1 ~Decoder	312
27.78.3 Member Function Documentation	312
27.78.3.1 CanDecode	312
27.78.3.2 Decode	312
27.78.3.3 DecodeByStreams	312
27.79gdcm::DefinedTerms Class Reference	313
27.79.1 Detailed Description	313
27.79.2 Constructor & Destructor Documentation	313
27.79.2.1 DefinedTerms	313
27.80gdcm::Defs Class Reference	313
27.80.1 Detailed Description	314

27.80.2 Constructor & Destructor Documentation	314
27.80.2.1 Defs	314
27.80.2.2 ~Defs	314
27.80.3 Member Function Documentation	314
27.80.3.1 GetIODFromFile	314
27.80.3.2 GetIODNameFromMediaStorage	314
27.80.3.3 GetIODs	315
27.80.3.4 GetIODs	315
27.80.3.5 GetMacros	315
27.80.3.6 GetMacros	315
27.80.3.7 GetModules	315
27.80.3.8 GetModules	315
27.80.3.9 GetTypeFromTag	315
27.80.3.10 IsEmpty	315
27.80.3.11 LoadDefaults	315
27.80.3.12 LoadFromFile	315
27.80.3.13 Verify	315
27.80.3.14 Verify	315
27.80.4 Friends And Related Function Documentation	315
27.80.4.1 Global	315
27.81 gdcmm::DeltaEncodingCodec Class Reference	316
27.81.1 Detailed Description	317
27.81.2 Constructor & Destructor Documentation	317
27.81.2.1 DeltaEncodingCodec	317
27.81.2.2 ~DeltaEncodingCodec	317
27.81.3 Member Function Documentation	317
27.81.3.1 CanDecode	317
27.81.3.2 Decode	317
27.81.3.3 Decode	317
27.82 gdcmm::DICOMDIR Class Reference	317
27.82.1 Detailed Description	317
27.82.2 Constructor & Destructor Documentation	318
27.82.2.1 DICOMDIR	318
27.82.2.2 DICOMDIR	318
27.83 gdcmm::DICOMDIRGenerator Class Reference	318
27.83.1 Detailed Description	319
27.83.2 Member Typedef Documentation	319

27.83.2.1	FilenameType	319
27.83.2.2	FilenameType	319
27.83.3	Constructor & Destructor Documentation	319
27.83.3.1	DICOMDIRGenerator	319
27.83.3.2	~DICOMDIRGenerator	319
27.83.4	Member Function Documentation	319
27.83.4.1	AddImageDirectoryRecord	319
27.83.4.2	AddPatientDirectoryRecord	319
27.83.4.3	AddSeriesDirectoryRecord	319
27.83.4.4	AddStudyDirectoryRecord	319
27.83.4.5	Generate	319
27.83.4.6	GetFile	320
27.83.4.7	GetScanner	320
27.83.4.8	SetDescriptor	320
27.83.4.9	SetFile	320
27.83.4.10	SetFilenames	320
27.83.4.11	SetRootDirectory	320
27.84	gdcm::Dict Class Reference	320
27.84.1	Detailed Description	321
27.84.2	Member Typedef Documentation	321
27.84.2.1	ConstIterator	321
27.84.2.2	Iterator	321
27.84.2.3	MapDictEntry	321
27.84.3	Constructor & Destructor Documentation	321
27.84.3.1	Dict	321
27.84.4	Member Function Documentation	321
27.84.4.1	AddDictEntry	321
27.84.4.2	Begin	321
27.84.4.3	End	322
27.84.4.4	GetDictEntry	322
27.84.4.5	GetDictEntryByKeyword	322
27.84.4.6	GetDictEntryByName	322
27.84.4.7	GetKeywordFromTag	322
27.84.4.8	IsEmpty	322
27.84.4.9	LoadDefault	322
27.84.5	Friends And Related Function Documentation	322
27.84.5.1	Dicts	322

27.84.5.2 operator<<	322
27.85gdcmm::DictConverter Class Reference	323
27.85.1 Detailed Description	323
27.85.2 Member Enumeration Documentation	324
27.85.2.1 OutputTypes	324
27.85.3 Constructor & Destructor Documentation	324
27.85.3.1 DictConverter	324
27.85.3.2 ~DictConverter	324
27.85.4 Member Function Documentation	324
27.85.4.1 AddGroupLength	324
27.85.4.2 Convert	324
27.85.4.3 ConvertToCXX	324
27.85.4.4 ConvertToXML	324
27.85.4.5 GetDictName	324
27.85.4.6 GetInputFilename	324
27.85.4.7 GetOutputFilename	324
27.85.4.8 GetOutputType	324
27.85.4.9 Readuint16	324
27.85.4.10ReadVM	324
27.85.4.11ReadVR	324
27.85.4.12SetDictName	324
27.85.4.13SetInputFileName	324
27.85.4.14SetOutputFileName	324
27.85.4.15SetOutputType	324
27.85.4.16WriteFooter	325
27.85.4.17WriteHeader	325
27.86gdcmm::DictEntry Class Reference	325
27.86.1 Detailed Description	326
27.86.2 Constructor & Destructor Documentation	326
27.86.2.1 DictEntry	326
27.86.3 Member Function Documentation	326
27.86.3.1 GetKeyword	326
27.86.3.2 GetName	326
27.86.3.3 GetRetired	326
27.86.3.4 GetVM	326
27.86.3.5 GetVR	327
27.86.3.6 IsUnique	327

27.86.3.7 SetElementXX	327
27.86.3.8 SetGroupXX	327
27.86.3.9 SetKeyword	327
27.86.3.10SetName	327
27.86.3.11SetRetired	327
27.86.3.12SetVM	327
27.86.3.13SetVR	327
27.86.4 Friends And Related Function Documentation	327
27.86.4.1 Dict	327
27.86.4.2 operator<<	327
27.87gdcmm::DictPrinter Class Reference	327
27.87.1 Detailed Description	329
27.87.2 Constructor & Destructor Documentation	329
27.87.2.1 DictPrinter	329
27.87.2.2 ~DictPrinter	329
27.87.3 Member Function Documentation	329
27.87.3.1 Print	329
27.87.3.2 PrintDataElement2	329
27.87.3.3 PrintDataSet2	329
27.88gdcmm::Dicts Class Reference	329
27.88.1 Detailed Description	330
27.88.2 Member Enumeration Documentation	330
27.88.2.1 ConstructorType	330
27.88.3 Constructor & Destructor Documentation	330
27.88.3.1 Dicts	330
27.88.3.2 ~Dicts	330
27.88.4 Member Function Documentation	330
27.88.4.1 GetConstructorString	331
27.88.4.2 GetCSAHeaderDict	331
27.88.4.3 GetDictEntry	331
27.88.4.4 GetDictEntry	331
27.88.4.5 GetPrivateDict	331
27.88.4.6 GetPrivateDict	331
27.88.4.7 GetPublicDict	331
27.88.4.8 IsEmpty	331
27.88.4.9 LoadDefaults	331
27.88.5 Friends And Related Function Documentation	331

27.88.5.1 Global	331
27.88.5.2 operator<<	331
27.89gdcmm::network::DIMSE Class Reference	332
27.89.1 Detailed Description	332
27.89.2 Member Enumeration Documentation	332
27.89.2.1 CommandTypes	332
27.90gdcmm::DirectionCosines Class Reference	333
27.90.1 Detailed Description	334
27.90.2 Constructor & Destructor Documentation	334
27.90.2.1 DirectionCosines	334
27.90.2.2 DirectionCosines	334
27.90.2.3 ~DirectionCosines	334
27.90.3 Member Function Documentation	334
27.90.3.1 ComputeDistAlongNormal	334
27.90.3.2 Cross	334
27.90.3.3 CrossDot	334
27.90.3.4 Dot	334
27.90.3.5 IsValid	334
27.90.3.6 Normalize	335
27.90.3.7 operator const double *	335
27.90.3.8 Print	335
27.90.3.9 SetFromString	335
27.91gdcmm::Directory Class Reference	335
27.91.1 Detailed Description	336
27.91.2 Member Typedef Documentation	336
27.91.2.1 FilenamesType	336
27.91.2.2 FilenameType	336
27.91.3 Constructor & Destructor Documentation	336
27.91.3.1 Directory	336
27.91.3.2 ~Directory	336
27.91.4 Member Function Documentation	336
27.91.4.1 Explore	336
27.91.4.2 GetDirectories	337
27.91.4.3 GetFilenames	337
27.91.4.4 GetToplevel	337
27.91.4.5 Load	337
27.91.4.6 Print	337

27.91.5 Friends And Related Function Documentation	337
27.91.5.1 operator<<	337
27.92gdcm::DirectoryHelper Class Reference	338
27.92.1 Detailed Description	338
27.92.2 Member Function Documentation	338
27.92.2.1 GetCTImageSeriesUIDs	338
27.92.2.2 GetFileNamesFromSeriesUIDs	338
27.92.2.3 GetFrameOfReference	338
27.92.2.4 GetMRImageSeriesUIDs	338
27.92.2.5 GetRTStructSeriesUIDs	339
27.92.2.6 GetSeriesUIDsBySOPClassUID	339
27.92.2.7 GetSOPClassUID	339
27.92.2.8 GetStringValuesFromTag	339
27.92.2.9 LoadImageFromFiles	339
27.92.2.10RetrieveSOPInstanceUIDFromIndex	339
27.92.2.11RetrieveSOPInstanceUIDFromZPosition	339
27.93gdcm::DummyValueGenerator Class Reference	339
27.93.1 Detailed Description	339
27.93.2 Member Function Documentation	339
27.93.2.1 Generate	340
27.94gdcm::Dumper Class Reference	340
27.94.1 Detailed Description	341
27.94.2 Constructor & Destructor Documentation	341
27.94.2.1 Dumper	341
27.94.2.2 ~Dumper	341
27.95gdcm::Element< TVR, TVM > Class Template Reference	342
27.95.1 Detailed Description	343
27.95.2 Member Typedef Documentation	344
27.95.2.1 Type	344
27.95.3 Member Function Documentation	344
27.95.3.1 GetAsDataElement	344
27.95.3.2 GetLength	344
27.95.3.3 GetValue	344
27.95.3.4 GetValue	344
27.95.3.5 GetValues	344
27.95.3.6 GetVM	344
27.95.3.7 GetVR	344

27.95.3.8 operator[]	344
27.95.3.9 Print	344
27.95.3.10Read	344
27.95.3.11Set	344
27.95.3.12SetFromDataElement	344
27.95.3.13SetNoSwap	344
27.95.3.14SetValue	344
27.95.3.15Write	344
27.95.4 Member Data Documentation	344
27.95.4.1 Internal	344
27.96gdcmm::Element< TVR, VM::VM1_2 > Class Template Reference	345
27.96.1 Member Typedef Documentation	346
27.96.1.1 Parent	346
27.96.2 Member Function Documentation	346
27.96.2.1 SetLength	346
27.97gdcmm::Element< TVR, VM::VM1_n > Class Template Reference	346
27.97.1 Member Typedef Documentation	347
27.97.1.1 Type	347
27.97.2 Constructor & Destructor Documentation	347
27.97.2.1 Element	347
27.97.2.2 ~Element	347
27.97.2.3 Element	347
27.97.3 Member Function Documentation	347
27.97.3.1 GetAsDataElement	347
27.97.3.2 GetLength	347
27.97.3.3 GetValue	347
27.97.3.4 GetValue	347
27.97.3.5 GetVM	347
27.97.3.6 GetVR	347
27.97.3.7 operator=	348
27.97.3.8 operator[]	348
27.97.3.9 Print	348
27.97.3.10Read	348
27.97.3.11Set	348
27.97.3.12SetArray	348
27.97.3.13SetFromDataElement	348
27.97.3.14SetLength	348

27.97.3.15SetNoSwap	348
27.97.3.16SetValue	348
27.97.3.17Write	348
27.97.3.18WriteASCII	348
27.98gdcmm::Element< TVR, VM::VM2_2n > Class Template Reference	348
27.98.1 Member Typedef Documentation	350
27.98.1.1 Parent	350
27.98.2 Member Function Documentation	350
27.98.2.1 SetLength	350
27.99gdcmm::Element< TVR, VM::VM2_n > Class Template Reference	350
27.99.1 Member Typedef Documentation	351
27.99.1.1 Parent	351
27.99.2 Member Function Documentation	351
27.99.2.1 SetLength	351
27.100gdcmm::Element< TVR, VM::VM3_3n > Class Template Reference	351
27.100.1 Member Typedef Documentation	353
27.100.1.1Parent	353
27.100.2Member Function Documentation	353
27.100.2.1SetLength	353
27.101gdcmm::Element< TVR, VM::VM3_n > Class Template Reference	353
27.101.1 Member Typedef Documentation	354
27.101.1.1Parent	354
27.101.2Member Function Documentation	354
27.101.2.1SetLength	354
27.102gdcmm::Element< VR::AS, VM::VM5 > Class Template Reference	354
27.102.1 Member Function Documentation	355
27.102.1.1GetLength	355
27.102.1.2Print	355
27.102.2Member Data Documentation	355
27.102.2.1Internal	355
27.103gdcmm::Element< VR::OB, VM::VM1 > Class Template Reference	355
27.104gdcmm::Element< VR::OW, VM::VM1 > Class Template Reference	356
27.105gdcmm::ElementDisableCombinations< TVR, TVM > Class Template Reference	358
27.105.1Detailed Description	358
27.106gdcmm::ElementDisableCombinations< VR::OB, VM::VM1_n > Class Template Reference	359
27.107gdcmm::ElementDisableCombinations< VR::OW, VM::VM1_n > Class Template Reference	359
27.108gdcmm::EncapsulatedDocument Class Reference	359

27.108.1	Detailed Description	359
27.108.2	Constructor & Destructor Documentation	359
27.108.2.1	EncapsulatedDocument	359
27.109	dcm::EncodingImplementation< T > Class Template Reference	360
27.109.1	Detailed Description	360
27.110	dcm::EncodingImplementation< VR::VRASCII > Class Template Reference	360
27.110.1	Member Function Documentation	360
27.110.1.1	Read	360
27.110.1.2	ReadComputeLength	361
27.110.1.3	ReadNoSwap	361
27.110.1.4	Write	361
27.110.1.5	Write	361
27.110.1.6	Write	361
27.111	dcm::EncodingImplementation< VR::VRBINARY > Class Template Reference	361
27.111.1	Member Function Documentation	361
27.111.1.1	Read	361
27.111.1.2	ReadComputeLength	361
27.111.1.3	ReadNoSwap	362
27.111.1.4	Write	362
27.112	dcm::EndEvent Class Reference	362
27.113	dcm::EnumeratedValues Class Reference	363
27.113.1	Detailed Description	363
27.113.2	Constructor & Destructor Documentation	364
27.113.2.1	EnumeratedValues	364
27.114	dcm::Event Class Reference	364
27.114.1	Detailed Description	366
27.114.2	Constructor & Destructor Documentation	366
27.114.2.1	Event	366
27.114.2.2	Event	366
27.114.2.3	~Event	366
27.114.3	Member Function Documentation	366
27.114.3.1	CheckEvent	366
27.114.3.2	GetEventName	366
27.114.3.3	MakeObject	366
27.114.3.4	Print	366
27.115	dcm::Exception Class Reference	367
27.115.1	Detailed Description	368

27.115.2	Constructor & Destructor Documentation	368
27.115.2.1	Exception	368
27.115.2.2	~Exception	368
27.115.3	Member Function Documentation	368
27.115.3.1	GetDescription	368
27.115.3.2	what	368
27.116	dcm::ExitEvent Class Reference	368
27.117	dcm::ExplicitDataElement Class Reference	370
27.117.1	Detailed Description	371
27.117.2	Member Function Documentation	371
27.117.2.1	GetLength	371
27.117.2.2	Read	371
27.117.2.3	ReadPreValue	371
27.117.2.4	ReadValue	371
27.117.2.5	ReadWithLength	371
27.117.2.6	Write	371
27.118	dcm::ExplicitImplicitDataElement Class Reference	371
27.118.1	Detailed Description	373
27.118.2	Member Function Documentation	373
27.118.2.1	GetLength	373
27.118.2.2	Read	373
27.118.2.3	ReadPreValue	373
27.118.2.4	ReadValue	373
27.118.2.5	ReadWithLength	373
27.119	dcm::Fiducials Class Reference	373
27.119.1	Detailed Description	373
27.119.2	Constructor & Destructor Documentation	374
27.119.2.1	Fiducials	374
27.120	dcm::File Class Reference	374
27.120.1	Detailed Description	375
27.120.2	Constructor & Destructor Documentation	376
27.120.2.1	File	376
27.120.2.2	~File	376
27.120.3	Member Function Documentation	376
27.120.3.1	GetDataSet	376
27.120.3.2	GetDataSet	376
27.120.3.3	GetHeader	376

27.120.3.4	GetHeader	376
27.120.3.5	Read	376
27.120.3.6	SetDataSet	377
27.120.3.7	SetHeader	377
27.120.3.8	Write	377
27.120.4	Friends And Related Function Documentation	377
27.120.4.1	operator<<	377
27.121	gdcm::FileAnonymizer Class Reference	377
27.121.1	Detailed Description	378
27.121.2	Constructor & Destructor Documentation	379
27.121.2.1	FileAnonymizer	379
27.121.2.2	~FileAnonymizer	379
27.121.3	Member Function Documentation	379
27.121.3.1	Empty	379
27.121.3.2	Remove	379
27.121.3.3	Replace	379
27.121.3.4	Replace	379
27.121.3.5	SetInputFileName	380
27.121.3.6	SetOutputFileName	380
27.121.3.7	Write	380
27.122	gdcm::FileChangeTransferSyntax Class Reference	380
27.122.1	Detailed Description	382
27.122.2	Constructor & Destructor Documentation	382
27.122.2.1	FileChangeTransferSyntax	382
27.122.2.2	~FileChangeTransferSyntax	382
27.122.3	Member Function Documentation	382
27.122.3.1	Change	382
27.122.3.2	GetCodec	382
27.122.3.3	New	382
27.122.3.4	SetInputFileName	383
27.122.3.5	SetOutputFileName	383
27.122.3.6	SetTransferSyntax	383
27.123	gdcm::FileDecompressLookupTable Class Reference	383
27.123.1	Detailed Description	384
27.123.2	Constructor & Destructor Documentation	384
27.123.2.1	FileDecompressLookupTable	384
27.123.2.2	~FileDecompressLookupTable	384

27.123.3	Member Function Documentation	384
27.123.3.1	Change	385
27.123.3.2	GetFile	385
27.123.3.3	GetPixmap	385
27.123.3.4	GetPixmap	385
27.123.3.5	SetFile	385
27.123.3.6	SetPixmap	385
27.124	dcm::FileDerivation Class Reference	385
27.124.1	Detailed Description	386
27.124.2	Constructor & Destructor Documentation	386
27.124.2.1	FileDerivation	386
27.124.2.2	~FileDerivation	386
27.124.3	Member Function Documentation	386
27.124.3.1	AddDerivationDescription	386
27.124.3.2	AddPurposeOfReferenceCodeSequence	386
27.124.3.3	AddReference	386
27.124.3.4	AddSourceImageSequence	386
27.124.3.5	Derive	386
27.124.3.6	GetFile	387
27.124.3.7	GetFile	387
27.124.3.8	SetDerivationCodeSequenceCodeValue	387
27.124.3.9	SetDerivationDescription	387
27.124.3.10	SetFile	387
27.124.3.11	SetPurposeOfReferenceCodeSequenceCodeValue	387
27.125	dcm::FileExplicitFilter Class Reference	387
27.125.1	Detailed Description	388
27.125.2	Constructor & Destructor Documentation	388
27.125.2.1	FileExplicitFilter	388
27.125.2.2	~FileExplicitFilter	388
27.125.3	Member Function Documentation	388
27.125.3.1	Change	389
27.125.3.2	ChangeFMI	389
27.125.3.3	GetFile	389
27.125.3.4	ProcessDataSet	389
27.125.3.5	SetChangePrivateTags	389
27.125.3.6	SetFile	389
27.125.3.7	SetRecomputeItemLength	389

27.125.3.8SetRecomputeSequenceLength	389
27.125.3.9SetUseVRUN	389
27.126.0dcm::FileMetaInformation Class Reference	389
27.126.1Detailed Description	392
27.126.2Constructor & Destructor Documentation	392
27.126.2.1FileMetaInformation	392
27.126.2.2~FileMetaInformation	392
27.126.2.3FileMetaInformation	392
27.126.3Member Function Documentation	392
27.126.3.1AppendImplementationClassUID	392
27.126.3.2ComputeDataSetMediaStorageSOPClass	392
27.126.3.3ComputeDataSetTransferSyntax	392
27.126.3.4Default	392
27.126.3.5FillFromDataSet	392
27.126.3.6GetDataSetTransferSyntax	392
27.126.3.7GetFileMetaInformationVersion	393
27.126.3.8GetFullLength	393
27.126.3.9GetGDCMImplementationClassUID	393
27.126.3.10GetGDCMImplementationVersionName	393
27.126.3.11GetGDCMSourceApplicationEntityTitle	393
27.126.3.12GetImplementationClassUID	393
27.126.3.13GetImplementationVersionName	393
27.126.3.14GetMediaStorage	393
27.126.3.15GetMediaStorageAsString	393
27.126.3.16GetMetaInformationTS	393
27.126.3.17GetPreamble	393
27.126.3.18GetPreamble	393
27.126.3.19GetSourceApplicationEntityTitle	393
27.126.3.20Insert	393
27.126.3.21IsValid	393
27.126.3.22Read	393
27.126.3.23ReadCompat	393
27.126.3.24ReadCompatInternal	393
27.126.3.25Replace	394
27.126.3.26SetDataSetTransferSyntax	394
27.126.3.27SetImplementationClassUID	394
27.126.3.28SetImplementationVersionName	394

27.126.3.2	SetPreamble	394
27.126.3.3	SetSourceApplicationEntityTitle	394
27.126.3.3	Write	394
27.126.4	Friends And Related Function Documentation	394
27.126.4.1	operator<<	394
27.126.5	Member Data Documentation	394
27.126.5.1	DataSetMS	394
27.126.5.2	DataSetTS	394
27.126.5.3	MetaInformationTS	395
27.127	gdcmm::Filename Class Reference	395
27.127.1	Detailed Description	395
27.127.2	Constructor & Destructor Documentation	396
27.127.2.1	Filename	396
27.127.3	Member Function Documentation	396
27.127.3.1	EndWith	396
27.127.3.2	GetExtension	396
27.127.3.3	GetFileName	396
27.127.3.4	GetName	396
27.127.3.5	GetPath	396
27.127.3.6	IsEmpty	396
27.127.3.7	IsIdentical	396
27.127.3.8	Join	396
27.127.3.9	operator const char *	396
27.127.3.10	ToUnixSlashes	397
27.127.3.11	ToWindowsSlashes	397
27.128	gdcmm::FileNameEvent Class Reference	397
27.128.1	Detailed Description	398
27.128.2	Member Typedef Documentation	399
27.128.2.1	Self	399
27.128.2.2	Superclass	399
27.128.3	Constructor & Destructor Documentation	399
27.128.3.1	FileNameEvent	399
27.128.3.2	~FileNameEvent	399
27.128.3.3	FileNameEvent	399
27.128.4	Member Function Documentation	399
27.128.4.1	CheckEvent	399
27.128.4.2	GetEventName	399

27.128.4.3	GetFileName	399
27.128.4.4	MakeObject	399
27.128.4.5	SetFileName	399
27.129	dcm::FilenameGenerator Class Reference	399
27.129.1	Detailed Description	400
27.129.2	Member Typedef Documentation	400
27.129.2.1	FileNamesType	400
27.129.2.2	FilenameType	400
27.129.2.3	SizeType	400
27.129.3	Constructor & Destructor Documentation	400
27.129.3.1	FilenameGenerator	401
27.129.3.2	~FilenameGenerator	401
27.129.4	Member Function Documentation	401
27.129.4.1	Generate	401
27.129.4.2	GetFilename	401
27.129.4.3	GetFileNames	401
27.129.4.4	GetNumberOfFileNames	401
27.129.4.5	GetPattern	401
27.129.4.6	GetPrefix	401
27.129.4.7	SetNumberOfFileNames	401
27.129.4.8	SetPattern	401
27.129.4.9	SetPrefix	402
27.130	dcm::FileSet Class Reference	402
27.130.1	Detailed Description	402
27.130.2	Member Typedef Documentation	402
27.130.2.1	FilesType	402
27.130.2.2	FileType	402
27.130.3	Constructor & Destructor Documentation	402
27.130.3.1	FileSet	402
27.130.4	Member Function Documentation	402
27.130.4.1	AddFile	403
27.130.4.2	AddFile	403
27.130.4.3	GetFiles	403
27.130.4.4	SetFiles	403
27.130.5	Friends And Related Function Documentation	403
27.130.5.1	operator<<	403
27.131	dcm::FileStreamer Class Reference	403

27.131.1Detailed Description	405
27.131.2Constructor & Destructor Documentation	405
27.131.2.1FileStreamer	405
27.131.2.2~FileStreamer	405
27.131.3Member Function Documentation	405
27.131.3.1AppendToDataElement	405
27.131.3.2AppendToGroupDataElement	405
27.131.3.3CheckDataElement	405
27.131.3.4CheckTemplateFileName	405
27.131.3.5New	405
27.131.3.6ReserveDataElement	406
27.131.3.7ReserveGroupDataElement	406
27.131.3.8SetOutputFileName	406
27.131.3.9SetTemplateFileName	406
27.131.3.10StartDataElement	406
27.131.3.11StartGroupDataElement	406
27.131.3.12StopDataElement	406
27.131.3.13StopGroupDataElement	406
27.132dcm::FileWithName Class Reference	407
27.132.1Detailed Description	408
27.132.2Constructor & Destructor Documentation	408
27.132.2.1FileWithName	408
27.132.3Member Data Documentation	408
27.132.3.1filename	408
27.133dcm::FindPatientRootQuery Class Reference	408
27.133.1Detailed Description	410
27.133.2Constructor & Destructor Documentation	410
27.133.2.1FindPatientRootQuery	410
27.133.3Member Function Documentation	410
27.133.3.1GetAbstractSyntaxUID	410
27.133.3.2GetTagListByLevel	410
27.133.3.3InitializeDataSet	410
27.133.3.4ValidateQuery	410
27.133.4Friends And Related Function Documentation	410
27.133.4.1QueryFactory	411
27.134dcm::FindStudyRootQuery Class Reference	411
27.134.1Detailed Description	412

27.134.2	Constructor & Destructor Documentation	412
27.134.2.1	FindStudyRootQuery	412
27.134.3	Member Function Documentation	412
27.134.3.1	GetAbstractSyntaxUID	412
27.134.3.2	GetTagListByLevel	412
27.134.3.3	InitializeDataSet	412
27.134.3.4	ValidateQuery	412
27.134.4	Friends And Related Function Documentation	413
27.134.4.1	QueryFactory	413
27.135	dcm::Fragment Class Reference	413
27.135.1	Detailed Description	414
27.135.2	Constructor & Destructor Documentation	415
27.135.2.1	Fragment	415
27.135.3	Member Function Documentation	415
27.135.3.1	ComputeLength	415
27.135.3.2	GetLength	415
27.135.3.3	Read	415
27.135.3.4	ReadBacktrack	415
27.135.3.5	ReadPreValue	415
27.135.3.6	ReadValue	415
27.135.3.7	Write	415
27.135.4	Friends And Related Function Documentation	415
27.135.4.1	operator<<	415
27.136	dcm::Global Class Reference	415
27.136.1	Detailed Description	416
27.136.2	Constructor & Destructor Documentation	416
27.136.2.1	Global	416
27.136.2.2	~Global	416
27.136.3	Member Function Documentation	416
27.136.3.1	Append	416
27.136.3.2	GetDefs	417
27.136.3.3	GetDicts	417
27.136.3.4	GetDicts	417
27.136.3.5	GetInstance	417
27.136.3.6	LoadResourcesFiles	417
27.136.3.7	Locate	417
27.136.3.8	Prepend	418

27.136.4	Friends And Related Function Documentation	. 418
27.136.4.1	operator<<	. 418
27.137	gdcm::GroupDict Class Reference	. 418
27.137.1	Detailed Description	. 418
27.137.2	Member Typedef Documentation	. 419
27.137.2.1	GroupStringVector	. 419
27.137.3	Constructor & Destructor Documentation	. 419
27.137.3.1	GroupDict	. 419
27.137.3.2	~GroupDict	. 419
27.137.4	Member Function Documentation	. 419
27.137.4.1	Add	. 419
27.137.4.2	GetAbbreviation	. 419
27.137.4.3	GetName	. 419
27.137.4.4	Insert	. 419
27.137.4.5	Size	. 419
27.137.5	Friends And Related Function Documentation	. 419
27.137.5.1	operator<<	. 419
27.138	gdcm::IconImageFilter Class Reference	. 419
27.138.1	Detailed Description	. 420
27.138.2	Constructor & Destructor Documentation	. 421
27.138.2.1	IconImageFilter	. 421
27.138.2.2	~IconImageFilter	. 421
27.138.3	Member Function Documentation	. 421
27.138.3.1	Extract	. 421
27.138.3.2	ExtractIconImages	. 421
27.138.3.3	ExtractVeprolIconImages	. 421
27.138.3.4	GetFile	. 421
27.138.3.5	GetFile	. 421
27.138.3.6	GetIconImage	. 421
27.138.3.7	GetNumberOfIconImages	. 421
27.138.3.8	SetFile	. 421
27.139	gdcm::IconImageGenerator Class Reference	. 422
27.139.1	Detailed Description	. 422
27.139.2	Constructor & Destructor Documentation	. 423
27.139.2.1	IconImageGenerator	. 423
27.139.2.2	~IconImageGenerator	. 423
27.139.3	Member Function Documentation	. 423

27.139.3.1AutoPixelMinMax	423
27.139.3.2ConvertRGBToPaletteColor	423
27.139.3.3Generate	423
27.139.3.4GetIconImage	423
27.139.3.5GetPixmap	423
27.139.3.6GetPixmap	423
27.139.3.7SetOutputDimensions	423
27.139.3.8SetOutsideValuePixel	424
27.139.3.9SetPixelMinMax	424
27.139.3.10SetPixmap	424
27.140gdcmm::ignore_char Struct Reference	424
27.140.1Constructor & Destructor Documentation	424
27.140.1.1ignore_char	424
27.140.2Member Data Documentation	424
27.140.2.1m_char	424
27.141gdcmm::Image Class Reference	425
27.141.1Detailed Description	426
27.141.2Constructor & Destructor Documentation	427
27.141.2.1Image	427
27.141.2.2~Image	427
27.141.3Member Function Documentation	427
27.141.3.1GetDirectionCosines	427
27.141.3.2GetDirectionCosines	427
27.141.3.3GetIntercept	427
27.141.3.4GetOrigin	427
27.141.3.5GetOrigin	427
27.141.3.6GetSlope	427
27.141.3.7GetSpacing	427
27.141.3.8GetSpacing	427
27.141.3.9Print	427
27.141.3.10SetDirectionCosines	428
27.141.3.11SetDirectionCosines	428
27.141.3.12SetDirectionCosines	428
27.141.3.13SetIntercept	428
27.141.3.14SetOrigin	428
27.141.3.15SetOrigin	428
27.141.3.16SetOrigin	428

27.141.3.1	SetSlope	428
27.141.3.1	SetSpacing	428
27.141.3.1	SetSpacing	428
27.142	dcml::ImageApplyLookupTable Class Reference	428
27.142.1	Detailed Description	431
27.142.2	Constructor & Destructor Documentation	431
27.142.2.1	ImageApplyLookupTable	431
27.142.2.2	~ImageApplyLookupTable	431
27.142.3	Member Function Documentation	431
27.142.3.1	Apply	431
27.143	dcml::ImageChangePhotometricInterpretation Class Reference	431
27.143.1	Detailed Description	433
27.143.2	Constructor & Destructor Documentation	433
27.143.2.1	ImageChangePhotometricInterpretation	433
27.143.2.2	~ImageChangePhotometricInterpretation	433
27.143.3	Member Function Documentation	433
27.143.3.1	Change	433
27.143.3.2	ChangeMonochrome	433
27.143.3.3	GetPhotometricInterpretation	433
27.143.3.4	RGB2YBR	433
27.143.3.5	SetPhotometricInterpretation	433
27.143.3.6	YBR2RGB	434
27.144	dcml::ImageChangePlanarConfiguration Class Reference	434
27.144.1	Detailed Description	436
27.144.2	Constructor & Destructor Documentation	436
27.144.2.1	ImageChangePlanarConfiguration	436
27.144.2.2	~ImageChangePlanarConfiguration	436
27.144.3	Member Function Documentation	436
27.144.3.1	Change	436
27.144.3.2	GetPlanarConfiguration	436
27.144.3.3	RGBPixelsToRGBPlanes	436
27.144.3.4	RGBPlanesToRGBPixels	436
27.144.3.5	SetPlanarConfiguration	436
27.145	dcml::ImageChangeTransferSyntax Class Reference	437
27.145.1	Detailed Description	439
27.145.2	Constructor & Destructor Documentation	439
27.145.2.1	ImageChangeTransferSyntax	439

27.145.2.2~ImageChangeTransferSyntax	439
27.145.3Member Function Documentation	439
27.145.3.1Change	439
27.145.3.2GetTransferSyntax	439
27.145.3.3SetCompressIconImage	440
27.145.3.4SetForce	440
27.145.3.5SetTransferSyntax	440
27.145.3.6SetUserCodec	440
27.145.3.7TryJPEG2000Codec	440
27.145.3.8TryJPEGCodec	440
27.145.3.9TryJPEGLSCodec	440
27.145.3.10TryRAWCodec	440
27.145.3.11TryRLECodec	440
27.146dcm::ImageCodec Class Reference	441
27.146.1Detailed Description	443
27.146.2Member Typedef Documentation	443
27.146.2.1LUTPtr	443
27.146.3Constructor & Destructor Documentation	443
27.146.3.1ImageCodec	443
27.146.3.2~ImageCodec	443
27.146.4Member Function Documentation	443
27.146.4.1AppendFrameEncode	443
27.146.4.2AppendRowEncode	443
27.146.4.3CanCode	444
27.146.4.4CanDecode	444
27.146.4.5Clone	444
27.146.4.6Decode	444
27.146.4.7DecodeByStreams	444
27.146.4.8DoByteSwap	444
27.146.4.9DoInvertMonochrome	444
27.146.4.10DoOverlayCleanup	444
27.146.4.11DoPaddedCompositePixelCode	444
27.146.4.12DoPlanarConfiguration	444
27.146.4.13DoSimpleCopy	444
27.146.4.14DoYBR	444
27.146.4.15GetDimensions	445
27.146.4.16GetHeaderInfo	445

27.146.4.1	GetLossyFlag	. 445
27.146.4.1	GetLUT	. 445
27.146.4.1	GetNeedByteSwap	. 445
27.146.4.2	GetNumberOfDimensions	. 445
27.146.4.2	GetPhotometricInterpretation	. 445
27.146.4.2	GetPixelFormat	. 445
27.146.4.2	GetPixelFormat	. 445
27.146.4.2	GetPlanarConfiguration	. 445
27.146.4.2	FrameEncoder	. 445
27.146.4.2	Lossy	. 445
27.146.4.2	RowEncoder	. 445
27.146.4.2	Valid	. 445
27.146.4.2	SetDimensions	. 445
27.146.4.3	SetDimensions	. 446
27.146.4.3	SetLossyFlag	. 446
27.146.4.3	SetLUT	. 446
27.146.4.3	SetNeedByteSwap	. 446
27.146.4.3	SetNeedOverlayCleanup	. 446
27.146.4.3	SetNumberOfDimensions	. 446
27.146.4.3	SetPhotometricInterpretation	. 446
27.146.4.3	SetPixelFormat	. 446
27.146.4.3	SetPlanarConfiguration	. 446
27.146.4.3	StartEncode	. 446
27.146.4.4	StopEncode	. 446
27.146.5	Friends And Related Function Documentation	. 446
27.146.5.1	FileChangeTransferSyntax	. 446
27.146.5.2	ImageChangePhotometricInterpretation	. 447
27.146.6	Member Data Documentation	. 447
27.146.6.1	Dimensions	. 447
27.146.6.2	LossyFlag	. 447
27.146.6.3	LUT	. 447
27.146.6.4	NeedByteSwap	. 447
27.146.6.5	NeedOverlayCleanup	. 447
27.146.6.6	NumberOfDimensions	. 447
27.146.6.7	PF	. 447
27.146.6.8	PI	. 447
27.146.6.9	PlanarConfiguration	. 447

27.146.6.1RequestPaddedCompositePixelCode	447
27.146.6.1RequestPlanarConfiguration	447
27.147dcm::ImageConverter Class Reference	447
27.147.1Detailed Description	447
27.147.2Constructor & Destructor Documentation	448
27.147.2.1ImageConverter	448
27.147.2.2~ImageConverter	448
27.147.3Member Function Documentation	448
27.147.3.1Convert	448
27.147.3.2GetOutput	448
27.147.3.3SetInput	448
27.148dcm::ImageFragmentSplitter Class Reference	448
27.148.1Detailed Description	451
27.148.2Constructor & Destructor Documentation	451
27.148.2.1ImageFragmentSplitter	451
27.148.2.2~ImageFragmentSplitter	451
27.148.3Member Function Documentation	451
27.148.3.1GetFragmentSizeMax	451
27.148.3.2SetForce	451
27.148.3.3SetFragmentSizeMax	451
27.148.3.4Split	451
27.149dcm::ImageHelper Class Reference	451
27.149.1Detailed Description	452
27.149.2Member Function Documentation	452
27.149.2.1ComputeMediaStorageFromModality	452
27.149.2.2ComputeSpacingFromImagePositionPatient	453
27.149.2.3GetDimensionsValue	453
27.149.2.4GetDirectionCosinesFromDataSet	453
27.149.2.5GetDirectionCosinesValue	453
27.149.2.6GetForcePixelSpacing	453
27.149.2.7GetForceRescaleInterceptSlope	453
27.149.2.8GetLUT	453
27.149.2.9GetOriginValue	453
27.149.2.10GetPhotometricInterpretationValue	453
27.149.2.11GetPixelFormatValue	453
27.149.2.12GetPlanarConfigurationValue	453
27.149.2.13GetPointerFromElement	453

27.149.2.10	GetRealWorldValueMappingContent	453
27.149.2.11	GetRescaleInterceptSlopeValue	454
27.149.2.12	GetSpacingTagFromMediaStorage	454
27.149.2.13	GetSpacingValue	454
27.149.2.14	GetZSpacingTagFromMediaStorage	454
27.149.2.15	SetDimensionsValue	454
27.149.2.16	SetDirectionCosinesValue	454
27.149.2.17	SetForcePixelSpacing	454
27.149.2.18	SetForceRescaleInterceptSlope	454
27.149.2.19	SetOriginValue	454
27.149.2.20	SetRescaleInterceptSlopeValue	454
27.149.2.21	SetSpacingValue	454
27.150	dcm::ImageReader Class Reference	455
27.150.1	Detailed Description	457
27.150.2	Constructor & Destructor Documentation	457
27.150.2.1	ImageReader	457
27.150.2.2	~ImageReader	457
27.150.3	Member Function Documentation	457
27.150.3.1	GetImage	457
27.150.3.2	GetImage	457
27.150.3.3	Read	457
27.150.3.4	ReadACRNEMAImage	458
27.150.3.5	ReadImage	458
27.151	dcm::ImageRegionReader Class Reference	458
27.151.1	Detailed Description	460
27.151.2	Constructor & Destructor Documentation	460
27.151.2.1	ImageRegionReader	460
27.151.2.2	~ImageRegionReader	460
27.151.3	Member Function Documentation	460
27.151.3.1	ComputeBufferLength	460
27.151.3.2	GetRegion	460
27.151.3.3	Read	460
27.151.3.4	ReadInformation	460
27.151.3.5	ReadIntoBuffer	461
27.151.3.6	SetRegion	461
27.152	dcm::ImageToImageFilter Class Reference	461
27.152.1	Detailed Description	462

27.152.2	Constructor & Destructor Documentation	. 462
27.152.2.1	ImageToImageFilter	. 463
27.152.2.2	~ImageToImageFilter	. 463
27.152.3	Member Function Documentation	. 463
27.152.3.1	GetInput	. 463
27.152.3.2	GetOutput	. 463
27.153	dcm::ImageWriter Class Reference	. 463
27.153.1	Detailed Description	. 465
27.153.2	Constructor & Destructor Documentation	. 465
27.153.2.1	ImageWriter	. 465
27.153.2.2	~ImageWriter	. 465
27.153.3	Member Function Documentation	. 465
27.153.3.1	GetImage	. 465
27.153.3.2	GetImage	. 465
27.153.3.3	Write	. 465
27.154	dcm::network::ImplementationClassUIDSub Class Reference	. 466
27.154.1	Detailed Description	. 466
27.154.2	Constructor & Destructor Documentation	. 466
27.154.2.1	ImplementationClassUIDSub	. 466
27.154.3	Member Function Documentation	. 466
27.154.3.1	Print	. 466
27.154.3.2	Read	. 466
27.154.3.3	Size	. 466
27.154.3.4	Write	. 466
27.155	dcm::network::ImplementationUIDSub Class Reference	. 466
27.155.1	Detailed Description	. 467
27.155.2	Constructor & Destructor Documentation	. 467
27.155.2.1	ImplementationUIDSub	. 467
27.155.3	Member Function Documentation	. 467
27.155.3.1	Write	. 467
27.156	dcm::network::ImplementationVersionNameSub Class Reference	. 467
27.156.1	Detailed Description	. 467
27.156.2	Constructor & Destructor Documentation	. 467
27.156.2.1	ImplementationVersionNameSub	. 467
27.156.3	Member Function Documentation	. 467
27.156.3.1	Print	. 467
27.156.3.2	Read	. 467

27.156.3.3Size	. 467
27.156.3.4Write	. 468
27.157dcm::ImplicitDataElement Class Reference	. 468
27.157.1Detailed Description	. 469
27.157.2Member Function Documentation	. 469
27.157.2.1GetLength	. 469
27.157.2.2Read	. 469
27.157.2.3ReadPreValue	. 469
27.157.2.4ReadValue	. 469
27.157.2.5ReadValueWithLength	. 469
27.157.2.6ReadWithLength	. 469
27.157.2.7Write	. 469
27.158dcm::InitializeEvent Class Reference	. 470
27.159dcm::IOD Class Reference	. 471
27.159.1Detailed Description	. 471
27.159.2Member Typedef Documentation	. 471
27.159.2.1MapIODEntry	. 471
27.159.2.2SizeType	. 471
27.159.3Constructor & Destructor Documentation	. 471
27.159.3.1IOD	. 471
27.159.4Member Function Documentation	. 472
27.159.4.1AddIODEntry	. 472
27.159.4.2Clear	. 472
27.159.4.3GetIODEntry	. 472
27.159.4.4GetNumberOfIODs	. 472
27.159.4.5GetTypeFromTag	. 472
27.159.5Friends And Related Function Documentation	. 472
27.159.5.1operator<<	. 472
27.160dcm::IODEntry Class Reference	. 472
27.160.1Detailed Description	. 473
27.160.2Constructor & Destructor Documentation	. 473
27.160.2.1IODEntry	. 473
27.160.3Member Function Documentation	. 473
27.160.3.1GetIE	. 473
27.160.3.2GetName	. 473
27.160.3.3GetRef	. 473
27.160.3.4GetUsage	. 474

27.160.3.5	GetUsageType	474
27.160.3.6	SetIE	474
27.160.3.7	SetName	474
27.160.3.8	SetRef	474
27.160.3.9	SetUsage	474
27.160.4	Friends And Related Function Documentation	474
27.160.4.1	operator<<	474
27.161	gdcm::IODs Class Reference	474
27.161.1	Detailed Description	475
27.161.2	Member Typedef Documentation	475
27.161.2.1	IODMapType	475
27.161.2.2	IODMapTypeConstIterator	475
27.161.2.3	IODName	475
27.161.3	Constructor & Destructor Documentation	475
27.161.3.1	IODs	475
27.161.4	Member Function Documentation	475
27.161.4.1	AddIOD	475
27.161.4.2	Begin	475
27.161.4.3	Clear	475
27.161.4.4	End	475
27.161.4.5	GetIOD	475
27.161.5	Friends And Related Function Documentation	475
27.161.5.1	operator<<	476
27.162	gdcm::IPPSorter Class Reference	476
27.162.1	Detailed Description	477
27.162.2	Constructor & Destructor Documentation	478
27.162.2.1	IPPSorter	478
27.162.3	Member Function Documentation	478
27.162.3.1	GetDirectionCosinesTolerance	478
27.162.3.2	GetZSpacing	478
27.162.3.3	GetZSpacingTolerance	478
27.162.3.4	SetComputeZSpacing	478
27.162.3.5	SetDirectionCosinesTolerance	478
27.162.3.6	SetDropDuplicatePositions	478
27.162.3.7	SetZSpacingTolerance	479
27.162.3.8	Sort	479
27.162.4	Member Data Documentation	479

27.162.4.1ComputeZSpacing	479
27.162.4.2DirCosTolerance	479
27.162.4.3DropDuplicatePositions	479
27.162.4.4ZSpacing	479
27.162.4.5ZTolerance	479
27.163dcm::Item Class Reference	479
27.163.1Detailed Description	481
27.163.2Constructor & Destructor Documentation	481
27.163.2.1Item	481
27.163.2.2Item	481
27.163.3Member Function Documentation	481
27.163.3.1Clear	481
27.163.3.2FindDataElement	481
27.163.3.3GetDataElement	482
27.163.3.4GetLength	482
27.163.3.5GetNestedDataSet	482
27.163.3.6GetNestedDataSet	482
27.163.3.7InsertDataElement	482
27.163.3.8Read	482
27.163.3.9SetNestedDataSet	482
27.163.3.10Write	482
27.163.4Friends And Related Function Documentation	482
27.163.4.1operator<<	482
27.164dcm::IterationEvent Class Reference	482
27.165dcm::JPEG12Codec Class Reference	484
27.165.1Detailed Description	485
27.165.2Constructor & Destructor Documentation	485
27.165.2.1JPEG12Codec	485
27.165.2.2~JPEG12Codec	485
27.165.3Member Function Documentation	485
27.165.3.1DecodeByStreams	485
27.165.3.2EncodeBuffer	485
27.165.3.3GetHeaderInfo	485
27.165.3.4InternalCode	485
27.165.3.5IsStateSuspension	485
27.166dcm::JPEG16Codec Class Reference	486
27.166.1Detailed Description	487

27.166.2	Constructor & Destructor Documentation	. 487
27.166.2.1	JPEG16Codec	. 487
27.166.2.2	~JPEG16Codec	. 487
27.166.3	Member Function Documentation	. 487
27.166.3.1	DecodeByStreams	. 487
27.166.3.2	EncodeBuffer	. 487
27.166.3.3	GetHeaderInfo	. 487
27.166.3.4	InternalCode	. 487
27.166.3.5	IsStateSuspension	. 487
27.167	gdcm::JPEG2000Codec Class Reference	. 488
27.167.1	Detailed Description	. 489
27.167.2	Constructor & Destructor Documentation	. 489
27.167.2.1	JPEG2000Codec	. 489
27.167.2.2	~JPEG2000Codec	. 489
27.167.3	Member Function Documentation	. 490
27.167.3.1	AppendFrameEncode	. 490
27.167.3.2	AppendRowEncode	. 490
27.167.3.3	CanCode	. 490
27.167.3.4	CanDecode	. 490
27.167.3.5	Clone	. 490
27.167.3.6	Code	. 490
27.167.3.7	Decode	. 490
27.167.3.8	DecodeByStreams	. 490
27.167.3.9	DecodeExtent	. 490
27.167.3.10	GetHeaderInfo	. 491
27.167.3.11	GetQuality	. 491
27.167.3.12	GetRate	. 491
27.167.3.13	FrameEncoder	. 491
27.167.3.14	RowEncoder	. 491
27.167.3.15	SetNumberOfResolutions	. 491
27.167.3.16	SetQuality	. 491
27.167.3.17	SetRate	. 491
27.167.3.18	SetReversible	. 491
27.167.3.19	SetTileSize	. 491
27.167.3.20	StartEncode	. 491
27.167.3.21	StopEncode	. 491
27.167.4	Friends And Related Function Documentation	. 491

27.167.4.1	Bitmap	491
27.167.4.2	ImageRegionReader	491
27.168	gdcm::JPEG8Codec Class Reference	491
27.168.1	Detailed Description	493
27.168.2	Constructor & Destructor Documentation	493
27.168.2.1	JPEG8Codec	493
27.168.2.2	~JPEG8Codec	493
27.168.3	Member Function Documentation	493
27.168.3.1	DecodeByStreams	493
27.168.3.2	EncodeBuffer	493
27.168.3.3	GetHeaderInfo	493
27.168.3.4	InternalCode	493
27.168.3.5	IsStateSuspension	493
27.169	gdcm::JPEGCodec Class Reference	494
27.169.1	Detailed Description	495
27.169.2	Constructor & Destructor Documentation	496
27.169.2.1	JPEGCodec	496
27.169.2.2	~JPEGCodec	496
27.169.3	Member Function Documentation	496
27.169.3.1	AppendFrameEncode	496
27.169.3.2	AppendRowEncode	496
27.169.3.3	CanCode	496
27.169.3.4	CanDecode	496
27.169.3.5	Clone	496
27.169.3.6	Code	497
27.169.3.7	ComputeOffsetTable	497
27.169.3.8	Decode	497
27.169.3.9	DecodeByStreams	497
27.169.3.10	DecodeExtent	497
27.169.3.11	EncodeBuffer	497
27.169.3.12	GetHeaderInfo	497
27.169.3.13	GetLossless	497
27.169.3.14	GetQuality	497
27.169.3.15	FrameEncoder	497
27.169.3.16	RowEncoder	497
27.169.3.17	IsStateSuspension	498
27.169.3.18	Valid	498

27.169.3.1SetBitSample	498
27.169.3.2SetLossless	498
27.169.3.3SetPixelFormat	498
27.169.3.4SetQuality	498
27.169.3.5StartEncode	498
27.169.3.6StopEncode	498
27.169.4Friends And Related Function Documentation	498
27.169.4.1ImageRegionReader	498
27.169.5Member Data Documentation	498
27.169.5.1BitSample	498
27.169.5.2Quality	498
27.170dcm::JPEGLSCodec Class Reference	498
27.170.1Detailed Description	500
27.170.2Constructor & Destructor Documentation	500
27.170.2.1JPEGLSCodec	500
27.170.2.2~JPEGLSCodec	500
27.170.3Member Function Documentation	500
27.170.3.1AppendFrameEncode	500
27.170.3.2AppendRowEncode	501
27.170.3.3CanCode	501
27.170.3.4CanDecode	501
27.170.3.5Clone	501
27.170.3.6Code	501
27.170.3.7Decode	501
27.170.3.8Decode	501
27.170.3.9DecodeExtent	501
27.170.3.10GetBufferLength	501
27.170.3.11GetHeaderInfo	501
27.170.3.12GetLossless	501
27.170.3.13FrameEncoder	501
27.170.3.14RowEncoder	502
27.170.3.15SetBufferLength	502
27.170.3.16SetLossless	502
27.170.3.17SetLossyError	502
27.170.3.18StartEncode	502
27.170.3.19StopEncode	502
27.170.4Friends And Related Function Documentation	502

27.170.4.1ImageRegionReader	502
27.171dcm::JSON Class Reference	502
27.171.1Detailed Description	502
27.171.2Constructor & Destructor Documentation	503
27.171.2.1JSON	503
27.171.2.2~JSON	503
27.171.3Member Function Documentation	503
27.171.3.1Code	503
27.171.3.2Decode	503
27.171.3.3GetPrettyPrint	503
27.171.3.4PrettyPrintOff	503
27.171.3.5PrettyPrintOn	503
27.171.3.6SetPrettyPrint	503
27.172dcm::KAKADUCodec Class Reference	503
27.172.1Detailed Description	505
27.172.2Constructor & Destructor Documentation	505
27.172.2.1KAKADUCodec	505
27.172.2.2~KAKADUCodec	505
27.172.3Member Function Documentation	505
27.172.3.1CanCode	505
27.172.3.2CanDecode	505
27.172.3.3Clone	505
27.172.3.4Code	505
27.172.3.5Decode	505
27.173dcm::LO Class Reference	505
27.173.1Detailed Description	507
27.173.2Member Typedef Documentation	507
27.173.2.1const_iterator	507
27.173.2.2const_reference	507
27.173.2.3const_reverse_iterator	507
27.173.2.4difference_type	507
27.173.2.5iterator	507
27.173.2.6pointer	507
27.173.2.7reference	507
27.173.2.8reverse_iterator	507
27.173.2.9size_type	507
27.173.2.10Superclass	507

27.173.2.1Value_type	507
27.173.3Constructor & Destructor Documentation	507
27.173.3.1LO	507
27.173.3.2LO	507
27.173.3.3LO	507
27.173.3.4LO	507
27.173.4Member Function Documentation	507
27.173.4.1IsValid	507
27.174dcm::LookupTable Class Reference	508
27.174.1Detailed Description	510
27.174.2Member Enumeration Documentation	510
27.174.2.1LookupTableType	510
27.174.3Constructor & Destructor Documentation	510
27.174.3.1LookupTable	510
27.174.3.2~LookupTable	510
27.174.3.3LookupTable	510
27.174.4Member Function Documentation	510
27.174.4.1Allocate	510
27.174.4.2Clear	510
27.174.4.3Decode	510
27.174.4.4Decode	511
27.174.4.5GetBitSample	511
27.174.4.6GetBufferAsRGBA	511
27.174.4.7GetLUT	511
27.174.4.8GetLUTDescriptor	511
27.174.4.9GetLUTLength	511
27.174.4.10GetPointer	511
27.174.4.11InitializeBlueLUT	511
27.174.4.12Initialized	511
27.174.4.13InitializeGreenLUT	511
27.174.4.14InitializeLUT	511
27.174.4.15InitializeRedLUT	511
27.174.4.16Print	512
27.174.4.17SetBlueLUT	512
27.174.4.18SetGreenLUT	512
27.174.4.19SetLUT	512
27.174.4.20SetRedLUT	512

27.174.4.2WriteBufferAsRGBA	512
27.174.5Member Data Documentation	512
27.174.5.1BitSample	512
27.174.5.2IncompleteLUT	512
27.174.5.3Internal	512
27.175dcm::Scanner::Itstr Struct Reference	512
27.175.1Member Function Documentation	512
27.175.1.1operator()	512
27.176dcm::StrictScanner::Itstr Struct Reference	513
27.176.1Member Function Documentation	513
27.176.1.1operator()	513
27.177dcm::Macro Class Reference	513
27.177.1Detailed Description	514
27.177.2Member Typedef Documentation	514
27.177.2.1ArrayIncludeMacrosType	514
27.177.2.2MapModuleEntry	514
27.177.3Constructor & Destructor Documentation	514
27.177.3.1Macro	514
27.177.4Member Function Documentation	514
27.177.4.1AddMacroEntry	514
27.177.4.2Clear	514
27.177.4.3FindMacroEntry	514
27.177.4.4GetMacroEntry	514
27.177.4.5GetName	514
27.177.4.6SetName	514
27.177.4.7Verify	514
27.177.5Friends And Related Function Documentation	514
27.177.5.1operator<<	514
27.178dcm::Macros Class Reference	515
27.178.1Detailed Description	515
27.178.2Member Typedef Documentation	515
27.178.2.1ModuleMapType	515
27.178.3Constructor & Destructor Documentation	515
27.178.3.1Macros	515
27.178.4Member Function Documentation	515
27.178.4.1AddMacro	515
27.178.4.2Clear	516

27.178.4.3	GetMacro	516
27.178.4.4	IsEmpty	516
27.178.5	Friends And Related Function Documentation	516
27.178.5.1	operator<<	516
27.179	dcm::network::MaximumLengthSub Class Reference	516
27.179.1	Detailed Description	516
27.179.2	Constructor & Destructor Documentation	516
27.179.2.1	MaximumLengthSub	516
27.179.3	Member Function Documentation	516
27.179.3.1	GetMaximumLength	516
27.179.3.2	Print	516
27.179.3.3	Read	516
27.179.3.4	SetMaximumLength	517
27.179.3.5	Size	517
27.179.3.6	Write	517
27.180	dcm::MD5 Class Reference	517
27.180.1	Detailed Description	517
27.180.2	Constructor & Destructor Documentation	517
27.180.2.1	MD5	517
27.180.2.2	~MD5	517
27.180.3	Member Function Documentation	517
27.180.3.1	Compute	517
27.180.3.2	ComputeFile	518
27.181	dcm::MediaStorage Class Reference	518
27.181.1	Detailed Description	521
27.181.2	Member Enumeration Documentation	521
27.181.2.1	MSType	521
27.181.2.2	ObjectType	523
27.181.3	Constructor & Destructor Documentation	524
27.181.3.1	MediaStorage	524
27.181.4	Member Function Documentation	524
27.181.4.1	GetModality	524
27.181.4.2	GetModalityDimension	524
27.181.4.3	GetMSString	524
27.181.4.4	GetMSType	524
27.181.4.5	GetNumberOfModality	524
27.181.4.6	GetNumberOfMSString	524

27.181.4.7	GetNumberOfMSType	524
27.181.4.8	GetString	524
27.181.4.9	GuessFromModality	524
27.181.4.10	Image	524
27.181.4.11	Undefined	525
27.181.4.12	operator MSType	525
27.181.4.13	SetFromDataSet	525
27.181.4.14	SetFromFile	525
27.181.4.15	SetFromHeader	525
27.181.4.16	SetFromModality	525
27.181.4.17	SetFromSourceImageSequence	525
27.181.5	Friends And Related Function Documentation	525
27.181.5.1	operator<<	525
27.182	dcm::MemberCommand< T > Class Template Reference	525
27.182.1	Detailed Description	527
27.182.2	Member Typedef Documentation	527
27.182.2.1	Self	527
27.182.2.2	TConstMemberFunctionPointer	527
27.182.2.3	TMemberFunctionPointer	527
27.182.3	Constructor & Destructor Documentation	528
27.182.3.1	MemberCommand	528
27.182.3.2	~MemberCommand	528
27.182.4	Member Function Documentation	528
27.182.4.1	Execute	528
27.182.4.2	Execute	528
27.182.4.3	New	528
27.182.4.4	SetCallbackFunction	528
27.182.4.5	SetCallbackFunction	528
27.182.5	Member Data Documentation	528
27.182.5.1	m_ConstMemberFunction	529
27.182.5.2	m_MemberFunction	529
27.182.5.3	m_This	529
27.183	dcm::MeshPrimitive Class Reference	529
27.183.1	Detailed Description	531
27.183.2	Member Typedef Documentation	531
27.183.2.1	PrimitivesData	531
27.183.3	Member Enumeration Documentation	531

27.183.3.1MPTType	531
27.183.4Constructor & Destructor Documentation	532
27.183.4.1MeshPrimitive	532
27.183.4.2~MeshPrimitive	532
27.183.5Member Function Documentation	532
27.183.5.1AddPrimitiveData	532
27.183.5.2GetMPTType	532
27.183.5.3GetMPTTypeString	532
27.183.5.4GetNumberOfPrimitivesData	532
27.183.5.5GetPrimitiveData	532
27.183.5.6GetPrimitiveData	532
27.183.5.7GetPrimitiveData	532
27.183.5.8GetPrimitiveData	532
27.183.5.9GetPrimitivesData	532
27.183.5.10GetPrimitivesData	532
27.183.5.11GetPrimitiveType	532
27.183.5.12SetPrimitiveData	532
27.183.5.13SetPrimitiveData	532
27.183.5.14SetPrimitivesData	532
27.183.5.15SetPrimitiveType	532
27.183.6Member Data Documentation	532
27.183.6.1PrimitiveData	532
27.183.6.2PrimitiveType	532
27.184dcm::ModalityPerformedProcedureStepCreateQuery Class Reference	533
27.184.1Detailed Description	534
27.184.2Constructor & Destructor Documentation	534
27.184.2.1ModalityPerformedProcedureStepCreateQuery	535
27.184.3Member Function Documentation	535
27.184.3.1GetAbstractSyntaxUID	535
27.184.3.2GetRequiredDataSet	535
27.184.3.3ValidateQuery	535
27.184.4Friends And Related Function Documentation	535
27.184.4.1QueryFactory	535
27.185dcm::ModalityPerformedProcedureStepSetQuery Class Reference	535
27.185.1Detailed Description	537
27.185.2Constructor & Destructor Documentation	537
27.185.2.1ModalityPerformedProcedureStepSetQuery	537

27.185.3	Member Function Documentation	537
27.185.3.1	GetAbstractSyntaxUID	537
27.185.3.2	GetRequiredDataSet	537
27.185.3.3	ValidateQuery	537
27.185.4	Friends And Related Function Documentation	537
27.185.4.1	QueryFactory	537
27.186	dcm::ModifiedEvent Class Reference	537
27.187	dcm::Module Class Reference	539
27.187.1	Detailed Description	539
27.187.2	Member Typedef Documentation	539
27.187.2.1	ArrayIncludeMacrosType	539
27.187.2.2	MapModuleEntry	539
27.187.3	Constructor & Destructor Documentation	540
27.187.3.1	Module	540
27.187.4	Member Function Documentation	540
27.187.4.1	AddMacro	540
27.187.4.2	AddModuleEntry	540
27.187.4.3	Clear	540
27.187.4.4	FindModuleEntryInMacros	540
27.187.4.5	GetModuleEntryInMacros	540
27.187.4.6	GetName	540
27.187.4.7	SetName	540
27.187.4.8	Verify	540
27.187.5	Friends And Related Function Documentation	540
27.187.5.1	operator<<	540
27.188	dcm::ModuleEntry Class Reference	540
27.188.1	Detailed Description	542
27.188.2	Member Typedef Documentation	542
27.188.2.1	Description	542
27.188.3	Constructor & Destructor Documentation	542
27.188.3.1	ModuleEntry	542
27.188.3.2	~ModuleEntry	542
27.188.4	Member Function Documentation	542
27.188.4.1	GetDescription	542
27.188.4.2	GetName	542
27.188.4.3	GetType	543
27.188.4.4	SetDescription	543

27.188.4.5SetName	543
27.188.4.6SetType	543
27.188.5Friends And Related Function Documentation	543
27.188.5.1operator<<	543
27.188.6Member Data Documentation	543
27.188.6.1DataElementType	543
27.188.6.2DescriptionField	543
27.188.6.3Name	543
27.189dcm::Modules Class Reference	543
27.189.1Detailed Description	544
27.189.2Member Typedef Documentation	544
27.189.2.1ModuleMapType	544
27.189.3Constructor & Destructor Documentation	544
27.189.3.1Modules	544
27.189.4Member Function Documentation	544
27.189.4.1AddModule	544
27.189.4.2Clear	544
27.189.4.3GetModule	544
27.189.4.4IsEmpty	544
27.189.5Friends And Related Function Documentation	544
27.189.5.1operator<<	544
27.190dcm::MovePatientRootQuery Class Reference	545
27.190.1Detailed Description	546
27.190.2Constructor & Destructor Documentation	546
27.190.2.1MovePatientRootQuery	546
27.190.3Member Function Documentation	546
27.190.3.1GetAbstractSyntaxUID	546
27.190.3.2GetTagListByLevel	546
27.190.3.3InitializeDataSet	546
27.190.3.4ValidateQuery	546
27.190.4Friends And Related Function Documentation	547
27.190.4.1QueryFactory	547
27.191dcm::MoveStudyRootQuery Class Reference	547
27.191.1Detailed Description	548
27.191.2Constructor & Destructor Documentation	548
27.191.2.1MoveStudyRootQuery	548
27.191.3Member Function Documentation	548

27.191.3.1	GetAbstractSyntaxUID	548
27.191.3.2	GetTagListByLevel	548
27.191.3.3	InitializeDataSet	549
27.191.3.4	ValidateQuery	549
27.191.4	Friends And Related Function Documentation	549
27.191.4.1	QueryFactory	549
27.192	dcm::network::NActionRQ Class Reference	549
27.192.1	Detailed Description	550
27.192.2	Member Function Documentation	550
27.192.2.1	ConstructPDV	550
27.193	dcm::network::NActionRSP Class Reference	551
27.193.1	Detailed Description	552
27.193.2	Member Function Documentation	552
27.193.2.1	ConstructPDVByDataSet	552
27.194	dcm::network::NCreateRQ Class Reference	552
27.194.1	Detailed Description	553
27.194.2	Member Function Documentation	553
27.194.2.1	ConstructPDV	553
27.195	dcm::network::NCreateRSP Class Reference	553
27.195.1	Detailed Description	554
27.195.2	Member Function Documentation	554
27.195.2.1	ConstructPDVByDataSet	554
27.196	dcm::network::NDeleteRQ Class Reference	555
27.196.1	Detailed Description	555
27.196.2	Member Function Documentation	555
27.196.2.1	ConstructPDV	556
27.197	dcm::network::NDeleteRSP Class Reference	556
27.197.1	Detailed Description	557
27.197.2	Member Function Documentation	557
27.197.2.1	ConstructPDVByDataSet	557
27.198	dcm::NestedModuleEntries Class Reference	557
27.198.1	Detailed Description	558
27.198.2	Member Typedef Documentation	559
27.198.2.1	SizeType	559
27.198.3	Constructor & Destructor Documentation	559
27.198.3.1	NestedModuleEntries	559
27.198.4	Member Function Documentation	559

27.198.4.1AddModuleEntry	559
27.198.4.2GetModuleEntry	559
27.198.4.3GetModuleEntry	559
27.198.4.4GetNumberOfModuleEntries	559
27.198.5Friends And Related Function Documentation	559
27.198.5.1operator<<	559
27.199gdcmm::network::NEventReportRQ Class Reference	559
27.199.1Detailed Description	560
27.199.2Member Function Documentation	560
27.199.2.1ConstructPDV	560
27.200gdcmm::network::NEventReportRSP Class Reference	561
27.200.1Detailed Description	562
27.200.2Member Function Documentation	562
27.200.2.1ConstructPDVByDataSet	562
27.201gdcmm::network::NGetRQ Class Reference	562
27.201.1Detailed Description	563
27.201.2Member Function Documentation	563
27.201.2.1ConstructPDV	563
27.202gdcmm::network::NGetRSP Class Reference	563
27.202.1Detailed Description	564
27.202.2Member Function Documentation	564
27.202.2.1ConstructPDVByDataSet	564
27.203gdcmm::NoEvent Class Reference	565
27.203.1Detailed Description	565
27.204gdcmm::network::NormalizedMessageFactory Class Reference	565
27.204.1Member Function Documentation	566
27.204.1.1ConstructNAction	566
27.204.1.2ConstructNCreate	566
27.204.1.3ConstructNDelete	566
27.204.1.4ConstructNEventReport	566
27.204.1.5ConstructNGet	566
27.204.1.6ConstructNSet	566
27.205gdcmm::NormalizedNetworkFunctions Class Reference	566
27.205.1Detailed Description	567
27.205.2Member Function Documentation	567
27.205.2.1ConstructQuery	567
27.205.2.2NAction	567

27.205.2.3NCreate	567
27.205.2.4NDelete	567
27.205.2.5NEventReport	568
27.205.2.6NGet	568
27.205.2.7NSet	568
27.206dcm::network::NSetRQ Class Reference	568
27.206.1Detailed Description	569
27.206.2Member Function Documentation	569
27.206.2.1ConstructPDV	569
27.207dcm::network::NSetRSP Class Reference	569
27.207.1Detailed Description	570
27.207.2Member Function Documentation	570
27.207.2.1ConstructPDVByDataSet	570
27.208dcm::Object Class Reference	571
27.208.1Detailed Description	572
27.208.2Constructor & Destructor Documentation	572
27.208.2.1Object	572
27.208.2.2~Object	572
27.208.2.3Object	572
27.208.3Member Function Documentation	572
27.208.3.1operator=	572
27.208.3.2Print	572
27.208.3.3Register	573
27.208.3.4UnRegister	573
27.208.4Friends And Related Function Documentation	573
27.208.4.1operator<<	573
27.208.4.2SmartPointer	573
27.209dcm::OpenSSLCryptoFactory Class Reference	573
27.209.1Constructor & Destructor Documentation	574
27.209.1.1OpenSSLCryptoFactory	574
27.209.2Member Function Documentation	574
27.209.2.1CreateCMSProvider	574
27.209.2.2InitOpenSSL	574
27.210dcm::OpenSSLCryptographicMessageSyntax Class Reference	574
27.210.1Constructor & Destructor Documentation	576
27.210.1.1OpenSSLCryptographicMessageSyntax	576
27.210.1.2~OpenSSLCryptographicMessageSyntax	576

27.210.2	Member Function Documentation	576
27.210.2.1	Decrypt	576
27.210.2.2	Encrypt	576
27.210.2.3	GetCipherType	576
27.210.2.4	ParseCertificateFile	576
27.210.2.5	ParseKeyFile	576
27.210.2.6	SetCipherType	576
27.210.2.7	SetPassword	576
27.210	gdcm::OpenSSLP7CryptoFactory Class Reference	577
27.211	Constructor & Destructor Documentation	577
27.211.1	OpenSSLP7CryptoFactory	577
27.211.2	Member Function Documentation	578
27.211.2.1	CreateCMSProvider	578
27.210	gdcm::OpenSSLP7CryptographicMessageSyntax Class Reference	578
27.212	Detailed Description	579
27.212	Constructor & Destructor Documentation	579
27.212.2	OpenSSLP7CryptographicMessageSyntax	579
27.212.2.2	~OpenSSLP7CryptographicMessageSyntax	579
27.212.3	Member Function Documentation	579
27.212.3.1	Decrypt	580
27.212.3.2	Encrypt	580
27.212.3.3	GetCipherType	580
27.212.3.4	ParseCertificateFile	580
27.212.3.5	ParseKeyFile	580
27.212.3.6	SetCipherType	580
27.212.3.7	SetPassword	580
27.210	gdcm::Orientation Class Reference	580
27.213	Detailed Description	581
27.213.2	Member Enumeration Documentation	581
27.213.2.1	OrientationType	581
27.213.3	Constructor & Destructor Documentation	582
27.213.3.1	Orientation	582
27.213.3.2	~Orientation	582
27.213.4	Member Function Documentation	582
27.213.4.1	GetLabel	582
27.213.4.2	GetMajorAxisFromPatientRelativeDirectionCosine	582
27.213.4.3	GetObliquityThresholdCosineValue	582

27.213.4.4	GetType	582
27.213.4.5	Print	582
27.213.4.6	SetObliquityThresholdCosineValue	582
27.213.5	Friends And Related Function Documentation	582
27.213.5.1	operator<<	582
27.214	dcm::Overlay Class Reference	583
27.214.1	Detailed Description	585
27.214.2	Member Enumeration Documentation	585
27.214.2.1	OverlayType	585
27.214.3	Constructor & Destructor Documentation	585
27.214.3.1	Overlay	585
27.214.3.2	~Overlay	585
27.214.3.3	Overlay	585
27.214.4	Member Function Documentation	586
27.214.4.1	Decompress	586
27.214.4.2	GetBitPosition	586
27.214.4.3	GetBitsAllocated	586
27.214.4.4	GetColumns	586
27.214.4.5	GetDescription	586
27.214.4.6	GetGroup	586
27.214.4.7	GetOrigin	586
27.214.4.8	GetOverlayData	586
27.214.4.9	GetOverlayTypeAsString	586
27.214.4.10	GetOverlayTypeFromString	586
27.214.4.11	GetRows	586
27.214.4.12	GetType	586
27.214.4.13	GetTypeAsEnum	587
27.214.4.14	GetUnpackBuffer	587
27.214.4.15	GetUnpackBufferLength	587
27.214.4.16	GrabOverlayFromPixelData	587
27.214.4.17	IsEmpty	587
27.214.4.18	InPixelData	587
27.214.4.19	InPixelData	587
27.214.4.20	Zero	587
27.214.4.21	operator=	587
27.214.4.22	Print	587
27.214.4.23	SetBitPosition	587

27.214.4.28	SetBitsAllocated	587
27.214.4.29	SetColumns	587
27.214.4.30	SetDescription	588
27.214.4.31	SetFrameOrigin	588
27.214.4.32	SetGroup	588
27.214.4.33	SetNumberOfFrames	588
27.214.4.34	SetOrigin	588
27.214.4.35	SetOverlay	588
27.214.4.36	SetRows	588
27.214.4.37	SetType	588
27.214.4.38	Update	588
27.215	dcm::ParseException Class Reference	588
27.215.1	Detailed Description	590
27.215.2	Constructor & Destructor Documentation	590
27.215.2.1	ParseException	590
27.215.2.2	~ParseException	590
27.215.3	Member Function Documentation	590
27.215.3.1	GetLastElement	590
27.215.3.2	operator=	590
27.215.3.3	SetLastElement	590
27.216	dcm::Parser Class Reference	590
27.216.1	Detailed Description	591
27.216.2	Member Typedef Documentation	591
27.216.2.1	EndElementHandler	591
27.216.2.2	StartElementHandler	591
27.216.3	Member Enumeration Documentation	591
27.216.3.1	ErrorType	591
27.216.4	Constructor & Destructor Documentation	592
27.216.4.1	Parser	592
27.216.4.2	~Parser	592
27.216.5	Member Function Documentation	592
27.216.5.1	GetBuffer	592
27.216.5.2	GetCurrentByteIndex	592
27.216.5.3	GetErrorCode	592
27.216.5.4	GetErrorString	592
27.216.5.5	GetUserData	592
27.216.5.6	Parse	592

27.216.5.7	ParseBuffer	592
27.216.5.8	Process	592
27.216.5.9	SetElementHandler	592
27.216.5.10	SetUserData	592
27.217	dcm::Patient Class Reference	592
27.217.1	Detailed Description	592
27.217.2	Constructor & Destructor Documentation	593
27.217.2.1	Patient	593
27.218	dcm::network::PDataTFPDU Class Reference	593
27.218.1	Detailed Description	594
27.218.2	Member Typedef Documentation	594
27.218.2.1	SizeType	594
27.218.3	Constructor & Destructor Documentation	594
27.218.3.1	PDataTFPDU	594
27.218.4	Member Function Documentation	594
27.218.4.1	AddPresentationDataValue	594
27.218.4.2	GetNumberOfPresentationDataValues	594
27.218.4.3	GetPresentationDataValue	594
27.218.4.4	IsLastFragment	594
27.218.4.5	Print	594
27.218.4.6	Read	594
27.218.4.7	ReadInto	595
27.218.4.8	Size	595
27.218.4.9	Write	595
27.219	dcm::PDBelement Class Reference	595
27.219.1	Detailed Description	596
27.219.2	Constructor & Destructor Documentation	596
27.219.2.1	PDBelement	596
27.219.3	Member Function Documentation	596
27.219.3.1	GetName	596
27.219.3.2	GetValue	596
27.219.3.3	operator==	596
27.219.3.4	SetName	596
27.219.3.5	SetValue	596
27.219.4	Friends And Related Function Documentation	596
27.219.4.1	operator<<	596
27.219.5	Member Data Documentation	597

27.219.5.1NameField	597
27.219.5.2ValueField	597
27.220dcm::PDBHeader Class Reference	597
27.220.1Detailed Description	598
27.220.2Constructor & Destructor Documentation	598
27.220.2.1PDBHeader	598
27.220.2.2~PDBHeader	598
27.220.3Member Function Documentation	598
27.220.3.1FindPDBElementByName	598
27.220.3.2GetPDBEEnd	598
27.220.3.3GetPDBElementByName	598
27.220.3.4GetPDBInfoTag	598
27.220.3.5LoadFromDataElement	598
27.220.3.6Print	599
27.220.4Friends And Related Function Documentation	599
27.220.4.1operator<<	599
27.221dcm::PDFCodec Class Reference	599
27.221.1Detailed Description	600
27.221.2Constructor & Destructor Documentation	600
27.221.2.1PDFCodec	600
27.221.2.2~PDFCodec	600
27.221.3Member Function Documentation	600
27.221.3.1CanCode	600
27.221.3.2CanDecode	601
27.221.3.3Decode	601
27.222dcm::network::PDUFactory Class Reference	601
27.222.1Detailed Description	602
27.222.2Member Function Documentation	602
27.222.2.1ConstructAbortPDU	602
27.222.2.2ConstructPDU	602
27.222.2.3ConstructReleasePDU	602
27.222.2.4CreateCEchoPDU	602
27.222.2.5CreateCFindPDU	602
27.222.2.6CreateCMovePDU	602
27.222.2.7CreateCStoreRQPDU	602
27.222.2.8CreateCStoreRSPPDU	602
27.222.2.9CreateNActionPDU	602

27.222.2.10createNCreatePDU	602
27.222.2.10createNDeletePDU	602
27.222.2.10createNEventReportPDU	602
27.222.2.10createNGetPDU	602
27.222.2.10createNSetPDU	602
27.222.2.10etermineEventByPDU	602
27.222.2.10GetPDVs	602
27.223dcm::PersonName Class Reference	603
27.223.1Detailed Description	603
27.223.2Member Function Documentation	603
27.223.2.1GetMaxLength	603
27.223.2.2GetNumberOfComponents	603
27.223.2.3Print	603
27.223.2.4SetBlob	603
27.223.2.5SetComponents	603
27.223.2.6SetComponents	603
27.223.3Member Data Documentation	604
27.223.3.1Component	604
27.223.3.2MaxLength	604
27.223.3.3MaxNumberOfComponents	604
27.223.3.4Padding	604
27.223.3.5Separator	604
27.224dcm::PGXCodec Class Reference	604
27.224.1Detailed Description	605
27.224.2Constructor & Destructor Documentation	605
27.224.2.1PGXCodec	605
27.224.2.2~PGXCodec	605
27.224.3Member Function Documentation	605
27.224.3.1CanCode	605
27.224.3.2CanDecode	606
27.224.3.3Clone	606
27.224.3.4GetHeaderInfo	606
27.224.3.5Read	606
27.224.3.6Write	606
27.225dcm::PhotometricInterpretation Class Reference	606
27.225.1Detailed Description	607
27.225.2Member Enumeration Documentation	607

27.225.2.1PIType	607
27.225.3Constructor & Destructor Documentation	608
27.225.3.1PhotometricInterpretation	608
27.225.4Member Function Documentation	608
27.225.4.1GetPIString	608
27.225.4.2GetPIType	608
27.225.4.3GetSamplesPerPixel	608
27.225.4.4GetString	608
27.225.4.5GetType	608
27.225.4.6IsLossless	608
27.225.4.7IsLossy	608
27.225.4.8IsRetired	608
27.225.4.9IsSameColorSpace	608
27.225.4.10Operator PIType	608
27.225.5Friends And Related Function Documentation	608
27.225.5.1operator<<	608
27.226gdcm::PixelFormat Class Reference	608
27.226.1Detailed Description	610
27.226.2Member Enumeration Documentation	610
27.226.2.1ScalarType	610
27.226.3Constructor & Destructor Documentation	611
27.226.3.1PixelFormat	611
27.226.3.2PixelFormat	611
27.226.4Member Function Documentation	611
27.226.4.1GetBitsAllocated	611
27.226.4.2GetBitsStored	611
27.226.4.3GetHighBit	611
27.226.4.4GetMax	611
27.226.4.5GetMin	612
27.226.4.6GetPixelRepresentation	612
27.226.4.7GetPixelSize	612
27.226.4.8GetSamplesPerPixel	612
27.226.4.9GetScalarType	612
27.226.4.10GetScalarTypeAsString	612
27.226.4.11IsCompatible	612
27.226.4.12IsValid	612
27.226.4.13Operator ScalarType	612

27.226.4.10operator"!=	612
27.226.4.15operator"!=	612
27.226.4.16operator==	613
27.226.4.17operator==	613
27.226.4.18Print	613
27.226.4.19SetBitsAllocated	613
27.226.4.20SetBitsStored	613
27.226.4.21SetHighBit	613
27.226.4.22SetPixelRepresentation	613
27.226.4.23SetSamplesPerPixel	613
27.226.4.24SetScalarType	613
27.226.4.25Validate	613
27.226.5Friends And Related Function Documentation	613
27.226.5.1Bitmap	613
27.226.5.2operator<<	613
27.227dcm::Pixmap Class Reference	614
27.227.1Detailed Description	615
27.227.2Constructor & Destructor Documentation	615
27.227.2.1Pixmap	615
27.227.2.2~Pixmap	615
27.227.3Member Function Documentation	615
27.227.3.1AreOverlaysInPixelData	616
27.227.3.2GetCurve	616
27.227.3.3GetCurve	616
27.227.3.4GetIconImage	616
27.227.3.5GetIconImage	616
27.227.3.6GetNumberOfCurves	616
27.227.3.7GetNumberOfOverlays	616
27.227.3.8GetOverlay	616
27.227.3.9GetOverlay	616
27.227.3.10Print	616
27.227.3.11RemoveOverlay	616
27.227.3.12SetIconImage	616
27.227.3.13SetNumberOfCurves	616
27.227.3.14SetNumberOfOverlays	616
27.227.4Member Data Documentation	616
27.227.4.1Curves	616

27.227.4.2	con	616
27.227.4.3	Overlays	616
27.228	gdcm::PixmapReader Class Reference	617
27.228.1	Detailed Description	619
27.228.2	Constructor & Destructor Documentation	619
27.228.2.1	PixmapReader	619
27.228.2.2	~PixmapReader	619
27.228.3	Member Function Documentation	619
27.228.3.1	GetPixmap	619
27.228.3.2	GetPixmap	619
27.228.3.3	Read	619
27.228.3.4	ReadACRNEMAIImage	619
27.228.3.5	ReadImage	619
27.228.3.6	ReadImageInternal	620
27.228.4	Member Data Documentation	620
27.228.4.1	PixelData	620
27.229	gdcm::PixmapToPixmapFilter Class Reference	620
27.229.1	Detailed Description	621
27.229.2	Constructor & Destructor Documentation	621
27.229.2.1	PixmapToPixmapFilter	621
27.229.2.2	~PixmapToPixmapFilter	622
27.229.3	Member Function Documentation	622
27.229.3.1	GetInput	622
27.229.3.2	GetOutput	622
27.229.3.3	GetOutputAsPixmap	622
27.230	gdcm::PixmapWriter Class Reference	622
27.230.1	Detailed Description	624
27.230.2	Constructor & Destructor Documentation	624
27.230.2.1	PixmapWriter	624
27.230.2.2	~PixmapWriter	624
27.230.3	Member Function Documentation	624
27.230.3.1	DolconImage	624
27.230.3.2	GetImage	624
27.230.3.3	GetImage	624
27.230.3.4	GetPixmap	624
27.230.3.5	GetPixmap	624
27.230.3.6	PrepareWrite	624

27.230.3.7PrepareWrite	624
27.230.3.8SetImage	625
27.230.3.9SetPixmap	625
27.230.3.10Write	625
27.230.4Member Data Documentation	625
27.230.4.1PixelData	625
27.231gdcmm::PNMCodec Class Reference	625
27.231.1Detailed Description	627
27.231.2Constructor & Destructor Documentation	627
27.231.2.1PNMCodec	627
27.231.2.2~PNMCodec	627
27.231.3Member Function Documentation	627
27.231.3.1CanCode	627
27.231.3.2CanDecode	627
27.231.3.3Clone	627
27.231.3.4GetBufferLength	627
27.231.3.5GetHeaderInfo	627
27.231.3.6Read	628
27.231.3.7SetBufferLength	628
27.231.3.8Write	628
27.232gdcmm::Preamble Class Reference	628
27.232.1Detailed Description	628
27.232.2Constructor & Destructor Documentation	629
27.232.2.1Preamble	629
27.232.2.2~Preamble	629
27.232.2.3Preamble	629
27.232.3Member Function Documentation	629
27.232.3.1Clear	629
27.232.3.2Create	629
27.232.3.3GetInternal	629
27.232.3.4GetLength	629
27.232.3.5IsEmpty	629
27.232.3.6IsValid	629
27.232.3.7operator=	629
27.232.3.8Print	629
27.232.3.9Read	629
27.232.3.10Remove	629

27.232.3.1Valid	629
27.232.3.1Write	629
27.232.4Friends And Related Function Documentation	629
27.232.4.1operator<<	629
27.233dcm::PresentationContext Class Reference	629
27.233.1Detailed Description	631
27.233.2Member Typedef Documentation	631
27.233.2.1SizeType	631
27.233.2.2TransferSyntaxArrayType	631
27.233.3Constructor & Destructor Documentation	631
27.233.3.1PresentationContext	631
27.233.3.2PresentationContext	631
27.233.4Member Function Documentation	631
27.233.4.1AddTransferSyntax	631
27.233.4.2GetAbstractSyntax	631
27.233.4.3GetNumberOfTransferSyntaxes	631
27.233.4.4GetPresentationContextID	631
27.233.4.5GetTransferSyntax	631
27.233.4.6operator==	631
27.233.4.7Print	631
27.233.4.8SetAbstractSyntax	631
27.233.4.9SetPresentationContextID	632
27.233.5Member Data Documentation	632
27.233.5.1AbstractSyntax	632
27.233.5.2ID	632
27.233.5.3TransferSyntaxes	632
27.234dcm::network::PresentationContextAC Class Reference	632
27.234.1Detailed Description	632
27.234.2Constructor & Destructor Documentation	633
27.234.2.1PresentationContextAC	633
27.234.3Member Function Documentation	633
27.234.3.1GetPresentationContextID	633
27.234.3.2GetReason	633
27.234.3.3GetTransferSyntax	633
27.234.3.4Print	633
27.234.3.5Read	633
27.234.3.6SetPresentationContextID	633

27.234.3.7SetReason	633
27.234.3.8SetTransferSyntax	633
27.234.3.9Size	633
27.234.3.10Write	633
27.235dcm::PresentationContextGenerator Class Reference	633
27.235.1Detailed Description	634
27.235.2Member Typedef Documentation	634
27.235.2.1PresentationContextArrayType	634
27.235.2.2SizeType	634
27.235.3Constructor & Destructor Documentation	634
27.235.3.1PresentationContextGenerator	634
27.235.4Member Function Documentation	634
27.235.4.1AddFromFile	634
27.235.4.2AddPresentationContext	635
27.235.4.3GenerateFromFilenames	635
27.235.4.4GenerateFromUID	635
27.235.4.5GetDefaultTransferSyntax	635
27.235.4.6GetPresentationContexts	635
27.235.4.7SetDefaultTransferSyntax	635
27.235.4.8SetMergeModeToAbstractSyntax	635
27.235.4.9SetMergeModeToTransferSyntax	635
27.236dcm::network::PresentationContextRQ Class Reference	635
27.236.1Detailed Description	636
27.236.2Member Typedef Documentation	636
27.236.2.1SizeType	636
27.236.3Constructor & Destructor Documentation	636
27.236.3.1PresentationContextRQ	636
27.236.3.2PresentationContextRQ	636
27.236.3.3PresentationContextRQ	636
27.236.4Member Function Documentation	636
27.236.4.1AddTransferSyntax	636
27.236.4.2GetAbstractSyntax	637
27.236.4.3GetAbstractSyntax	637
27.236.4.4GetNumberOfTransferSyntaxes	637
27.236.4.5GetPresentationContextID	637
27.236.4.6GetTransferSyntax	637
27.236.4.7GetTransferSyntax	637

27.236.4.8	GetTransferSyntaxes	637
27.236.4.9	operator==	637
27.236.4.10	Print	637
27.236.4.11	Read	637
27.236.4.12	SetAbstractSyntax	637
27.236.4.13	SetPresentationContextID	637
27.236.4.14	Size	637
27.236.4.15	Write	637
27.237	gdcm::network::PresentationDataValue Class Reference	637
27.237.1	Detailed Description	638
27.237.2	Constructor & Destructor Documentation	638
27.237.2.1	PresentationDataValue	638
27.237.3	Member Function Documentation	638
27.237.3.1	ConcatenatePDVBlobs	638
27.237.3.2	ConcatenatePDVBlobsAsExplicit	638
27.237.3.3	GetBlob	638
27.237.3.4	GetIsCommand	638
27.237.3.5	GetIsLastFragment	638
27.237.3.6	GetMessageHeader	638
27.237.3.7	GetPresentationContextID	638
27.237.3.8	Print	638
27.237.3.9	Read	638
27.237.3.10	ReadInto	639
27.237.3.11	SetBlob	639
27.237.3.12	SetCommand	639
27.237.3.13	SetDataSet	639
27.237.3.14	SetLastFragment	639
27.237.3.15	SetMessageHeader	639
27.237.3.16	SetPresentationContextID	639
27.237.3.17	Size	639
27.237.3.18	Write	639
27.238	gdcm::Printer Class Reference	639
27.238.1	Detailed Description	641
27.238.2	Member Enumeration Documentation	641
27.238.2.1	PrintStyles	641
27.238.3	Constructor & Destructor Documentation	641
27.238.3.1	Printer	641

27.238.3.2~Printer	641
27.238.4Member Function Documentation	641
27.238.4.1GetPrintStyle	641
27.238.4.2Print	641
27.238.4.3PrintDataElement	642
27.238.4.4PrintDataSet	642
27.238.4.5PrintSQ	642
27.238.4.6SetColor	642
27.238.4.7SetFile	642
27.238.4.8SetStyle	642
27.238.5Member Data Documentation	642
27.238.5.1F	642
27.238.5.2MaxPrintLength	642
27.238.5.3PrintStyle	642
27.239dcm::PrivateDict Class Reference	642
27.239.1Detailed Description	643
27.239.2Constructor & Destructor Documentation	643
27.239.2.1PrivateDict	643
27.239.2.2~PrivateDict	643
27.239.3Member Function Documentation	643
27.239.3.1AddDictEntry	643
27.239.3.2FindDictEntry	643
27.239.3.3GetDictEntry	643
27.239.3.4IsEmpty	643
27.239.3.5LoadDefault	643
27.239.3.6PrintXML	643
27.239.3.7RemoveDictEntry	644
27.239.4Friends And Related Function Documentation	644
27.239.4.1Dicts	644
27.239.4.2operator<<	644
27.240dcm::PrivateTag Class Reference	644
27.240.1Detailed Description	645
27.240.2Constructor & Destructor Documentation	645
27.240.2.1PrivateTag	645
27.240.2.2PrivateTag	646
27.240.3Member Function Documentation	646
27.240.3.1GetAsDataElement	646

27.240.3.2	GetOwner	646
27.240.3.3	operator<	646
27.240.3.4	ReadFromCommaSeparatedString	646
27.240.3.5	SetOwner	646
27.240.4	Friends And Related Function Documentation	646
27.240.4.1	operator<<	646
27.241	gdcm::ProgressEvent Class Reference	646
27.241.1	Detailed Description	648
27.241.2	Member Typedef Documentation	648
27.241.2.1	Self	648
27.241.2.2	Superclass	648
27.241.3	Constructor & Destructor Documentation	648
27.241.3.1	ProgressEvent	648
27.241.3.2	~ProgressEvent	648
27.241.3.3	ProgressEvent	648
27.241.4	Member Function Documentation	648
27.241.4.1	CheckEvent	648
27.241.4.2	GetEventName	648
27.241.4.3	GetProgress	648
27.241.4.4	MakeObject	648
27.241.4.5	SetProgress	648
27.242	gdcm::PVRGCodec Class Reference	649
27.242.1	Detailed Description	650
27.242.2	Constructor & Destructor Documentation	650
27.242.2.1	PVRGCodec	650
27.242.2.2	~PVRGCodec	650
27.242.3	Member Function Documentation	650
27.242.3.1	CanCode	650
27.242.3.2	CanDecode	650
27.242.3.3	Clone	650
27.242.3.4	Code	651
27.242.3.5	Decode	651
27.242.3.6	SetLossyFlag	651
27.243	gdcm::PythonFilter Class Reference	651
27.243.1	Detailed Description	651
27.243.2	Constructor & Destructor Documentation	651
27.243.2.1	PythonFilter	651

27.243.2.2~PythonFilter	651
27.243.3Member Function Documentation	651
27.243.3.1GetFile	652
27.243.3.2GetFile	652
27.243.3.3SetDicts	652
27.243.3.4SetFile	652
27.243.3.5ToPyObject	652
27.243.3.6UseDictAlways	652
27.244dcm::QueryBase Class Reference	652
27.244.1Detailed Description	652
27.244.2Constructor & Destructor Documentation	653
27.244.2.1~QueryBase	653
27.244.3Member Function Documentation	653
27.244.3.1GetAllRequiredTags	653
27.244.3.2GetAllTags	653
27.244.3.3GetHierachicalSearchTags	653
27.244.3.4GetName	653
27.244.3.5GetOptionalTags	653
27.244.3.6GetQueryLevel	654
27.244.3.7GetRequiredTags	654
27.244.3.8GetUniqueTags	654
27.245dcm::QueryFactory Class Reference	654
27.245.1Detailed Description	654
27.245.2Member Function Documentation	655
27.245.2.1GetCharacterFromCurrentLocale	655
27.245.2.2ListCharSets	655
27.245.2.3ProduceCharacterSetDataElement	655
27.245.2.4ProduceQuery	655
27.245.2.5ProduceQuery	655
27.246dcm::QueryImage Class Reference	655
27.246.1Detailed Description	656
27.246.2Member Function Documentation	656
27.246.2.1GetHierachicalSearchTags	657
27.246.2.2GetName	657
27.246.2.3GetOptionalTags	657
27.246.2.4GetQueryLevel	657
27.246.2.5GetRequiredTags	657

27.246.2.6GetUniqueTags	657
27.247dcm::QueryPatient Class Reference	657
27.247.1Detailed Description	658
27.247.2Member Function Documentation	658
27.247.2.1GetHierarchicalSearchTags	659
27.247.2.2GetName	659
27.247.2.3GetOptionalTags	659
27.247.2.4GetQueryLevel	659
27.247.2.5GetRequiredTags	659
27.247.2.6GetUniqueTags	659
27.248dcm::QuerySeries Class Reference	659
27.248.1Detailed Description	660
27.248.2Member Function Documentation	660
27.248.2.1GetHierarchicalSearchTags	661
27.248.2.2GetName	661
27.248.2.3GetOptionalTags	661
27.248.2.4GetQueryLevel	661
27.248.2.5GetRequiredTags	661
27.248.2.6GetUniqueTags	661
27.249dcm::QueryStudy Class Reference	661
27.249.1Detailed Description	662
27.249.2Member Function Documentation	662
27.249.2.1GetHierarchicalSearchTags	663
27.249.2.2GetName	663
27.249.2.3GetOptionalTags	663
27.249.2.4GetQueryLevel	663
27.249.2.5GetRequiredTags	663
27.249.2.6GetUniqueTags	663
27.250dcm::RAWCodec Class Reference	663
27.250.1Detailed Description	665
27.250.2Constructor & Destructor Documentation	665
27.250.2.1RAWCodec	665
27.250.2.2~RAWCodec	665
27.250.3Member Function Documentation	665
27.250.3.1CanCode	665
27.250.3.2CanDecode	665
27.250.3.3Clone	665

27.250.3.4	Code	. 665
27.250.3.5	Decode	. 665
27.250.3.6	DecodeByStreams	. 666
27.250.3.7	DecodeBytes	. 666
27.250.3.8	GetHeaderInfo	. 666
27.250	gdcm::Reader Class Reference	. 666
27.251.1	Detailed Description	. 668
27.251.2	Constructor & Destructor Documentation	. 669
27.251.2.1	Reader	. 669
27.251.2.2	~Reader	. 669
27.251.3	Member Function Documentation	. 669
27.251.3.1	CanRead	. 669
27.251.3.2	GetFile	. 669
27.251.3.3	GetFile	. 669
27.251.3.4	GetStreamCurrentPosition	. 669
27.251.3.5	GetStreamPtr	. 669
27.251.3.6	Read	. 669
27.251.3.7	ReadDataSet	. 670
27.251.3.8	ReadMetaInformation	. 670
27.251.3.9	ReadPreamble	. 670
27.251.3.10	ReadSelectedPrivateTags	. 670
27.251.3.11	ReadSelectedTags	. 670
27.251.3.12	ReadUpToTag	. 670
27.251.3.13	SetFile	. 670
27.251.3.14	SetFileName	. 670
27.251.3.15	SetStream	. 671
27.251.4	Friends And Related Function Documentation	. 671
27.251.4.1	StreamImageReader	. 671
27.251.5	Member Data Documentation	. 671
27.251.5.1	F	. 671
27.250	gdcm::RealWorldValueMappingContent Struct Reference	. 671
27.252.1	Member Data Documentation	. 672
27.252.1.1	CodeMeaning	. 672
27.252.1.2	CodeValue	. 672
27.252.1.3	RealWorldValueIntercept	. 672
27.252.1.4	RealWorldValueSlope	. 672
27.250	gdcm::Region Class Reference	. 672

27.253.1Detailed Description	673
27.253.2Constructor & Destructor Documentation	673
27.253.2.1Region	673
27.253.2.2~Region	673
27.253.3Member Function Documentation	673
27.253.3.1Area	673
27.253.3.2Clone	674
27.253.3.3ComputeBoundingBox	674
27.253.3.4Empty	674
27.253.3.5IsValid	674
27.253.3.6Print	674
27.254dcm::Rescaler Class Reference	674
27.254.1Detailed Description	675
27.254.2Constructor & Destructor Documentation	676
27.254.2.1Rescaler	676
27.254.2.2~Rescaler	676
27.254.3Member Function Documentation	676
27.254.3.1ComputeInterceptSlopePixelType	676
27.254.3.2ComputePixelTypeFromMinMax	676
27.254.3.3GetIntercept	676
27.254.3.4GetSlope	676
27.254.3.5InverseRescale	676
27.254.3.6InverseRescaleFunctionIntoBestFit	676
27.254.3.7Rescale	676
27.254.3.8RescaleFunctionIntoBestFit	676
27.254.3.9SetIntercept	676
27.254.3.10SetMinMaxForPixelType	677
27.254.3.11SetPixelFormat	677
27.254.3.12SetSlope	677
27.254.3.13SetTargetPixelType	677
27.254.3.14SetUseTargetPixelType	677
27.255dcm::RLECodec Class Reference	677
27.255.1Detailed Description	679
27.255.2Constructor & Destructor Documentation	679
27.255.2.1RLECodec	679
27.255.2.2~RLECodec	679
27.255.3Member Function Documentation	679

27.255.3.1AppendFrameEncode	679
27.255.3.2AppendRowEncode	680
27.255.3.3CanCode	680
27.255.3.4CanDecode	680
27.255.3.5Clone	680
27.255.3.6Code	680
27.255.3.7Decode	680
27.255.3.8DecodeByStreams	680
27.255.3.9DecodeExtent	680
27.255.3.10GetBufferLength	680
27.255.3.11GetHeaderInfo	680
27.255.3.12FrameEncoder	680
27.255.3.13RowEncoder	681
27.255.3.14SetBufferLength	681
27.255.3.15SetLength	681
27.255.3.16StartEncode	681
27.255.3.17StopEncode	681
27.255.4Friends And Related Function Documentation	681
27.255.4.1ImageRegionReader	681
27.256dcm::network::RoleSelectionSub Class Reference	681
27.256.1Detailed Description	681
27.256.2Constructor & Destructor Documentation	682
27.256.2.1RoleSelectionSub	682
27.256.3Member Function Documentation	682
27.256.3.1Print	682
27.256.3.2Read	682
27.256.3.3SetTuple	682
27.256.3.4Size	682
27.256.3.5Write	682
27.257dcm::SerieHelper::Rule Struct Reference	682
27.257.1Member Data Documentation	683
27.257.1.1elem	683
27.257.1.2group	683
27.257.1.3op	683
27.257.1.4value	683
27.258dcm::Scanner Class Reference	683
27.258.1Detailed Description	685

27.258.2	Member Typedef Documentation	686
27.258.2.1	ConstIterator	686
27.258.2.2	MappingType	686
27.258.2.3	TagToValue	686
27.258.2.4	TagToValueValueType	686
27.258.2.5	ValuesType	686
27.258.3	Constructor & Destructor Documentation	686
27.258.3.1	Scanner	686
27.258.3.2	~Scanner	686
27.258.4	Member Function Documentation	686
27.258.4.1	AddPrivateTag	686
27.258.4.2	AddSkipTag	686
27.258.4.3	AddTag	686
27.258.4.4	Begin	687
27.258.4.5	ClearSkipTags	687
27.258.4.6	ClearTags	687
27.258.4.7	End	687
27.258.4.8	GetAllFileNamesFromTagToValue	687
27.258.4.9	GetFilenameFromTagToValue	687
27.258.4.10	GetFileNames	687
27.258.4.11	GetKeys	687
27.258.4.12	GetMapping	687
27.258.4.13	GetMappingFromTagToValue	687
27.258.4.14	GetMappings	687
27.258.4.15	GetOrderedValues	687
27.258.4.16	GetValue	688
27.258.4.17	GetValues	688
27.258.4.18	GetValues	688
27.258.4.19	Key	688
27.258.4.20	New	688
27.258.4.21	Print	688
27.258.4.22	ProcessPublicTag	688
27.258.4.23	Scan	688
27.258.5	Friends And Related Function Documentation	689
27.258.5.1	operator<<	689
27.258.6	dcm::Segment Class Reference	689
27.259.1	Detailed Description	691

27.259.2	Member Typedef Documentation	. 691
27.259.2.1	SurfaceVector	. 691
27.259.3	Member Enumeration Documentation	. 691
27.259.3.1	ALGOType	. 691
27.259.4	Constructor & Destructor Documentation	. 691
27.259.4.1	Segment	. 691
27.259.4.2	~Segment	. 691
27.259.5	Member Function Documentation	. 691
27.259.5.1	AddSurface	. 691
27.259.5.2	GetALGOType	. 691
27.259.5.3	GetALGOTypeString	. 691
27.259.5.4	GetAnatomicRegion	. 691
27.259.5.5	GetAnatomicRegion	. 691
27.259.5.6	GetPropertyCategory	. 691
27.259.5.7	GetPropertyCategory	. 691
27.259.5.8	GetPropertyType	. 691
27.259.5.9	GetPropertyType	. 691
27.259.5.10	GetSegmentAlgorithmName	. 692
27.259.5.11	GetSegmentAlgorithmType	. 692
27.259.5.12	GetSegmentDescription	. 692
27.259.5.13	GetSegmentLabel	. 692
27.259.5.14	GetSegmentNumber	. 692
27.259.5.15	GetSurface	. 692
27.259.5.16	GetSurfaceCount	. 692
27.259.5.17	GetSurfaces	. 692
27.259.5.18	GetSurfaces	. 692
27.259.5.19	SetAnatomicRegion	. 692
27.259.5.20	SetPropertyCategory	. 692
27.259.5.21	SetPropertyType	. 692
27.259.5.22	SetSegmentAlgorithmName	. 692
27.259.5.23	SetSegmentAlgorithmType	. 692
27.259.5.24	SetSegmentAlgorithmType	. 692
27.259.5.25	SetSegmentDescription	. 692
27.259.5.26	SetSegmentLabel	. 692
27.259.5.27	SetSegmentNumber	. 692
27.259.5.28	SetSurfaceCount	. 692
27.259.6	Member Data Documentation	. 692

27.259.6.1AnatomicRegion	692
27.259.6.2PropertyCategory	692
27.259.6.3PropertyType	692
27.259.6.4SegmentAlgorithmName	692
27.259.6.5SegmentAlgorithmType	692
27.259.6.6SegmentDescription	692
27.259.6.7SegmentLabel	693
27.259.6.8SegmentNumber	693
27.259.6.9SurfaceCount	693
27.259.6.10Surfaces	693
27.260dcm::SegmentedPaletteColorLookupTable Class Reference	693
27.260.1Detailed Description	694
27.260.2Constructor & Destructor Documentation	694
27.260.2.1SegmentedPaletteColorLookupTable	694
27.260.2.2~SegmentedPaletteColorLookupTable	694
27.260.3Member Function Documentation	694
27.260.3.1Print	694
27.260.3.2SetLUT	695
27.261dcm::SegmentReader Class Reference	695
27.261.1Detailed Description	697
27.261.2Member Typedef Documentation	697
27.261.2.1SegmentMap	697
27.261.2.2SegmentVector	697
27.261.3Constructor & Destructor Documentation	697
27.261.3.1SegmentReader	697
27.261.3.2~SegmentReader	697
27.261.4Member Function Documentation	697
27.261.4.1GetSegments	697
27.261.4.2GetSegments	697
27.261.4.3Read	697
27.261.4.4ReadSegment	697
27.261.4.5ReadSegments	697
27.261.5Member Data Documentation	697
27.261.5.1Segments	697
27.262dcm::SegmentWriter Class Reference	698
27.262.1Detailed Description	699
27.262.2Member Typedef Documentation	699

27.262.2.1SegmentVector	699
27.262.3Constructor & Destructor Documentation	699
27.262.3.1SegmentWriter	699
27.262.3.2~SegmentWriter	699
27.262.4Member Function Documentation	699
27.262.4.1AddSegment	699
27.262.4.2GetNumberOfSegments	699
27.262.4.3GetSegment	699
27.262.4.4GetSegments	699
27.262.4.5GetSegments	699
27.262.4.6PrepareWrite	699
27.262.4.7SetNumberOfSegments	699
27.262.4.8SetSegments	699
27.262.4.9Write	699
27.262.5Member Data Documentation	700
27.262.5.1Segments	700
27.263dcm::SequenceOfFragments Class Reference	700
27.263.1Detailed Description	702
27.263.2Member Typedef Documentation	702
27.263.2.1ConstIterator	702
27.263.2.2FragmentVector	702
27.263.2.3Iterator	702
27.263.2.4SizeType	702
27.263.3Constructor & Destructor Documentation	702
27.263.3.1SequenceOfFragments	702
27.263.4Member Function Documentation	703
27.263.4.1AddFragment	703
27.263.4.2Begin	703
27.263.4.3Begin	703
27.263.4.4Clear	703
27.263.4.5ComputeByteLength	703
27.263.4.6ComputeLength	703
27.263.4.7End	703
27.263.4.8End	703
27.263.4.9GetBuffer	703
27.263.4.10GetFragBuffer	703
27.263.4.11GetFragment	703

27.263.4.10	GetLength	. 703
27.263.4.10	GetNumberOfFragments	. 703
27.263.4.10	GetTable	. 704
27.263.4.10	GetTable	. 704
27.263.4.10	New	. 704
27.263.4.10	operator==	. 704
27.263.4.10	Print	. 704
27.263.4.10	Read	. 704
27.263.4.20	ReadPreValue	. 704
27.263.4.20	ReadValue	. 704
27.263.4.20	SetLength	. 704
27.263.4.20	Write	. 704
27.263.4.20	WriteBuffer	. 704
27.264	gdcm::SequenceOfItems Class Reference	. 705
27.264.1	Detailed Description	. 707
27.264.2	Member Typedef Documentation	. 708
27.264.2.1	ConstIterator	. 708
27.264.2.2	ItemVector	. 708
27.264.2.3	Iterator	. 708
27.264.2.4	SizeType	. 708
27.264.3	Constructor & Destructor Documentation	. 708
27.264.3.1	SequenceOfItems	. 708
27.264.4	Member Function Documentation	. 708
27.264.4.1	AddItem	. 708
27.264.4.2	AddNewUndefinedLengthItem	. 708
27.264.4.3	Begin	. 708
27.264.4.4	Begin	. 708
27.264.4.5	Clear	. 708
27.264.4.6	ComputeLength	. 709
27.264.4.7	End	. 709
27.264.4.8	End	. 709
27.264.4.9	FindDataElement	. 709
27.264.4.10	GetItem	. 709
27.264.4.10	GetItem	. 709
27.264.4.10	GetLength	. 709
27.264.4.10	GetNumberOfItems	. 709
27.264.4.10	UndefinedLength	. 709

27.264.4.1New	709
27.264.4.1Operator=	709
27.264.4.1Operator==	709
27.264.4.1Print	710
27.264.4.1Read	710
27.264.4.2RemoveItemByIndex	710
27.264.4.2SetLength	710
27.264.4.2SetLengthToUndefined	710
27.264.4.2SetNumberOfItems	710
27.264.4.2Write	710
27.264.5Member Data Documentation	710
27.264.5.1Items	710
27.264.5.2SequenceLengthField	711
27.265dcm::SerieHelper Class Reference	711
27.265.1Detailed Description	712
27.265.2Member Typedef Documentation	712
27.265.2.1SerieRestrictions	712
27.265.2.2SingleSerieUIDFileSetmap	712
27.265.3Constructor & Destructor Documentation	712
27.265.3.1SerieHelper	712
27.265.3.2~SerieHelper	713
27.265.4Member Function Documentation	713
27.265.4.1AddFile	713
27.265.4.2AddFileName	713
27.265.4.3AddRestriction	713
27.265.4.4AddRestriction	713
27.265.4.5AddRestriction	713
27.265.4.6Clear	713
27.265.4.7CreateDefaultUniqueSeriesIdentifier	713
27.265.4.8CreateUniqueSeriesIdentifier	713
27.265.4.9FileNameOrdering	713
27.265.4.10GetFirstSingleSerieUIDFileSet	713
27.265.4.10GetNextSingleSerieUIDFileSet	713
27.265.4.11ImagePositionPatientOrdering	713
27.265.4.10OrderFileList	713
27.265.4.13SetDirectory	713
27.265.4.15SetLoadMode	713

27.265.4.1	Set UseSeriesDetails	713
27.265.4.1	User Ordering	713
27.265.5	Member Data Documentation	713
27.265.5.1	ItFileSetHt	713
27.265.5.2	SingleSerieUIDFileSetHT	713
27.266	dcm::Series Class Reference	713
27.266.1	Detailed Description	714
27.266.2	Constructor & Destructor Documentation	714
27.266.2.1	Series	714
27.267	dcm::network::ServiceClassApplicationInformation Class Reference	714
27.267.1	Detailed Description	714
27.267.2	Constructor & Destructor Documentation	714
27.267.2.1	ServiceClassApplicationInformation	714
27.267.3	Member Function Documentation	714
27.267.3.1	Print	714
27.267.3.2	Read	714
27.267.3.3	SetTuple	715
27.267.3.4	Size	715
27.267.3.5	Write	715
27.268	dcm::ServiceClassUser Class Reference	715
27.268.1	Detailed Description	717
27.268.2	Constructor & Destructor Documentation	717
27.268.2.1	ServiceClassUser	717
27.268.2.2	~ServiceClassUser	717
27.268.3	Member Function Documentation	717
27.268.3.1	GetAETitle	717
27.268.3.2	GetCalledAETitle	718
27.268.3.3	GetTimeout	718
27.268.3.4	InitializeConnection	718
27.268.3.5	IsPresentationContextAccepted	718
27.268.3.6	New	718
27.268.3.7	SendEcho	718
27.268.3.8	SendFind	718
27.268.3.9	SendMove	718
27.268.3.10	SendMove	718
27.268.3.11	SendMove	718
27.268.3.12	SendStore	718

27.268.3.1SendStore	719
27.268.3.1SendStore	719
27.268.3.1SetAETitle	719
27.268.3.1SetCalledAETitle	719
27.268.3.1SetHostname	719
27.268.3.1SetPort	719
27.268.3.1SetPortSCP	719
27.268.3.2SetPresentationContexts	719
27.268.3.2SetTimeout	720
27.268.3.2StartAssociation	720
27.268.3.2StopAssociation	720
27.269dcm::SHA1 Class Reference	720
27.269.1Detailed Description	721
27.269.2Constructor & Destructor Documentation	721
27.269.2.1SHA1	721
27.269.2.2~SHA1	721
27.269.3Member Function Documentation	721
27.269.3.1Compute	721
27.269.3.2ComputeFile	721
27.270dcm::SimpleMemberCommand< T > Class Template Reference	721
27.270.1Detailed Description	723
27.270.2Member Typedef Documentation	723
27.270.2.1Self	723
27.270.2.2TMemberFunctionPointer	723
27.270.3Constructor & Destructor Documentation	724
27.270.3.1SimpleMemberCommand	724
27.270.3.2~SimpleMemberCommand	724
27.270.4Member Function Documentation	724
27.270.4.1Execute	724
27.270.4.2Execute	724
27.270.4.3New	724
27.270.4.4SetCallbackFunction	724
27.270.5Member Data Documentation	724
27.270.5.1m_MemberFunction	724
27.270.5.2m_This	725
27.271dcm::SimpleSubjectWatcher Class Reference	725
27.271.1Detailed Description	725

27.271.2	Constructor & Destructor Documentation	725
27.271.2.1	SimpleSubjectWatcher	725
27.271.2.2	~SimpleSubjectWatcher	725
27.271.3	Member Function Documentation	725
27.271.3.1	EndFilter	726
27.271.3.2	ShowAbort	726
27.271.3.3	ShowAnonymization	726
27.271.3.4	ShowData	726
27.271.3.5	ShowDataSet	726
27.271.3.6	ShowFileName	726
27.271.3.7	ShowIteration	726
27.271.3.8	ShowProgress	726
27.271.3.9	StartFilter	726
27.271.3.10	TestAbortOff	726
27.271.3.11	TestAbortOn	726
27.272	gdcm::SmartPointer< ObjectType > Class Template Reference	726
27.272.1	Detailed Description	728
27.272.2	Constructor & Destructor Documentation	728
27.272.2.1	SmartPointer	728
27.272.2.2	SmartPointer	728
27.272.2.3	SmartPointer	728
27.272.2.4	SmartPointer	728
27.272.2.5	~SmartPointer	728
27.272.3	Member Function Documentation	728
27.272.3.1	GetPointer	729
27.272.3.2	operator ObjectType *	729
27.272.3.3	operator*	729
27.272.3.4	operator->	729
27.272.3.5	operator=	729
27.272.3.6	operator=	729
27.272.3.7	operator=	729
27.273	gdcm::network::SOPClassExtendedNegociationSub Class Reference	729
27.273.1	Detailed Description	730
27.273.2	Constructor & Destructor Documentation	730
27.273.2.1	ISOPClassExtendedNegociationSub	730
27.273.3	Member Function Documentation	730
27.273.3.1	Print	730

27.273.3.2	Read	730
27.273.3.3	SetTuple	730
27.273.3.4	Size	730
27.273.3.5	Write	730
27.274	dcm::SOPClassUIDToIOD Class Reference	730
27.274.1	Detailed Description	731
27.274.2	Member Typedef Documentation	731
27.274.2.1	const	731
27.274.3	Member Function Documentation	731
27.274.3.1	GetIOD	731
27.274.3.2	GetIODFromSOPClassUID	731
27.274.3.3	GetNumberOfSOPClassToIOD	731
27.274.3.4	GetSOPClassUIDFromIOD	731
27.274.3.5	GetSOPClassUIDToIOD	731
27.274.3.6	GetSOPClassUIDToIODs	731
27.275	dcm::Sorter Class Reference	731
27.275.1	Detailed Description	733
27.275.2	Member Typedef Documentation	733
27.275.2.1	SelectionMap	733
27.275.2.2	SortFunction	733
27.275.3	Constructor & Destructor Documentation	734
27.275.3.1	Sorter	734
27.275.3.2	~Sorter	734
27.275.4	Member Function Documentation	734
27.275.4.1	AddSelect	734
27.275.4.2	GetFileNames	734
27.275.4.3	Print	734
27.275.4.4	SetSortFunction	734
27.275.4.5	Sort	734
27.275.4.6	StableSort	734
27.275.5	Friends And Related Function Documentation	735
27.275.5.1	operator<<	735
27.275.6	Member Data Documentation	735
27.275.6.1	FileNames	735
27.275.6.2	Selection	735
27.275.6.3	SortFunc	735
27.276	dcm::Spacing Class Reference	735

27.276.1Detailed Description	735
27.276.2Member Enumeration Documentation	736
27.276.2.1SpacingType	736
27.276.3Constructor & Destructor Documentation	736
27.276.3.1Spacing	736
27.276.3.2~Spacing	736
27.276.4Member Function Documentation	736
27.276.4.1ComputePixelAspectRatioFromPixelSpacing	736
27.277gdcmm::Spectroscopy Class Reference	737
27.277.1Detailed Description	737
27.277.2Constructor & Destructor Documentation	737
27.277.2.1Spectroscopy	737
27.278gdcmm::SplitMosaicFilter Class Reference	737
27.278.1Detailed Description	738
27.278.2Constructor & Destructor Documentation	738
27.278.2.1SplitMosaicFilter	738
27.278.2.2~SplitMosaicFilter	738
27.278.3Member Function Documentation	738
27.278.3.1ComputeMOSAICDimensions	738
27.278.3.2GetFile	738
27.278.3.3GetFile	738
27.278.3.4GetImage	738
27.278.3.5GetImage	738
27.278.3.6SetFile	738
27.278.3.7SetImage	738
27.278.3.8Split	738
27.279gdcmm::StartEvent Class Reference	738
27.280gdcmm::static_assert_test< x > Struct Template Reference	740
27.281gdcmm::STATIC_ASSERTION_FAILURE< x > Struct Template Reference	740
27.282gdcmm::STATIC_ASSERTION_FAILURE< true > Struct Template Reference	740
27.282.1Member Enumeration Documentation	740
27.282.1.1anonymous enum	740
27.283gdcmm::StreamImageReader Class Reference	740
27.283.1Detailed Description	741
27.283.2Constructor & Destructor Documentation	741
27.283.2.1StreamImageReader	741
27.283.2.2~StreamImageReader	741

27.283.3	Member Function Documentation	741
27.283.3.1	CanReadImage	741
27.283.3.2	DefinePixelExtent	741
27.283.3.3	DefineProperBufferLength	742
27.283.3.4	GetDimensionsValueForResolution	742
27.283.3.5	GetFile	742
27.283.3.6	Read	742
27.283.3.7	ReadImageInformation	742
27.283.3.8	SetFileName	743
27.283.3.9	SetStream	743
27.284	dcm::StreamImageWriter Class Reference	743
27.284.1	Detailed Description	745
27.284.2	Constructor & Destructor Documentation	745
27.284.2.1	StreamImageWriter	745
27.284.2.2	~StreamImageWriter	745
27.284.3	Member Function Documentation	745
27.284.3.1	CanWriteFile	746
27.284.3.2	DefinePixelExtent	746
27.284.3.3	DefineProperBufferLength	746
27.284.3.4	SetFile	746
27.284.3.5	SetFileName	746
27.284.3.6	SetStream	746
27.284.3.7	Write	747
27.284.3.8	WriteImageInformation	747
27.284.3.9	WriteImageSubregionRAW	747
27.284.3.10	WriteRawHeader	747
27.284.4	Member Data Documentation	747
27.284.4.1	mElementOffsets	747
27.284.4.2	mElementOffsets1	748
27.284.4.3	mospFile	748
27.284.4.4	mWriter	748
27.284.4.5	mXMax	748
27.284.4.6	mXMin	748
27.284.4.7	mYMax	748
27.284.4.8	mYMin	748
27.284.4.9	mZMax	748
27.284.4.10	mZMin	748

27.285.5dcm::StrictScanner Class Reference	748
27.285.1Detailed Description	750
27.285.2Member Typedef Documentation	751
27.285.2.1ConstIterator	751
27.285.2.2MappingType	751
27.285.2.3TagToValue	751
27.285.2.4TagToValueValueType	751
27.285.2.5ValuesType	751
27.285.3Constructor & Destructor Documentation	751
27.285.3.1StrictScanner	751
27.285.3.2~StrictScanner	751
27.285.4Member Function Documentation	751
27.285.4.1AddPrivateTag	751
27.285.4.2AddSkipTag	751
27.285.4.3AddTag	751
27.285.4.4Begin	752
27.285.4.5ClearSkipTags	752
27.285.4.6ClearTags	752
27.285.4.7End	752
27.285.4.8GetAllFileNamesFromTagToValue	752
27.285.4.9GetFilenameFromTagToValue	752
27.285.4.10GetFileNames	752
27.285.4.11GetKeys	752
27.285.4.12GetMapping	752
27.285.4.13GetMappingFromTagToValue	752
27.285.4.14GetMappings	752
27.285.4.15GetOrderedValues	752
27.285.4.16GetValue	752
27.285.4.17GetValues	753
27.285.4.18GetValues	753
27.285.4.19Key	753
27.285.4.20New	753
27.285.4.21Print	753
27.285.4.22ProcessPublicTag	753
27.285.4.23Scan	753
27.285.5Friends And Related Function Documentation	753
27.285.5.1operator<<	753

27.286.0	gdcmm::String< TDelimiter, TMaxLength, TPadChar > Class Template Reference	754
27.286.1	Detailed Description	755
27.286.2	Member Typedef Documentation	755
27.286.2.1	const_iterator	755
27.286.2.2	const_reference	756
27.286.2.3	const_reverse_iterator	756
27.286.2.4	difference_type	756
27.286.2.5	iterator	756
27.286.2.6	pointer	756
27.286.2.7	reference	756
27.286.2.8	reverse_iterator	756
27.286.2.9	size_type	756
27.286.2.10	value_type	756
27.286.3	Constructor & Destructor Documentation	756
27.286.3.1	String	756
27.286.3.2	String	756
27.286.3.3	String	756
27.286.3.4	String	756
27.286.4	Member Function Documentation	756
27.286.4.1	IsValid	756
27.286.4.2	operator const char *	757
27.286.4.3	Trim	757
27.286.4.4	Trim	757
27.286.4.5	Truncate	757
27.287.0	gdcmm::StringFilter Class Reference	757
27.287.1	Detailed Description	758
27.287.2	Constructor & Destructor Documentation	758
27.287.2.1	StringFilter	758
27.287.2.2	~StringFilter	758
27.287.3	Member Function Documentation	758
27.287.3.1	ExecuteQuery	758
27.287.3.2	ExecuteQuery	758
27.287.3.3	FromString	758
27.287.3.4	FromString	758
27.287.3.5	GetFile	758
27.287.3.6	GetFile	758
27.287.3.7	SetDicts	758

27.287.3.8SetFile	758
27.287.3.9ToString	759
27.287.3.10ToString	759
27.287.3.11ToStringPair	759
27.287.3.12ToStringPair	759
27.287.3.13ToStringPair	759
27.287.3.14UseDictAlways	759
27.288dcm::Study Class Reference	759
27.288.1Detailed Description	759
27.288.2Constructor & Destructor Documentation	760
27.288.2.1Study	760
27.289dcm::Subject Class Reference	760
27.289.1Detailed Description	761
27.289.2Constructor & Destructor Documentation	761
27.289.2.1Subject	761
27.289.2.2~Subject	761
27.289.3Member Function Documentation	761
27.289.3.1AddObserver	762
27.289.3.2AddObserver	762
27.289.3.3GetCommand	762
27.289.3.4HasObserver	762
27.289.3.5InvokeEvent	762
27.289.3.6InvokeEvent	762
27.289.3.7RemoveAllObservers	762
27.289.3.8RemoveObserver	762
27.290dcm::Surface Class Reference	762
27.290.1Detailed Description	765
27.290.2Member Enumeration Documentation	765
27.290.2.1STATES	765
27.290.2.2VIEWType	765
27.290.3Constructor & Destructor Documentation	766
27.290.3.1Surface	766
27.290.3.2~Surface	766
27.290.4Member Function Documentation	766
27.290.4.1GetAlgorithmFamily	766
27.290.4.2GetAlgorithmFamily	766
27.290.4.3GetAlgorithmName	766

27.290.4.4	GetAlgorithmVersion	. 766
27.290.4.5	GetAxisOfRotation	. 766
27.290.4.6	GetCenterOfRotation	. 766
27.290.4.7	GetFiniteVolume	. 766
27.290.4.8	GetManifold	. 766
27.290.4.9	GetMaximumPointDistance	. 766
27.290.4.10	GetMeanPointDistance	. 766
27.290.4.10	GetMeshPrimitive	. 766
27.290.4.10	GetMeshPrimitive	. 766
27.290.4.10	GetNumberOfSurfacePoints	. 767
27.290.4.10	GetNumberOfVectors	. 767
27.290.4.10	GetPointCoordinatesData	. 767
27.290.4.10	GetPointCoordinatesData	. 767
27.290.4.10	GetPointPositionAccuracy	. 767
27.290.4.10	GetPointsBoundingBoxCoordinates	. 767
27.290.4.10	GetProcessingAlgorithm	. 767
27.290.4.20	GetProcessingAlgorithm	. 767
27.290.4.20	GetRecommendedDisplayCIELabValue	. 767
27.290.4.20	GetRecommendedDisplayCIELabValue	. 767
27.290.4.20	GetRecommendedDisplayGrayscaleValue	. 767
27.290.4.20	GetRecommendedPresentationOpacity	. 767
27.290.4.20	GetRecommendedPresentationType	. 767
27.290.4.20	GetSTATES	. 767
27.290.4.20	GetSTATESString	. 767
27.290.4.20	GetSurfaceComments	. 767
27.290.4.20	GetSurfaceNumber	. 767
27.290.4.30	GetSurfaceProcessing	. 767
27.290.4.30	GetSurfaceProcessingDescription	. 767
27.290.4.30	GetSurfaceProcessingRatio	. 767
27.290.4.30	GetVectorAccuracy	. 767
27.290.4.30	GetVectorCoordinateData	. 767
27.290.4.30	GetVectorCoordinateData	. 768
27.290.4.30	GetVectorDimensionality	. 768
27.290.4.30	GetVIEWType	. 768
27.290.4.30	GetVIEWTypeString	. 768
27.290.4.30	GetAlgorithmFamily	. 768
27.290.4.40	GetAlgorithmName	. 768

27.290.4.49	SetAlgorithmVersion	768
27.290.4.49	SetAxisOfRotation	768
27.290.4.49	SetCenterOfRotation	768
27.290.4.49	SetFiniteVolume	768
27.290.4.49	SetManifold	768
27.290.4.49	SetMaximumPointDistance	768
27.290.4.49	SetMeanPointDistance	768
27.290.4.49	SetMeshPrimitive	768
27.290.4.49	SetNumberOfSurfacePoints	768
27.290.4.50	SetNumberOfVectors	768
27.290.4.53	SetPointCoordinatesData	768
27.290.4.53	SetPointPositionAccuracy	768
27.290.4.53	SetPointsBoundingBoxCoordinates	768
27.290.4.53	SetProcessingAlgorithm	768
27.290.4.55	SetRecommendedDisplayCIELabValue	768
27.290.4.55	SetRecommendedDisplayCIELabValue	768
27.290.4.55	SetRecommendedDisplayCIELabValue	768
27.290.4.58	SetRecommendedDisplayGrayscaleValue	768
27.290.4.59	SetRecommendedPresentationOpacity	768
27.290.4.60	SetRecommendedPresentationType	768
27.290.4.63	SetSurfaceComments	769
27.290.4.63	SetSurfaceNumber	769
27.290.4.63	SetSurfaceProcessing	769
27.290.4.63	SetSurfaceProcessingDescription	769
27.290.4.65	SetSurfaceProcessingRatio	769
27.290.4.66	SetVectorAccuracy	769
27.290.4.66	SetVectorCoordinateData	769
27.290.4.68	SetVectorDimensionality	769
27.290.4	gdcm::SurfaceHelper Class Reference	769
27.291.1	Detailed Description	770
27.291.2	Member Typedef Documentation	770
27.291.2.1	ColorArray	770
27.291.3	Member Function Documentation	770
27.291.3.1	RecommendedDisplayCIELabToRGB	770
27.291.3.2	RecommendedDisplayCIELabToRGB	770
27.291.3.3	RGBToRecommendedDisplayCIELab	770
27.291.3.4	RGBToRecommendedDisplayGrayscale	771

27.292	gdcmm::SurfaceReader Class Reference	. 771
27.292.1	Detailed Description	. 773
27.292.2	Constructor & Destructor Documentation	. 773
27.292.2.1	SurfaceReader	. 773
27.292.2.2	~SurfaceReader	. 773
27.292.3	Member Function Documentation	. 773
27.292.3.1	GetNumberOfSurfaces	. 773
27.292.3.2	Read	. 773
27.292.3.3	ReadPointMacro	. 773
27.292.3.4	ReadSurface	. 773
27.292.3.5	ReadSurfaces	. 773
27.293	gdcmm::SurfaceWriter Class Reference	. 774
27.293.1	Detailed Description	. 775
27.293.2	Constructor & Destructor Documentation	. 775
27.293.2.1	SurfaceWriter	. 775
27.293.2.2	~SurfaceWriter	. 775
27.293.3	Member Function Documentation	. 775
27.293.3.1	ComputeNumberOfSurfaces	. 775
27.293.3.2	GetNumberOfSurfaces	. 775
27.293.3.3	PrepareWrite	. 775
27.293.3.4	PrepareWritePointMacro	. 775
27.293.3.5	SetNumberOfSurfaces	. 775
27.293.3.6	Write	. 775
27.293.4	Member Data Documentation	. 775
27.293.4.1	NumberOfSurfaces	. 775
27.294	gdcmm::SwapCode Class Reference	. 775
27.294.1	Detailed Description	. 776
27.294.2	Member Enumeration Documentation	. 776
27.294.2.1	SwapCodeType	. 776
27.294.3	Constructor & Destructor Documentation	. 777
27.294.3.1	SwapCode	. 777
27.294.4	Member Function Documentation	. 777
27.294.4.1	GetIndex	. 777
27.294.4.2	GetSwapCodeString	. 777
27.294.4.3	operator SwapCode::SwapCodeType	. 777
27.294.5	Friends And Related Function Documentation	. 777
27.294.5.1	operator<<	. 777

27.295	dcm::SwapperDoOp Class Reference	777
27.295.1	Member Function Documentation	777
27.295.1.1	Swap	777
27.295.1.2	SwapArray	777
27.296	dcm::SwapperNoOp Class Reference	777
27.296.1	Detailed Description	778
27.296.2	Member Function Documentation	778
27.296.2.1	Swap	778
27.296.2.2	SwapArray	778
27.297	dcm::System Class Reference	778
27.297.1	Detailed Description	779
27.297.2	Member Function Documentation	779
27.297.2.1	DeleteDirectory	779
27.297.2.2	EncodeBytes	779
27.297.2.3	FileExists	779
27.297.2.4	FilesDirectory	780
27.297.2.5	FilesSymlink	780
27.297.2.6	FileSize	780
27.297.2.7	FileTime	780
27.297.2.8	FormatDateTime	780
27.297.2.9	GetCurrentDateTime	780
27.297.2.10	GetCurrentModuleFileName	780
27.297.2.11	GetCurrentProcessFileName	780
27.297.2.12	GetCurrentResourcesDirectory	781
27.297.2.13	GetCWD	781
27.297.2.14	GetHostName	781
27.297.2.15	GetLastSystemError	781
27.297.2.16	GetLocaleCharset	781
27.297.2.17	GetPermissions	781
27.297.2.18	GetTimezoneOffsetFromUTC	781
27.297.2.19	MakeDirectory	781
27.297.2.20	ParseDateTime	781
27.297.2.21	ParseDateTime	781
27.297.2.22	RemoveFile	782
27.297.2.23	SetPermissions	782
27.297.2.24	StrCaseCmp	782
27.297.2.25	StrNCaseCmp	782

27.297.2.28	StrSep	782
27.297.2.29	StrTokR	782
27.298	gdcmm::Table Class Reference	782
27.298.1	Detailed Description	783
27.298.2	Member Typedef Documentation	783
27.298.2.1	MapTableEntry	783
27.298.3	Constructor & Destructor Documentation	783
27.298.3.1	Table	783
27.298.3.2	~Table	783
27.298.4	Member Function Documentation	783
27.298.4.1	GetTableEntry	783
27.298.4.2	InsertEntry	783
27.298.5	Friends And Related Function Documentation	783
27.298.5.1	operator<<	783
27.299	gdcmm::TableEntry Class Reference	783
27.299.1	Detailed Description	784
27.299.2	Constructor & Destructor Documentation	784
27.299.2.1	TableEntry	784
27.299.2.2	~TableEntry	784
27.300	gdcmm::TableReader Class Reference	784
27.300.1	Detailed Description	785
27.300.2	Constructor & Destructor Documentation	785
27.300.2.1	TableReader	785
27.300.2.2	~TableReader	785
27.300.3	Member Function Documentation	785
27.300.3.1	CharacterDataHandler	785
27.300.3.2	EndElement	785
27.300.3.3	GetDefs	785
27.300.3.4	GetFilename	785
27.300.3.5	HandleIOD	785
27.300.3.6	HandleIODEntry	785
27.300.3.7	HandleMacro	785
27.300.3.8	HandleMacroEntry	785
27.300.3.9	HandleMacroEntryDescription	785
27.300.3.10	HandleModule	785
27.300.3.11	HandleModuleEntry	785
27.300.3.12	HandleModuleEntryDescription	786

27.300.3.1	HandleModuleInclude	786
27.300.3.1	Read	786
27.300.3.1	SetFilename	786
27.300.3.1	StartElement	786
27.300	gdcm::network::TableRow Class Reference	786
27.301	Constructor & Destructor Documentation	787
27.301.1.1	TableRow	787
27.301.1.2	~TableRow	787
27.301.2	Member Data Documentation	787
27.301.2.1	transitions	787
27.300	gdcm::Tag Class Reference	787
27.302	Detailed Description	789
27.302.2	Constructor & Destructor Documentation	789
27.302.2.1	Tag	789
27.302.2.2	Tag	789
27.302.2.3	Tag	789
27.302.3	Member Function Documentation	789
27.302.3.1	GetElement	789
27.302.3.2	GetElementTag	790
27.302.3.3	GetGroup	790
27.302.3.4	GetLength	790
27.302.3.5	GetPrivateCreator	790
27.302.3.6	IsGroupLength	790
27.302.3.7	IsGroupXX	790
27.302.3.8	IsIllegal	790
27.302.3.9	IsPrivate	790
27.302.3.10	IsPrivateCreator	791
27.302.3.11	IsPublic	791
27.302.3.12	operator!="	791
27.302.3.13	operator<	791
27.302.3.14	operator<=	791
27.302.3.15	operator=	791
27.302.3.16	operator==	791
27.302.3.17	operator[]	791
27.302.3.18	operator[]	791
27.302.3.19	PrintAsContinuousString	792
27.302.3.20	PrintAsContinuousUpperCaseString	792

27.302.3.2PrintAsPipeSeparatedString	792
27.302.3.2Read	792
27.302.3.2ReadFromCommaSeparatedString	792
27.302.3.2ReadFromContinuousString	792
27.302.3.2ReadFromPipeSeparatedString	792
27.302.3.2SetElement	792
27.302.3.2SetElementTag	793
27.302.3.2SetElementTag	793
27.302.3.2SetGroup	793
27.302.3.2SetPrivateCreator	793
27.302.3.3Write	793
27.302.4Friends And Related Function Documentation	793
27.302.4.1operator<<	793
27.302.4.2operator>>	793
27.302.5Member Data Documentation	793
27.302.5.1bytes	793
27.302.5.2tag	793
27.302.5.3tags	793
27.303dcm::TagPath Class Reference	794
27.303.1Detailed Description	794
27.303.2Constructor & Destructor Documentation	794
27.303.2.1TagPath	794
27.303.2.2~TagPath	794
27.303.3Member Function Documentation	794
27.303.3.1ConstructFromString	794
27.303.3.2ConstructFromTagList	794
27.303.3.3IsValid	794
27.303.3.4Print	795
27.303.3.5Push	795
27.303.3.6Push	795
27.304dcm::Testing Class Reference	795
27.304.1Detailed Description	796
27.304.2Member Typedef Documentation	796
27.304.2.1MD5DataImagesType	796
27.304.2.2MediaStorageDataFilesType	796
27.304.3Constructor & Destructor Documentation	796
27.304.3.1Testing	796

27.304.3.2~Testing	796
27.304.4Member Function Documentation	796
27.304.4.1ComputeFileMD5	796
27.304.4.2ComputeMD5	797
27.304.4.3GetDataExtraRoot	797
27.304.4.4GetDataRoot	797
27.304.4.5GetFileName	797
27.304.4.6GetFileNames	797
27.304.4.7GetLossyFlagFromFile	797
27.304.4.8GetMD5DataImage	797
27.304.4.9GetMD5DataImages	797
27.304.4.10GetMD5FromBrokenFile	797
27.304.4.11GetMD5FromFile	798
27.304.4.12GetMediaStorageDataFile	798
27.304.4.13GetMediaStorageDataFiles	798
27.304.4.14GetMediaStorageFromFile	798
27.304.4.15GetNumberOfFileNames	798
27.304.4.16GetNumberOfMD5DataImages	798
27.304.4.17GetNumberOfMediaStorageDataFiles	798
27.304.4.18GetPixelSpacingDataRoot	798
27.304.4.19GetSelectedPrivateGroupOffsetFromFile	798
27.304.4.20GetSelectedTagsOffsetFromFile	798
27.304.4.21GetSourceDirectory	798
27.304.4.22GetStreamOffsetFromFile	798
27.304.4.23GetTempDirectory	799
27.304.4.24GetTempDirectoryW	799
27.304.4.25GetTempFilename	799
27.304.4.26GetTempFilenameW	799
27.304.4.27Print	799
27.305dcm::Trace Class Reference	799
27.305.1Detailed Description	800
27.305.2Constructor & Destructor Documentation	800
27.305.2.1Trace	800
27.305.2.2~Trace	800
27.305.3Member Function Documentation	800
27.305.3.1DebugOff	801
27.305.3.2DebugOn	801

27.305.3.3ErrorOff	801
27.305.3.4ErrorOn	801
27.305.3.5GetDebugFlag	801
27.305.3.6GetDebugStream	801
27.305.3.7GetErrorFlag	801
27.305.3.8GetErrorStream	801
27.305.3.9GetStream	801
27.305.3.10GetWarningFlag	801
27.305.3.10GetWarningStream	801
27.305.3.12SetDebug	801
27.305.3.13SetDebugStream	801
27.305.3.13SetError	801
27.305.3.13SetErrorStream	802
27.305.3.13SetStream	802
27.305.3.13SetStreamToFile	802
27.305.3.13SetWarning	802
27.305.3.13SetWarningStream	802
27.305.3.20WarningOff	802
27.305.3.20WarningOn	802
27.306dcm::TransferSyntax Class Reference	802
27.306.1Detailed Description	804
27.306.2Member Enumeration Documentation	804
27.306.2.1NegociatedType	804
27.306.2.2TSType	804
27.306.3Constructor & Destructor Documentation	805
27.306.3.1TransferSyntax	805
27.306.4Member Function Documentation	805
27.306.4.1CanStoreLossy	805
27.306.4.2GetNegociatedType	805
27.306.4.3GetString	805
27.306.4.4GetSwapCode	805
27.306.4.5GetTSString	806
27.306.4.6GetTSType	806
27.306.4.7IsEncapsulated	806
27.306.4.8IsEncoded	806
27.306.4.9IsExplicit	806
27.306.4.10IsImplicit	806

27.306.4.11\$Lossless	806
27.306.4.12\$Lossy	806
27.306.4.13\$Valid	806
27.306.4.14operator TSType	806
27.306.5Friends And Related Function Documentation	806
27.306.5.1operator<<	806
27.307dcm::network::TransferSyntaxSub Class Reference	806
27.307.1Detailed Description	807
27.307.2Constructor & Destructor Documentation	807
27.307.2.1TransferSyntaxSub	807
27.307.3Member Function Documentation	807
27.307.3.1GetName	807
27.307.3.2operator==	807
27.307.3.3Print	807
27.307.3.4Read	807
27.307.3.5SetName	807
27.307.3.6SetNameFromUID	807
27.307.3.7Size	807
27.307.3.8Write	807
27.308dcm::network::Transition Struct Reference	807
27.308.1Constructor & Destructor Documentation	808
27.308.1.1Transition	808
27.308.1.2~Transition	808
27.308.1.3Transition	808
27.308.2Member Function Documentation	808
27.308.2.1MakeNew	809
27.308.3Member Data Documentation	809
27.308.3.1mAction	809
27.308.3.2mEnd	809
27.309dcm::Type Class Reference	809
27.309.1Detailed Description	810
27.309.2Member Enumeration Documentation	810
27.309.2.1TypeType	810
27.309.3Constructor & Destructor Documentation	810
27.309.3.1Type	810
27.309.4Member Function Documentation	810
27.309.4.1GetTypeString	810

27.309.4.2GetTypeType	810
27.309.4.3operator TypeType	810
27.309.5Friends And Related Function Documentation	810
27.309.5.1operator<<	811
27.310dcm::UI Struct Reference	811
27.310.1Friends And Related Function Documentation	811
27.310.1.1operator<<	811
27.310.2Member Data Documentation	811
27.310.2.1Internal	811
27.311dcm::UIDGenerator Class Reference	811
27.311.1Detailed Description	812
27.311.2Constructor & Destructor Documentation	812
27.311.2.1UIDGenerator	812
27.311.3Member Function Documentation	812
27.311.3.1Generate	812
27.311.3.2GenerateUUID	812
27.311.3.3GetGDCMUID	812
27.311.3.4GetRoot	813
27.311.3.5IsValid	813
27.311.3.6SetRoot	813
27.312dcm::UIDs Class Reference	813
27.312.1Detailed Description	818
27.312.2Member Typedef Documentation	818
27.312.2.1TransferSyntaxStringsType	818
27.312.3Member Enumeration Documentation	818
27.312.3.1TSName	818
27.312.3.2TSType	825
27.312.4Member Function Documentation	831
27.312.4.1GetName	831
27.312.4.2GetNumberOfTransferSyntaxStrings	832
27.312.4.3GetString	832
27.312.4.4GetTransferSyntaxString	832
27.312.4.5GetTransferSyntaxStrings	832
27.312.4.6GetUIDName	832
27.312.4.7GetUIDString	832
27.312.4.8operator TSType	832
27.312.4.9SetFromUID	832

27.313	dcm::network::ULAction Class Reference	832
27.313.1	Detailed Description	834
27.313.2	Constructor & Destructor Documentation	834
27.313.2.1	ULAction	834
27.313.2.2	~ULAction	834
27.313.3	Member Function Documentation	834
27.313.3.1	PerformAction	834
27.314	dcm::network::ULActionAA1 Class Reference	835
27.314.1	Member Function Documentation	835
27.314.1.1	PerformAction	835
27.315	dcm::network::ULActionAA2 Class Reference	836
27.315.1	Member Function Documentation	836
27.315.1.1	PerformAction	836
27.316	dcm::network::ULActionAA3 Class Reference	837
27.316.1	Member Function Documentation	837
27.316.1.1	PerformAction	837
27.317	dcm::network::ULActionAA4 Class Reference	838
27.317.1	Member Function Documentation	838
27.317.1.1	PerformAction	838
27.318	dcm::network::ULActionAA5 Class Reference	839
27.318.1	Member Function Documentation	839
27.318.1.1	PerformAction	839
27.319	dcm::network::ULActionAA6 Class Reference	840
27.319.1	Member Function Documentation	840
27.319.1.1	PerformAction	840
27.320	dcm::network::ULActionAA7 Class Reference	841
27.320.1	Member Function Documentation	841
27.320.1.1	PerformAction	841
27.321	dcm::network::ULActionAA8 Class Reference	842
27.321.1	Member Function Documentation	842
27.321.1.1	PerformAction	842
27.322	dcm::network::ULActionAE1 Class Reference	843
27.322.1	Member Function Documentation	843
27.322.1.1	PerformAction	843
27.323	dcm::network::ULActionAE2 Class Reference	844
27.323.1	Member Function Documentation	844
27.323.1.1	PerformAction	844

27.324	dcm::network::ULActionAE3 Class Reference	845
27.324.1	Member Function Documentation	845
27.324.1.1	PerformAction	845
27.325	dcm::network::ULActionAE4 Class Reference	846
27.325.1	Member Function Documentation	846
27.325.1.1	PerformAction	846
27.326	dcm::network::ULActionAE5 Class Reference	847
27.326.1	Member Function Documentation	847
27.326.1.1	PerformAction	847
27.327	dcm::network::ULActionAE6 Class Reference	848
27.327.1	Member Function Documentation	848
27.327.1.1	PerformAction	848
27.328	dcm::network::ULActionAE7 Class Reference	849
27.328.1	Member Function Documentation	849
27.328.1.1	PerformAction	849
27.329	dcm::network::ULActionAE8 Class Reference	850
27.329.1	Member Function Documentation	850
27.329.1.1	PerformAction	850
27.330	dcm::network::ULActionAR1 Class Reference	851
27.330.1	Member Function Documentation	851
27.330.1.1	PerformAction	851
27.331	dcm::network::ULActionAR10 Class Reference	852
27.331.1	Member Function Documentation	852
27.331.1.1	PerformAction	852
27.332	dcm::network::ULActionAR2 Class Reference	853
27.332.1	Member Function Documentation	853
27.332.1.1	PerformAction	853
27.333	dcm::network::ULActionAR3 Class Reference	854
27.333.1	Member Function Documentation	854
27.333.1.1	PerformAction	854
27.334	dcm::network::ULActionAR4 Class Reference	855
27.334.1	Member Function Documentation	855
27.334.1.1	PerformAction	855
27.335	dcm::network::ULActionAR5 Class Reference	856
27.335.1	Member Function Documentation	856
27.335.1.1	PerformAction	856
27.336	dcm::network::ULActionAR6 Class Reference	857

27.336.1Member Function Documentation	857
27.336.1.1PerformAction	857
27.337gdcmm::network::ULActionAR7 Class Reference	858
27.337.1Member Function Documentation	858
27.337.1.1PerformAction	858
27.338gdcmm::network::ULActionAR8 Class Reference	859
27.338.1Member Function Documentation	859
27.338.1.1PerformAction	859
27.339gdcmm::network::ULActionAR9 Class Reference	860
27.339.1Member Function Documentation	860
27.339.1.1PerformAction	860
27.340gdcmm::network::ULActionDT1 Class Reference	861
27.340.1Member Function Documentation	861
27.340.1.1PerformAction	861
27.341gdcmm::network::ULActionDT2 Class Reference	862
27.341.1Member Function Documentation	862
27.341.1.1PerformAction	862
27.342gdcmm::network::ULBasicCallback Class Reference	863
27.342.1Detailed Description	864
27.342.2Constructor & Destructor Documentation	864
27.342.2.1ULBasicCallback	864
27.342.2.2~ULBasicCallback	864
27.342.3Member Function Documentation	864
27.342.3.1GetDataSets	864
27.342.3.2GetResponses	864
27.342.3.3HandleDataSet	864
27.342.3.4HandleResponse	864
27.343gdcmm::network::ULConnection Class Reference	864
27.343.1Detailed Description	865
27.343.2Constructor & Destructor Documentation	865
27.343.2.1ULConnection	865
27.343.2.2~ULConnection	865
27.343.3Member Function Documentation	865
27.343.3.1AddAcceptedPresentationContext	865
27.343.3.2FindContext	866
27.343.3.3GetAcceptedPresentationContexts	866
27.343.3.4GetAcceptedPresentationContexts	866

27.343.3.5	GetConnectionInfo	866
27.343.3.6	GetMaxPDUSize	866
27.343.3.7	GetPresentationContextACByID	866
27.343.3.8	GetPresentationContextIDFromPresentationContext	866
27.343.3.9	GetPresentationContextRQByID	866
27.343.3.10	GetPresentationContexts	866
27.343.3.11	GetProtocol	866
27.343.3.12	GetState	866
27.343.3.13	GetTimer	866
27.343.3.14	InitializeConnection	866
27.343.3.15	InitializeIncomingConnection	866
27.343.3.16	SetMaxPDUSize	866
27.343.3.17	SetPresentationContexts	866
27.343.3.18	SetPresentationContexts	866
27.343.3.19	SetState	866
27.343.3.20	StopProtocol	866
27.343.4	Friends And Related Function Documentation	866
27.343.4.1	ULActionAE6	867
27.343.4.2	ULConnectionManager	867
27.344	dcm::network::ULConnectionCallback Class Reference	867
27.344.1	Detailed Description	868
27.344.2	Constructor & Destructor Documentation	868
27.344.2.1	ULConnectionCallback	868
27.344.2.2	~ULConnectionCallback	868
27.344.3	Member Function Documentation	868
27.344.3.1	DataSetHandled	868
27.344.3.2	DataSetHandles	868
27.344.3.3	HandleDataSet	868
27.344.3.4	HandleResponse	868
27.344.3.5	ResetHandledDataSet	868
27.344.3.6	SetImplicitFlag	868
27.344.4	Member Data Documentation	868
27.344.4.1	Implicit	868
27.345	dcm::network::ULConnectionInfo Class Reference	868
27.345.1	Detailed Description	869
27.345.2	Constructor & Destructor Documentation	869
27.345.2.1	ULConnectionInfo	869

27.345.3	Member Function Documentation	869
27.345.3.1	GetCalledAETitle	869
27.345.3.2	GetCalledComputerName	869
27.345.3.3	GetCalledIPAddress	869
27.345.3.4	GetCalledIPPort	869
27.345.3.5	GetCallingAETitle	869
27.345.3.6	GetMaxPDULength	869
27.345.3.7	Initialize	869
27.345.3.8	SetMaxPDULength	869
27.346	dcm::network::ULConnectionManager Class Reference	870
27.346.1	Detailed Description	871
27.346.2	Constructor & Destructor Documentation	872
27.346.2.1	ULConnectionManager	872
27.346.2.2	ULConnectionManager	872
27.346.2.3	~ULConnectionManager	872
27.346.3	Member Function Documentation	872
27.346.3.1	BreakConnection	872
27.346.3.2	BreakConnectionNow	872
27.346.3.3	EstablishConnection	872
27.346.3.4	EstablishConnectionMove	872
27.346.3.5	RunEventLoop	872
27.346.3.6	RunMoveEventLoop	872
27.346.3.7	SendEcho	872
27.346.3.8	SendFind	872
27.346.3.9	SendFind	872
27.346.3.10	SendMove	872
27.346.3.11	SendMove	872
27.346.3.12	SendNAction	873
27.346.3.13	SendNAction	873
27.346.3.14	SendNCreate	873
27.346.3.15	SendNCreate	873
27.346.3.16	SendNDelete	873
27.346.3.17	SendNDelete	873
27.346.3.18	SendNEventReport	873
27.346.3.19	SendNEventReport	873
27.346.3.20	SendNGet	873
27.346.3.21	SendNGet	873

27.346.3.2SendNSet	873
27.346.3.2SendNSet	873
27.346.3.2SendStore	873
27.346.3.2SendStore	873
27.346.4Member Data Documentation	873
27.346.4.1mConnection	873
27.346.4.2mSecondaryConnection	873
27.346.4.3mTransitions	873
27.347dcm::network::ULEvent Class Reference	874
27.347.1Detailed Description	874
27.347.2Constructor & Destructor Documentation	874
27.347.2.1ULEvent	874
27.347.2.2ULEvent	874
27.347.2.3~ULEvent	874
27.347.3Member Function Documentation	874
27.347.3.1GetDataSetPos	874
27.347.3.2GetEvent	874
27.347.3.3GetIStream	874
27.347.3.4GetPDUs	874
27.347.3.5SetEvent	874
27.347.3.6SetPDU	875
27.348dcm::network::ULTransitionTable Class Reference	875
27.348.1Detailed Description	875
27.348.2Constructor & Destructor Documentation	875
27.348.2.1ULTransitionTable	875
27.348.3Member Function Documentation	875
27.348.3.1HandleEvent	875
27.348.3.2PrintTable	875
27.349dcm::network::ULWritingCallback Class Reference	875
27.349.1Constructor & Destructor Documentation	876
27.349.1.1ULWritingCallback	877
27.349.1.2~ULWritingCallback	877
27.349.2Member Function Documentation	877
27.349.2.1HandleDataSet	877
27.349.2.2HandleResponse	877
27.349.2.3SetDirectory	877
27.350dcm::UNExplicitDataElement Class Reference	877

27.350.1Detailed Description	878
27.350.2Member Function Documentation	878
27.350.2.1GetLength	878
27.350.2.2Read	879
27.350.2.3ReadPreValue	879
27.350.2.4ReadValue	879
27.350.2.5ReadWithLength	879
27.351dcm::UNExplicitImplicitDataElement Class Reference	879
27.351.1Detailed Description	880
27.351.2Member Function Documentation	880
27.351.2.1GetLength	880
27.351.2.2Read	881
27.351.2.3ReadPreValue	881
27.351.2.4ReadValue	881
27.352dcm::Unpacker12Bits Class Reference	881
27.352.1Detailed Description	881
27.352.2Member Function Documentation	881
27.352.2.1Pack	881
27.352.2.2Unpack	881
27.353dcm::Usage Class Reference	882
27.353.1Detailed Description	882
27.353.2Member Enumeration Documentation	883
27.353.2.1UsageType	883
27.353.3Constructor & Destructor Documentation	883
27.353.3.1Usage	883
27.353.4Member Function Documentation	883
27.353.4.1GetUsageString	883
27.353.4.2GetUsageType	883
27.353.4.3operator UsageType	883
27.353.5Friends And Related Function Documentation	883
27.353.5.1operator<<	883
27.354dcm::UserEvent Class Reference	883
27.355dcm::network::UserInformation Class Reference	885
27.355.1Detailed Description	885
27.355.2Constructor & Destructor Documentation	885
27.355.2.1UserInformation	885
27.355.2.2~UserInformation	885

27.355.3	Member Function Documentation	885
27.355.3.1	AddRoleSelectionSub	885
27.355.3.2	AddSOPClassExtendedNegociationSub	885
27.355.3.3	GetMaximumLengthSub	885
27.355.3.4	GetMaximumLengthSub	885
27.355.3.5	operator=	885
27.355.3.6	Print	885
27.355.3.7	Read	885
27.355.3.8	Size	886
27.355.3.9	Write	886
27.356	dcm::UUIDGenerator Class Reference	886
27.356.1	Detailed Description	886
27.356.2	Member Function Documentation	886
27.356.2.1	Generate	886
27.356.2.2	IsValid	886
27.357	dcm::Validate Class Reference	886
27.357.1	Detailed Description	887
27.357.2	Constructor & Destructor Documentation	887
27.357.2.1	Validate	887
27.357.2.2	~Validate	887
27.357.3	Member Function Documentation	887
27.357.3.1	GetValidatedFile	887
27.357.3.2	SetFile	888
27.357.3.3	Validation	888
27.357.4	Member Data Documentation	888
27.357.4.1	F	888
27.357.4.2	V	888
27.358	dcm::Value Class Reference	888
27.358.1	Detailed Description	889
27.358.2	Constructor & Destructor Documentation	889
27.358.2.1	Value	889
27.358.2.2	~Value	889
27.358.3	Member Function Documentation	889
27.358.3.1	Clear	889
27.358.3.2	GetLength	889
27.358.3.3	operator==	889
27.358.3.4	SetLength	890

27.358.3.5SetLengthOnly	890
27.358.4Friends And Related Function Documentation	890
27.358.4.1DataElement	890
27.359dcm::ValueIO< TDE, TSwap, TType > Class Template Reference	890
27.359.1Detailed Description	890
27.359.2Member Function Documentation	890
27.359.2.1Read	890
27.359.2.2Write	890
27.360dcm::Version Class Reference	891
27.360.1Detailed Description	891
27.360.2Constructor & Destructor Documentation	891
27.360.2.1Version	891
27.360.2.2~Version	891
27.360.3Member Function Documentation	891
27.360.3.1GetBuildVersion	891
27.360.3.2GetMajorVersion	891
27.360.3.3GetMinorVersion	891
27.360.3.4GetVersion	891
27.360.3.5Print	891
27.360.4Friends And Related Function Documentation	892
27.360.4.1operator<<	892
27.361dcm::VL Class Reference	892
27.361.1Detailed Description	893
27.361.2Member Typedef Documentation	893
27.361.2.1Type	893
27.361.3Constructor & Destructor Documentation	893
27.361.3.1VL	893
27.361.4Member Function Documentation	893
27.361.4.1GetLength	893
27.361.4.2GetVL16Max	893
27.361.4.3GetVL32Max	893
27.361.4.4IsOdd	893
27.361.4.5IsUndefined	893
27.361.4.6operator uint32_t	893
27.361.4.7operator++	893
27.361.4.8operator++	893
27.361.4.9operator+=	893

27.361.4.1Read	893
27.361.4.1Read16	893
27.361.4.1SetToUndefined	894
27.361.4.1Write	894
27.361.4.1Write16	894
27.361.5Friends And Related Function Documentation	894
27.361.5.operator<<	894
27.362gdcmm::VM Class Reference	894
27.362.1Detailed Description	896
27.362.2Member Enumeration Documentation	896
27.362.2.1VMType	896
27.362.3Constructor & Destructor Documentation	897
27.362.3.1VM	897
27.362.4Member Function Documentation	897
27.362.4.1Compatible	897
27.362.4.2GetIndex	897
27.362.4.3GetLength	897
27.362.4.4GetNumberOfElementsFromArray	897
27.362.4.5GetVMString	897
27.362.4.6GetVMType	898
27.362.4.7GetVMTypeFromLength	898
27.362.4.8IsValid	898
27.362.4.9operator VMType	898
27.362.5Friends And Related Function Documentation	898
27.362.5.operator<<	898
27.363gdcmm::VMToLength< T > Struct Template Reference	898
27.364gdcmm::VR Class Reference	898
27.364.1Detailed Description	900
27.364.2Member Enumeration Documentation	900
27.364.2.1VRType	900
27.364.3Constructor & Destructor Documentation	902
27.364.3.1VR	902
27.364.4Member Function Documentation	902
27.364.4.1CanDisplay	902
27.364.4.2Compatible	902
27.364.4.3GetLength	902
27.364.4.4GetLength	902

27.364.4.5	GetSize	902
27.364.4.6	GetSizeof	902
27.364.4.7	GetVRString	902
27.364.4.8	GetVRStringFromFile	902
27.364.4.9	GetVRType	902
27.364.4.10	GetVRTypeFromFile	902
27.364.4.11	ASCII	902
27.364.4.12	ASCII2	902
27.364.4.13	Binary	902
27.364.4.14	Binary2	902
27.364.4.15	Dual	902
27.364.4.16	Swap	902
27.364.4.17	Valid	902
27.364.4.18	Valid	902
27.364.4.19	VRFile	902
27.364.4.20	operator VRType	903
27.364.4.21	Read	903
27.364.4.22	Write	903
27.364.5	Friends And Related Function Documentation	903
27.364.5.1	operator<<	903
27.365	dcm::VR16ExplicitDataElement Class Reference	903
27.365.1	Detailed Description	904
27.365.2	Member Function Documentation	904
27.365.2.1	GetLength	904
27.365.2.2	Read	905
27.365.2.3	ReadPreValue	905
27.365.2.4	ReadValue	905
27.365.2.5	ReadWithLength	905
27.366	dcm::VRToEncoding< T > Struct Template Reference	905
27.367	dcm::VRToType< T > Struct Template Reference	905
27.367.1	Detailed Description	905
27.368	dcm::VRVLSIZE< T > Class Template Reference	906
27.369	dcm::VRVLSIZE< 0 > Class Template Reference	906
27.369.1	Member Function Documentation	906
27.369.1.1	Read	906
27.369.1.2	Write	906
27.370	dcm::VRVLSIZE< 1 > Class Template Reference	906

27.370.1 Member Function Documentation	906
27.370.1.1 Read	906
27.370.1.2 Write	906
27.371.1 vtkGDCMImageReader Class Reference	907
27.371.1.1 Detailed Description	909
27.371.2 Constructor & Destructor Documentation	909
27.371.2.1 vtkGDCMImageReader	909
27.371.2.2 ~vtkGDCMImageReader	910
27.371.3 Member Function Documentation	910
27.371.3.1 CanReadFile	910
27.371.3.2 ExecuteData	910
27.371.3.3 ExecuteInformation	910
27.371.3.4 FillMedicalImageInformation	910
27.371.3.5 GetDescriptiveName	910
27.371.3.6 GetFileExtensions	910
27.371.3.7 GetIconImage	910
27.371.3.8 GetOverlay	910
27.371.3.9 LoadSingleFile	910
27.371.3.10 New	910
27.371.3.11 PrintSelf	910
27.371.3.12 RequestDataCompat	910
27.371.3.13 RequestInformationCompat	910
27.371.3.14 SetCurve	910
27.371.3.15 SetFileNames	911
27.371.3.16 SetFilePattern	911
27.371.3.17 SetFilePrefix	911
27.371.3.18 SetMedicalImageProperties	911
27.371.3.19 SetBooleanMacro	911
27.371.3.20 GetBooleanMacro	911
27.371.3.21 SetBooleanMacro	911
27.371.3.22 GetBooleanMacro	911
27.371.3.23 SetBooleanMacro	911
27.371.3.24 GetMacro	911
27.371.3.25 SetMacro	911
27.371.3.26 GetMacro	911
27.371.3.27 SetMacro	911
27.371.3.28 GetMacro	911

27.371.3.29kGetMacro	911
27.371.3.30kGetMacro	911
27.371.3.31kGetMacro	911
27.371.3.32kGetMacro	911
27.371.3.33kGetMacro	911
27.371.3.34kGetMacro	911
27.371.3.35kGetObjectMacro	911
27.371.3.36kGetObjectMacro	911
27.371.3.37kGetObjectMacro	911
27.371.3.38kGetObjectMacro	912
27.371.3.39kGetStringMacro	912
27.371.3.40kGetStringMacro	912
27.371.3.41kGetVector3Macro	912
27.371.3.42kGetVector6Macro	912
27.371.3.43kSetMacro	912
27.371.3.44kSetMacro	912
27.371.3.45kSetMacro	912
27.371.3.46kSetMacro	912
27.371.3.47kSetVector6Macro	912
27.371.3.48kTypeRevisionMacro	912
27.371.4Member Data Documentation	912
27.371.4.1ApplyInverseVideo	912
27.371.4.2ApplyLookupTable	912
27.371.4.3ApplyPlanarConfiguration	912
27.371.4.4ApplyShiftScale	912
27.371.4.5ApplyYBRToRGB	912
27.371.4.6Curve	912
27.371.4.7DirectionCosines	912
27.371.4.8FileNames	912
27.371.4.9ForceRescale	912
27.371.4.10IconDataScalarType	912
27.371.4.11IconImageDataExtent	912
27.371.4.12IconNumberOfScalarComponents	912
27.371.4.13ImageFormat	912
27.371.4.14ImageOrientationPatient	912
27.371.4.15ImagePositionPatient	913
27.371.4.16LoadIconImage	913

27.371.4.1	LoadOverlays	913
27.371.4.1	BossyFlag	913
27.371.4.1	MedicalImageProperties	913
27.371.4.2	NumberOfIconImages	913
27.371.4.2	NumberOfOverlays	913
27.371.4.2	PlanarConfiguration	913
27.371.4.2	Scale	913
27.371.4.2	Shift	913
27.372	vtkGDCMImageReader2 Class Reference	913
27.372.1	Detailed Description	916
27.372.2	Constructor & Destructor Documentation	916
27.372.2.1	vtkGDCMImageReader2	916
27.372.2.2	~vtkGDCMImageReader2	916
27.372.3	Member Function Documentation	916
27.372.3.1	CanReadFile	916
27.372.3.2	FillMedicalImageInformation	916
27.372.3.3	GetDescriptiveName	916
27.372.3.4	GetFileExtensions	916
27.372.3.5	GetIconImage	916
27.372.3.6	GetIconImagePort	916
27.372.3.7	GetOverlay	916
27.372.3.8	GetOverlayPort	916
27.372.3.9	LoadSingleFile	916
27.372.3.10	New	916
27.372.3.11	PrintSelf	916
27.372.3.12	ProcessRequest	916
27.372.3.13	RequestData	916
27.372.3.14	RequestDataCompat	916
27.372.3.15	RequestInformation	916
27.372.3.16	RequestInformationCompat	917
27.372.3.17	SetCurve	917
27.372.3.18	SetFilePattern	917
27.372.3.19	SetFilePrefix	917
27.372.3.20	SetMedicalImageProperties	917
27.372.3.21	tkBooleanMacro	917
27.372.3.22	tkBooleanMacro	917
27.372.3.23	tkBooleanMacro	917

27.372.3.24kBooleanMacro	917
27.372.3.25kBooleanMacro	917
27.372.3.26kGetMacro	917
27.372.3.27kGetMacro	917
27.372.3.28kGetMacro	917
27.372.3.29kGetMacro	917
27.372.3.30kGetMacro	917
27.372.3.31kGetMacro	917
27.372.3.32kGetMacro	917
27.372.3.33kGetMacro	917
27.372.3.34kGetMacro	917
27.372.3.35kGetMacro	917
27.372.3.36kGetMacro	917
27.372.3.37kGetObjectMacro	917
27.372.3.38kGetObjectMacro	917
27.372.3.39kGetStringMacro	917
27.372.3.40kGetStringMacro	917
27.372.3.41kGetVector3Macro	917
27.372.3.42kGetVector6Macro	918
27.372.3.43kSetMacro	918
27.372.3.44kSetMacro	918
27.372.3.45kSetMacro	918
27.372.3.46kSetMacro	918
27.372.3.47kSetVector6Macro	918
27.372.3.48kTypeRevisionMacro	918
27.372.4Member Data Documentation	918
27.372.4.1ApplyInverseVideo	918
27.372.4.2ApplyLookupTable	918
27.372.4.3ApplyPlanarConfiguration	918
27.372.4.4ApplyShiftScale	918
27.372.4.5ApplyYBRToRGB	918
27.372.4.6Curve	918
27.372.4.7DirectionCosines	918
27.372.4.8ForceRescale	918
27.372.4.9IconDataScalarType	918
27.372.4.10IconImageDataExtent	918
27.372.4.11IconNumberOfScalarComponents	918

27.372.4.12	ImageFormat	918
27.372.4.13	ImageOrientationPatient	918
27.372.4.14	ImagePositionPatient	918
27.372.4.15	LoadIconImage	918
27.372.4.16	LoadOverlays	918
27.372.4.17	LossyFlag	918
27.372.4.18	NumberOfIconImages	918
27.372.4.19	NumberOfOverlays	919
27.372.4.20	PlanarConfiguration	919
27.372.4.21	Scale	919
27.372.4.22	Shift	919
27.373.1	vtkGDCMImageWriter Class Reference	919
27.373.1	Detailed Description	921
27.373.2	Member Enumeration Documentation	921
27.373.2.1	CompressionTypes	921
27.373.3	Constructor & Destructor Documentation	921
27.373.3.1	vtkGDCMImageWriter	921
27.373.3.2	~vtkGDCMImageWriter	921
27.373.4	Member Function Documentation	921
27.373.4.1	GetDescriptiveName	921
27.373.4.2	GetFileExtensions	921
27.373.4.3	GetFileName	921
27.373.4.4	New	921
27.373.4.5	PrintSelf	922
27.373.4.6	SetDirectionCosines	922
27.373.4.7	SetDirectionCosinesFromImageOrientationPatient	922
27.373.4.8	SetFileNames	922
27.373.4.9	SetMedicalImageProperties	922
27.373.4.10	BooleanMacro	922
27.373.4.11	BooleanMacro	922
27.373.4.12	GetMacro	922
27.373.4.13	GetMacro	922
27.373.4.14	GetMacro	922
27.373.4.15	GetMacro	922
27.373.4.16	GetMacro	922
27.373.4.17	GetMacro	922
27.373.4.18	GetMacro	922

27.373.4.10	GetObjectMacro	922
27.373.4.20	GetObjectMacro	922
27.373.4.21	GetObjectMacro	922
27.373.4.22	GetStringMacro	922
27.373.4.23	GetStringMacro	923
27.373.4.24	SetMacro	923
27.373.4.25	SetMacro	923
27.373.4.26	SetMacro	923
27.373.4.27	SetMacro	923
27.373.4.28	SetMacro	923
27.373.4.29	SetMacro	923
27.373.4.30	SetMacro	923
27.373.4.31	SetStringMacro	923
27.373.4.32	SetStringMacro	923
27.373.4.33	TypeRevisionMacro	923
27.373.4.34	Write	923
27.373.4.35	WriteGDCMData	923
27.373.4.36	WriteSlice	923
27.374	vtkGDCMMedicalImageProperties Class Reference	923
27.374.1	Constructor & Destructor Documentation	925
27.374.1.1	vtkGDCMMedicalImageProperties	925
27.374.1.2	~vtkGDCMMedicalImageProperties	925
27.374.2	Member Function Documentation	925
27.374.2.1	Clear	925
27.374.2.2	GetFile	925
27.374.2.3	New	925
27.374.2.4	PrintSelf	925
27.374.2.5	PushBackFile	925
27.374.2.6	vtkTypeRevisionMacro	925
27.374.3	Friends And Related Function Documentation	925
27.374.3.1	vtkGDCMImageReader	925
27.374.3.2	vtkGDCMImageReader2	925
27.374.3.3	vtkGDCMImageWriter	925
27.375	vtkGDCMPolyDataReader Class Reference	925
27.375.1	Detailed Description	927
27.375.2	Constructor & Destructor Documentation	927
27.375.2.1	vtkGDCMPolyDataReader	927

27.375.2.2~vtkGDCMPolyDataReader	927
27.375.3Member Function Documentation	927
27.375.3.1FillMedicalImageInformation	927
27.375.3.2New	927
27.375.3.3PrintSelf	927
27.375.3.4RequestData	927
27.375.3.5RequestData_HemodynamicWaveformStorage	927
27.375.3.6RequestData_RTStructureSetStorage	928
27.375.3.7RequestInformation	928
27.375.3.8RequestInformation_HemodynamicWaveformStorage	928
27.375.3.9RequestInformation_RTStructureSetStorage	928
27.375.3.10GetObjectMacro	928
27.375.3.11GetObjectMacro	928
27.375.3.12GetStringMacro	928
27.375.3.13SetStringMacro	928
27.375.3.14TypeRevisionMacro	928
27.375.4Member Data Documentation	928
27.375.4.1FileName	928
27.375.4.2MedicalImageProperties	928
27.375.4.3RTStructSetProperties	928
27.376.1vtkGDCMPolyDataWriter Class Reference	928
27.376.1Detailed Description	930
27.376.2Constructor & Destructor Documentation	930
27.376.2.1vtkGDCMPolyDataWriter	930
27.376.2.2~vtkGDCMPolyDataWriter	930
27.376.3Member Function Documentation	930
27.376.3.1InitializeRTStructSet	930
27.376.3.2New	930
27.376.3.3PrintSelf	930
27.376.3.4SetMedicalImageProperties	930
27.376.3.5SetNumberOfInputPorts	931
27.376.3.6SetRTStructSetProperties	931
27.376.3.7vtkTypeRevisionMacro	931
27.376.3.8WriteData	931
27.376.3.9WriteRTSTRUCTData	931
27.376.3.10WriteRTSTRUCTInfo	931
27.376.4Member Data Documentation	931

27.376.4.1MedicalImageProperties	931
27.376.4.2RTStructSetProperties	931
27.377.vtkGDCMTesting Class Reference	931
27.377.1Detailed Description	932
27.377.2Member Typedef Documentation	933
27.377.2.1MD5MetalImagesType	933
27.377.3Constructor & Destructor Documentation	933
27.377.3.1vtkGDCMTesting	933
27.377.3.2~vtkGDCMTesting	933
27.377.4Member Function Documentation	933
27.377.4.1GetGDCMDataRoot	933
27.377.4.2GetMD5MetalImage	933
27.377.4.3GetMHDMD5FromFile	933
27.377.4.4GetNumberOfMD5MetalImages	933
27.377.4.5GetRAWMD5FromFile	933
27.377.4.6GetVTKDataRoot	933
27.377.4.7New	933
27.377.4.8PrintSelf	934
27.377.4.9vtkTypeRevisionMacro	934
27.378.vtkGDCMThreadedImageReader Class Reference	934
27.378.1Constructor & Destructor Documentation	935
27.378.1.1vtkGDCMThreadedImageReader	935
27.378.1.2~vtkGDCMThreadedImageReader	936
27.378.2Member Function Documentation	936
27.378.2.1ExecuteData	936
27.378.2.2ExecuteInformation	936
27.378.2.3New	936
27.378.2.4PrintSelf	936
27.378.2.5ReadFiles	936
27.378.2.6RequestDataCompat	936
27.378.2.7vtkBooleanMacro	936
27.378.2.8vtkGetMacro	936
27.378.2.9vtkSetMacro	936
27.378.2.10vtkSetMacro	936
27.378.2.11vtkSetMacro	936
27.378.2.12vtkTypeRevisionMacro	936
27.379.vtkGDCMThreadedImageReader2 Class Reference	936

27.379.1	Constructor & Destructor Documentation	938
27.379.1.1	vtkGDCMThreadedImageReader2	938
27.379.1.2	~vtkGDCMThreadedImageReader2	938
27.379.2	Member Function Documentation	938
27.379.2.1	GetFileName	938
27.379.2.2	New	938
27.379.2.3	PrintSelf	938
27.379.2.4	RequestInformation	938
27.379.2.5	SetFileName	938
27.379.2.6	SetFileNames	938
27.379.2.7	SplitExtent	939
27.379.2.8	ThreadedRequestData	939
27.379.2.9	vtkBooleanMacro	939
27.379.2.10	vtkBooleanMacro	939
27.379.2.11	vtkBooleanMacro	939
27.379.2.12	vtkGetMacro	939
27.379.2.13	vtkGetMacro	939
27.379.2.14	vtkGetMacro	939
27.379.2.15	vtkGetMacro	939
27.379.2.16	vtkGetMacro	939
27.379.2.17	vtkGetMacro	939
27.379.2.18	vtkGetMacro	939
27.379.2.19	vtkGetMacro	939
27.379.2.20	vtkGetObjectMacro	939
27.379.2.21	vtkGetVector3Macro	939
27.379.2.22	vtkGetVector3Macro	939
27.379.2.23	vtkGetVector6Macro	939
27.379.2.24	vtkSetMacro	939
27.379.2.25	vtkSetMacro	939
27.379.2.26	vtkSetMacro	939
27.379.2.27	vtkSetMacro	939
27.379.2.28	vtkSetMacro	939
27.379.2.29	vtkSetMacro	939
27.379.2.30	vtkSetMacro	939
27.379.2.31	vtkSetVector3Macro	939
27.379.2.32	vtkSetVector3Macro	940
27.379.2.33	vtkSetVector6Macro	940

27.379.2.34	tkTypeRevisionMacro	940
27.380.0	tkImageColorViewer Class Reference	940
27.380.1	Detailed Description	943
27.380.2	Member Enumeration Documentation	943
27.380.2.1	anonymous enum	943
27.380.3	Constructor & Destructor Documentation	943
27.380.3.1	tkImageColorViewer	943
27.380.3.2	~tkImageColorViewer	943
27.380.4	Member Function Documentation	943
27.380.4.1	AddInput	943
27.380.4.2	AddInputConnection	943
27.380.4.3	GetColorLevel	943
27.380.4.4	GetColorWindow	943
27.380.4.5	GetInput	943
27.380.4.6	GetOffScreenRendering	944
27.380.4.7	GetOverlayVisibility	944
27.380.4.8	GetPosition	944
27.380.4.9	GetSize	944
27.380.4.10	GetSliceMax	944
27.380.4.11	GetSliceMin	944
27.380.4.12	GetSliceRange	944
27.380.4.13	GetSliceRange	944
27.380.4.14	GetSliceRange	944
27.380.4.15	GetWindowName	944
27.380.4.16	InstallPipeline	944
27.380.4.17	New	944
27.380.4.18	PrintSelf	944
27.380.4.19	Render	944
27.380.4.20	SetColorLevel	944
27.380.4.21	SetColorWindow	944
27.380.4.22	SetDisplayId	944
27.380.4.23	SetInput	944
27.380.4.24	SetInputConnection	945
27.380.4.25	SetOffScreenRendering	945
27.380.4.26	SetOverlayVisibility	945
27.380.4.27	SetParentId	945
27.380.4.28	SetPosition	945

27.380.4.29	SetPosition	. 945
27.380.4.30	SetRenderer	. 945
27.380.4.31	SetRenderWindow	. 945
27.380.4.32	SetSize	. 945
27.380.4.33	SetSize	. 945
27.380.4.34	SetSlice	. 945
27.380.4.35	SetSliceOrientation	. 945
27.380.4.36	SetSliceOrientationToXY	. 945
27.380.4.37	SetSliceOrientationToXZ	. 945
27.380.4.38	SetSliceOrientationToYZ	. 945
27.380.4.39	SetupInteractor	. 946
27.380.4.40	SetWindowId	. 946
27.380.4.41	UninstallPipeline	. 946
27.380.4.42	UpdateDisplayExtent	. 946
27.380.4.43	UpdateOrientation	. 946
27.380.4.44	TK_LEGACY	. 946
27.380.4.45	TK_LEGACY	. 946
27.380.4.46	TK_LEGACY	. 946
27.380.4.47	TK_LEGACY	. 946
27.380.4.48	BooleanMacro	. 946
27.380.4.49	GetMacro	. 946
27.380.4.50	GetMacro	. 946
27.380.4.51	GetObjectMacro	. 946
27.380.4.52	GetObjectMacro	. 946
27.380.4.53	GetObjectMacro	. 946
27.380.4.54	GetObjectMacro	. 946
27.380.4.55	GetObjectMacro	. 946
27.380.4.56	TypeRevisionMacro	. 946
27.380.5	Friends And Related Function Documentation	. 946
27.380.5.1	vtkImageColorViewerCallback	. 946
27.380.6	Member Data Documentation	. 946
27.380.6.1	FirstRender	. 946
27.380.6.2	ImageActor	. 946
27.380.6.3	Interactor	. 946
27.380.6.4	InteractorStyle	. 947
27.380.6.5	OverlayImageActor	. 947
27.380.6.6	Renderer	. 947

27.380.6.7RenderWindow	947
27.380.6.8Slice	947
27.380.6.9SliceOrientation	947
27.380.6.10WindowLevel	947
27.381.vtkImageMapToColors16 Class Reference	947
27.381.1.Constructor & Destructor Documentation	949
27.381.1.1vtkImageMapToColors16	949
27.381.1.2~vtkImageMapToColors16	949
27.381.2.Member Function Documentation	949
27.381.2.1GetMTime	949
27.381.2.2New	949
27.381.2.3PrintSelf	949
27.381.2.4RequestData	949
27.381.2.5RequestInformation	949
27.381.2.6SetLookupTable	949
27.381.2.7SetOutputFormatToLuminance	949
27.381.2.8SetOutputFormatToLuminanceAlpha	949
27.381.2.9SetOutputFormatToRGB	949
27.381.2.10SetOutputFormatToRGBA	949
27.381.2.11ThreadedRequestData	949
27.381.2.12VtkBooleanMacro	949
27.381.2.13VtkGetMacro	949
27.381.2.14VtkGetMacro	949
27.381.2.15VtkGetMacro	949
27.381.2.16VtkGetObjectMacro	949
27.381.2.17VtkSetMacro	950
27.381.2.18VtkSetMacro	950
27.381.2.19VtkSetMacro	950
27.381.2.20VtkTypeRevisionMacro	950
27.381.3.Member Data Documentation	950
27.381.3.1ActiveComponent	950
27.381.3.2DataWasPassed	950
27.381.3.3LookupTable	950
27.381.3.4OutputFormat	950
27.381.3.5PassAlphaToOutput	950
27.382.vtkImageMapToWindowLevelColors2 Class Reference	950
27.382.1.Constructor & Destructor Documentation	952

27.382.1.1	vtkImageMapToWindowLevelColors2	952
27.382.1.2	~vtkImageMapToWindowLevelColors2	952
27.382.2	Member Function Documentation	952
27.382.2.1	New	952
27.382.2.2	PrintSelf	952
27.382.2.3	RequestData	952
27.382.2.4	RequestInformation	952
27.382.2.5	ThreadedRequestData	952
27.382.2.6	vtkGetMacro	952
27.382.2.7	vtkGetMacro	952
27.382.2.8	vtkSetMacro	952
27.382.2.9	vtkSetMacro	952
27.382.2.10	vtkTypeRevisionMacro	952
27.382.3	Member Data Documentation	952
27.382.3.1	Level	952
27.382.3.2	Window	952
27.383	vtkImagePlanarComponentsToComponents Class Reference	952
27.383.1	Constructor & Destructor Documentation	954
27.383.1.1	vtkImagePlanarComponentsToComponents	954
27.383.1.2	~vtkImagePlanarComponentsToComponents	954
27.383.2	Member Function Documentation	954
27.383.2.1	New	954
27.383.2.2	PrintSelf	954
27.383.2.3	RequestData	954
27.383.2.4	vtkTypeRevisionMacro	954
27.384	vtkImageRGBToYBR Class Reference	954
27.384.1	Constructor & Destructor Documentation	955
27.384.1.1	vtkImageRGBToYBR	955
27.384.1.2	~vtkImageRGBToYBR	955
27.384.2	Member Function Documentation	955
27.384.2.1	New	955
27.384.2.2	PrintSelf	955
27.384.2.3	ThreadedExecute	955
27.384.2.4	vtkTypeRevisionMacro	955
27.385	vtkImageYBRToRGB Class Reference	956
27.385.1	Constructor & Destructor Documentation	957
27.385.1.1	vtkImageYBRToRGB	957

27.385.1.2~vtkImageYBRToRGB	957
27.385.2Member Function Documentation	957
27.385.2.1New	957
27.385.2.2PrintSelf	957
27.385.2.3ThreadedExecute	957
27.385.2.4vtkTypeRevisionMacro	957
27.386.1vtkLookupTable16 Class Reference	957
27.386.1.1Constructor & Destructor Documentation	958
27.386.1.1.1vtkLookupTable16	958
27.386.1.2~vtkLookupTable16	958
27.386.2Member Function Documentation	958
27.386.2.1Build	959
27.386.2.2GetPointer	959
27.386.2.3MapScalarsThroughTable2	959
27.386.2.4New	959
27.386.2.5PrintSelf	959
27.386.2.6SetNumberOfTableValues	959
27.386.2.7vtkTypeRevisionMacro	959
27.386.2.8WritePointer	959
27.386.3Member Data Documentation	959
27.386.3.1Table16	959
27.387.1vtkRTStructSetProperties Class Reference	959
27.387.1.1Detailed Description	961
27.387.2Constructor & Destructor Documentation	961
27.387.2.1vtkRTStructSetProperties	961
27.387.2.2~vtkRTStructSetProperties	961
27.387.3Member Function Documentation	961
27.387.3.1AddContourReferencedFrameOfReference	962
27.387.3.2AddReferencedFrameOfReference	962
27.387.3.3AddStructureSetROI	962
27.387.3.4AddStructureSetROIObservation	962
27.387.3.5Clear	962
27.387.3.6DeepCopy	962
27.387.3.7GetContourReferencedFrameOfReferenceClassUID	962
27.387.3.8GetContourReferencedFrameOfReferenceInstanceUID	962
27.387.3.9GetNumberOfContourReferencedFrameOfReferences	962
27.387.3.10GetNumberOfContourReferencedFrameOfReferences	962

27.387.3.1	GetNumberOfReferencedFrameOfReferences962
27.387.3.10	GetNumberOfStructureSetROIs962
27.387.3.13	GetReferencedFrameOfReferenceClassUID962
27.387.3.14	GetReferencedFrameOfReferenceInstanceUID962
27.387.3.15	GetStructureSetObservationNumber962
27.387.3.16	GetStructureSetROIDescription962
27.387.3.17	GetStructureSetROIGenerationAlgorithm962
27.387.3.18	GetStructureSetROIName962
27.387.3.19	GetStructureSetROINumber962
27.387.3.20	GetStructureSetROI ObservationLabel962
27.387.3.23	GetStructureSetROIRefFrameRefUID962
27.387.3.24	GetStructureSetRTROIInterpretedType962
27.387.3.25	New963
27.387.3.26	PrintSelf963
27.387.3.27k	GetStringMacro963
27.387.3.28k	GetStringMacro963
27.387.3.29k	GetStringMacro963
27.387.3.30k	GetStringMacro963
27.387.3.31k	GetStringMacro963
27.387.3.32k	GetStringMacro963
27.387.3.33k	GetStringMacro963
27.387.3.34k	SetStringMacro963
27.387.3.35k	SetStringMacro963
27.387.3.36k	SetStringMacro963
27.387.3.37k	SetStringMacro963
27.387.3.38k	SetStringMacro963
27.387.3.39k	SetStringMacro963
27.387.3.40k	SetStringMacro963
27.387.3.41k	SetStringMacro963
27.387.3.42k	SetStringMacro963
27.387.3.43k	TypeRevisionMacro963
27.387.4.	Member Data Documentation963
27.387.4.1	Internals963
27.387.4.2	ReferenceFrameOfReferenceUID963
27.387.4.3	ReferenceSeriesInstanceUID964

27.387.4.4SeriesInstanceUID	964
27.387.4.5SOPInstanceUID	964
27.387.4.6StructureSetDate	964
27.387.4.7StructureSetLabel	964
27.387.4.8StructureSetName	964
27.387.4.9StructureSetTime	964
27.387.4.10StudyInstanceUID	964
27.388dcm::Waveform Class Reference	964
27.388.1Detailed Description	964
27.388.2Constructor & Destructor Documentation	964
27.388.2.1Waveform	964
27.389dcm::WLMFindQuery Class Reference	964
27.389.1Detailed Description	966
27.389.2Constructor & Destructor Documentation	966
27.389.2.1WLMFindQuery	966
27.389.3Member Function Documentation	966
27.389.3.1GetAbstractSyntaxUID	966
27.389.3.2GetTagListByLevel	966
27.389.3.3GetValidDataSet	966
27.389.3.4InitializeDataSet	966
27.389.3.5ValidateQuery	966
27.389.4Friends And Related Function Documentation	967
27.389.4.1QueryFactory	967
27.390dcm::Writer Class Reference	967
27.390.1Detailed Description	969
27.390.2Constructor & Destructor Documentation	970
27.390.2.1Writer	970
27.390.2.2~Writer	970
27.390.3Member Function Documentation	970
27.390.3.1CheckFileMetaInformationOff	970
27.390.3.2CheckFileMetaInformationOn	970
27.390.3.3GetFile	970
27.390.3.4GetStreamPtr	970
27.390.3.5SetCheckFileMetaInformation	970
27.390.3.6SetFile	970
27.390.3.7SetFileName	971
27.390.3.8SetStream	971

27.390.3.9SetWriteDataSetOnly	971
27.390.3.10Write	971
27.390.4Friends And Related Function Documentation	971
27.390.4.1StreamImageWriter	971
27.390.5Member Data Documentation	971
27.390.5.1Ofstream	971
27.390.5.2Stream	971
27.391gdcmm::XMLDictReader Class Reference	972
27.391.1Detailed Description	973
27.391.2Constructor & Destructor Documentation	973
27.391.2.1XMLDictReader	973
27.391.2.2~XMLDictReader	973
27.391.3Member Function Documentation	973
27.391.3.1CharacterDataHandler	973
27.391.3.2EndElement	973
27.391.3.3GetDict	973
27.391.3.4HandleDescription	973
27.391.3.5HandleEntry	973
27.391.3.6StartElement	973
27.392gdcmm::XMLPrinter Class Reference	973
27.392.1Member Enumeration Documentation	975
27.392.1.1PrintStyles	975
27.392.2Constructor & Destructor Documentation	975
27.392.2.1XMLPrinter	975
27.392.2.2~XMLPrinter	975
27.392.3Member Function Documentation	975
27.392.3.1GetPrintStyle	975
27.392.3.2HandleBulkData	975
27.392.3.3Print	975
27.392.3.4PrintDataElement	975
27.392.3.5PrintDataSet	975
27.392.3.6PrintSQ	975
27.392.3.7SetFile	975
27.392.3.8SetStyle	975
27.392.4Member Data Documentation	975
27.392.4.1F	975
27.392.4.2PrintStyle	975

27.393gdcm::XMLPrivateDictReader Class Reference	976
27.393.1Detailed Description	977
27.393.2Constructor & Destructor Documentation	977
27.393.2.1XMLPrivateDictReader	977
27.393.2.2~XMLPrivateDictReader	977
27.393.3Member Function Documentation	977
27.393.3.1CharacterDataHandler	977
27.393.3.2EndElement	977
27.393.3.3GetPrivateDict	977
27.393.3.4HandleDescription	977
27.393.3.5HandleEntry	977
27.393.3.6StartElement	977
28 File Documentation	979
28.1 gdcm2pnm.dox File Reference	979
28.2 gdcm2vtk.dox File Reference	979
28.3 gdcmAAbortPDU.h File Reference	979
28.4 gdcmAAssociateACPDU.h File Reference	980
28.5 gdcmAAssociateRJPDU.h File Reference	980
28.6 gdcmAAssociateRQPDU.h File Reference	981
28.7 gdcmAbstractSyntax.h File Reference	982
28.8 gdcmanon.dox File Reference	983
28.9 gdcmAnonymizeEvent.h File Reference	983
28.10gdcmAnonymizer.h File Reference	984
28.11gdcmApplicationContext.h File Reference	985
28.12gdcmApplicationEntity.h File Reference	986
28.13gdcmAReleaseRPPDU.h File Reference	987
28.14gdcmAReleaseRQPDU.h File Reference	988
28.15gdcmARTIMTimer.h File Reference	989
28.16gdcmASN1.h File Reference	990
28.17gdcmAsynchronousOperationsWindowSub.h File Reference	990
28.18gdcmAttribute.h File Reference	991
28.19gdcmAudioCodec.h File Reference	993
28.20gdcmBase64.h File Reference	993
28.21gdcmBaseCompositeMessage.h File Reference	994
28.22gdcmBaseNormalizedMessage.h File Reference	995
28.23gdcmBasePDU.h File Reference	996

28.24gdcmBaseQuery.h File Reference	997
28.25gdcmBaseRootQuery.h File Reference	998
28.26gdcmBasicOffsetTable.h File Reference	1000
28.27gdcmBitmap.h File Reference	1001
28.28gdcmBitmapToBitmapFilter.h File Reference	1002
28.29gdcmBoxRegion.h File Reference	1003
28.30gdcmByteBuffer.h File Reference	1003
28.31gdcmByteSwap.h File Reference	1004
28.32gdcmByteSwapFilter.h File Reference	1005
28.33gdcmByteValue.h File Reference	1006
28.34gdcmCAPICryptoFactory.h File Reference	1007
28.35gdcmCAPICryptographicMessageSyntax.h File Reference	1007
28.36gdcmCEchoMessages.h File Reference	1008
28.37gdcmCFindMessages.h File Reference	1009
28.38gdcmCMoveMessages.h File Reference	1010
28.39gdcmCodec.h File Reference	1011
28.40gdcmCoder.h File Reference	1012
28.41gdcmCodeString.h File Reference	1013
28.42gdcmCommand.h File Reference	1014
28.43gdcmCommandDataSet.h File Reference	1016
28.44gdcmCompositeMessageFactory.h File Reference	1016
28.45gdcmCompositeNetworkFunctions.h File Reference	1017
28.46gdcmConstCharWrapper.h File Reference	1018
28.47gdcmconv.dox File Reference	1019
28.48gdcmCP246ExplicitDataElement.h File Reference	1019
28.49gdcmCryptoFactory.h File Reference	1019
28.50gdcmCryptographicMessageSyntax.h File Reference	1020
28.51gdcmCSAElement.h File Reference	1021
28.52gdcmCSAHeader.h File Reference	1023
28.53gdcmCSAHeaderDict.h File Reference	1023
28.54gdcmCSAHeaderDictEntry.h File Reference	1025
28.55gdcmCStoreMessages.h File Reference	1026
28.56gdcmCurve.h File Reference	1027
28.57gdcmDataElement.h File Reference	1028
28.58gdcmDataEvent.h File Reference	1029
28.59gdcmDataSet.h File Reference	1030
28.60gdcmDataSetEvent.h File Reference	1031

28.61gdcmDataSetHelper.h File Reference	1032
28.62gdcmDecoder.h File Reference	1033
28.63gdcmDefinedTerms.h File Reference	1034
28.64gdcmDeflateStream.h File Reference	1035
28.65gdcmDefs.h File Reference	1035
28.66gdcmDeltaEncodingCodec.h File Reference	1037
28.67gdcmDICOMDIR.h File Reference	1037
28.68gdcmDICOMDIRGenerator.h File Reference	1038
28.69gdcmDict.h File Reference	1039
28.70gdcmDictConverter.h File Reference	1041
28.71gdcmDictEntry.h File Reference	1041
28.72gdcmDictPrinter.h File Reference	1043
28.73gdcmDicts.h File Reference	1043
28.74gdcmdiff.dox File Reference	1044
28.75gdcmDIMSE.h File Reference	1045
28.76gdcmDirectionCosines.h File Reference	1045
28.77gdcmDirectory.h File Reference	1046
28.78gdcmDirectoryHelper.h File Reference	1047
28.79gdcmDummyValueGenerator.h File Reference	1048
28.80gdcmdump.dox File Reference	1048
28.81gdcmDumper.h File Reference	1048
28.82gdcmElement.h File Reference	1049
28.82.1 Macro Definition Documentation	1051
28.82.1.1 VRDS16ILLEGAL	1051
28.83gdcmEncapsulatedDocument.h File Reference	1051
28.84gdcmEnumeratedValues.h File Reference	1052
28.85gdcmEvent.h File Reference	1052
28.85.1 Macro Definition Documentation	1054
28.85.1.1 gdcmEventMacro	1054
28.86gdcmException.h File Reference	1054
28.87gdcmExplicitDataElement.h File Reference	1055
28.88gdcmExplicitImplicitDataElement.h File Reference	1056
28.89gdcmFiducials.h File Reference	1056
28.90gdcmFile.h File Reference	1057
28.91gdcmFileAnonymizer.h File Reference	1058
28.92gdcmFileChangeTransferSyntax.h File Reference	1059
28.93gdcmFileDecompressLookupTable.h File Reference	1059

28.94gdcmlFileDerivation.h File Reference	1060
28.95gdcmlFileExplicitFilter.h File Reference	1061
28.96gdcmlFileMetaInformation.h File Reference	1061
28.97gdcmlFilename.h File Reference	1062
28.98gdcmlFileNameEvent.h File Reference	1063
28.99gdcmlFilenameGenerator.h File Reference	1064
28.100gdcmlFileSet.h File Reference	1065
28.101gdcmlFileStreamer.h File Reference	1066
28.102gdcmlFindPatientRootQuery.h File Reference	1067
28.103gdcmlFindStudyRootQuery.h File Reference	1068
28.104gdcmlFragment.h File Reference	1069
28.105gdcmlgendir.dox File Reference	1071
28.106gdcmlGlobal.h File Reference	1071
28.107gdcmlGroupDict.h File Reference	1072
28.108gdcmlIconImage.h File Reference	1072
28.109gdcmlIconImageFilter.h File Reference	1073
28.110gdcmlIconImageGenerator.h File Reference	1074
28.111gdcmlImage.h File Reference	1075
28.112gdcmlImageApplyLookupTable.h File Reference	1076
28.113gdcmlImageChangePhotometricInterpretation.h File Reference	1077
28.114gdcmlImageChangePlanarConfiguration.h File Reference	1077
28.115gdcmlImageChangeTransferSyntax.h File Reference	1078
28.116gdcmlImageCodec.h File Reference	1079
28.117gdcmlImageConverter.h File Reference	1080
28.118gdcmlImageFragmentSplitter.h File Reference	1081
28.119gdcmlImageHelper.h File Reference	1082
28.120gdcmlImageReader.h File Reference	1083
28.121gdcmlImageRegionReader.h File Reference	1083
28.122gdcmlImageToImageFilter.h File Reference	1084
28.123gdcmlImageWriter.h File Reference	1085
28.124gdcmlimg.dox File Reference	1086
28.125gdcmlImplementationClassUIDSub.h File Reference	1086
28.126gdcmlImplementationUIDSub.h File Reference	1087
28.127gdcmlImplementationVersionNameSub.h File Reference	1088
28.128gdcmlImplicitDataElement.h File Reference	1089
28.129gdcmlinfo.dox File Reference	1089
28.130gdcmlIOD.h File Reference	1090

28.131dcmIODEntry.h File Reference	1091
28.132dcmIODs.h File Reference	1093
28.133dcmIPPSorter.h File Reference	1095
28.134dcmItem.h File Reference	1096
28.135dcmJPEG12Codec.h File Reference	1097
28.136dcmJPEG16Codec.h File Reference	1097
28.137dcmJPEG2000Codec.h File Reference	1098
28.138dcmJPEG8Codec.h File Reference	1099
28.139dcmJPEGCodec.h File Reference	1100
28.140dcmJPEGLSCodec.h File Reference	1101
28.141dcmJSON.h File Reference	1102
28.142dcmKAKADUCodec.h File Reference	1103
28.143dcmLegacyMacro.h File Reference	1104
28.143.1Macro Definition Documentation	1104
28.143.1.1GDCM_LEGACY	1104
28.143.1.2GDCM_LEGACY_BODY	1105
28.143.1.3GDCM_LEGACY_REPLACED_BODY	1105
28.144dcmLO.h File Reference	1105
28.145dcmLookupTable.h File Reference	1105
28.146dcmMacro.h File Reference	1106
28.147dcmMacroEntry.h File Reference	1108
28.147.1Macro Definition Documentation	1109
28.147.1.1GDCMMACROENTRY_H	1109
28.148dcmMacros.h File Reference	1109
28.149dcmMaximumLengthSub.h File Reference	1111
28.150dcmMD5.h File Reference	1112
28.151dcmMediaStorage.h File Reference	1113
28.152dcmMeshPrimitive.h File Reference	1114
28.153dcmModalityPerformedProcedureStepCreateQuery.h File Reference	1115
28.154dcmModalityPerformedProcedureStepSetQuery.h File Reference	1116
28.155dcmModule.h File Reference	1117
28.156dcmModuleEntry.h File Reference	1118
28.157dcmModules.h File Reference	1120
28.158dcmMovePatientRootQuery.h File Reference	1121
28.159dcmMoveStudyRootQuery.h File Reference	1122
28.160dcmNActionMessages.h File Reference	1123
28.161dcmNCreateMessages.h File Reference	1124

28.162dcmNDeleteMessages.h File Reference	1125
28.163dcmNestedModuleEntries.h File Reference	1125
28.164dcmNetworkEvents.h File Reference	1127
28.165dcmNetworkStateID.h File Reference	1128
28.166dcmNEventReportMessages.h File Reference	1129
28.167dcmNGetMessages.h File Reference	1129
28.168dcmNormalizedMessageFactory.h File Reference	1130
28.169dcmNormalizedNetworkFunctions.h File Reference	1131
28.170dcmNSetMessages.h File Reference	1132
28.171dcmObject.h File Reference	1132
28.172dcmOpenSSLCryptoFactory.h File Reference	1133
28.173dcmOpenSSLCryptographicMessageSyntax.h File Reference	1134
28.174dcmOpenSSL7CryptoFactory.h File Reference	1135
28.175dcmOpenSSL7CryptographicMessageSyntax.h File Reference	1136
28.176dcmOrientation.h File Reference	1138
28.177dcmOverlay.h File Reference	1138
28.178dcmpap3.dox File Reference	1139
28.179dcmParseException.h File Reference	1139
28.180dcmParser.h File Reference	1141
28.181dcmPatient.h File Reference	1141
28.182dcmPDataTFPDU.h File Reference	1142
28.183dcmPDBElement.h File Reference	1143
28.184dcmPDBHeader.h File Reference	1145
28.185dcmpdf.dox File Reference	1145
28.186dcmPDFCodec.h File Reference	1145
28.187dcmPDUFactory.h File Reference	1146
28.188dcmPersonName.h File Reference	1147
28.189dcmPGXCodec.h File Reference	1147
28.190dcmPhotometricInterpretation.h File Reference	1148
28.191dcmPixelFormat.h File Reference	1149
28.192dcmPixmap.h File Reference	1150
28.193dcmPixmapReader.h File Reference	1151
28.194dcmPixmapToPixmapFilter.h File Reference	1152
28.195dcmPixmapWriter.h File Reference	1153
28.196dcmPNMCodec.h File Reference	1154
28.197dcmPreamble.h File Reference	1155
28.198dcmPresentationContext.h File Reference	1156

28.199dcmPresentationContextAC.h File Reference	1157
28.200dcmPresentationContextGenerator.h File Reference	1159
28.201dcmPresentationContextRQ.h File Reference	1159
28.202dcmPresentationDataValue.h File Reference	1160
28.203dcmPrinter.h File Reference	1161
28.204dcmPrivateTag.h File Reference	1162
28.205dcmProgressEvent.h File Reference	1164
28.206dcmPVRGCodec.h File Reference	1164
28.207dcmPythonFilter.h File Reference	1165
28.208dcmQueryBase.h File Reference	1166
28.209dcmQueryFactory.h File Reference	1167
28.210dcmQueryImage.h File Reference	1168
28.211dcmQueryPatient.h File Reference	1169
28.212dcmQuerySeries.h File Reference	1170
28.213dcmQueryStudy.h File Reference	1170
28.214dcmraw.dox File Reference	1171
28.215dcmRAWCodec.h File Reference	1171
28.216dcmReader.h File Reference	1172
28.217dcmRegion.h File Reference	1174
28.218dcmRescaler.h File Reference	1175
28.219dcmRLECodec.h File Reference	1175
28.220dcmRoleSelectionSub.h File Reference	1176
28.221dcmscanner.dox File Reference	1177
28.222dcmScanner.h File Reference	1177
28.223dcmscu.dox File Reference	1178
28.224dcmSegment.h File Reference	1178
28.225dcmSegmentedPaletteColorLookupTable.h File Reference	1179
28.226dcmSegmentHelper.h File Reference	1180
28.227dcmSegmentReader.h File Reference	1182
28.228dcmSegmentWriter.h File Reference	1183
28.229dcmSequenceOfFragments.h File Reference	1184
28.230dcmSequenceOfItems.h File Reference	1185
28.231dcmSerieHelper.h File Reference	1185
28.232dcmSeries.h File Reference	1187
28.233dcmServiceClassApplicationInformation.h File Reference	1188
28.234dcmServiceClassUser.h File Reference	1189
28.235dcmSHA1.h File Reference	1189

28.236dcmSimpleSubjectWatcher.h File Reference	1190
28.237dcmSmartPointer.h File Reference	1191
28.238dcmSOPClassExtendedNegociationSub.h File Reference	1192
28.239dcmSOPClassUIDToIOD.h File Reference	1193
28.240dcmSorter.h File Reference	1194
28.241dcmSpacing.h File Reference	1196
28.242dcmSpectroscopy.h File Reference	1196
28.243dcmSplitMosaicFilter.h File Reference	1197
28.244dcmStaticAssert.h File Reference	1198
28.244.1Macro Definition Documentation	1198
28.244.1.1GDCM_DO_JOIN	1198
28.244.1.2GDCM_DO_JOIN2	1198
28.244.1.3GDCM_JOIN	1198
28.244.1.4GDCM_STATIC_ASSERT	1199
28.245dcmStreamImageReader.h File Reference	1199
28.246dcmStreamImageWriter.h File Reference	1199
28.247dcmStrictScanner.h File Reference	1200
28.248dcmString.h File Reference	1201
28.249dcmStringFilter.h File Reference	1202
28.250dcmStudy.h File Reference	1203
28.251dcmSubject.h File Reference	1204
28.252dcmSurface.h File Reference	1205
28.253dcmSurfaceHelper.h File Reference	1206
28.254dcmSurfaceReader.h File Reference	1207
28.255dcmSurfaceWriter.h File Reference	1208
28.256dcmSwapCode.h File Reference	1208
28.257dcmSwapper.h File Reference	1209
28.258dcmSystem.h File Reference	1210
28.259dcmTable.h File Reference	1211
28.260dcmTableEntry.h File Reference	1212
28.261dcmTableReader.h File Reference	1213
28.262dcmTag.h File Reference	1215
28.263dcmTagPath.h File Reference	1215
28.264dcmTagToVR.h File Reference	1216
28.265dcmTar.dox File Reference	1217
28.266dcmTerminal.h File Reference	1217
28.267dcmTestDriver.h File Reference	1219

28.269dcmTesting.h File Reference	1219
28.269dcmTrace.h File Reference	1220
28.269.1Macro Definition Documentation	1221
28.269.1.1GDCM_FUNCTION	1221
28.269.1.2gdcmAssertAlwaysMacro	1221
28.269.1.3gdcmAssertMacro	1221
28.269.1.4gdcmDebugMacro	1222
28.269.1.5gdcmErrorMacro	1222
28.269.1.6gdcmWarningMacro	1222
28.270dcmTransferSyntax.h File Reference	1223
28.270dcmTransferSyntaxSub.h File Reference	1224
28.270dcmType.h File Reference	1225
28.270dcmTypes.h File Reference	1226
28.270dcmUIDGenerator.h File Reference	1227
28.270dcmUIDs.h File Reference	1228
28.270dcmULAction.h File Reference	1228
28.270dcmULActionAA.h File Reference	1229
28.270dcmULActionAE.h File Reference	1230
28.270dcmULActionAR.h File Reference	1231
28.280dcmULActionDT.h File Reference	1232
28.280dcmULBasicCallback.h File Reference	1232
28.280dcmULConnection.h File Reference	1233
28.280dcmULConnectionCallback.h File Reference	1234
28.280dcmULConnectionInfo.h File Reference	1235
28.280dcmULConnectionManager.h File Reference	1237
28.280dcmULEvent.h File Reference	1237
28.280dcmULTransitionTable.h File Reference	1238
28.280dcmULWritingCallback.h File Reference	1240
28.280dcmUNExplicitDataElement.h File Reference	1240
28.290dcmUNExplicitImplicitDataElement.h File Reference	1241
28.290dcmUnpacker12Bits.h File Reference	1241
28.290dcmUsage.h File Reference	1242
28.290dcmUserInformation.h File Reference	1244
28.290dcmUIDGenerator.h File Reference	1245
28.290dcmValidate.h File Reference	1245
28.290dcmValue.h File Reference	1246
28.290dcmValueIO.h File Reference	1247

28.290	gdcmVersion.h File Reference	1248
28.290	gdcmviewer.dox File Reference	1249
28.300	gdcmVL.h File Reference	1249
28.300	gdcmVM.h File Reference	1250
28.301	Macro Definition Documentation	1251
28.301.1	1.TYPETOLENGTH	1251
28.300	gdcmVR.h File Reference	1251
28.302	Macro Definition Documentation	1253
28.302.1	1.TYPETOENCODING	1253
28.302.1.2	VRTemplateCase	1253
28.300	gdcmVR16ExplicitDataElement.h File Reference	1253
28.300	gdcmWaveform.h File Reference	1254
28.305	gdcmWin32.h File Reference	1254
28.305	Macro Definition Documentation	1254
28.305.1	1.GDCM_EXPORT	1254
28.300	gdcmWLMFindQuery.h File Reference	1255
28.300	gdcmWriter.h File Reference	1255
28.300	gdcmxml.dox File Reference	1256
28.300	gdcmXMLDictReader.h File Reference	1257
28.310	gdcmXMLPrinter.h File Reference	1257
28.310	gdcmXMLPrivateDictReader.h File Reference	1258
28.310	README.txt File Reference	1259
28.310	TestsList.txt File Reference	1259
28.310	tkGDCMImageReader.h File Reference	1259
28.314	Macro Definition Documentation	1260
28.314.1	1.VTK_CMYK	1260
28.314.1.2	VTK_INVERSE_LUMINANCE	1260
28.314.1.3	VTK_LOOKUP_TABLE	1260
28.314.1.4	VTK_YBR	1260
28.315	tkGDCMImageReader2.h File Reference	1260
28.315	Macro Definition Documentation	1261
28.315.1	1.VTK_CMYK	1261
28.315.1.2	VTK_INVERSE_LUMINANCE	1261
28.315.1.3	VTK_LOOKUP_TABLE	1261
28.315.1.4	VTK_YBR	1261
28.310	tkGDCMImageWriter.h File Reference	1261
28.317	tkGDCMMedicalImageProperties.h File Reference	1261

28.318tkGDCMPolyDataReader.h File Reference	1262
28.319tkGDCMPolyDataWriter.h File Reference	1263
28.320tkGDCMTesting.h File Reference	1263
28.321tkGDCMThreadedImageReader.h File Reference	1264
28.322tkGDCMThreadedImageReader2.h File Reference	1265
28.323tkImageColorViewer.h File Reference	1265
28.324tkImageMapToColors16.h File Reference	1266
28.325tkImageMapToWindowLevelColors2.h File Reference	1266
28.326tkImagePlanarComponentsToComponents.h File Reference	1267
28.327tkImageRGBToYBR.h File Reference	1267
28.328tkImageYBRToRGB.h File Reference	1268
28.329tkLookupTable16.h File Reference	1268
28.330tkRTStructSetProperties.h File Reference	1269
29 Example Documentation	1271
29.1 AWTMedical3.java	1271
29.2 BasicAnonymizer.cs	1275
29.3 BasicImageAnonymizer.cs	1276
29.4 CastConvertPhilips.py	1278
29.5 ChangePrivateTags.cxx	1280
29.6 ChangeSequenceUltrasound.cxx	1281
29.7 CheckBigEndianBug.cxx	1282
29.8 ClinicalTrialAnnotate.cxx	1284
29.9 ClinicalTrialIdentificationWorkflow.cs	1285
29.10CompressImage.cxx	1288
29.11CompressLossyJPEG.cs	1289
29.12Compute3DSpacing.cxx	1290
29.13Convert16BitsTo8Bits.cxx	1291
29.14ConvertMPL.py	1292
29.15ConvertMultiFrameToSingleFrame.cxx	1293
29.16ConvertNumpy.py	1294
29.17ConvertPIL.py	1295
29.18ConvertRGBToLuminance.cxx	1297
29.19ConvertSingleBitTo8Bits.cxx	1297
29.20ConvertToQImage.cxx	1299
29.21CreateARGBImage.cxx	1300
29.22CreateCMYKImage.cxx	1301

29.23CreateFakePET.cxx	1302
29.24CreateFakeRTDOSE.cxx	1304
29.25CreateJPIPDataSet.cxx	1306
29.26CreateRAWStorage.py	1307
29.27csa2img.cxx	1309
29.28CStoreQtProgress.cxx	1311
29.29DecompressImage.cs	1313
29.30DecompressImage.java	1314
29.31DecompressImage.py	1315
29.32DecompressImageMultiframe.cs	1316
29.33DecompressJPEGFile.cs	1318
29.34DecompressPixmap.java	1319
29.35DiffFile.cxx	1320
29.36DiscriminateVolume.cxx	1321
29.37DumbAnonymizer.py	1325
29.38DumpADAC.cxx	1326
29.39DumpExamCard.cxx	1331
29.40DumpGEMSMovieGroup.cxx	1338
29.41DumpImageHeaderInfo.cxx	1344
29.42DumpPhilipsECHO.cxx	1346
29.43DumpToshibaDTI.cxx	1351
29.44DumpToSQLITE3.cxx	1353
29.45DuplicatePCDE.cxx	1355
29.46ELSCINT1WaveToText.cxx	1357
29.47EncapsulateFileInRawData.cxx	1359
29.48ExtractEncapsulatedFile.cs	1360
29.49ExtractEncryptedContent.cxx	1361
29.50ExtractIconFromFile.cxx	1362
29.51ExtractImageRegion.cs	1363
29.52ExtractImageRegion.java	1365
29.53ExtractImageRegionWithLUT.cs	1366
29.54Extracting_All_Resolution.cxx	1367
29.55ExtractOneFrame.cs	1373
29.56Fake_Image_Using_Stream_Image_Writer.cxx	1374
29.57FileAnonymize.cs	1377
29.58FileAnonymize.java	1378
29.59FileChangeTS.cs	1379

29.60FileChangeTSLossy.cs	1381
29.61 FileStreaming.cs	1384
29.62FindAllPatientName.py	1385
29.63FixBrokenJ2K.cxx	1385
29.64FixCommaBug.py	1387
29.65FixJAIBugJPEGLS.cxx	1388
29.66FixOrientation.cxx	1391
29.67gdcmmorthoplanes.cxx	1392
29.68gdcmmreslice.cxx	1398
29.69gdcmmrtionplan.cxx	1400
29.70gdcmmrtplan.cxx	1405
29.71 gdcmmscene.cxx	1409
29.72gdcmmtexture.cxx	1410
29.73gdcmmvolume.cxx	1412
29.74GenAllIVR.cxx	1413
29.75GenerateDICOMDIR.cs	1416
29.76GenerateRTSTRUCT.cxx	1417
29.77GenerateStandardSOPClasses.cxx	1419
29.78GenFakeIdentifyFile.cxx	1420
29.79GenFakeImage.cxx	1423
29.80GenLongSeqs.cxx	1424
29.81 GenSeqs.cxx	1426
29.82GetArray.cs	1427
29.83GetJPEGSamplePrecision.cxx	1428
29.84GetPortionCSAHeader.py	1430
29.85GetSequenceUltrasound.cxx	1431
29.86GetSubSequenceData.cxx	1433
29.87headsq2dcm.py	1435
29.88HelloActiviz.cs	1436
29.89HelloActiviz2.cs	1437
29.90HelloActiviz3.cs	1439
29.91 HelloActiviz4.cs	1439
29.92HelloActiviz5.cs	1440
29.93HelloSimple.java	1441
29.94HelloVizWorld.cxx	1442
29.95HelloVTKWorld.cs	1443
29.96HelloVTKWorld.java	1444

29.97HelloVTKWorld2.cs	1445
29.98HelloWorld.cxx	1446
29.99HelloWorld.py	1447
29.100J22tomultisc.cxx	1448
29.101LargeVRDSExplicit.cxx	1449
29.102MagnifyFile.cxx	1451
29.103MakeTemplate.cxx	1452
29.104ManipulateFile.cs	1453
29.105ManipulateFile.py	1454
29.106ManipulateSequence.py	1456
29.107MergeFile.py	1457
29.108MergeTwoFiles.cxx	1457
29.109MetalImageMD5Activiz.cs	1459
29.110MIPViewer.java	1460
29.111MpegVideoInfo.cs	1462
29.112MPRViewer.java	1467
29.113MPRViewer2.java	1469
29.114MrProtocol.cxx	1473
29.115NewSequence.cs	1480
29.116NewSequence.py	1481
29.117OffscreenImage.cxx	1482
29.118PatchFile.cxx	1483
29.119PhilipsPrivateRescaleInterceptSlope.py	1485
29.120PlaySound.py	1486
29.121pmsct_rgb1.cxx	1487
29.122PrivateDict.py	1490
29.123PublicDict.cxx	1491
29.124QIDO-RS.cxx	1492
29.125ReadAndDumpDICOMDIR.cxx	1492
29.126ReadAndDumpDICOMDIR.py	1496
29.127ReadAndPrintAttributes.cxx	1498
29.128ReadExplicitLengthSQIVR.cxx	1500
29.129ReadFiles.java	1500
29.130ReadGEMSSDO.cxx	1501
29.131ReadMultiTimesException.cxx	1504
29.132ReadSeriesIntoVTK.java	1505
29.133ReadUTF8QtDir.cxx	1506

29.134RefCounting.cs	1507
29.135ReformatFile.cs	1508
29.136RemovePrivateTags.py	1509
29.137RescaleImage.cs	1510
29.138Reslicesphere.cxx	1511
29.139ReWriteSCAsMR.py	1519
29.140Re2img.cxx	1520
29.141tstructapp.cxx	1523
29.142ScanDirectory.cs	1524
29.143ScanDirectory.java	1525
29.144ScanDirectory.py	1529
29.145SendFileSCU.cs	1530
29.146SimplePrint.cs	1531
29.147SimplePrintPatientName.cs	1532
29.148SimpleScanner.cxx	1532
29.149SortImage.cxx	1534
29.150SortImage.py	1536
29.151SortImage2.cs	1536
29.152StandardizeFiles.cs	1537
29.153StreamImageReaderTest.cxx	1539
29.154TestByteSwap.cxx	1542
29.155TestReader.cxx	1544
29.156TestReader.py	1545
29.157Threadgdcn.cxx	1546
29.158TraverseModules.cxx	1550
29.159id_unique.cxx	1551
29.160VolumeSorter.cxx	1551
29.161WriteBuffer.py	1554

Index**1557**

Chapter 1

GDCM Documentation

This is the developpers documentation.

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

<http://gdcm.sourceforge.net/2.6/gdcm-2.6.1.pdf>

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

<http://gdcm.sourceforge.net/2.6/gdcm-2.6.1-doc.tar.gz>

Author

Mathieu Malaterre

Chapter 2

off-screen rendering of DICOM images

2.1 SYNOPSIS

```
gdcm2pnm [options] file-in bitmap-out
```

2.2 DESCRIPTION

The **gdcm2pnm** command line program takes as input a DICOM file and produces a rendered bitmap file.

2.3 PARAMETERS

file-in DICOM input filename

bitmap-out Bitmap output filename

2.4 OPTIONS

2.4.1 OPTIONS

2.4.2 general options

```
-h    --help  
      print this help text and exit  
  
-v    --version  
      print version information and exit  
  
-V    --verbose  
      verbose mode (warning+error).  
  
-W    --warning  
      warning mode, print warning information
```

```
-E  --error
    error mode, print error information

-D  --debug
    debug mode, print debug information
```

2.5 Simple usage

gdcm2pnm will take as input DICOM and render it into a bitmap file using the window/level attributes value.

```
$ gdcm2pnm input.dcm output.png
```

It is much different from the **gdcmraw** or **gdcmimg** command line tool as it will render a DICOM image. This means that the output will be rendered in 8bits ready for display.

2.6 SEE ALSO

gdcm2vtk(1), **gdcmimg(1)**

2.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 3

Convert a file supported by VTK into DICOM.

3.1 SYNOPSIS

```
gdc2vtk [options] file-in file-out
```

3.2 DESCRIPTION

The **gdc2vtk** takes as input any file supported by VTK (including DICOM file) and will generate as output a DICOM file.

3.3 PARAMETERS

```
file-in    input filename (DICOM or VTK supported)

file-out    output filename (DICOM or VTK supported)
```

3.4 OPTIONS

3.4.1 OPTIONS

--force-rescale	force rescale.
--force-spacing	force spacing.
--palette-color	when supported generate a PALETTE COLOR file.
--argb	when supported generate a ARGB file.
--compress	when supported generate a compressed file.
--use-vtkdicom	Use vtkDICOMImageReader (instead of GDCM).
--modality	set Modality.
--lower-left	set lower left.
--shift	set shift.
--scale	set scale.
--compress	set compression (MetaIO).
-T --study-uid	Study UID.
-S --series-uid	Series UID.
--root-uid	Root UID.

3.4.2 compression options

```
-J --jpeg          Compress image in jpeg.
-K --j2k          Compress image in j2k.
-L --jpegls       Compress image in jpeg-ls.
-R --rle          Compress image in rle (lossless only).
```

3.4.3 general options

```
-h  --help          print this help text and exit
-v  --version       print version information and exit
-V  --verbose       verbose mode (warning+error).
-W  --warning       warning mode, print warning information
-E  --error         error mode, print error information
-D  --debug         debug mode, print debug information
```

3.4.4 environment variable

```
GDCM_ROOT_UID Root UID
```

3.5 DESCRIPTION

Convert a file supported by VTK into DICOM.

Typical usage is:

```
$ gdcmm2vtk inputfile output.dcm
```

It uses the internal factory mechanism of VTK to recognize a file (CanRead function). See VTK supported file here:

What image file formats can VTK read and write? http://www.vtk.org/Wiki/VTK_FAQ#What_image_file_formats_can_VTK

If your input file has 4 components, the 4th comp (alpha) will be removed from the output file as DICOM does not support alpha component anymore (see `--argb` option).

Special care was taken for the following file format:

1. DICOM: Direction Cosines and `vtkMedicalImageInformation` are passed to the output
2. BMP: The file can be saved with a Lookup Table (see `--palette-color`)
3. GE Signa: `vtkMedicalImageProperties` is passed to the output
4. MINC: Direction Cosines is passed to the output
5. TIFF: `vtkTIFFReader` is currently in bad shape in VTK (different behavior in VTK 5.2 and `git/master`). Only u

3.5.1 CONVERT MetaImage (mhd, mha)

```
$ gdcmm2vtk inputfile output.mha
```

This command will convert the input DICOM file: inputfile into a MetaImage .mha file. Same goes for .mhd file.

3.5.2 CONVERT MHA/MHD

```
$ gdc2vtk inputfile output.mha
```

or

```
$ gdc2vtk inputfile output.mhd
```

This command will convert the input DICOM file: inputfile into a MetaImageData .mha/.mhd file.

3.5.3 CONVERT VTI

```
$ gdc2vtk inputfile output.vti
```

This command will convert the input DICOM file: inputfile into a XML VTK ImageData .vti file.

3.5.4 CONVERT VTK

```
$ gdc2vtk inputfile output.vtk
```

This command will convert the input DICOM file: inputfile into an old VTK Structured PointSets .vtk file.

3.6 CONVERT DICOM

```
$ gdc2vtk input.dcm output.dcm
```

[vtkGDCMImageReader](#) will be used to read in a DICOM file, not the default `vtkDICOMImageReader`. See option `--use-vtkdicom` to use `vtkDICOMImageReader`.

3.7 RoundTrip DICOM to MHD to DICOM

```
$ gdc2vtk input_ybr.dcm output.mhd
$ gdc2vtk --modality US --imageformat 7 output.mhd output.dcm
```

The above section shows how to convert a DICOM using the Photometric Interpretation of YBR_FULL (or even YBR_FULL_422 is lossy) into another file format: MetaImage (mhd). Since this file format does not handle color space, we have to explicitly set it using the `--imageformat` command line option. The `--modality` command line option is required in this case since the default Secondary Capture Image Storage Class family does not allow for YBR Photometric Interpretation.

3.8 gdc2vtk notes

IMPORTANT NOTE: The internal VTK structured will be filled from the input DICOM, and then pass to the output DICOM writer. Some information might be lost during the conversion DICOM to VTK to DICOM. This option is mostly used to test the `vtkGDCMImageReader/vtkGDCMImageWriter` combination.

IMPORTANT NOTE: When converting from a lossy format such as JPEG, the information of lossiness is important. The output DICOM will contains the required Lossy Image Compression attribute that indicates that image was lossy-compressed somewhere along the pipeline. See also `gdcmimg` (better handling of JPEG in general).

IMPORTANT NOTE: When using `-use-vtkdicom` the output DICOM file will always be written as MR Image Storage as this information is not available from the reader itself. This allow setting the Image Orientation (Patient) properly.

3.9 SEE ALSO

`gdcmdump(1)`, `gdcmviewer(1)`, `gdcmimg(1)`

3.10 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 4

Tool to anonymize a DICOM file.

4.1 SYNOPSIS

```
gdcmanon [options] file-in file-out
gdcmanon [options] dir-in  dir-out
```

4.2 DESCRIPTION

The **gdcmanon** tool is an implementation of PS 3.15 / E.1 / Basic Application Level Confidentiality Profile (Implementation of E.1.1 De-identify & E.1.2 Re-identify)

This tool is split into two very different operating mode:

- An implementation of PS 3.15, see `-e` and `-d` flags
- A dumb mode, see `-dumb`

Dumb mode and PS 3.15 do not work well together, you should really only use one type of anonymization. In case of doubt, avoid using `-dumb`.

In order to use the PS 3.15 implementation (`-d` & `-e` flag), you'll need a certificate to do de-identification operations, and the associated private key to do the re-identification operation. If you are only doing a one-shot anonymization and do not need to properly re-identify the DICOM file, you can safely discard the private key and only keep the certificate. See OpenSSL section below for an example on how to generate the private key/certificate pair.

`gdcmanon` will exit early if OpenSSL was not configured/build properly into the library (see `GDCM_USE_SYSTEM_OPENSSL` in `CMakeLists.txt`).

4.3 PARAMETERS

```
file-in  DICOM input filename
```

```
file-out  DICOM output filename
```

or

```
file-in  DICOM input directory
```

```
file-out  DICOM output directory
```

4.4 OPTIONS

You need to specify at least one operating mode, from the following list (and only one):

4.4.1 Required parameters

-e --de-identify	De-identify DICOM (default)
-d --re-identify	Re-identify DICOM
--dumb	Dumb mode anonymizer

Warning when operating in dumb mode, you need to also specify an operation to do, such as 'remove' or 'empty' a tag, see below the dumb mode options.

4.4.2 OPTIONS

-i --input	DICOM filename / directory
-o --output	DICOM filename / directory
-r --recursive	recursively process (sub-)directories.
--continue	Do not stop when file found is not DICOM.
--root-uid	Root UID.
--resources-path	Resources path.
-k --key	Path to RSA Private Key.
-c --certificate	Path to Certificate.

4.4.3 encryption options

--des	DES.
--des3	Triple DES.
--aes128	AES 128.
--aes192	AES 192.
--aes256	AES 256.

4.4.4 dumb mode options

--empty %d,%d	DICOM tag(s) to empty
--remove %d,%d	DICOM tag(s) to remove
--replace %d,%d,%s	DICOM tag(s) to replace

4.4.5 general options

-h --help	print this help text and exit
-v --version	print version information and exit
-V --verbose	verbose mode (warning+error).
-W --warning	warning mode, print warning information
-E --error	error mode, print error information
-D --debug	debug mode, print debug information

4.4.6 environment variable

GDCM_ROOT_UID Root UID
GDCM_RESOURCES_PATH path pointing to resources files (Part3.xml, ...)

4.5 Typical usage

4.5.1 De-identification (anonymization, encrypt)

The only thing required for this operation is a certificate file (in PEM format).

```
$ gdcmanon --certificate certificate.pem -e original.dcm original_anonymized.dcm
```

You can use `--asn1` option from `gdcmdump` to dump the generated DataSet as ASN1 structure (see `gdcmdump(1)` for example).

4.5.2 Re-identification (de-anonymization, decrypt)

The only thing required for this operation is a private key (in PEM format). It is required that the private key used for the re-identification process, was the actual private key used to generate the certificate file (certificate.pem) used during the de-identification step.

```
$ gdcmanon --key privatekey.pem -d original_anonymized.dcm original_copy.dcm
```

You can then check that `original.dcm` and `original_copy.dcm` are identical.

4.5.3 Multiple files caveat

It is very important to understand the following section, when anonymizing more than one single file. When anonymizing multiple DICOM files, you are required to use the directory input. You cannot call multiple time the `gdcmanon` command line tool. Indeed the tool stores in memory during the process only a hash table of conversion so that each time a particular value is found it get always replaced by the same de-identified value (think: consistent Series Instance UID).

4.5.4 Dumb mode

This functionality is not described in the DICOM standard. Users are advised that improper use of that mode is not recommended, meaning that important tag can be emptied/removed/replaced resulting in illegal/invalid DICOM file. Only use when you know what you are doing. If you delete a Type 1 attribute, chance is that your DICOM file will be not accepted in most DICOM third party viewer. Unfortunately this is often this mode that is implemented in popular DICOM Viewer, always prefer what the DICOM standard describes, and avoid the dumb mode.

The following example shows how to use dumb mode and achieve 5 operations at the same time:

- Empty the tag (0010,0010) Patient's Name,
- Empty the tag (0010,0020) Patient ID,
- Remove the tag (0010,0040) Patient's Sex
- Remove the tag (0010,1010) Patient's Age

- Replace the tag (0010,1030) Patient's Weight with the value '10'

You are required to check which DICOM attribute is Type 1 and Type 1C, before trying to '**Empty**' or '**Remove**' a particular DICOM attribute. For the same reason, you are required to check what are valid value in a replace operation.

```
$ gdcmanon --dumb --empty 10,10 --empty 10,20 --remove 10,40 --remove 10,1010 --replace 10,1030,10 012345.002.050
```

Multiple operation of `--dumb` mode can take place, just reuse the output of the previous operation. Always use `gdcmdump` on the input and output file to check what was actually achieved. You can use a diff program to check only what changed (see `gdcmdiff(1)` for example).

4.5.4.1 Irreversible Anonymization

In some very rare cases, one would want to anonymize using the PS 3.15 mode so as to take benefit of the automatic conversion of all content that could contain Patient related information.

In the end all Patient related information has been removed and has been secretly stored in the 0400,0500 DICOM attribute. However to make sure that no-one ever try to break that security using brute-force algorithm, one want want to remove completely this DICOM attribute. This will make the DICOM:

- Completely free of any Patient related information (as per PS 3.15 specification)
- Remove any mean of people to brute force attack the file to find out the identity of the Patient

In this case one could simply do, as a first step execute the reversible anonymizer:

```
$ gdcmanon -c certificate.pem input.dcm anonymized_reversible.dcm
```

and now completely remove the DICOM attribute containing the secretly encrypted Patient related information:

```
$ gdcmanon --dumb --remove 400,500 --remove 12,62 --remove 12,63 anonymized_reversible.dcm anonymized_irreversible.dcm
```

Remarks

As mentionned in DICOM Sup 142, this anonymization is preferred over de-identification since: It is not required that the Encrypted Attributes Data Set be created; indeed, there may be circumstances where the Dataset is expected to be archived long enough that any contemporary encryption technology may be inadequate to provide long term protection against unauthorized recovery of identification

4.6 OpenSSL

On most system you can have access to OpenSSL to generate the Private Key/Certificate pair.

4.6.1 Generating a Private Key

Command line to generate a rsa key (512bit)

```
$ openssl genrsa -out CA_key.pem
```

Command line to generate a rsa key (2048bit)

```
$ openssl genrsa -out CA_key.pem 2048
```

Command line to generate a rsa key (2048bit) + passphrase

```
$ openssl genrsa -des3 -out CA_key.pem 2048
```

4.6.2 Generating a Certificate

From your previously generated Private Key, you can now generate a certificate in PEM (DER format is currently not supported).

```
$ openssl req -new -key CA_key.pem -x509 -days 365 -out CA_cert.cer
```

4.7 DICOM Standard:

Page to the DICOM Standard:

<http://dicom.nema.org/>

The DICOM Standard at the time of releasing gdcmanon is:

<ftp://medical.nema.org/medical/dicom/2008/>

Direct link to PS 3.15-2008:

ftp://medical.nema.org/medical/dicom/2008/08_15pu.pdf

4.8 Warnings

Certain attributes may still contains Protected Health Information (PHI) after an anonymization step. This is typically the case for Patient's Address (0010,1040). The reason is that this particular attribute is not supposed to be in the composite IODs in the first place. DICOM Supp 142 includes it (however gdcmanon does not implement it).

4.9 SEE ALSO

gdcconv(1), gdcmdump(1), gdcmdiff(1), openssl(1), dumpasn1(1)

4.10 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 5

Tool to convert DICOM to DICOM.

5.1 SYNOPSIS

```
gdcmconv [options] file-in file-out
```

5.2 DESCRIPTION

The **gdcmconv** command line program takes as input a DICOM file (file-in) and process it to generate an output DICOM file (file-out). The command line option dictate the type of operation(s) gdcmconv will use to generate the output file.

5.3 PARAMETERS

```
file-in    DICOM input filename
```

```
file-out   DICOM output filename
```

5.4 OPTIONS

5.4.1 PARAMETERS

```
-i --input      DICOM filename
-o --output     DICOM filename
```

5.4.2 OPTIONS

```
-X --explicit      Change Transfer Syntax to explicit.
-M --implicit      Change Transfer Syntax to implicit.
-U --use-dict       Use dict for VR (only public by default).
  --with-private-dict Use private dict for VR (advanced user only).
-C --check-meta     Check File Meta Information (advanced user only).
  --root-uid        Root UID.
  --remove-gl       Remove group length (deprecated in DICOM 2008).
  --remove-private-tags Remove private tags.
  --remove-retired  Remove retired tags.
```

5.4.3 image options

```
-l --apply-lut           Apply LUT (non-standard, advanced user only).
-P --photometric-interpretation %s  Change Photometric Interpretation (when possible).
-w --raw                Decompress image.
-d --deflated           Compress using deflated (gzip).
-J --jpeg               Compress image in jpeg.
-K --j2k                Compress image in j2k.
-L --jpegls             Compress image in jpeg-ls.
-R --rle                Compress image in rle (lossless only).
-F --force              Force decompression/merging before recompression/splitting.
  --generate-icon       Generate icon.
  --icon-minmax %d,%d   Min/Max value for icon.
  --icon-auto-minmax    Automatically compute best Min/Max values for icon.
  --compress-icon       Decide whether icon follows main TransferSyntax or remains uncompressed.
  --planar-configuration [01] Change planar configuration.
-Y --lossy              Use the lossy (if possible) compressor.
-S --split %d           Write 2D image with multiple fragments (using max size)
```

5.4.4 JPEG options

```
-q --quality %*f        set quality.
```

5.4.5 JPEG-LS options

```
-e --lossy-error %*i    set error.
```

5.4.6 J2K options

```
-r --rate %*f           set rate.
-q --quality %*f        set quality.
-t --tile %d,%d         set tile size.
-n --number-resolution %d set number of resolution.
  --irreversible        set irreversible.
```

5.4.7 general options

```
-h --help               print this help text and exit
-v --version            print version information and exit
-V --verbose            verbose mode (warning+error).
-W --warning            warning mode, print warning information
-E --error              error mode, print error information
-D --debug              debug mode, print debug information
```

5.4.8 special options

```
-I --ignore-errors      convert even if file is corrupted (advanced users only, see disclaimers).
```

5.4.9 environment variable

```
GDCM_ROOT_UID Root UID
```

5.5 Simple usage

gdcmmconv is a great tool to convert broken DICOM implementation into properly parsable DICOM file. Usage is simply:

```
$ gdcmmconv input.dcm output.dcm
```

or if you prefer being explicit:

```
$ gdcmmconv -i input.dcm -o output.dcm
```

Even though **gdcmmconv** can overwrite directly on the same file (`input.dcm = output.dcm`), it is recommended that user should first convert into a different file to make sure the bug is properly handled by GDCM.

Typical cases where you would want to use **gdcmmconv** in its simple form:

- convert non-cp246 conforming file into conforming cp246,
- convert implicit little endian transfer syntax file meta header into proper explicit little endian transfer syntax,
- convert the GE-13 bytes bug,
- convert dual syntax file: implicit/explicit,
- convert Philips dual Little Endian/Big Endian file,
- convert GDCM 1.2.0 broken UN-2-bytes fields,
- &...
- All other broken files listed in the supported section.

When no option other is used, only the dataset is inspected. So encapsulated Pixel Data, for instance, is not inspected for well known bugs.

When doing this kind of work, this is usually a good idea to perform some kind of quality control, see **gdcmmconv** Quality Control section (down below).

5.6 Typical usage

5.6.1 File Meta Header

Running

```
$ gdcmmconv input.dcm output.dcm
```

Is not enough to recompute file meta header, when input file is buggy. You may want to use: `--check-meta`

```
$ gdcmmconv --check-meta input.dcm output.dcm
```

See typical cases such as: `GE_DLX-8-MONO2-PrivateSyntax.dcm` or `PICKER-16-MONO2-No_DicomV3_Preamble.dcm` from `gdcmmData`.

5.6.2 Conversion to Explicit Transfer Syntax

To convert a file that was written using Implicit Transfer Syntax into Explicit Transfer Syntax simply use:

```
$ gdcmmconv --explicit uncompressed.dcm compressed.dcm
```

5.6.3 Compressing to lossless JPEG

To compress an uncompressed DICOM file to a JPEG Lossless encapsulated format:

```
$ gdcmmconv --jpeg uncompressed.dcm compressed.dcm
```

5.6.4 Compressing to lossy JPEG

To compress an uncompressed DICOM file to a JPEG Lossy encapsulated format:

```
$ gdcmmconv --lossy --jpeg -q 90 uncompressed.dcm compressed.dcm
```

Note:

`-q` is just one of the many way to specify lossy quality, you need to inspect the other cmd line flag to specify

5.6.5 Compressing to lossless JPEG-LS

To compress an uncompressed DICOM file to a JPEG-LS Lossless encapsulated format:

```
$ gdcmmconv --jpegls uncompressed.dcm compressed.dcm
```

5.6.6 Compressing to lossy JPEG-LS

To compress an uncompressed DICOM file to a JPEG-LS Lossy encapsulated format:

```
$ gdcmmconv --lossy --jpegls -e 2 uncompressed.dcm lossy_compressed.dcm
```

Note:

`-e` (or `--lossy-error`) means that the maximum tolerate error is 2 for each pixel value

5.6.7 Compressing to lossless J2K

To compress an uncompressed DICOM file to a JPEG-2000 Lossless encapsulated format:

```
$ gdcmmconv --j2k uncompressed.dcm compressed.dcm
```

5.6.8 Compressing to lossy J2K

To compress an uncompressed DICOM file to a JPEG-2000 Lossy encapsulated format:

```
$ gdcmmconv --lossy -q 55,50,45 --j2k uncompressed.dcm lossy_compressed.dcm
```

Note:

`-q` is just one of the many way to specify lossy quality, you need to inspect the other cmd line flag to specify

5.6.9 Compressing to lossless RLE

To compress an uncompressed DICOM file to a RLE Lossless encapsulated format:

```
$ gdcconv --rle uncompressed.dcm compressed.dcm
```

There is no such thing as lossy RLE compression.

5.6.10 Split encapsulated DICOM:

To split an encapsulated stream into smaller chunk (1024 bytes each):

```
$ gdcconv --split 1024 rle.dcm rle_1024.dcm
```

If an odd number of bytes is passed it will be rounded down to the next even number (eg. 1025 -> 1024) since DICOM only allow even number for Value Length.

5.6.11 Forcing (re)compression

Sometime it is necessary to use the `-force` option. By default when user specify `-j2k` and input file is already in JPEG 2000 encapsulated DICOM format then no operation takes places. By using `-force` you make sure that (re)compression operation takes places.

Real life example of why you would use `-force`:

- When Pixel Data is missing data / is padded with junk
- When you would like to make sure GDCM can handle decompression & recompression cycle

5.6.12 Decompressing a Compressed DICOM

```
$ gdcconv --raw compressed.dcm uncompressed.dcm
```

5.6.13 Compressing an uncompressed Icon

By default when compressing a DICOM Image file, `gdcconv` will not compress the icon. A user option needs to be turned on to explicitly force the compression of the Icon Image Sequence Pixel Data

For example, by default we will not compress the Icon Image Sequence Pixel Data attribute:

```
$ gdcconv --jpeg gdcData/simpleImageWithIcon.dcm uncompressed_icon.dcm
```

In the following example we will explicitly compress the Icon Image Sequence Pixel Data attribute. In that case the same Transfer Syntax is being used for both the main Pixel Data and the Pixel Data from the Icon Image Sequence:

```
$ gdcconv --jpeg --compress-icon gdcData/simpleImageWithIcon.dcm compressed_icon.dcm
```

5.6.14 Generating an Icon

For some application it might be necessary to produce a small preview of the main image to be able to quickly load that short preview instead of the main image. In that case:

```
$ gdcconv --raw --generate-icon gdcData/test.acr test_icon.dcm
```

In some cases the main Pixel Data element is expressed as pixel defined on 16bits. Since Icon can only store at most pixel of size 8bits, a rescale operation needs to take place. In order to properly select a better interval for doing the rescale operation user can specify the min max used for the rescale operation:

```
$ gdcconv --raw --generate-icon --icon-minmax 0,192 gdcData/012345.002.050.dcm icon_minmax.dcm
```

5.6.15 Changing the planar Configuration

Often RLE files are compressed using a different Planar Configuration (RRR ... GGG... BBB...) instead of the usual triplet (RGB ... RGB ... RGB). So upon decompression the Planar Configuration is 1. This is perfectly legal in DICOM, however this is unconventional, and thus it may be a good idea to also change the planar configuration and set it to the default :

```
$ gdcconv --raw --planar-configuration 0 compressed.dcm uncompressed1.dcm
```

To reinvert the planar configuration of file 'uncompressed1.dcm', simply do:

```
$ gdcconv --raw --planar-configuration 1 uncompressed1.dcm uncompressed2.dcm
```

5.7 Lossless Conversion

When talking about lossless conversion, there is an ambiguity that need to be understood. To achieve higher compression ratio, the RGB color space is usually not used, in favor of a YBR one. Changing from one color space to the other is (bit level) not lossless.

For more detail, see what are the true lossless transformations as described:

http://gdc.sourceforge.net/wiki/index.php/Color_Space_Transformations

5.8 Quality Control

One important part when using gdcconv it to have a way to quality control the output.

You can use 3rd party tool to check the output of gdcconv is correct.

5.8.1 DCMTK / dicom3tools

Using another DICOM implementation such as the one from DCMTK or dicom3tools can be a good process to check the output of gdcconv.

- For DCMTK use: dcmdump
- For dicom3tools use: dcdump

For reference, gdcconv --raw will act as dcmdjpeg +cn +px, since it never tries to convert color space.

5.8.2 VIM: vimdiff

You can setup your favorite editor to compare the output, for instance in vim:

```
autocmd BufReadPre *.dcm set ro
autocmd BufReadPost *.dcm silent %!gdcm dump -M +uc "%"
```

then simply do:

```
$ vimdiff input.dcm output.dcm
```

5.8.3 vbindiff

On UNIX you can visually compare binary file using the vbindiff command:

```
$ vbindiff input.dcm output.dcm
```

5.9 SEE ALSO

gdcmdump(1), **gdcmmraw(1)**, **gdcminfo(1)**, **gdcmdiff(1)**

5.10 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 6

dumps differences of two DICOM files

6.1 SYNOPSIS

```
gdcmdiff [options] file1 file2
```

6.2 DESCRIPTION

The **gdcmdiff** command line program takes as input two DICOM files: file1 and file2.

6.3 PARAMETERS

```
file1    DICOM input filename
file2    DICOM output filename
```

6.4 OPTIONS

6.4.1 OPTIONS

```
-m      --meta          Compare metainformation. Default is off.
-t <n>  --truncate <n>  String values trimmed to n characters.
```

6.4.2 general options

```
-h      --help          print this help text and exit
-v      --version       print version information and exit
-V      --verbose       verbose mode (warning+error).
-W      --warning       warning mode, print warning information
```

```
-E  --error  
    error mode, print error information  
  
-D  --debug  
    debug mode, print debug information
```

6.5 Simple usage

gdcmdiff is a great tool to produce a diff in between two DICOM files. Usage is simply:

```
$ gdcmdiff input1.dcm input2.dcm
```

6.6 SEE ALSO

gdcmdump(1), **gdcminfo(1)**

6.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 7

dumps a DICOM file, it will display the structure and values contained in the specified DICOM file.

7.1 SYNOPSIS

```
gdcm dump [options] dcm_file
gdcm dump [options] dcm_directory
```

7.2 DESCRIPTION

The **gdcm dump** command line program dumps a DICOM file to the console. For those familiar with **dcm dump** (DCMTK) output, **gdcm dump** has some minor differences. Namely:

- For Implicit Transfer Syntax **gdcm dump** will print ?? instead of the dictionary VR

gdcm dump has a limited private dictionary that is used to lookup private element whenever possible.

7.3 PARAMETERS

```
dcm_file          DICOM input filename
dcm_directory     DICOM input directory
```

7.4 OPTIONS

7.4.1 OPTIONS

-x --xml-dict	generate the XML dict (only private elements for now).
-r --recursive	recursive (input is a directory)
-d --dump	dump value (limited use).
-p --print	print value instead of simply dumping (default).
-c --color	print in color.
-C --csa	print SIEMENS CSA Header (0029,[12]0,SIEMENS CSA HEADER).
-P --pdb	print GEMS Protocol Data Block (0025,1b,GEMS_SERS_01).
--elscint	print ELSCINT Protocol Information (01f7,26,ELSCINT1).
--vepro	print VEPRO Protocol Information (0055,20,VEPRO VIF 3.0 DATA).

```

                or VEPRO Protocol Information (0055,20,VEPRO VIM 5.0 DATA).
--sds           print Philips MR Series Data Storage (1.3.46.670589.11.0.0.12.2) Information (2005,32,Philips)
-A --asn1       print encapsulated ASN1 structure >(0400,0520).
--map-uid-names map UID to names.

```

7.4.2 general options

```

-h --help
    print this help text and exit

-v --version
    print version information and exit

-V --verbose
    verbose mode (warning+error).

-W --warning
    warning mode, print warning information

-E --error
    error mode, print error information

-D --debug
    debug mode, print debug information

```

7.4.3 special options

```

-I --ignore-errors  dumps even if file is corrupted (advanced users only, see disclaimers).

```

7.5 Typical usage

7.5.1 Printing Implicit Transfer Syntax

The VR are not found in the file, thus are presented with a "(??)", and right next to it (if found) the correct VR.

Eg.:

```
$ gdcmdump GE_DLX-8-MONO2-PrivateSyntax.dcm
```

```

# Dicom-File-Format
\&...
(0008,0000) ?? (UL) 434                                # 4,1 Generic Group Length
(0008,0005) ?? (CS) [ISO_IR 100]                        # 10,1-n Specific Character Set
(0008,0008) ?? (CS) [ORIGINAL\\PRIMARY\\SINGLE PLANE ]   # 30,2-n Image Type
(0008,0016) ?? (UI) [1.2.840.10008.5.1.4.1.1.12.1]      # 28,1 SOP Class UID
(0008,0018) ?? (UI) [1.2.840.113619.2.16.1.0.906539207.1.24207] # 42,1 SOP Instance UID
(0008,0020) ?? (DA) [19980923]                          # 8,1 Study Date
(0008,0021) ?? (DA) [19980923]                          # 8,1 Series Date
(0008,0022) ?? (DA) [19980923]                          # 8,1 Acquisition Date
(0008,0023) ?? (DA) [19980923]                          # 8,1 Content Date
(0008,0030) ?? (TM) [101229.000]                        # 10,1 Study Time
(0008,0031) ?? (TM) [101229.000]                        # 10,1 Series Time
(0008,0032) ?? (TM) [102653.000]                        # 10,1 Acquisition Time
(0008,0033) ?? (TM) [102653.000]                        # 10,1 Content Time
\&...

```


7.5.2 Print Private Attributes

GDCM has a limited private dictionary. Whenever possible, it will try to lookup the private data element.

```
$ gdcmdump 012345.002.050.dcm
```

```
\&...
(0009,0010) LO [GEMS_IDEN_01] # 12,1 Private Creator
(0009,1001) LO [GE_GENESIS_FF ] # 14,1 Full fidelity
(0009,1002) SH [MRCV] # 4,1 Suite id
(0009,1004) SH [SIGNA ] # 6,1 Product id
(0009,1027) SL 985968524 # 4,1 Image actual date
(0009,1030) SH [19356UMR2 ] # 10,1 Service id
(0009,1031) SH [999 ] # 4,1 Mobile location number
(0009,10e3) UI [1.2.840.113619.1.1.4.1762386977] # 32,1 Equipment UID
(0009,10e6) SH [08] # 2,1 Genesis Version - now
(0009,10e7) UL 2757786872 # 4,1 Exam Record checksum
(0009,10e9) SL 985968523 # 4,1 Actual series data time stamp
\&...
(0019,0000) UL 1208 # 4,1 Generic Group Length
(0019,0010) LO [GEMS_ACQU_01] # 12,1 Private Creator
(0019,100f) DS [424.399994] # 10,1 Horiz. Frame of ref.
(0019,1011) SS 0 # 2,1 Series contrast
\&...
(0019,10e0) DS [0.000000] # 8,1 User data 24 {# DTI Diffusion Dir., relea
(0019,10e2) DS [0.000000] # 8,1 Velocity Encode Scale
(0019,10f2) SS 0 # 2,1 Fast phases
(0019,10f9) DS [98] # 2,1 Transmit gain
\&...
(0021,0000) UL 372 # 4,1 Generic Group Length
(0021,0010) LO [GEMS_RELA_01] # 12,1 Private Creator
(0021,1003) SS 0 # 2,1 Series from which Prescribed
\&...
```

7.5.3 SIEMENS CSA Header

Using this option it is possible to dump as a readable text what is contained in the private attribute as found in typical SIEMENS MR DICOM file.

Eg.:

```
$ gdcmdump --csa MR_SIEMENS_forceLoad29-1010_29-1020.dcm
```

```
(0029,0010)siemens csa header
Image shadow data (0029,xx10)

0 - 'EchoLinePosition' VM 1, VR IS, SyngoDT 6, NoOfItems 6, Data '64      '
1 - 'EchoColumnPosition' VM 1, VR IS, SyngoDT 6, NoOfItems 6, Data '64      '
2 - 'EchoPartitionPosition' VM 1, VR IS, SyngoDT 6, NoOfItems 6, Data '32      '
3 - 'UsedChannelMask' VM 1, VR UL, SyngoDT 9, NoOfItems 6, Data '255      '
4 - 'Actual3DImaPartNumber' VM 1, VR IS, SyngoDT 6, NoOfItems 0, Data
5 - 'ICE_Dims' VM 1, VR LO, SyngoDT 19, NoOfItems 6, Data 'X_1_1_1_1_1_31_1_1_1_1_19'
6 - 'B_value' VM 1, VR IS, SyngoDT 6, NoOfItems 6, Data '0      '
7 - 'Filter1' VM 1, VR IS, SyngoDT 6, NoOfItems 0, Data
8 - 'Filter2' VM 1, VR IS, SyngoDT 6, NoOfItems 0, Data
\&...
```

7.5.4 GEMS Protocol Data Block

Using this option it is possible to dump as a readable text what is contained in the private attribute as found in typical GEMS MR DICOM file.

Protocol Data Block : 0025,xx1b,GEMS_SERS_01

```
$ gdcmdump --pdb GE_MR_0025xx1bProtocolDataBlock.dcm
```

```
ENTRY "Head First"
POSITION "Supine"
ANREF "NA"
COIL "HEAD"
PLANE "OBLIQUE"
SEDESCFLAG "1"
SEDESC "AX FSE T2"
IMODE "2D"
PSEQ "FSE-XL"
IOPT "FC, EDR, TRF, Fast"
PLUG "22"
FILTCHOICE "None"
BWRT "-1"
TRICKSIMG "1"
TAG_SPACE "7"
TAG_TYPE "None"
\&...
```

7.5.5 ELSCINT Protocol Information

Using this option it is possible to dump as a readable text what is contained in the private attribute as found in typical ELSCINT CT DICOM file.

ELSCINT Protocol Information: (01f7,26,ELSCINT1)

```
$ gdcmdump --elscint ELSCINT1_ProtocolInformation.dcm
```

```
ELSCINT1 Dumping info from tag (01f7,26,elscint1)
```

```
ELSCINT1/Item name: []
  ApprovedStep [yes]
  RefSurview [1\0]
  STD-first-img-pos [11.5]
  current-step [yes]
  ntimed-steps [0]
  orig-n-slices [390]
  protocol-file [Head_Multi_1032_usr.proc]
  protocol-name [FACE-TRAUMA/Head/Hx]
  protocol-path [/usr/diamond.root/spr/]
  protocol-step [1]
  protocol-version [2.51]
```

```
ELSCINT1/Item name: [doseright]
```

```
  ACS [n/a]
  ACS-bed-position [0]
  ACS-calc-mas [0]
  ACS-ig-parameter [0]
  ACS-learn-allowed [no]
  ACS-water-radius [-1.000000]
  ACS-water-radius-scan [-1]
\&...
```

7.5.6 VEPRO Protocol Information

Using this option it is possible to dump as a readable text what is contained in the private attribute as found in typical VEPRO CT DICOM file.

ELSCINT Protocol Information: (0055,20,VEPRO VIM 5.0 DATA)

```
$ gdcmdump --vepro VEPRO_ProtocolInformation.dcm

VIMDATA2: (0055,20,VEPRO VIM 5.0 DATA)
  ID: VIM
  Version: 5.0
  UserName:
  UserAdress1: Name of Institution
  UserAdress2: Street of Institution
  UserAdress3: City of Institution
  UserAdress4:
  UserAdress5:
  RecDate: 20101001
  RecTime: 211321
  RecPlace:
  RecSource: DICOM Distributor
  DF1: P-09/10-41808
  DF2: Sultana Razia
  DF3: 19411001
  DF4: F
  DF5:
  DF6:
  DF7:
  DF8: CT Scan Brain without Contrast
  DF9: 10/10-0034873
  DF10: 10/10-00348
  DF11:
  DF12:
  DF13:
  DF14: Head 0.5
  DF15: 4
  DF16:
  DF17:
  DF18:
  DF19:
  DF20:
  StudyUID: 1.2.392.200036.9116.2.6.1.48.1214228007.1285934880.206831
  SeriesUID: 1.2.392.200036.9116.2.6.1.48.1214228007.1285935201.938653
  Modality: CT
```

7.5.7 Philips Private MR Series Data Storage (1.3.46.670589.11.0.0.12.2)

Using this option it is possible to dump as a readable text what is contained in the private attribute as found in typical Philips Private MR Series Data Storage file.

PMS Series Data Storage (2005,32,Philips MR Imaging DD 002)

```
$ gdcmdump --sds PMS_SeriesDataStorage.dcm

\&...
PMS/Item name: [PDF_CONTROL_GEN_PARS/IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_CONTROL_PREP_PARS /IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_CONTROL_RECON_PARS/IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_CONTROL_SCAN_PARS /IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_EXAM_PARS /IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_HARDWARE_PARS /IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_PREP_PARS /IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_SPT_PARS/IEEE_PDF/Y ]
```

```

SP_scan_resol [256\256] # 2
SP_pda_profiles [0\0] # 2
SP_filter [324074] # 1
SP_analyse_with_iqt [0] # 1
SP_main_system_type [3] # 1
SP_gradient_system [6] # 1
SP_coil_type [2\2\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
SP_coil_id [2\34\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
SP_coil_part [0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
SP_act_q [0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
SP_act_coil_freq [0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
SP_coil_m_pos [255\255\255\0\0\0\0\0\0\0\0\0\0\0\0\255] # 16
SP_coil_t_pos [255\128\255\0\0\0\0\0\0\0\0\0\0\0\0\255] # 16
SP_surface_coil_con [0\1\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
SP_proton_freq [127801349] # 1
SP_tm_result [2\2\2\2\2\2\2\2\2\2\2\2\2\2\2\2] # 16
SP_f0_result [0] # 1
SP_as_result [0] # 1
SP_po_result [0] # 1
SP_rg_result [0] # 1
SP_dc_result [0] # 1
SP_ph_result [0] # 1
\&...

```

7.5.8 Encapsulated ASN1 Structure

This option is mainly used for dumping the ASN1 structure of the encrypted Attribute (0040,0520)

```
$ gdcmdump encrypted.dcm
```

```

\&...
(0040,0500) SQ # u/1,1 Encrypted Attributes Sequence
  (fffe,e000) na (Item with undefined length)
    (0040,0510) UI [1.2.840.10008.1.2] # 18,1 Encrypted Content Transfer Syntax UID
    (0040,0520) OB 30\82\03\ba\06\09\2a\86\48\55\04\08\13 # 958,1 Encrypted Content
  (fffe,e00d)
(fffe,e0dd)
\&...

```

```
$ gdcmdump --asn1 encrypted.dcm
```

```

0:d=0 hl=4 l= 954 cons: SEQUENCE
4:d=1 hl=2 l= 9 prim: OBJECT :pkcs7-envelopedData
15:d=1 hl=4 l= 939 cons: cont [ 0 ]
19:d=2 hl=4 l= 935 cons: SEQUENCE
23:d=3 hl=2 l= 1 prim: INTEGER :00
26:d=3 hl=4 l= 366 cons: SET
30:d=4 hl=4 l= 362 cons: SEQUENCE
34:d=5 hl=2 l= 1 prim: INTEGER :00
37:d=5 hl=2 l= 82 cons: SEQUENCE
39:d=6 hl=2 l= 69 cons: SEQUENCE
41:d=7 hl=2 l= 11 cons: SET
43:d=8 hl=2 l= 9 cons: SEQUENCE
45:d=9 hl=2 l= 3 prim: OBJECT :countryName
50:d=9 hl=2 l= 2 prim: PRINTABLESTRING :AU
54:d=7 hl=2 l= 19 cons: SET
56:d=8 hl=2 l= 17 cons: SEQUENCE
58:d=9 hl=2 l= 3 prim: OBJECT :stateOrProvinceName
63:d=9 hl=2 l= 10 prim: PRINTABLESTRING :Some-State
75:d=7 hl=2 l= 33 cons: SET
77:d=8 hl=2 l= 31 cons: SEQUENCE
79:d=9 hl=2 l= 3 prim: OBJECT :organizationName
84:d=9 hl=2 l= 24 prim: PRINTABLESTRING :Internet Widgits Pty Ltd
110:d=6 hl=2 l= 9 prim: INTEGER :AC966D88787A51B4

```

```
121:d=5 hl=2 l= 13 cons: SEQUENCE
123:d=6 hl=2 l= 9 prim: OBJECT :rsaEncryption
134:d=6 hl=2 l= 0 prim: NULL
136:d=5 hl=4 l= 256 prim: OCTET STRING [HEX DUMP]:822368070285AD756C962ECB973514B291F946...
396:d=3 hl=4 l= 558 cons: SEQUENCE
400:d=4 hl=2 l= 9 prim: OBJECT :pkcs7-data
411:d=4 hl=2 l= 29 cons: SEQUENCE
413:d=5 hl=2 l= 9 prim: OBJECT :aes-256-cbc
424:d=5 hl=2 l= 16 prim: OCTET STRING [HEX DUMP]:3B49AFE71749F2BFF1519EBAEA95A393
442:d=4 hl=4 l= 512 prim: cont [ 0 ]
```

7.6 SEE ALSO

gdcmdump(1), **gdcmrw(1)**, **gdcmanon(1)**

7.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 8

Tool to generate a DICOMDIR file from a File-Set.

8.1 SYNOPSIS

```
gdcmgendir [options] file-in file-out
```

8.2 DESCRIPTION

8.3 PARAMETERS

file-in DICOM input filename

file-out DICOM output filename

8.4 OPTIONS

8.4.1 Parameters

8.4.2 OPTIONS

-i --input	DICOM filename or directory
-o --output	DICOM filename or directory
-r --recursive	recursive.
--descriptor	descriptor.
--root-uid	Root UID.

8.4.3 general options

-h --help	print this help text and exit
-v --version	print version information and exit

```
-V  --verbose
    verbose mode (warning+error).

-W  --warning
    warning mode, print warning information

-E  --error
    error mode, print error information

-D  --debug
    debug mode, print debug information
```

8.4.4 environment variable

```
GDCM_ROOT_UID Root UID
```

8.5 Typical usage

8.6 NOTE

One may have to run some preliminary steps in order to get gdcmgendir to generate the DICOMDIR file. Namely two steps:

- Batch renaming of the DICOM filename into something compatible with ISO 9660 filename convention
- Convert all DICOM file into the Explicit VR Little Endian Uncompressed (1.2.840.10008.1.2.1)

Step 1. can be solved in a numerous way. Eg. on UNIX environment this could either be solved using the `mkisofs` command line tool. Filenames should not contains any extension since the VR CS does not allow for the '.' character. Only upper case, digit 0-9, the space ' ' and the underscore '_' character are valid in VR CS, with a maximum of 8 bytes. Another simple tool that can be handy is 'rename' in conjunction with 'basename'.

Step 2. can simply be achieved using the `gdcconv` command line tool:

```
$ for i in `ls IMG*`; do gdcconv --raw --force $i /tmp/out/$i; done
```

8.7 SEE ALSO

gdcconv(1), **gdcmanon(1)**, **rename(1)**, **mkisofs(1)**

8.8 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 9

Manipulate DICOM image file.

`gdcmimg` is a low level tool to allow de-/encapsulation from/to DICOM image. This tool does not understand Transfer Syntax conversion. It will encapsulate the raw data as-is. This has some impact in some cases, see special warnings below.

It is important to note that `gdcmimg` can only encapsulate proper input file, for instance JPG and or JP2 are accepted since an associated DICOM Transfer Syntax can be found. However input such as TIFF and/or PNG are not, since DICOM does not support those. See instead a tool such as `gdcm2vtk`.

9.1 SYNOPSIS

```
gdcmimg [options] file-in file-out
```

9.2 DESCRIPTION

The **`gdcmimg`** command line tool can be used in two fashions:

- 1. Converting a recognized file format into its encapsulated DICOM counterpart,
- 2. Anonymizing a rectangular portion of a DICOM file.

9.3 PARAMETERS

```
file-in    input filename (non-DICOM)
```

```
file-out   DICOM output filename
```

9.4 OPTIONS

9.4.1 PARAMETERS

```
-i --input      Input filename  
-o --output     Output filename
```

9.4.2 OPTIONS

```

--endian %s      Endianness (LSB/MSB).
-d --depth %d    Depth (Either 8/16/32 or BitsAllocated eg. 12 when known).
--sign %s        Pixel sign (0/1).
--spp %d         Sample Per Pixel (1/3).
-s --size %d,%d  Size.
-C --sop-class-uid SOP Class UID (name or value).
-T --study-uid   Study UID.
-S --series-uid  Series UID.
--root-uid       Root UID.

```

9.4.3 fill options

```

-R --region %d,%d Region.
-F --fill %d      Fill with pixel value specified.

```

9.4.4 general options

```

-h --help
    print this help text and exit

-v --version
    print version information and exit

-V --verbose
    verbose mode (warning+error).

-W --warning
    warning mode, print warning information

-E --error
    error mode, print error information

-D --debug
    debug mode, print debug information

```

9.4.5 environment variable

```
GDCM_ROOT_UID Root UID
```

9.5 Supported File Format (appropriate file extension) gdcming

will base it's conversion process based on the file extension. Follows the list of recognized file extension. When no extension is found, DICOM file is assumed.

input format

```

* RAW      (raw, rawl, gray, rgb)
* RLE      (rle)
* PNM      (pgm, pnm, ppm)
* JPEG-LS  (jls)
* JPEG 2000 (jp2, j2k, j2c, jpx, jpc)
* JPEG     (jpg, jpeg, ljpg, ljpeg)
* DICOM    ()

```

output format:

```
* PGM      (pgm, pnm, ppm)
* DICOM    ()
```

For RAW file format, you should take special care of the `--endian` option. For the (old) JPEG file format, both the lossy and lossless format are supported, user should pay attention to the `--sign` option. For file format such as RLE or RAW, user is expected to fill in information required to find the dimension and type of input data as there is no other way to find this information. For all other file format, the properties are derived from the file format itself.

PNM file are supposed to be big endian (important for depth > 8)

9.6 Typical usage

9.6.1 Remove a rectangular part of the image

To fill the region [0,100]x[0,100] of a DICOM image simply do:

```
$ gdcimg --fill 0 --region 0,100,0,100 -i input.dcm -o output_black.dcm
```

Warning: if the Pixel Data is compressed, the image is first decompressed so that pixel can be set to 0, but it is not re-compressed.

9.6.2 Convert RAW to DICOM

Recognized extension is .raw, .rawl, .gray or .rgb (case insensitive)

```
$ gdcimg --size 512,512 --depth 16 -i input.raw -o output.dcm
```

the image will be a Secondary Capture.

When the input is 3 component, one need to specify explicitly the Samples Per Pixel:

```
$ gdcimg --size 512,512 --spp 3 input_rgb.raw output_rgb.dcm
```

When the filename contains .rgb as file extension output is automatically recognized as RGB no need to specify `--spp`

```
$ gdcimg --size 512,512 input.rgb output_rgb.dcm
```

You can use the `dd` cmd line to skip any header you would like to discard, for instance, if you would like to skip the first 108 bytes, simply do:

```
$ dd skip=108 bs=1 if=input.raw of=output.raw
```

.raw and .rawl extension are equivalent. You need to explicitly specify the endianness manually:

```
$ gdcimg --endian MSB --size 512,512 --depth 16 -i input.raw -o output.dcm
```

or

```
$ gdcimg --endian LSB --size 512,512 --depth 16 -i input.raw -o output.dcm
```

9.6.3 Convert PGM/PNM/PPM to DICOM

Recognized extensions are .pgm, .pnm, .ppm (case insensitive)

```
$ gdcming -i input.pgm -o output.dcm
```

the image will be a Secondary Capture

9.6.4 Convert RLE to DICOM

Recognized extension is .rle (case insensitive)

```
$ gdcming --size 512,512 --depth 16 -i input.rle -o output.dcm
```

the image will be a Secondary Capture

9.6.5 Convert JPEG to DICOM

Recognized extensions are .jpg, .jpeg, .ljpg, .ljpeg (case insensitive)

```
$ gdcming -i input.ljpeg -o output.dcm
```

the image will be a Secondary Capture

9.6.6 Convert J2K to DICOM

Recognized extensions are .j2k, .jp2, .jpc, .jpx, .j2c (case insensitive)

```
$ gdcming -i input.j2k -o output.dcm
```

the image will be a Secondary Capture.

All Pixel information (Bits Stored/Allocated...) will be derived from the image itself, and not from the command line options.

9.6.7 Specifying a SOP Class UID

Instead of the default Secondary Capture Image Storage, one may want to specify, say VL Photographic Image Storage.

```
$ gdcming --sop-class-uid 1.2.840.10008.5.1.4.1.1.77.1.4 input.jpg output.dcm
```

9.7 Multiple Files

gdcming handle nicely a set of files (for instance jpeg):

```
$ gdcming -C 1.2.840.10008.5.1.4.1.1.12.1 1.jpg 2.jpg 3.jpg 4.jpg output.dcm
```

It is important to specify an SOP Class that supports multi-frames images otherwise gdcming will fail.

9.8 Start Offset

In some case, one may want to create a 2D slice from an arbitrary volume (e.g 3D). In which case `--offset` becomes handy:

```
$ gdcming --offset 4954104330 --size 1673,1673 Input3D_1673_1673_1775.raw slice_1770.dcm
```

9.9 Warning

There are a couple of issues with `gdcming` implementation:

For RAW file, one should pay attention that when using `--endian MSB` the Pixel Data will be encapsulated as is (not touched by `gdcming`). Therefore the only possible transfer syntax available is Implicit VR Big Endian DLX (G.E Private). GDCM does handle this private Transfer Syntax. So if you need to convert this Transfer Syntax to another one (and allow Pixel Data manipulation), you can use:

```
$ gdcconv --raw --force input_big_endian_dlx.raw -o output_implicit_vr_little_endian.dcm
```

For JFIF file and JP2 file (with header) the header is copied into the Pixel Data element which is illegal for JP2. Use `gdcconv` to properly re-encode a JP2/JFIF file into J2K/JPG.

```
$ gdcming input.jp2 output_jp2.dcm
$ gdcconv --j2k --force output_jp2.dcm output_j2k.dcm
```

For RLE file, no check is done for crossing the row boundary. It is recommended to use `gdcconv -rle` to re-encode into a proper RLE file in case of doubt.

Of course if the compression is not ok with your setup, you can always de-encapsulated the DICOM file (typically JPEG) to a non-encapsulated form, using `gdcconv`:

```
$ gdcconv --raw input_jpeg.dcm output_raw.dcm
```

9.10 SEE ALSO

`gdcmdump(1)`, `gdc2vtk(1)`, `gdcmrw(1)`, `convert(1)`, `dd(1)`

9.11 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 10

Display meta info about the input DICOM file.

10.1 SYNOPSIS

```
gdcminfo [options] file-in
```

10.2 DESCRIPTION

The **gdcminfo** command line program takes as input a DICOM file, or a directory and process it to extract meta-information about the DICOM file processed.

10.3 PARAMETERS

```
file-in    DICOM input filename
```

10.4 OPTIONS

10.4.1 OPTIONS

-r --recursive	recursive.
-d --check-deflated	check if file is proper deflated syntax.
--resources-path	Resources path.
--md5sum	Compute md5sum of Pixel Data attribute value.
--check-compression	check the encapsulated stream compression (lossless/lossy).

10.4.2 general options

-h	--help	print this help text and exit
-v	--version	print version information and exit
-V	--verbose	verbose mode (warning+error).

```
-W  --warning
    warning mode, print warning information

-E  --error
    error mode, print error information

-D  --debug
    debug mode, print debug information
```

10.4.3 environment variable

GDCM_RESOURCES_PATH path pointing to resources files (Part3.xml, ...)

10.5 Simple usage

10.5.1 gdcmdata

Using data from gdcmdata:

```
$ gdcminfo gdcmdata/012345.002.050.dcm
```

```
MediaStorage is 1.2.840.10008.5.1.4.1.1.4 [MR Image Storage]
NumberOfDimensions: 2
Dimensions: (256,256)
Origin: (-85,21.6,108.7)
Spacing: (0.664062,0.664062,1.5)
DirectionCosines: (1,0,0,0,0,-1)
Rescale Intercept/Slope: (0,1)
SamplesPerPixel      :1
BitsAllocated        :16
BitsStored           :16
HighBit              :15
PixelRepresentation:1
Orientation Label: CORONAL
```

10.5.2 Davie Clunie datasets:

Using data from David Clunie datasets:

```
$ gdcminfo BRTUM001.dcm
```

```
MediaStorage is 1.2.840.10008.5.1.4.1.1.4.1 [Enhanced MR Image Storage]
NumberOfDimensions: 3
Dimensions: (256,256,15)
Origin: (40,-105,105)
Spacing: (0.820312,0.820312,6)
DirectionCosines: (0,1,0,0,0,-1)
Rescale Intercept/Slope: (0,1)
SamplesPerPixel      :1
BitsAllocated        :16
BitsStored           :16
HighBit              :15
PixelRepresentation:1
Orientation Label: SAGITTAL
```


10.5.3 Checking the md5sum of the Pixel Data

After compressing a DICOM file (see `gdcmconv`) using a lossless compression algorithm, it is fairly easy to compare the two files for differences at DICOM attribute level. However one operation is slightly easier to do: how to make sure the compression was actually lossless ? In this case one could use the `--md5sum` operation.

Take an uncompressed DICOM image file:

```
$ gdcminfo --md5sum SIEMENS_ImageLocationUN.dcm
```

The tool return: 0621954acd5815e0b4f7b65fcc6506b1

Now compress this file:

```
$ gdcmmconv --jpegls SIEMENS_ImageLocationUN.dcm lossless_compressed.dcm
```

and then check again the md5sum:

```
$ gdcminfo --md5sum lossless_compressed.dcm
```

The tool return: 0621954acd5815e0b4f7b65fcc6506b1

10.5.4 Checking if Pixel Data is lossless

In some environment one wish to check whether or not the DICOM file is lossless. It is fairly easy to do that in most cases. Only in two occasion this is not clear from the sole DICOM Attribute. When the Transfer Syntax is JPEG 2000 Image Compression (1.2.840.10008.1.2.4.91) and when the Transfer Syntax is JPEG-LS Lossy (Near-Lossless) Image Compression (1.2.840.10008.1.2.4.81).

In this case, the only solution is to open the Pixel Data element, read the specific JPEG header and check whether or not the JPEG transformation was lossless or not:

```
$ gdcminfo --check-compression gdcmmData/MAROTECH_CT_JP2Lossy.dcm
```

The tool returns: "Encapsulated Stream was found to be: lossy"

10.6 SEE ALSO

`gdcmdump(1)`, `gdcmraw(1)`, `gdcmconv(1)`

10.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 11

Tool to convert PAPYRUS 3.0 to DICOM.

11.1 SYNOPSIS

```
gdcmconv [options] file-in file-out
```

11.2 DESCRIPTION

The **gdcmconv** command line program takes as input a PAPYRUS 3.0 file (file-in) and process it to generate an output (pseudo) DICOM file (file-out). The command line option dictate the type of operation(s) gdcmconv will use to generate the output file.

11.3 PARAMETERS

```
file-in    DICOM input filename
```

```
file-out   DICOM output filename
```

11.4 OPTIONS

11.4.1 PARAMETERS

```
-i --input      DICOM filename
-o --output     DICOM filename
```

11.4.2 OPTIONS

```
-S --split      Split multiframe PAPYRUS 3.0 into multiples DICOM files
--decomp-pap3   Use PAPYRUS 3.0 for decompressing (can be combined with --split).
--check-iop     Check that the Image Orientation (Patient) Attribute is ok (see --split).
```

11.4.3 general options

```
-h --help
```

```
    print this help text and exit

-v  --version
    print version information and exit

-V  --verbose
    verbose mode (warning+error).

-W  --warning
    warning mode, print warning information

-E  --error
    error mode, print error information

-D  --debug
    debug mode, print debug information
```

11.4.4 environment variable

GDCM_ROOT_UID Root UID

11.5 Simple usage

gdcmap3 is a great tool to convert broken PAPYRUS 3.0 implementation into properly parsable DICOM file. Usage is simply:

```
$ gdcmap3 input.pa3 output.dcm
```

or if you prefer being explicit:

```
$ gdcmap3 -i input.pa3 -o output.dcm
```

Even though **gdcmap3** can overwrite directly on the same file (input.pa3 = output.dcm), it is recommended that user should first convert into a different file to make sure the bug is properly handled by GDCM.

11.6 SEE ALSO

gdcmdump(1), **gdcmap3**(1), **gdcminfo**(1)

11.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 12

Tool to convert PDF to PDF/DICOM.

12.1 SYNOPSIS

```
gdcmpdf [options] file-in file-out
```

12.2 DESCRIPTION

The **gdcmpdf** tool convert a PDF file (any PDF version) into an encapsulated PDF/DICOM file. By default it will try to read the PDF meta information stored in the PDF and convert this information to some specific DICOM fields (see below). However it may fails (eg. wrong password on encrypted PDF file) in which case empty value are used.

12.3 PARAMETERS

file-in PDF input filename

file-out DICOM output filename

12.4 OPTIONS

12.4.1 general options

```
-h    --help  
      print this help text and exit  
  
-v    --version  
      print version information and exit  
  
-V    --verbose  
      verbose mode (warning+error).  
  
-W    --warning  
      warning mode, print warning information  
  
-E    --error  
      error mode, print error information  
  
-D    --debug
```

```
debug mode, print debug information
```

12.5 Usage Example

```
$ wget http://gdcm.sourceforge.net/gdcm.pdf
$ gdcmpdf gdcm.pdf gdcm.dcm
```

To re-extract the encapsulated PDF file:

```
$ gdcmmraw -i gdcm.dcm -t 42,11 -o gdcm.dcm.pdf
$ diff gdcm.pdf gdcm.dcm.pdf
```

12.6 PDF Info Mapping

Here is how the PDF info is mapped to DICOM information (typical pdftinfo output):

```
Title:      GDCM Reference Manual
Subject:    Grassroots DICOM API reference
Keywords:   GDCM,DICOM,JPEG, Lossless JPEG,JPEG-LS,J2K, JPEG 2000,RLE
Author:     Mathieu Malaterre and co.
Creator:    LaTeX with hyperref package
Producer:   pdfTeX-1.21a
CreationDate: Tue Apr 28 15:34:26 2009
Tagged:     no
Pages:      1188
Encrypted:  no
Page size:  612 x 792 pts (letter)
File size:  13756841 bytes
Optimized:  yes
PDF version: 1.4
```

Converted to DICOM this leads to:

```
# Dicom-Data-Set
# Used TransferSyntax: Little Endian Explicit
(0008,0005) CS [ISO_IR 100] # 10, 1 SpecificCharacterSet
(0008,0012) DA [20090428] # 8, 1 InstanceCreationDate
(0008,0013) TM [182550.302631] # 14, 1 InstanceCreationTime
(0008,0016) UI =EncapsulatedPDFStorage # 30, 1 SOPClassUID
(0008,0018) UI [1.2.826.0.1.3680043.2.1143.776842935192792959289022034349197114] # 64, 1 SOPInstanceUID
(0008,0020) DA [20090428] # 8, 1 StudyDate
(0008,0023) DA [20090428] # 8, 1 ContentDate
(0008,002a) DT [20090428153437.000000] # 22, 1 AcquisitionDateTime
(0008,0030) TM [182550.302160] # 14, 1 StudyTime
(0008,0033) TM [153426.000000] # 14, 1 ContentTime
(0008,0050) SH (no value available) # 0, 0 AccessionNumber
(0008,0060) CS [OT] # 2, 1 Modality
(0008,0064) CS [WSD] # 4, 1 ConversionType
(0008,0070) LO [LaTeX with hyperref package] # 28, 1 Manufacturer
(0008,0090) PN (no value available) # 0, 0 ReferringPhysiciansName
(0010,0010) PN [Mathieu Malaterre and co.] # 26, 1 PatientsName
(0010,0020) LO (no value available) # 0, 0 PatientID
(0010,0030) DA (no value available) # 0, 0 PatientsBirthDate
(0010,0040) CS (no value available) # 0, 0 PatientsSex
(0018,1020) LO [pdfTeX-1.21a] # 14, 1 SoftwareVersions
(0020,000d) UI [1.2.826.0.1.3680043.2.1143.1868121832223417351654232480755123133] # 64, 1 StudyInstanceUID
(0020,000e) UI [1.2.826.0.1.3680043.2.1143.1330099150825746617507846107663964311] # 64, 1 SeriesInstanceUID
(0020,0010) SH (no value available) # 0, 0 StudyID
(0020,0011) IS [1] # 2, 1 SeriesNumber
```

```

(0020,0013) IS [1] # 2, 1 InstanceNumber
(0028,0301) CS [YES] # 4, 1 BurnedInAnnotation
(0040,a043) SQ (Sequence with explicit length #=0) # 0, 1 ConceptNameCodeSequence
(ffff,e0dd) na (SequenceDelimitationItem for re-encod.) # 0, 0 SequenceDelimitationItem
(0042,0010) ST [GDCM Reference Manual] # 22, 1 DocumentTitle
(0042,0011) OB 25\\50\\44\\46\\2d\\31\\2e\\34\\0a\\25\\e7\\f3\\cf\\d3\\0a\\33\\32\\30\\37\\37\\20\\30... # 137568
(0042,0012) LO [application/pdf] # 16, 1 MIMETimeTypeOfEncapsulatedDocument

```

```

$ stat gdc.m.pdf
  File: `gdc.m.pdf'
  Size: 13756841      Blocks: 26912      IO Block: 4096   regular file
Device: fe01h/65025d Inode: 2675750      Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1002/mmalaterre)   Gid: ( 1002/mmalaterre)
Access: 2009-04-28 16:05:00.000000000 +0200
Modify: 2009-04-28 15:34:37.000000000 +0200
Change: 2009-04-28 16:05:00.000000000 +0200

```

Explanation for the different Date/Time mappings:

- Study Date/Time, Instance Creation Date/Time are both equal to the current time gdc.mpdf tool was run,
- Acquisition Date Time is set to the Modify Time of the actual PDF file,
- Content Date/Time are set from the actual PDF header info: CreationDate.

12.7 SEE ALSO

gdc.mconv(1), gdc.mraw(1), pdfinfo(1)

12.8 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 13

Extract Data Element Value Field.

13.1 SYNOPSIS

```
gdcmmraw [options] file-in file-out
```

13.2 DESCRIPTION

The **gdcmmraw** tool is mostly used for development purpose. It is used to extract a specific binary field from a DICOM DataSet.

13.3 PARAMETERS

```
file-in    DICOM input filename
```

```
file-out    output filename
```

13.4 OPTIONS

13.4.1 PARAMETERS

```
-i --input      Input filename
-o --output      Output filename
-t --tag        Specify tag to extract value from.
```

13.4.2 OPTIONS

```
-S --split-frags  Split fragments into multiple files.
-p --pattern      Specify trailing file pattern (see split-frags).
-P --pixel-data   Pixel Data trailing 0.
```

13.4.3 general options

```
-h    --help
```



```
-rw-r--r-- 1 mathieu mathieu 81512 2008-08-08 22:10 jpeg03.ljpeg  
-rw-r--r-- 1 mathieu mathieu 81694 2008-08-08 22:10 jpeg02.ljpeg  
-rw-r--r-- 1 mathieu mathieu 81564 2008-08-08 22:10 jpeg01.ljpeg  
-rw-r--r-- 1 mathieu mathieu 79970 2008-08-08 22:10 jpeg00.ljpeg
```

13.6 Footnote about JPEG files

It is a common misunderstanding to interchange 'JPEG 8bits lossy' with simply JPEG file. The JPEG specification is much broader than simply the common lossy 8bits file (as found on internet).

You can have:

- JPEG Lossy 8bits
- JPEG Lossy 12bits
- JPEG Lossless 2-16bits

Those are what is defined in ITU-T T.81, ISO/IEC IS 10918-1.

13.7 SEE ALSO

gdcmdump(1), gdcmrw(1)

13.8 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 14

Scan a directory containing DICOM files.

14.1 SYNOPSIS

```
gdcmscanner [options] directory
```

14.2 DESCRIPTION

The **gdcmscanner** is a command line tool to quickly extract value from a set of DICOM attribute in a DICOM File-Set.

14.2.1 PARAMETERS

```
-d --dir          DICOM directory
-t --tag %d,%d    DICOM tag(s) to look for
```

14.2.2 OPTIONS

```
-p --print        Print output.
-r --recursive    Recursively descend directory.
```

14.2.3 general options

```
-h  --help
     print this help text and exit

-v  --version
     print version information and exit

-V  --verbose
     verbose mode (warning+error).

-W  --warning
     warning mode, print warning information

-E  --error
     error mode, print error information

-D  --debug
     debug mode, print debug information
```

14.3 Typical usage

14.4 Simple usage

In order to display all the value for Patient Name (0010,0010) in the directory name **gdcmData**, simply do:

```
$ gdcmscanner -t 10,10 -d gdcmData -p
```

14.5 Complex usage

Because gdcmscanner does not support progress, you have to wait until all files are traversed to see any results. This is quite cumbersome, on UNIX this can be worked around with the following trick:

```
$ find gdcmData -type d -exec gdcmscanner -t 10,10 -d {} -p ';'
```

So all directory are locally traversed (no child directory are recursively traversed), which means results comes out much faster.

14.6 SEE ALSO

gdcmdump(1), **gdcmrw(1)**

14.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 15

Tool to execute a DICOM Query/Retrieve operation

15.1 SYNOPSIS

```
gdcmscu [OPTION]...[OPERATION]...HOSTNAME...[PORT]...
```

Execute a DICOM Q/R operation to HOSTNAME, using port PORT (104 when not specified)

15.2 DESCRIPTION

The **gdcmscu** command line program is the tool to execute DICOM Query/Retrieve operation. It supports:

- C-ECHO (SCU)
- C-FIND (SCU)
- C-STORE (SCU)
- C-MOVE (SCU/SCP) C-MOVE operation are executed using two different ports (one for the SCU and one for the SCP).

15.3 PARAMETERS

15.4 OPTIONS

15.4.1 OPTIONS

```
-H --hostname    %s  Hostname.  
-p --port        %d  Port number.  
    --aetitle    %s  Set calling AE Title.  
    --call       %s  Set called AE Title.
```

15.4.2 mode options

```
--echo          C-ECHO (default when none).  
--store          C-STORE.
```

```
--find      C-FIND.
--move      C-MOVE.
```

15.4.3 C-STORE options

```
-i --input      %s  DICOM filename
-r --recursive  recursively process (sub-)directories
--store-query %s  Store constructed query in file
```

15.4.4 C-FIND/C-MOVE options

```
--patientroot  C-FIND Patient Root Model.
--studyroot    C-FIND Study Root Model.

--patient      C-FIND Query on Patient Info (cannot be used with --studyroot).
--study        C-FIND Query on Study Info.
--series       C-FIND Query on Series Info.
--image        C-FIND Query on Image Info.
--key %d,%d[%s] 0123,4567=VALUE for specifying search criteria (wildcard allowed)
                With --key, leave blank (ie, --key 10,20="" or --key 10,20) to retrieve values
```

15.4.5 C-MOVE options

```
-o --output      %s  DICOM filename / directory
--port-scp %d      Port for incoming associations
--key %d,%d[%s]    0123,4567=VALUE for specifying search criteria (wildcard not allowed)
                Note that C-MOVE supports the same queries as C-FIND, but no wildcards are allowed
```

15.4.6 general options

```
-h --help
    print this help text and exit

-v --version
    print version information and exit

-V --verbose
    verbose mode (warning+error).

-W --warning
    warning mode, print warning information

-E --error
    error mode, print error information

-D --debug
    debug mode, print debug information

-L --log-file
    specify a filename where to write logs

--queryhelp
    print query help
```

15.4.7 environment variable

```
GDCM_ROOT_UID Root UID
```


15.5 C-ECHO usage

gdcm SCU is a great tool to test if a DICOM server is up. For example to send a C-ECHO to server `dicom.example.com` using port 104, use:

```
$ gdcm SCU dicom.example.com
```

or if you prefer being explicit:

```
$ gdcm SCU --echo dicom.example.com 104
```

Using basic security your DICOM server might require that you set the appropriate called AE-TITLE

```
$ gdcm SCU --echo dicom.example.com 11112 --call SERVSCP
```

If you want to specify your own AE-TITLE (default is GDCMSCU), simply use:

```
$ gdcm SCU --echo dicom.example.com 11112 --call SERVSCP --aetitle MYSCU
```

For example you could test on the DICOM server provided by DICOMObject team:

```
$ gdcm SCU www.dicomserver.co.uk 11112
```

15.6 C-STORE usage

C-STORE is the operation that allow sending a DICOM file to a remote DICOM server. For instance to send a file called `myfile.dcm`

```
$ gdcm SCU --store dicom.example.com 104 myfile.dcm
```

or if you prefer being explicit:

```
$ gdcm SCU --store dicom.example.com 104 -i myfile.dcm
```

You can even send multiple files using the same association:

```
$ gdcm SCU --store dicom.example.com 104 myfile1.dcm myfile2.dcm myfile3.dcm ...
```

15.7 C-FIND usage

gdcm SCU also allow querying a DICOM server. This is the C-FIND operation, for example to find all DICOM Instance where PatientsName match a particular pattern, usage is simply:

```
$ gdcm SCU --find --patient dicom.example.com 11112 --patientroot --key 10,10,"A*"
```

We also support a DCMTK compatible convention:

```
$ gdcm SCU --find --patient dicom.example.com 11112 --patientroot --key 10,10="A*"
```

When an attribute is set without a value it will be part of the output result:

```
$ gdcm SCU --find --patient dicom.example.com 11112 --call MI2B2 --patientroot -k 10,10="A*" -k 10,20
```

15.8 C-MOVE usage

C-MOVE is the operation to retrieve a DICOM instance from a remote DICOM server. Most of the time, it is a subsequent operation after a C-FIND query. To retrieve a DICOM instance where PatientID is ABCD1234, simply execute:

```
$ gdcmscu --move --patient --aetitle ACME1 --call ACME_STORE dicom.example.com 5678 --patientroot -k 10,20="ABCD1234"
```

WARNING For this operation to work you need information from the DICOM server you are communicating with. Only the DICOM server you are sending a C-MOVE query will be responsible for sending back incoming associations (the actual C-STORE SCP). Therefore you need to make sure that you mapping of (AE-TITLE,PortNumber) is properly set on the DICOM server side as well as the port for incoming association (`--port-scp`).

gdcmscu does not currently support external C-STORE association (C-STORE request sent to an external SCP application).

15.9 patientroot notes

The flag `--patientroot` is just simply a wrapper around the syntax `--key 8,52=PATIENT` For instance one would write using DCMTK syntax:

```
$ findscu --patient dicom.example.com 11112 --key 8,52=PATIENT --key 10,10="F*"
```

This would become using GDCM syntax:

```
$ gdcmscu --find --patient dicom.example.com 11112 --patientroot --key 10,10="F*"
```

15.10 Debugging

This is sometime difficult to investigate why a connection to a remote DICOM server cannot be done. Some recommendations follows:

Always try to do a simple C-ECHO at first. If you cannot get the C-ECHO to work none of the other operations will work. Before trying to a C-MOVE operation, make sure you can execute the C-FIND equivalent query first.

When doing a C-MOVE operation you really need to communicate with the PACS admin as the C-MOVE operation is different from the other lower level operation such as HTTP/GET. When doing a C-MOVE, the server will communicate back using another channel (could be different port) using it's internal database to map an AE-TITLE back to the destination IP.

Indeed the C-MOVE operation by design does not always use your incoming IP address to send back the resulting dataset. Instead it uses a mapping of AE-TITLE to IP address to send back any results. So pay particular attention to the spelling of your AE-TITLE and your incoming port (which may be different from the port to connect to the server).

15.11 Port Warning

Watch out that port ranging [1-1024] are reserved for admin and not easily accessible unless granted special privileges. Therefore the default 104 DICOM port might not be accessible to all your users.

15.12 C-STORE Warnings

When constructing a C-STORE operation, `gdcm SCU` will always use the Media Storage SOP Class UID as found in the file to be sent. For encapsulated DICOM file (eg. RLE Lossless) the receiving SCP server might not support this compression and will legitimately refuse the C-STORE operation. In this case users have to manually convert to a non-compressed form this particular file:

```
$ gdcmconv --raw compressed.dcm non_compressed.dcm
```

15.13 C-MOVE Warnings

At the moment `gdcm SCU` only supports non-compressed transfer syntax. It will always request DataSet using Implicit VR Little Endian Transfer Syntax during a C-MOVE operation (both incoming and outgoing associations). This make `gdcm SCU -move` equivalent to DCMTK `movescu` syntax:

```
$ movescu -xi +xi ...
```

15.14 C-FIND IMAGE level (Composite Object Instance)

One should pay attention that `gdcm SCU -find` and `findscu` are not completely equivalent. Using `gdcm SCU -find`, all Unique Keys will be added automatically. One can therefore execute something like this:

```
$ gdcm SCU --find --patientroot --image --key 8,18=1.2.3.4.5.6 dicom.example.com 11112
```

instead of the more explicit form

```
$ gdcm SCU --find --patientroot --image --key 8,18=1.2.3.4.5.6 dicom.example.com 11112 --key 10,20 --key 20,d --key
```

This would also be equivalent to:

```
$ findscu --patient --key 8,52=IMAGE --key 8,18=1.2.3.4.5.6 dicom.example.com 11112 --key 10,20 --key 20,d --key
```

15.15 Storing the Query

It is also possible to store the query:

```
gdcm SCU --find --patient --patientroot dicom.example.com 11112 --key 10,20="*" --key 10,10 --store-query query.dcm
```

One can then check the DataSet values send for the query:

```
$ gdcmdump query.dcm
# Dicom-File-Format

# Dicom-Meta-Information-Header
# Used TransferSyntax:

# Dicom-Data-Set
# Used TransferSyntax: 1.2.840.10008.1.2
(0008,0005) ?? (CS) [ISO_IR 192] # 10,1-n Specific Character Set
(0008,0052) ?? (CS) [PATIENT ] # 8,1 Query/Retrieve Level
(0010,0010) ?? (PN) (no value) # 0,1 Patient's Name
(0010,0020) ?? (LO) [* ] # 2,1 Patient ID
```

The Specific Character Set was set to "ISO_IR 192" as the locale encoding of the system was found automatically by gdcmscu to be UTF-8.

This means that the following command line will properly setup the Query with the appropriate Charset to be executed correctly:

```
$ gdcmscu --find --patient --patientroot dicom.example.com 11112 --key 10,10="*Jérôme*
```

The query is always executed on the server side (SCP), some implementations does not support string matching with different Character Set.

15.16 DICOM Public Servers

An up to date list of DICOM Public Servers can be found at:

<http://www.dclunie.com/medical-image-faq/html/part8.html#DICOMPublicServers>

15.17 SEE ALSO

gdcmscu(1)

15.18 COPYRIGHT

Copyright Insight Software Consortium

Chapter 16

Concatenate/Extract DICOM files.

16.1 SYNOPSIS

```
gdcmtar [options] file-in file-out
```

16.2 DESCRIPTION

The **gdcmtar** is a command line tool used to tar/untar multi-frames images (including SIEMENS MOSAIC file)

16.3 PARAMETERS

file-in DICOM input filename

file-out DICOM output filename

16.4 OPTIONS

16.4.1 OPTIONS

```
--enhance      enhance (default)
-U --unenhance  unenhance
-M --mosaic     Split SIEMENS Mosaic image into multiple frames.
-p --pattern    Specify trailing file pattern.
--root-uid      Root UID.
```

16.4.2 general options

```
-h --help      print this help text and exit
-v --version    print version information and exit
-V --verbose    verbose mode (warning+error).
```

```
-W  --warning
    warning mode, print warning information

-E  --error
    error mode, print error information

-D  --debug
    debug mode, print debug information
```

16.4.3 environment variable

GDCM_ROOT_UID Root UID

16.5 Typical usage

16.5.1 SIEMENS Mosaic

```
$ gdcminfo MR-sonata-3D-as-Tile.dcm
```

```
MediaStorage is 1.2.840.10008.5.1.4.1.1.4 [MR Image Storage]
TransferSyntax is 1.2.840.10008.1.2.1 [Explicit VR Little Endian]
NumberOfDimensions: 2
Dimensions: (384,384,1)
\&...
```

```
$ gdcmtar --mosaic -i MR-sonata-3D-as-Tile.dcm -o mosaic --pattern %03d.dcm
```

Will output:

```
-rw-r--r-- 1 mathieu mathieu 72882 2009-08-10 11:14 mosaic000.dcm
-rw-r--r-- 1 mathieu mathieu 72886 2009-08-10 11:14 mosaic001.dcm
-rw-r--r-- 1 mathieu mathieu 72886 2009-08-10 11:14 mosaic002.dcm
-rw-r--r-- 1 mathieu mathieu 72886 2009-08-10 11:14 mosaic003.dcm
-rw-r--r-- 1 mathieu mathieu 72886 2009-08-10 11:14 mosaic004.dcm
-rw-r--r-- 1 mathieu mathieu 72886 2009-08-10 11:14 mosaic005.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic006.dcm
-rw-r--r-- 1 mathieu mathieu 72882 2009-08-10 11:14 mosaic007.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic008.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic009.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic010.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic011.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic012.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic013.dcm
-rw-r--r-- 1 mathieu mathieu 72882 2009-08-10 11:14 mosaic014.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic015.dcm
-rw-r--r-- 1 mathieu mathieu 72882 2009-08-10 11:14 mosaic016.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic017.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic018.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic019.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic020.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic021.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic022.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic023.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic024.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic025.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic026.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic027.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic028.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic029.dcm
-rw-r--r-- 1 mathieu mathieu 72882 2009-08-10 11:14 mosaic030.dcm
```

```
$ gdcminfo mosaic000.dcm
```

```
MediaStorage is 1.2.840.10008.5.1.4.1.1.4 [MR Image Storage]  
TransferSyntax is 1.2.840.10008.1.2.1 [Explicit VR Little Endian]  
NumberOfDimensions: 2  
Dimensions: (64,64,1)  
\&...
```

16.6 SEE ALSO

gdcmdump(1), **gdcmrw(1)**, **gdcminfo(1)**

16.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 17

Simple DICOM viewer.

17.1 SYNOPSIS

```
gdcviewer [options] file-in
```

17.2 DESCRIPTION

The **gdcviewer** is a simple tool that show how to use [vtkGDCMImageReader](#). The class that use gdc to make a layer to VTK. **gdcviewer** is basically only just a wrapper around VTK/GDCM.

This tool is meant for testing integration of GDCM in VTK. You should see it as a demo tool. It does compile with VTK ranging from 4.2 to 5.10, but only with VTK 5.2 (or above) can play with the widgets (as described below).

17.3 PARAMETERS

```
file-in    DICOM input filename
```

17.4 OPTIONS

17.4.1 OPTIONS

<code>--force-rescale</code>	force rescale (advanced users)
<code>--force-spacing</code>	force spacing (advanced users)
<code>-r --recursive</code>	Recursively descend directory

17.4.2 general options

<code>-h</code>	<code>--help</code>	print this help text and exit
<code>-v</code>	<code>--version</code>	print version information and exit
<code>-V</code>	<code>--verbose</code>	verbose mode (warning+error).

```
-W  --warning
    warning mode, print warning information

-E  --error
    error mode, print error information

-D  --debug
    debug mode, print debug information
```

17.5 Typical usage

17.6 Simple usage

For now `gdcmviewer` should be started from a command line prompt. The next argument should be the name of the DICOM file you wish to read. For instance:

```
$ gdcmviewer -V 012345.002.050.dcm
```

`gdcmviewer` will try to read your file, and then print the `vtk` information associated with this file. Basically what kind of image you are looking at.

- `ScalarType` is the DICOM Real World Value type
- `Dimensions` is the dimension of the image
- `Spacing` is the spacing of the image
- `NumberOfScalarComponents` should be 1 for grayscale & `PALETTE COLOR` and 3 for `RGB`, `YBR` data.

17.7 Wiki Link

The wiki page, with color pictures can be found at: <http://gdcm.sourceforge.net/wiki/index.php/Gdcmviewer>

17.8 SEE ALSO

`gdcmdump(1)`, `gdcm2vtk(1)`

17.9 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 18

provides a tool to convert a DICOM file into a XML info set and vice-versa.

18.1 SYNOPSIS

```
gdcxml [options] file-in[DICOM or XML] file-out[XML or DICOM]
```

18.2 DESCRIPTION

The **gdcxml** command line program converts a DICOM file (DataSet) into an XML file (according to the Native DICOM Model) or vice-versa. For those familiar with DCMTK, this provides binary capabilities (i.e. functionality of both dcm2xml and xml2dcm).

The XML info set which is from the DICOM file gdcXMLPrintet Class. This is in strict compliance with the Native DICOM Model as given in Supp 118.

18.3 PARAMETERS

```
file-in    DICOM or XML input filename ( cannot be absent)
```

```
file-out    output filename (can be absent)
```

18.4 OPTIONS

18.4.1 PARAMETERS

```
-i --input      DICOM filename
-o --output      DICOM filename
```

18.4.2 Options for DICOM to XML:

```
-B --loadBulkData  Loads bulk data into a binary file named "UUID" (by default UUID are written).
```

18.4.3 Options for XML to DICOM:

```
-B --loadBulkData  Loads bulk data from a binary file named as the "UUID" in XML file (by default UUID are writ  
-T --TransferSyntax Loads transfer syntax from file (default is LittleEndianImplicit)
```

18.4.4 general options

```
-h  --help  
    print this help text and exit  
  
-v  --version  
    print version information and exit  
  
-V  --verbose  
    verbose mode (warning+error).  
  
-W  --warning  
    warning mode, print warning information  
  
-E  --error  
    error mode, print error information  
  
-D  --debug  
    debug mode, print debug information
```

18.5 SEE ALSO

gdcmdump(1), gdcconv(1)

18.6 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 19

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 20

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 21

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:

- Gantry Tilt is not considered (always an error)
- Application programmer should only sort valid `DataSet` (eg. `MRImageStorage`, `CTImageStorage`, `PETImageStorage`)

Chapter 22

Namespace Index

22.1 Namespace List

Here is a list of all namespaces with brief descriptions:

gdc	113
gdc::network	136
gdc::SegmentHelper	142
gdc::terminal	
Class for Terminal Allow one to print in color in a shell	142

Chapter 23

Hierarchical Index

23.1 Class Hierarchy

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

gdcn::network::AbstractSyntax	156
gdcn::network::ApplicationContext	166
gdcn::ApplicationEntity	167
gdcn::network::ARTIMTimer	172
gdcn::ASN1	173
gdcn::network::AsynchronousOperationsWindowSub	174
gdcn::Attribute< Group, Element, TVR, TVM >	175
gdcn::Attribute< Group, Element, TVR, VM::VM1 >	180
gdcn::Attribute< Group, Element, TVR, VM::VM1_n >	187
gdcn::Attribute< Group, Element, TVR, VM::VM1_3 >	184
gdcn::Attribute< Group, Element, TVR, VM::VM1_8 >	185
gdcn::Attribute< Group, Element, TVR, VM::VM2_n >	191
gdcn::Attribute< Group, Element, TVR, VM::VM2_2n >	190
gdcn::Attribute< Group, Element, TVR, VM::VM3_n >	194
gdcn::Attribute< Group, Element, TVR, VM::VM3_3n >	193
gdcn::Base64	198
gdcn::network::BaseCompositeMessage	199
gdcn::network::CEchoRQ	240
gdcn::network::CEchoRSP	241
gdcn::network::CFindCancelRQ	243
gdcn::network::CFindRQ	244
gdcn::network::CFindRSP	245
gdcn::network::CMoveCancelRq	246
gdcn::network::CMoveRQ	248
gdcn::network::CMoveRSP	249
gdcn::network::CStoreRQ	284
gdcn::network::CStoreRSP	285
gdcn::network::BaseNormalizedMessage	201
gdcn::network::NActionRQ	549
gdcn::network::NActionRSP	551
gdcn::network::NCreateRQ	552
gdcn::network::NCreateRSP	553
gdcn::network::NDeleteRQ	555

gdcmm::network::NDeleteRSP	556
gdcmm::network::NEventReportRQ	559
gdcmm::network::NEventReportRSP	561
gdcmm::network::NGetRQ	562
gdcmm::network::NGetRSP	563
gdcmm::network::NSetRQ	568
gdcmm::network::NSetRSP	569
gdcmm::network::BasePDU	203
gdcmm::network::AAabortPDU	145
gdcmm::network::AAAssociateACPDU	147
gdcmm::network::AAAssociateRJPDU	150
gdcmm::network::AAAssociateRQPDU	151
gdcmm::network::AReleaseRPPDU	169
gdcmm::network::AReleaseRQPDU	170
gdcmm::network::PDataTFPDU	593
std::basic_string< Char >	
std::string	
gdcmm::String< TDelimiter, TMaxLength, TPadChar >	754
gdcmm::SegmentHelper::BasicCodedEntry	212
gdcmm::BitmapToBitmapFilter	224
gdcmm::PixmapToPixmapFilter	620
gdcmm::ImageToImageFilter	461
gdcmm::ImageApplyLookupTable	428
gdcmm::ImageChangePhotometricInterpretation	431
gdcmm::ImageChangePlanarConfiguration	434
gdcmm::ImageChangeTransferSyntax	437
gdcmm::ImageFragmentSplitter	448
gdcmm::ByteBuffer	229
gdcmm::ByteSwap< T >	230
gdcmm::ByteSwapFilter	231
gdcmm::network::CFind	242
gdcmm::Coder	251
gdcmm::Codec	250
gdcmm::AudioCodec	196
gdcmm::ImageCodec	441
gdcmm::DeltaEncodingCodec	316
gdcmm::JPEG2000Codec	488
gdcmm::JPEGCodec	494
gdcmm::JPEG12Codec	484
gdcmm::JPEG16Codec	486
gdcmm::JPEG8Codec	491
gdcmm::JPEGLSCCodec	498
gdcmm::KAKADUCCodec	503
gdcmm::PGXCodec	604
gdcmm::PNMCodec	625
gdcmm::PVRGCodec	649
gdcmm::RAWCodec	663
gdcmm::RLECodec	677
gdcmm::PDFCodec	599
gdcmm::CodeString	253
gdcmm::network::CompositeMessageFactory	259
gdcmm::CompositeNetworkFunctions	260
gdcmm::ConstCharWrapper	265

gdcmm::CryptoFactory	267
gdcmm::CAPICryptoFactory	236
gdcmm::OpenSSLCryptoFactory	573
gdcmm::OpenSSLP7CryptoFactory	577
gdcmm::CryptographicMessageSyntax	269
gdcmm::CAPICryptographicMessageSyntax	237
gdcmm::OpenSSLCryptographicMessageSyntax	574
gdcmm::OpenSSLP7CryptographicMessageSyntax	578
gdcmm::CSAElement	271
gdcmm::CSAHeader	275
gdcmm::CSAHeaderDict	279
gdcmm::CSAHeaderDictEntry	281
gdcmm::DataElement	289
gdcmm::CP246ExplicitDataElement	265
gdcmm::ExplicitDataElement	370
gdcmm::ExplicitImplicitDataElement	371
gdcmm::Fragment	413
gdcmm::BasicOffsetTable	214
gdcmm::ImplicitDataElement	468
gdcmm::Item	479
gdcmm::UNExplicitDataElement	877
gdcmm::UNExplicitImplicitDataElement	879
gdcmm::VR16ExplicitDataElement	903
gdcmm::DataSet	301
gdcmm::CommandDataSet	257
gdcmm::FileMetaInformation	389
gdcmm::DataSetHelper	311
gdcmm::Decoder	311
gdcmm::Codec	250
gdcmm::DefinedTerms	313
gdcmm::Defs	313
gdcmm::DICOMDIR	317
gdcmm::DICOMDIRGenerator	318
gdcmm::Dict	320
gdcmm::DictConverter	323
gdcmm::DictEntry	325
gdcmm::Dicts	329
gdcmm::network::DIMSE	332
gdcmm::DirectionCosines	333
gdcmm::Directory	335
gdcmm::DirectoryHelper	338
gdcmm::DummyValueGenerator	339
gdcmm::Element< TVR, TVM >	342
gdcmm::Element< TVR, VM::VM1_n >	346
gdcmm::Element< TVR, VM::VM1_2 >	345
gdcmm::Element< TVR, VM::VM2_n >	350
gdcmm::Element< TVR, VM::VM2_2n >	348
gdcmm::Element< TVR, VM::VM3_n >	353
gdcmm::Element< TVR, VM::VM3_3n >	351
gdcmm::Element< VR::AS, VM::VM5 >	354
gdcmm::Element< VR::OB, VM::VM1_n >	342
gdcmm::Element< VR::OB, VM::VM1 >	355

gdcm::Element< VR::OW, VM::VM1_n >	342
gdcm::Element< VR::OW, VM::VM1 >	356
gdcm::ElementDisableCombinations< TVR, TVM >	358
gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n >	359
gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n >	359
gdcm::EncapsulatedDocument	359
gdcm::EncodingImplementation< T >	360
gdcm::EncodingImplementation< VR::VRASCII >	360
gdcm::EncodingImplementation< VR::VRBINARY >	361
gdcm::EnumeratedValues	363
gdcm::Event	364
gdcm::AnyEvent	164
gdcm::AbortEvent	155
gdcm::AnonymizeEvent	157
gdcm::DataEvent	299
gdcm::DataSetEvent	308
gdcm::EndEvent	362
gdcm::ExitEvent	368
gdcm::FileNameEvent	397
gdcm::InitializeEvent	470
gdcm::IterationEvent	482
gdcm::ModifiedEvent	537
gdcm::ProgressEvent	646
gdcm::StartEvent	738
gdcm::UserEvent	883
gdcm::NoEvent	565
std::exception	
gdcm::CSAHeaderDictException	283
gdcm::DataElementException	298
gdcm::Exception	367
gdcm::ParseException	588
gdcm::Fiducials	373
gdcm::FileDerivation	385
gdcm::FileExplicitFilter	387
gdcm::Filename	395
gdcm::FilenameGenerator	399
gdcm::FileSet	402
gdcm::Global	415
gdcm::GroupDict	418
gdcm::IconImageFilter	419
gdcm::IconImageGenerator	422
gdcm::ignore_char	424
gdcm::ImageConverter	447
gdcm::ImageHelper	451
gdcm::network::ImplementationClassUIDSub	466
gdcm::network::ImplementationUIDSub	466
gdcm::network::ImplementationVersionNameSub	467
gdcm::IOD	471
gdcm::IODEntry	472
gdcm::IODs	474
gdcm::JSON	502
gdcm::Scanner::ltstr	512
gdcm::StrictScanner::ltstr	513

gdcmmacro::Macro	513
gdcmmacro::Macros	515
gdcmmacro::network::MaximumLengthSub	516
gdcmmacro::MD5	517
gdcmmacro::MediaStorage	518
gdcmmacro::Module	539
gdcmmacro::ModuleEntry	540
gdcmmacro::NestedModuleEntries	557
gdcmmacro::Modules	543
gdcmmacro::network::NormalizedMessageFactory	565
gdcmmacro::NormalizedNetworkFunctions	566
gdcmmacro::Object	571
gdcmmacro::BaseQuery	205
gdcmmacro::BaseRootQuery	208
gdcmmacro::FindPatientRootQuery	408
gdcmmacro::FindStudyRootQuery	411
gdcmmacro::MovePatientRootQuery	545
gdcmmacro::MoveStudyRootQuery	547
gdcmmacro::WLMFindQuery	964
gdcmmacro::ModalityPerformedProcedureStepCreateQuery	533
gdcmmacro::ModalityPerformedProcedureStepSetQuery	535
gdcmmacro::Bitmap	216
gdcmmacro::Pixmap	614
gdcmmacro::Image	425
gdcmmacro::Curve	287
gdcmmacro::File	374
gdcmmacro::FileWithName	407
gdcmmacro::LookupTable	508
gdcmmacro::SegmentedPaletteColorLookupTable	693
gdcmmacro::MeshPrimitive	529
gdcmmacro::Overlay	583
gdcmmacro::Segment	689
gdcmmacro::Subject	760
gdcmmacro::Anonymizer	160
gdcmmacro::Command	255
gdcmmacro::MemberCommand< T >	525
gdcmmacro::SimpleMemberCommand< T >	721
gdcmmacro::FileAnonymizer	377
gdcmmacro::FileChangeTransferSyntax	380
gdcmmacro::FileDecompressLookupTable	383
gdcmmacro::FileStreamer	403
gdcmmacro::network::ULConnectionManager	870
gdcmmacro::Scanner	683
gdcmmacro::ServiceClassUser	715
gdcmmacro::StrictScanner	748
gdcmmacro::Surface	762
gdcmmacro::Value	888
gdcmmacro::ByteValue	231
gdcmmacro::SequenceOfFragments	700
gdcmmacro::SequenceOfItems	705
gdcmmacro::Orientation	580
gdcmmacro::Parser	590
gdcmmacro::Patient	592

gdcmm::PDBelement	595
gdcmm::PDBHeader	597
gdcmm::network::PDUFactory	601
gdcmm::PersonName	603
gdcmm::PhotometricInterpretation	606
gdcmm::PixelFormat	608
gdcmm::Preamble	628
gdcmm::PresentationContext	629
gdcmm::network::PresentationContextAC	632
gdcmm::PresentationContextGenerator	633
gdcmm::network::PresentationContextRQ	635
gdcmm::network::PresentationDataValue	637
gdcmm::Printer	639
gdcmm::DictPrinter	327
gdcmm::Dumper	340
gdcmm::PrivateDict	642
gdcmm::PythonFilter	651
gdcmm::QueryBase	652
gdcmm::QueryImage	655
gdcmm::QueryPatient	657
gdcmm::QuerySeries	659
gdcmm::QueryStudy	661
gdcmm::QueryFactory	654
gdcmm::Reader	666
gdcmm::PixmapReader	617
gdcmm::ImageReader	455
gdcmm::ImageRegionReader	458
gdcmm::SegmentReader	695
gdcmm::SurfaceReader	771
gdcmm::RealWorldValueMappingContent	671
gdcmm::Region	672
gdcmm::BoxRegion	226
gdcmm::Rescaler	674
gdcmm::network::RoleSelectionSub	681
gdcmm::SerieHelper::Rule	682
gdcmm::SerieHelper	711
gdcmm::Series	713
gdcmm::network::ServiceClassApplicationInformation	714
gdcmm::SHA1	720
gdcmm::SimpleSubjectWatcher	725
gdcmm::SmartPointer< ObjectType >	726
gdcmm::SmartPointer< gdcmm::Bitmap >	726
gdcmm::SmartPointer< gdcmm::File >	726
gdcmm::SmartPointer< gdcmm::Image >	726
gdcmm::SmartPointer< gdcmm::MemberCommand >	726
gdcmm::SmartPointer< gdcmm::MeshPrimitive >	726
gdcmm::SmartPointer< gdcmm::Pixmap >	726
gdcmm::SmartPointer< gdcmm::SimpleMemberCommand >	726
gdcmm::SmartPointer< gdcmm::Subject >	726
gdcmm::SmartPointer< LookupTable >	726
gdcmm::SmartPointer< Segment >	726
gdcmm::SmartPointer< Surface >	726
gdcmm::SmartPointer< Value >	726

gdcm::network::SOPClassExtendedNegociationSub	729
gdcm::SOPClassUIDToIOD	730
gdcm::Sorter	731
gdcm::IPPSorter	476
gdcm::Spacing	735
gdcm::Spectroscopy	737
gdcm::SplitMosaicFilter	737
gdcm::static_assert_test< x >	740
gdcm::STATIC_ASSERTION_FAILURE< x >	740
gdcm::STATIC_ASSERTION_FAILURE< true >	740
gdcm::StreamImageReader	740
gdcm::StreamImageWriter	743
String<'\', 64 >	
gdcm::LO	505
gdcm::StringFilter	757
gdcm::Study	759
gdcm::SurfaceHelper	769
gdcm::SwapCode	775
gdcm::SwapperDoOp	777
gdcm::SwapperNoOp	777
gdcm::System	778
gdcm::Table	782
gdcm::TableEntry	783
gdcm::TableReader	784
gdcm::XMLDictReader	972
gdcm::XMLPrivateDictReader	976
gdcm::network::TableRow	786
gdcm::Tag	787
gdcm::PrivateTag	644
gdcm::TagPath	794
gdcm::Testing	795
gdcm::Trace	799
gdcm::TransferSyntax	802
gdcm::network::TransferSyntaxSub	806
gdcm::network::Transition	807
gdcm::Type	809
gdcm::UI	811
gdcm::UIDGenerator	811
gdcm::UIDs	813
gdcm::network::ULAction	832
gdcm::network::ULActionAA1	835
gdcm::network::ULActionAA2	836
gdcm::network::ULActionAA3	837
gdcm::network::ULActionAA4	838
gdcm::network::ULActionAA5	839
gdcm::network::ULActionAA6	840
gdcm::network::ULActionAA7	841
gdcm::network::ULActionAA8	842
gdcm::network::ULActionAE1	843
gdcm::network::ULActionAE2	844
gdcm::network::ULActionAE3	845
gdcm::network::ULActionAE4	846
gdcm::network::ULActionAE5	847

gdcmm::network::ULActionAE6	848
gdcmm::network::ULActionAE7	849
gdcmm::network::ULActionAE8	850
gdcmm::network::ULActionAR1	851
gdcmm::network::ULActionAR10	852
gdcmm::network::ULActionAR2	853
gdcmm::network::ULActionAR3	854
gdcmm::network::ULActionAR4	855
gdcmm::network::ULActionAR5	856
gdcmm::network::ULActionAR6	857
gdcmm::network::ULActionAR7	858
gdcmm::network::ULActionAR8	859
gdcmm::network::ULActionAR9	860
gdcmm::network::ULActionDT1	861
gdcmm::network::ULActionDT2	862
gdcmm::network::ULConnection	864
gdcmm::network::ULConnectionCallback	867
gdcmm::network::ULBasicCallback	863
gdcmm::network::ULWritingCallback	875
gdcmm::network::ULConnectionInfo	868
gdcmm::network::ULEvent	874
gdcmm::network::ULTransitionTable	875
gdcmm::Unpacker12Bits	881
gdcmm::Usage	882
gdcmm::network::UserInformation	885
gdcmm::UUIDGenerator	886
gdcmm::Validate	886
gdcmm::ValueIO< TDE, TSwap, TType >	890
gdcmm::Version	891
gdcmm::VL	892
gdcmm::VM	894
gdcmm::VMToLength< T >	898
gdcmm::VR	898
gdcmm::VRToEncoding< T >	905
gdcmm::VRToType< T >	905
gdcmm::VRToType< TVR >	905
gdcmm::VRVLSIZE< T >	906
gdcmm::VRVLSIZE< 0 >	906
gdcmm::VRVLSIZE< 1 >	906
vtkImageAlgorithm	
vtkImagePlanarComponentsToComponents	952
vtkImageMapToColors	
vtkImageMapToWindowLevelColors2	950
vtkImageWriter	
vtkGDCMImageWriter	919
vtkLookupTable	
vtkLookupTable16	957
vtkMedicalImageProperties	
vtkGDCMMedicalImageProperties	923
vtkMedicalImageReader2	
vtkGDCMImageReader	907
vtkGDCMThreadedImageReader	934
vtkGDCMImageReader2	913
vtkObject	

vtkGDCMTesting	931
vtkImageColorViewer	940
vtkRTStructSetProperties	959
vtkPolyDataAlgorithm	
vtkGDCMPolyDataReader	925
vtkPolyDataWriter	
vtkGDCMPolyDataWriter	928
vtkThreadedImageAlgorithm	
vtkGDCMThreadedImageReader2	936
vtkImageMapToColors16	947
vtkImageRGBToYBR	954
vtkImageYBRToRGB	956
gdcm::Waveform	964
gdcm::Writer	967
gdcm::PixmapWriter	622
gdcm::ImageWriter	463
gdcm::SegmentWriter	698
gdcm::SurfaceWriter	774
gdcm::XMLPrinter	973

Chapter 24

Class Index

24.1 Class List

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

gdcmm::network::AAabortPDU	
AAabortPDU Table 9-26 A-ABORT PDU FIELDS	145
gdcmm::network::AAssociateACPDU	
AAssociateACPDU Table 9-17 ASSOCIATE-AC PDU fields	147
gdcmm::network::AAssociateRJPDU	
AAssociateRJPDU Table 9-21 ASSOCIATE-RJ PDU FIELDS	150
gdcmm::network::AAssociateRQPDU	
AAssociateRQPDU Table 9-11 ASSOCIATE-RQ PDU fields	151
gdcmm::AbortEvent	155
gdcmm::network::AbstractSyntax	
AbstractSyntax Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS	156
gdcmm::AnonymizeEvent	
AnonymizeEvent Special type of event triggered during the Anonymization process	157
gdcmm::Anonymizer	
Anonymizer This class is a multi purpose anonymizer. It can work in 2 mode:	160
gdcmm::AnyEvent	164
gdcmm::network::ApplicationContext	
ApplicationContext Table 9-12 APPLICATION CONTEXT ITEM FIELDS	166
gdcmm::ApplicationEntity	
ApplicationEntity	167
gdcmm::network::AReleaseRPPDU	
AReleaseRPPDU Table 9-25 A-RELEASE-RP PDU fields	169
gdcmm::network::AReleaseRQPDU	
AReleaseRQPDU Table 9-24 A-RELEASE-RQ PDU FIELDS	170
gdcmm::network::ARTIMTimer	
ARTIMTimer This file contains the code for the ARTIM timer	172
gdcmm::ASN1	
Class for ASN1	173
gdcmm::network::AsynchronousOperationsWindowSub	
AsynchronousOperationsWindowSub PS 3.7 Table D.3-7 ASYNCHRONOUS OPERATIONS WIN↔	
DOW SUB-ITEM FIELDS (A-ASSOCIATE-RQ)	174
gdcmm::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	175

gdcm::Attribute< Group, Element, TVR, VM::VM1 >	180
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >	184
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >	185
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >	187
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >	190
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >	191
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >	193
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >	194
gdcm::AudioCodec	
AudioCodec	196
gdcm::Base64	
Class for Base64	198
gdcm::network::BaseCompositeMessage	
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	199
gdcm::network::BaseNormalizedMessage	
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	201
gdcm::network::BasePDU	
BasePDU base class for PDUs	203
gdcm::BaseQuery	
BaseQuery contains: a baseclass which will produce a dataset for all dimse messages	205
gdcm::BaseRootQuery	
BaseRootQuery contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root	208
gdcm::SegmentHelper::BasicCodedEntry	
This structure defines a basic coded entry with all of its attributes	212
gdcm::BasicOffsetTable	
Class to represent a BasicOffsetTable	214
gdcm::Bitmap	
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)	216
gdcm::BitmapToBitmapFilter	
BitmapToBitmapFilter class Super class for all filter taking an image and producing an output image	224
gdcm::BoxRegion	
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)	226
gdcm::ByteBuffer	
ByteBuffer	229
gdcm::ByteSwap< T >	
ByteSwap	230
gdcm::ByteSwapFilter	
ByteSwapFilter In place byte-swapping of a dataset FIXME: FL status ??	231
gdcm::ByteValue	
Class to represent binary value (array of bytes)	231
gdcm::CAPICryptoFactory	236
gdcm::CAPICryptographicMessageSyntax	237
gdcm::network::CEchoRQ	
CEchoRQ this file defines the messages for the cecho action	240
gdcm::network::CEchoRSP	
CEchoRSP this file defines the messages for the cecho action	241
gdcm::network::CFind	242

gdcmm::network::CFindCancelRQ	
CFindCancelRQ this file defines the messages for the cfind action	243
gdcmm::network::CFindRQ	
CFindRQ this file defines the messages for the cfind action	244
gdcmm::network::CFindRSP	
CFindRSP this file defines the messages for the cfind action	245
gdcmm::network::CMoveCancelRq	246
gdcmm::network::CMoveRQ	
CMoveRQ this file defines the messages for the cmove action	248
gdcmm::network::CMoveRSP	
CMoveRSP this file defines the messages for the cmove action	249
gdcmm::Codec	
Codec class	250
gdcmm::Coder	
Coder	251
gdcmm::CodeString	
CodeString This is an implementation of DICOM VR: CS The ctor will properly Trim so that operator== is correct	253
gdcmm::Command	
Command superclass for callback/observer methods	255
gdcmm::CommandDataSet	
Class to represent a Command DataSet	257
gdcmm::network::CompositeMessageFactory	
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)	259
gdcmm::CompositeNetworkFunctions	
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:	260
gdcmm::ConstCharWrapper	
Do not use me	265
gdcmm::CP246ExplicitDataElement	
Class to read/write a DataElement as CP246Explicit Data Element	265
gdcmm::CryptoFactory	
Class to do handle the crypto factory	267
gdcmm::CryptographicMessageSyntax	269
gdcmm::CSAElement	
Class to represent a CSA Element	271
gdcmm::CSAHeader	
Class for CSAHeader	275
gdcmm::CSAHeaderDict	
Class to represent a map of CSAHeaderDictEntry	279
gdcmm::CSAHeaderDictEntry	
Class to represent an Entry in the Dict Does not really exist within the DICOM definition, just a way to minimize storage and have a mapping from gdcmm::Tag to the needed information	281

gdcmm::CSAHeaderDictException	283
gdcmm::network::CStoreRQ	
CStoreRQ this file defines the messages for the cecho action	284
gdcmm::network::CStoreRSP	
CStoreRSP this file defines the messages for the cecho action	285
gdcmm::Curve	
Curve class to handle element 50xx,3000 Curve Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004	287
gdcmm::DataElement	
Class to represent a Data Element either Implicit or Explicit	289
gdcmm::DataElementException	298
gdcmm::DataEvent	
DataEvent	299
gdcmm::DataSet	
Class to represent a Data Set (which contains Data Elements) A Data Set represents an instance of a real world Information Object	301
gdcmm::DataSetEvent	
DataSetEvent Special type of event triggered during the DataSet store/move process	308
gdcmm::DataSetHelper	
DataSetHelper (internal class, not intended for user level)	311
gdcmm::Decoder	
Decoder	311
gdcmm::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	313
gdcmm::Defs	
FIXME I do not like the name 'Defs'	313
gdcmm::DeltaEncodingCodec	
DeltaEncodingCodec compression used by some private vendor	316
gdcmm::DICOMDIR	
DICOMDIR class	317
gdcmm::DICOMDIRGenerator	
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	318
gdcmm::Dict	
Class to represent a map of DictEntry	320
gdcmm::DictConverter	
Class to convert a .dic file into something else:	323
gdcmm::DictEntry	
Class to represent an Entry in the Dict Does not really exist within the DICOM definition, just a way to minimize storage and have a mapping from gdcmm::Tag to the needed information	325
gdcmm::DictPrinter	
DictPrinter class	327
gdcmm::Dicts	
Class to manipulate the sum of knowledge (all the dict user load)	329

gdcm::network::DIMSE	
DIMSE PS 3.7 - 2009 Annex E Command Dictionary (Normative) E.1 REGISTRY OF DICOM CO↔	
MMAND ELEMENTS Table E.1-1 COMMAND FIELDS (PART 1)	332
gdcm::DirectionCosines	
Class to handle DirectionCosines	333
gdcm::Directory	
Class for manipulation directories	335
gdcm::DirectoryHelper	
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	338
gdcm::DummyValueGenerator	
Class for generating dummy value	339
gdcm::Dumper	
Codec class	340
gdcm::Element< TVR, TVM >	
Element class	342
gdcm::Element< TVR, VM::VM1_2 >	345
gdcm::Element< TVR, VM::VM1_n >	346
gdcm::Element< TVR, VM::VM2_2n >	348
gdcm::Element< TVR, VM::VM2_n >	350
gdcm::Element< TVR, VM::VM3_3n >	351
gdcm::Element< TVR, VM::VM3_n >	353
gdcm::Element< VR::AS, VM::VM5 >	354
gdcm::Element< VR::OB, VM::VM1 >	355
gdcm::Element< VR::OW, VM::VM1 >	356
gdcm::ElementDisableCombinations< TVR, TVM >	
A class which is used to produce compile errors for an invalid combination of template parameters	358
gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n >	359
gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n >	359
gdcm::EncapsulatedDocument	
EncapsulatedDocument	359
gdcm::EncodingImplementation< T >	
EncodingImplementation	360
gdcm::EncodingImplementation< VR::VRASCII >	360
gdcm::EncodingImplementation< VR::VRBINARY >	361
gdcm::EndEvent	362
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:	363
gdcm::Event	
Superclass for callback/observer methods	364
gdcm::Exception	
Exception	367
gdcm::ExitEvent	368
gdcm::ExplicitDataElement	
Class to read/write a DataElement as Explicit Data Element	370
gdcm::ExplicitImplicitDataElement	
Class to read/write a DataElement as ExplicitImplicit Data Element	371

gdcm::Fiducials	
Fiducials	373
gdcm::File	
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	374
gdcm::FileAnonymizer	
FileAnonymizer	377
gdcm::FileChangeTransferSyntax	
FileChangeTransferSyntax	380
gdcm::FileDecompressLookupTable	
FileDecompressLookupTable class It decompress the segmented LUT into linearized one (only P←ALLETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image	383
gdcm::FileDerivation	
FileDerivation class See PS 3.16 - 2008 For the list of Code Value that can be used for in Derivation Code Sequence	385
gdcm::FileExplicitFilter	
FileExplicitFilter class After changing a file from Implicit to Explicit representation (see Image←ChangeTransferSyntax) 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	387
gdcm::FileMetaInformation	
Class to represent a File Meta Information	389
gdcm::Filename	
Class to manipulate file name's	395
gdcm::FileNameEvent	
FileNameEvent Special type of event triggered during processing of FileSet	397
gdcm::FilenameGenerator	
FilenameGenerator	399
gdcm::FileSet	
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	402
gdcm::FileStreamer	
FileStreamer This class let a user create a massive DICOM DataSet from a template DICOM file, by appending chunks of data	403
gdcm::FileWithName	
FileWithName	407
gdcm::FindPatientRootQuery	
PatientRootQuery contains: the class which will produce a dataset for c-find with patient root	408
gdcm::FindStudyRootQuery	
FindStudyRootQuery contains: the class which will produce a dataset for C-FIND with study root	411
gdcm::Fragment	
Class to represent a Fragment	413
gdcm::Global	
Global	415
gdcm::GroupDict	
Class to represent the mapping from group number to its abbreviation and name	418
gdcm::IconImageFilter	
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	419

gdcm::IconImageGenerator	
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:
gdcm::ignore_char 422
gdcm::Image	
Image	This is the container for an Image in the general sense. From this container you should be able to request information like:
gdcm::ImageApplyLookupTable	
ImageApplyLookupTable	class It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a PhotometricInterpretation =RGB image
gdcm::ImageChangePhotometricInterpretation	
ImageChangePhotometricInterpretation	class Class to change the Photometric Interpretation of an input DICOM
gdcm::ImageChangePlanarConfiguration	
ImageChangePlanarConfiguration	class Class to change the Planar configuration of an input DICOM By default it will change into the more usual representation: PlanarConfiguration = 0
gdcm::ImageChangeTransferSyntax	
ImageChangeTransferSyntax	class Class to change the transfer syntax of an input DICOM
gdcm::ImageCodec	
ImageCodec 441
gdcm::ImageConverter	
Image	Converter 447
gdcm::ImageFragmentSplitter	
ImageFragmentSplitter	class For single frame image, DICOM standard allow splitting the frame into multiple fragments
gdcm::ImageHelper	
ImageHelper	(internal class, not intended for user level) 451
gdcm::ImageReader	
ImageReader 455
gdcm::ImageRegionReader	
ImageRegionReader 458
gdcm::ImageToImageFilter	
ImageToImageFilter	class Super class for all filter taking an image and producing an output image . 461
gdcm::ImageWriter	
ImageWriter 463
gdcm::network::ImplementationClassUIDSub	
ImplementationClassUIDSub	PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE-RQ)
gdcm::network::ImplementationUIDSub	
ImplementationUIDSub	Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-RQ)
gdcm::network::ImplementationVersionNameSub	
ImplementationVersionNameSub	Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)
gdcm::ImplicitDataElement	
Class to represent an <i>Implicit VR</i> Data Element 468
gdcm::InitializeEvent 470
gdcm::IOD	
Class for representing a IOD 471
gdcm::IODEntry	
Class for representing a IODEntry 472
gdcm::IODs	
Class for representing a IODs 474

gdcm::IPPSorter	
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	476
gdcm::Item	
Class to represent an Item A component of the value of a Data Element that is of Value Representation Sequence of Items. An Item contains a Data Set . See PS 3.5 7.5.1 Item Encoding Rules Each Item of a Data Element of VR SQ shall be encoded as a DICOM Standard Data Element with a specific Data Element Tag of Value (FFFE,E000). The Item Tag is followed by a 4 byte Item Length field encoded in one of the following two ways Explicit/ Implicit	479
gdcm::IterationEvent	482
gdcm::JPEG12Codec	
Class to do JPEG 12bits (lossy & lossless)	484
gdcm::JPEG16Codec	
Class to do JPEG 16bits (lossless)	486
gdcm::JPEG2000Codec	
Class to do JPEG 2000	488
gdcm::JPEG8Codec	
Class to do JPEG 8bits (lossy & lossless)	491
gdcm::JPEGCodec	
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	494
gdcm::JPEGLSCodec	
JPEG-LS	498
gdcm::JSON	502
gdcm::KAKADUCodec	
KAKADUCodec	503
gdcm::LO	
LO	505
gdcm::LookupTable	
LookupTable class	508
gdcm::Scanner::ltstr	512
gdcm::StrictScanner::ltstr	513
gdcm::Macro	
Class for representing a Macro	513
gdcm::Macros	
Class for representing a Modules	515
gdcm::network::MaximumLengthSub	
MaximumLengthSub Annex D Table D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIA↔TE-RQ)	516
gdcm::MD5	
Class for MD5	517
gdcm::MediaStorage	
MediaStorage	518
gdcm::MemberCommand< T >	
Command subclass that calls a pointer to a member function	525
gdcm::MeshPrimitive	
This class defines surface mesh primitives. It is designed from surface mesh primitives macro	529
gdcm::ModalityPerformedProcedureStepCreateQuery	
ModalityPerformedProcedureStepCreateQuery contains: the class which will produce a dataset for n-create for Modality Performed Procedure Step sop class	533

gdcm::ModalityPerformedProcedureStepSetQuery	
ModalityPerformedProcedureStepSetQuery contains: the class which will produce a dataset for n-set for Modality Performed Procedure Step sop class	535
gdcm::ModifiedEvent	537
gdcm::Module	
Class for representing a Module	539
gdcm::ModuleEntry	
Class for representing a ModuleEntry	540
gdcm::Modules	
Class for representing a Modules	543
gdcm::MovePatientRootQuery	
MovePatientRootQuery contains: the class which will produce a dataset for c-move with patient root	545
gdcm::MoveStudyRootQuery	
MoveStudyRootQuery contains: the class which will produce a dataset for C-MOVE with study root	547
gdcm::network::NActionRQ	
NActionRQ this file defines the messages for the NAction action	549
gdcm::network::NActionRSP	
NActionRSP this file defines the messages for the NAction action	551
gdcm::network::NCreateRQ	
NCreateRQ this file defines the messages for the ncreate action	552
gdcm::network::NCreateRSP	
NCreateRSP this file defines the messages for the ncreate action	553
gdcm::network::NDeleteRQ	
NDeleteRQ this file defines the messages for the ndelete action	555
gdcm::network::NDeleteRSP	
NDeleteRSP this file defines the messages for the ndelete action	556
gdcm::NestedModuleEntries	
Class for representing a NestedModuleEntries	557
gdcm::network::NEventReportRQ	
NEventReportRQ this file defines the messages for the neventreport action	559
gdcm::network::NEventReportRSP	
NEventReportRSP this file defines the messages for the neventreport action	561
gdcm::network::NGetRQ	
NGetRQ this file defines the messages for the nget action	562
gdcm::network::NGetRSP	
NGetRSP this file defines the messages for the nget action	563
gdcm::NoEvent	565
gdcm::network::NormalizedMessageFactory	565
gdcm::NormalizedNetworkFunctions	
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:	566
gdcm::network::NSetRQ	
NSetRQ this file defines the messages for the nset action	568
gdcm::network::NSetRSP	
NSetRSP this file defines the messages for the nset action	569
gdcm::Object	
Object	571

gdcm::OpenSSLCryptoFactory	573
gdcm::OpenSSLCryptographicMessageSyntax	574
gdcm::OpenSSLP7CryptoFactory	577
gdcm::OpenSSLP7CryptographicMessageSyntax	
Class for CryptographicMessageSyntax encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities	578
gdcm::Orientation	
Class to handle Orientation	580
gdcm::Overlay	
Overlay class	583
gdcm::ParseException	
ParseException Standard exception handling object	588
gdcm::Parser	
Parser ala XML_Parser from expat (SAX)	590
gdcm::Patient	
See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54	592
gdcm::network::PDataTFPDU	
PDataTFPDU Table 9-22 P-DATA-TF PDU FIELDS	593
gdcm::PDBElement	
Class to represent a PDB Element	595
gdcm::PDBHeader	
Class for PDBHeader	597
gdcm::PDFCodec	
PDFCodec class	599
gdcm::network::PDUFactory	
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	601
gdcm::PersonName	
PersonName class	603
gdcm::PGXCodec	
Class to do PGX See PGX as used in JPEG 2000 implementation and reference images	604
gdcm::PhotometricInterpretation	
Class to represent an PhotometricInterpretation	606
gdcm::PixelFormat	
PixelFormat	608
gdcm::Pixmap	
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)	614
gdcm::PixmapReader	
PixmapReader	617
gdcm::PixmapToPixmapFilter	
PixmapToPixmapFilter class Super class for all filter taking an image and producing an output image	620
gdcm::PixmapWriter	
PixmapWriter This class will takes two inputs:	622
gdcm::PNMCodec	
Class to do PNM PNM is the Portable anymap file format. The main web page can be found at: http://netpbm.sourceforge.net/	625
gdcm::Preamble	
DICOM Preamble (Part 10)	628
gdcm::PresentationContext	
PresentationContext	629
gdcm::network::PresentationContextAC	
PresentationContextAC Table 9-18 PRESENTATION CONTEXT ITEM FIELDS	632

gdcm::PresentationContextGenerator	
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
	633
gdcm::network::PresentationContextRQ	
PresentationContextRQ	Table 9-13 PRESENTATION CONTEXT ITEM FIELDS
	635
gdcm::network::PresentationDataValue	
PresentationDataValue	Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS
	637
gdcm::Printer	
Printer	class
	639
gdcm::PrivateDict	
Private	Dict
	642
gdcm::PrivateTag	
Class to represent a Private DICOM Data	Element (Attribute) Tag (Group, Element , Owner)
	644
gdcm::ProgressEvent	
ProgressEvent	Special type of event triggered during
	646
gdcm::PVRGCodec	
PVRGCodec
	649
gdcm::PythonFilter	
PythonFilter	PythonFilter is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a DataElement into a string, typically this is a nice feature to have for wrapped language
	651
gdcm::QueryBase	
QueryBase	contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE
	652
gdcm::QueryFactory	
QueryFactory.h
	654
gdcm::QueryImage	
QueryImage	contains: class to construct an image-based query for C-FIND and C-MOVE
	655
gdcm::QueryPatient	
QueryPatient	contains: class to construct a patient-based query for c-find and c-move
	657
gdcm::QuerySeries	
QuerySeries	contains: class to construct a series-based query for c-find and c-move
	659
gdcm::QueryStudy	
QueryStudy.h	contains: class to construct a study-based query for C-FIND and C-MOVE
	661
gdcm::RAWCodec	
RAWCodec	class
	663
gdcm::Reader	
Reader	ala DOM (Document Object Model)
	666
gdcm::RealWorldValueMappingContent
	671
gdcm::Region	
Class for manipulation region	672
gdcm::Rescaler	
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
	674
gdcm::RLECodec	
Class to do RLE	677

gdcm::network::RoleSelectionSub	
RoleSelectionSub PS 3.7 Table D.3-9 SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASSOCIATE-RQ)	681
gdcm::SerieHelper::Rule	682
gdcm::Scanner	
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	683
gdcm::Segment	
This class defines a segment. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface	689
gdcm::SegmentedPaletteColorLookupTable	
SegmentedPaletteColorLookupTable class	693
gdcm::SegmentReader	
This class defines a segment reader. It reads attributes of group 0x0062	695
gdcm::SegmentWriter	
This class defines a segment writer. It writes attributes of group 0x0062	698
gdcm::SequenceOfFragments	
Class to represent a Sequence Of Fragments	700
gdcm::SequenceOfItems	
Class to represent a Sequence Of Items (value representation : SQ)	705
gdcm::SerieHelper	
SerieHelper DO NOT USE this class, it is only a temporary solution for ITK migration from GDCM 1.x to GDCM 2.x It will disappear soon, you've been warned	711
gdcm::Series	
Series	713
gdcm::network::ServiceClassApplicationInformation	714
gdcm::ServiceClassUser	
ServiceClassUser	715
gdcm::SHA1	
Class for SHA1	720
gdcm::SimpleMemberCommand< T >	
Command subclass that calls a pointer to a member function	721
gdcm::SimpleSubjectWatcher	
SimpleSubjectWatcher This is a typical Subject Watcher class. It will observe all events	725
gdcm::SmartPointer< ObjectType >	
Class for Smart Pointer	726
gdcm::network::SOPClassExtendedNegociationSub	
SOPClassExtendedNegociationSub PS 3.7 Table D.3-11 SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-ASSOCIATE-AC)	729
gdcm::SOPClassUIDToIOD	
Class convert a class SOP Class UID into IOD	730
gdcm::Sorter	
Sorter General class to do sorting using a custom function You simply need to provide a function of type: Sorter::SortFunction	731
gdcm::Spacing	
Class for Spacing	735
gdcm::Spectroscopy	
Spectroscopy class	737
gdcm::SplitMosaicFilter	
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	737
gdcm::StartEvent	738

gdcm::static_assert_test< x >	740
gdcm::STATIC_ASSERTION_FAILURE< x >	740
gdcm::STATIC_ASSERTION_FAILURE< true >	740
gdcm::StreamImageReader	
StreamImageReader	740
gdcm::StreamImageWriter	
StreamImageReader	743
gdcm::StrictScanner	
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	748
gdcm::String< TDelimiter, TMaxLength, TPadChar >	
String	754
gdcm::StringFilter	
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	757
gdcm::Study	
Study	759
gdcm::Subject	
Subject	760
gdcm::Surface	
This class defines a SURFACE IE. This members are taken from required surface mesh module attributes	762
gdcm::SurfaceHelper	
SurfaceHelper Helper class for Surface object	769
gdcm::SurfaceReader	
This class defines a SURFACE IE reader. It reads surface mesh module attributes	771
gdcm::SurfaceWriter	
This class defines a SURFACE IE writer. It writes surface mesh module attributes	774
gdcm::SwapCode	
SwapCode representation	775
gdcm::SwapperDoOp	777
gdcm::SwapperNoOp	777
gdcm::System	
Class to do system operation	778
gdcm::Table	
Table	782
gdcm::TableEntry	
TableEntry	783
gdcm::TableReader	
Class for representing a TableReader	784
gdcm::network::TableRow	786
gdcm::Tag	
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)	787
gdcm::TagPath	
Class to handle a path of tag	794
gdcm::Testing	
Class for testing	795
gdcm::Trace	
Trace	799
gdcm::TransferSyntax	
Class to manipulate Transfer Syntax	802

gdcmm::network::TransferSyntaxSub	
TransferSyntaxSub Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS	806
gdcmm::network::Transition	807
gdcmm::Type	
Type	809
gdcmm::UI	811
gdcmm::UIDGenerator	
Class for generating unique UID	811
gdcmm::UIDs	
All known uids	813
gdcmm::network::ULAction	
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	832
gdcmm::network::ULActionAA1	835
gdcmm::network::ULActionAA2	836
gdcmm::network::ULActionAA3	837
gdcmm::network::ULActionAA4	838
gdcmm::network::ULActionAA5	839
gdcmm::network::ULActionAA6	840
gdcmm::network::ULActionAA7	841
gdcmm::network::ULActionAA8	842
gdcmm::network::ULActionAE1	843
gdcmm::network::ULActionAE2	844
gdcmm::network::ULActionAE3	845
gdcmm::network::ULActionAE4	846
gdcmm::network::ULActionAE5	847
gdcmm::network::ULActionAE6	848
gdcmm::network::ULActionAE7	849
gdcmm::network::ULActionAE8	850
gdcmm::network::ULActionAR1	851
gdcmm::network::ULActionAR10	852
gdcmm::network::ULActionAR2	853
gdcmm::network::ULActionAR3	854
gdcmm::network::ULActionAR4	855
gdcmm::network::ULActionAR5	856
gdcmm::network::ULActionAR6	857
gdcmm::network::ULActionAR7	858
gdcmm::network::ULActionAR8	859
gdcmm::network::ULActionAR9	860
gdcmm::network::ULActionDT1	861
gdcmm::network::ULActionDT2	862
gdcmm::network::ULBasicCallback	
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	863
gdcmm::network::ULConnection	
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	864
gdcmm::network::ULConnectionCallback	867

gdcm::network::ULConnectionInfo	
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	868
gdcm::network::ULConnectionManager	
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)	870
gdcm::network::ULEvent	
ULEvent base class for network events	874
gdcm::network::ULTransitionTable	
ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates	875
gdcm::network::ULWritingCallback	875
gdcm::UNExplicitDataElement	
Class to read/write a DataElement as UNExplicit Data Element	877
gdcm::UNExplicitImplicitDataElement	
Class to read/write a DataElement as ExplicitImplicit Data Element This class gather two known bugs:	879
gdcm::Unpacker12Bits	
Pack/Unpack 12 bits pixel into 16bits	881
gdcm::Usage	
Usage	882
gdcm::UserEvent	883
gdcm::network::UserInformation	
UserInformation Table 9-16 USER INFORMATION ITEM FIELDS	885
gdcm::UUIDGenerator	
Class for generating unique UUID generate DCE 1.1 uid	886
gdcm::Validate	
Validate class	886
gdcm::Value	
Class to represent the value of a Data Element	888
gdcm::ValueIO< TDE, TSwap, TType >	
Class to dispatch template calls	890
gdcm::Version	
Major/minor and build version	891
gdcm::VL	
Value Length	892
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	894
gdcm::VMToLength< T >	898
gdcm::VR	
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	898
gdcm::VR16ExplicitDataElement	
Class to read/write a DataElement as Explicit Data Element	903
gdcm::VRToEncoding< T >	905
gdcm::VRToType< T >	905
gdcm::VRVLSize< T >	906
gdcm::VRVLSize< 0 >	906
gdcm::VRVLSize< 1 >	906
vtkGDCMImageReader	907

vtkGDCMImageReader2	913
vtkGDCMImageWriter	919
vtkGDCMMedicalImageProperties	923
vtkGDCMPolyDataReader	925
vtkGDCMPolyDataWriter	928
vtkGDCMTesting	931
vtkGDCMThreadedImageReader	934
vtkGDCMThreadedImageReader2	936
vtkImageColorViewer	940
vtkImageMapToColors16	947
vtkImageMapToWindowLevelColors2	950
vtkImagePlanarComponentsToComponents	952
vtkImageRGBToYBR	954
vtkImageYBRToRGB	956
vtkLookupTable16	957
vtkRTStructSetProperties	959
gdcm::Waveform	
Waveform class	964
gdcm::WLMFindQuery	
PatientRootQuery contains: the class which will produce a dataset for c-find with patient root	964
gdcm::Writer	
Writer ala DOM (Document Object Model) This class is a non-validating writer, it will only performs well- formedness check only	967
gdcm::XMLDictReader	
Class for representing a XMLDictReader	972
gdcm::XMLPrinter	973
gdcm::XMLPrivateDictReader	
Class for representing a XMLPrivateDictReader	976

Chapter 25

File Index

25.1 File List

Here is a list of all files with brief descriptions:

gdcmAAbortPDU.h	979
gdcmAAssociateACPDU.h	980
gdcmAAssociateRJPDU.h	980
gdcmAAssociateRQPDU.h	981
gdcmAbstractSyntax.h	982
gdcmAnonymizeEvent.h	983
gdcmAnonymizer.h	984
gdcmApplicationContext.h	985
gdcmApplicationEntity.h	986
gdcmAReleaseRPPDU.h	987
gdcmAReleaseRQPDU.h	988
gdcmARTIMTimer.h	989
gdcmASN1.h	990
gdcmAsynchronousOperationsWindowSub.h	990
gdcmAttribute.h	991
gdcmAudioCodec.h	993
gdcmBase64.h	993
gdcmBaseCompositeMessage.h	994
gdcmBaseNormalizedMessage.h	995
gdcmBasePDU.h	996
gdcmBaseQuery.h	997
gdcmBaseRootQuery.h	998
gdcmBasicOffsetTable.h	1000
gdcmBitmap.h	1001
gdcmBitmapToBitmapFilter.h	1002
gdcmBoxRegion.h	1003
gdcmByteBuffer.h	1003
gdcmByteSwap.h	1004
gdcmByteSwapFilter.h	1005
gdcmByteValue.h	1006
gdcmCAPICryptoFactory.h	1007
gdcmCAPICryptographicMessageSyntax.h	1007
gdcmCEchoMessages.h	1008
gdcmCFindMessages.h	1009

gdcmCMoveMessages.h	1010
gdcmCodec.h	1011
gdcmCoder.h	1012
gdcmCodeString.h	1013
gdcmCommand.h	1014
gdcmCommandDataSet.h	1016
gdcmCompositeMessageFactory.h	1016
gdcmCompositeNetworkFunctions.h	1017
gdcmConstCharWrapper.h	1018
gdcmCP246ExplicitDataElement.h	1019
gdcmCryptoFactory.h	1019
gdcmCryptographicMessageSyntax.h	1020
gdcmCSAElement.h	1021
gdcmCSAHeader.h	1023
gdcmCSAHeaderDict.h	1023
gdcmCSAHeaderDictEntry.h	1025
gdcmCStoreMessages.h	1026
gdcmCurve.h	1027
gdcmDataElement.h	1028
gdcmDataEvent.h	1029
gdcmDataSet.h	1030
gdcmDataSetEvent.h	1031
gdcmDataSetHelper.h	1032
gdcmDecoder.h	1033
gdcmDefinedTerms.h	1034
gdcmDeflateStream.h	1035
gdcmDefs.h	1035
gdcmDeltaEncodingCodec.h	1037
gdcmDICOmdir.h	1037
gdcmDICOmdirGenerator.h	1038
gdcmDict.h	1039
gdcmDictConverter.h	1041
gdcmDictEntry.h	1041
gdcmDictPrinter.h	1043
gdcmDicts.h	1043
gdcmDIMSE.h	1045
gdcmDirectionCosines.h	1045
gdcmDirectory.h	1046
gdcmDirectoryHelper.h	1047
gdcmDummyValueGenerator.h	1048
gdcmDumper.h	1048
gdcmElement.h	1049
gdcmEncapsulatedDocument.h	1051
gdcmEnumeratedValues.h	1052
gdcmEvent.h	1052
gdcmException.h	1054
gdcmExplicitDataElement.h	1055
gdcmExplicitImplicitDataElement.h	1056
gdcmFiducials.h	1056
gdcmFile.h	1057
gdcmFileAnonymizer.h	1058
gdcmFileChangeTransferSyntax.h	1059
gdcmFileDecompressLookupTable.h	1059
gdcmFileDerivation.h	1060

gdcmFileExplicitFilter.h	1061
gdcmFileMetaInformation.h	1061
gdcmFilename.h	1062
gdcmFileNameEvent.h	1063
gdcmFilenameGenerator.h	1064
gdcmFileSet.h	1065
gdcmFileStreamer.h	1066
gdcmFindPatientRootQuery.h	1067
gdcmFindStudyRootQuery.h	1068
gdcmFragment.h	1069
gdcmGlobal.h	1071
gdcmGroupDict.h	1072
gdcmIconImage.h	1072
gdcmIconImageFilter.h	1073
gdcmIconImageGenerator.h	1074
gdcmImage.h	1075
gdcmImageApplyLookupTable.h	1076
gdcmImageChangePhotometricInterpretation.h	1077
gdcmImageChangePlanarConfiguration.h	1077
gdcmImageChangeTransferSyntax.h	1078
gdcmImageCodec.h	1079
gdcmImageConverter.h	1080
gdcmImageFragmentSplitter.h	1081
gdcmImageHelper.h	1082
gdcmImageReader.h	1083
gdcmImageRegionReader.h	1083
gdcmImageToImageFilter.h	1084
gdcmImageWriter.h	1085
gdcmImplementationClassUIDSub.h	1086
gdcmImplementationUIDSub.h	1087
gdcmImplementationVersionNameSub.h	1088
gdcmImplicitDataElement.h	1089
gdcmIOD.h	1090
gdcmIODEntry.h	1091
gdcmIODs.h	1093
gdcmIPPSorter.h	1095
gdcmItem.h	1096
gdcmJPEG12Codec.h	1097
gdcmJPEG16Codec.h	1097
gdcmJPEG2000Codec.h	1098
gdcmJPEG8Codec.h	1099
gdcmJPEGCodec.h	1100
gdcmJPEGLSCCodec.h	1101
gdcmJSON.h	1102
gdcmKAKADUCodec.h	1103
gdcmLegacyMacro.h	1104
gdcmLO.h	1105
gdcmLookupTable.h	1105
gdcmMacro.h	1106
gdcmMacroEntry.h	1108
gdcmMacros.h	1109
gdcmMaximumLengthSub.h	1111
gdcmMD5.h	1112
gdcmMediaStorage.h	1113

gdcmMeshPrimitive.h	1114
gdcmModalityPerformedProcedureStepCreateQuery.h	1115
gdcmModalityPerformedProcedureStepSetQuery.h	1116
gdcmModule.h	1117
gdcmModuleEntry.h	1118
gdcmModules.h	1120
gdcmMovePatientRootQuery.h	1121
gdcmMoveStudyRootQuery.h	1122
gdcmNActionMessages.h	1123
gdcmNCreateMessages.h	1124
gdcmNDeleteMessages.h	1125
gdcmNestedModuleEntries.h	1125
gdcmNetworkEvents.h	1127
gdcmNetworkStateID.h	1128
gdcmNEventReportMessages.h	1129
gdcmNGetMessages.h	1129
gdcmNormalizedMessageFactory.h	1130
gdcmNormalizedNetworkFunctions.h	1131
gdcmNSetMessages.h	1132
gdcmObject.h	1132
gdcmOpenSSLCryptoFactory.h	1133
gdcmOpenSSLCryptographicMessageSyntax.h	1134
gdcmOpenSSLP7CryptoFactory.h	1135
gdcmOpenSSLP7CryptographicMessageSyntax.h	1136
gdcmOrientation.h	1138
gdcmOverlay.h	1138
gdcmParseException.h	1139
gdcmParser.h	1141
gdcmPatient.h	1141
gdcmPDataTFPDU.h	1142
gdcmPDBelement.h	1143
gdcmPDBHeader.h	1145
gdcmPDFCodec.h	1145
gdcmPDUFactory.h	1146
gdcmPersonName.h	1147
gdcmPGXCodec.h	1147
gdcmPhotometricInterpretation.h	1148
gdcmPixelFormat.h	1149
gdcmPixmap.h	1150
gdcmPixmapReader.h	1151
gdcmPixmapToPixmapFilter.h	1152
gdcmPixmapWriter.h	1153
gdcmPNMCodec.h	1154
gdcmPreamble.h	1155
gdcmPresentationContext.h	1156
gdcmPresentationContextAC.h	1157
gdcmPresentationContextGenerator.h	1159
gdcmPresentationContextRQ.h	1159
gdcmPresentationDataValue.h	1160
gdcmPrinter.h	1161
gdcmPrivateTag.h	1162
gdcmProgressEvent.h	1164
gdcmPVRGCodec.h	1164
gdcmPythonFilter.h	1165

gdcmQueryBase.h	1166
gdcmQueryFactory.h	1167
gdcmQueryImage.h	1168
gdcmQueryPatient.h	1169
gdcmQuerySeries.h	1170
gdcmQueryStudy.h	1170
gdcmRAWCodec.h	1171
gdcmReader.h	1172
gdcmRegion.h	1174
gdcmRescaler.h	1175
gdcmRLECodec.h	1175
gdcmRoleSelectionSub.h	1176
gdcmScanner.h	1177
gdcmSegment.h	1178
gdcmSegmentedPaletteColorLookupTable.h	1179
gdcmSegmentHelper.h	1180
gdcmSegmentReader.h	1182
gdcmSegmentWriter.h	1183
gdcmSequenceOfFragments.h	1184
gdcmSequenceOfItems.h	1185
gdcmSerieHelper.h	1185
gdcmSeries.h	1187
gdcmServiceClassApplicationInformation.h	1188
gdcmServiceClassUser.h	1189
gdcmSHA1.h	1189
gdcmSimpleSubjectWatcher.h	1190
gdcmSmartPointer.h	1191
gdcmSOPClassExtendedNegociationSub.h	1192
gdcmSOPClassUIDToIOD.h	1193
gdcmSorter.h	1194
gdcmSpacing.h	1196
gdcmSpectroscopy.h	1196
gdcmSplitMosaicFilter.h	1197
gdcmStaticAssert.h	1198
gdcmStreamImageReader.h	1199
gdcmStreamImageWriter.h	1199
gdcmStrictScanner.h	1200
gdcmString.h	1201
gdcmStringFilter.h	1202
gdcmStudy.h	1203
gdcmSubject.h	1204
gdcmSurface.h	1205
gdcmSurfaceHelper.h	1206
gdcmSurfaceReader.h	1207
gdcmSurfaceWriter.h	1208
gdcmSwapCode.h	1208
gdcmSwapper.h	1209
gdcmSystem.h	1210
gdcmTable.h	1211
gdcmTableEntry.h	1212
gdcmTableReader.h	1213
gdcmTag.h	1215
gdcmTagPath.h	1215
gdcmTagToVR.h	1216

gdcmTerminal.h	1217
gdcmTestDriver.h	1219
gdcmTesting.h	1219
gdcmTrace.h	1220
gdcmTransferSyntax.h	1223
gdcmTransferSyntaxSub.h	1224
gdcmType.h	1225
gdcmTypes.h	1226
gdcmUIDGenerator.h	1227
gdcmUIDs.h	1228
gdcmULAction.h	1228
gdcmULActionAA.h	1229
gdcmULActionAE.h	1230
gdcmULActionAR.h	1231
gdcmULActionDT.h	1232
gdcmULBasicCallback.h	1232
gdcmULConnection.h	1233
gdcmULConnectionCallback.h	1234
gdcmULConnectionInfo.h	1235
gdcmULConnectionManager.h	1237
gdcmULEvent.h	1237
gdcmULTransitionTable.h	1238
gdcmULWritingCallback.h	1240
gdcmUNExplicitDataElement.h	1240
gdcmUNExplicitImplicitDataElement.h	1241
gdcmUnpacker12Bits.h	1241
gdcmUsage.h	1242
gdcmUserInformation.h	1244
gdcmUUIDGenerator.h	1245
gdcmValidate.h	1245
gdcmValue.h	1246
gdcmValueIO.h	1247
gdcmVersion.h	1248
gdcmVL.h	1249
gdcmVM.h	1250
gdcmVR.h	1251
gdcmVR16ExplicitDataElement.h	1253
gdcmWaveform.h	1254
gdcmWin32.h	1254
gdcmWLMFindQuery.h	1255
gdcmWriter.h	1255
gdcmXMLDictReader.h	1257
gdcmXMLPrinter.h	1257
gdcmXMLPrivateDictReader.h	1258
vtkGDCMImageReader.h	1259
vtkGDCMImageReader2.h	1260
vtkGDCMImageWriter.h	1261
vtkGDCMMedicalImageProperties.h	1261
vtkGDCMPolyDataReader.h	1262
vtkGDCMPolyDataWriter.h	1263
vtkGDCMTesting.h	1263
vtkGDCMThreadedImageReader.h	1264
vtkGDCMThreadedImageReader2.h	1265
vtkImageColorViewer.h	1265

vtkImageMapToColors16.h	1266
vtkImageMapToWindowLevelColors2.h	1266
vtkImagePlanarComponentsToComponents.h	1267
vtkImageRGBToYBR.h	1267
vtkImageYBRToRGB.h	1268
vtkLookupTable16.h	1268
vtkRTStructSetProperties.h	1269

Chapter 26

Namespace Documentation

26.1 gdcM Namespace Reference

Namespaces

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

Class for Terminal Allow one to print in color in a shell.

Classes

- class [AbortEvent](#)
- class [AnonymizeEvent](#)
[AnonymizeEvent](#) Special type of event triggered during the Anonymization process.
- class [Anonymizer](#)
[Anonymizer](#) This class is a multi purpose anonymizer. It can work in 2 mode:
- 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 instanciation 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](#) contains: a baseclass which will produce a dataset for all dimse messages.*
- class [BaseRootQuery](#)
 - [BaseRootQuery](#) contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root.*
- class [BasicOffsetTable](#)
 - Class to represent a [BasicOffsetTable](#).*
- class [Bitmap](#)
 - [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)*
- class [BitmapToBitmapFilter](#)
 - [BitmapToBitmapFilter](#) class Super class for all filter taking an image and producing an output image.*
- class [BoxRegion](#)
 - 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)*
- class [ByteBuffer](#)
 - [ByteBuffer](#).*
- class [ByteSwap](#)
 - [ByteSwap](#).*
- class [ByteSwapFilter](#)
 - [ByteSwapFilter](#) In place byte-swapping of a dataset FIXME: FL status ??*
- class [ByteValue](#)
 - Class to represent binary value (array of bytes)*
- class [CAPICryptoFactory](#)
- class [CAPICryptographicMessageSyntax](#)
- class [Codec](#)
 - [Codec](#) class.*
- class [Coder](#)
 - [Coder](#).*
- class [CodeString](#)
 - [CodeString](#) This is an implementation of DICOM [VR](#): CS The ctor will properly Trim so that operator== is correct.*
- class [Command](#)
 - [Command](#) superclass for callback/observer methods.*
- class [CommandDataSet](#)
 - Class to represent a [Command DataSet](#).*
- class [CompositeNetworkFunctions](#)
 - 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:*
- 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](#) 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.
- class [CSAHeaderDictException](#)
- class [Curve](#)
[Curve](#) class to handle element 50xx,3000 [Curve](#) Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004.
- 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) A Data Set represents an instance of a real world Information [Object](#).
- class [DataSetEvent](#)
[DataSetEvent](#) Special type of event triggered during the [DataSet](#) store/move process.
- 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 This is a STD-GEN-CD [DICOMDIR](#) generator. ref: PS 3.11-2008 Annex D (Normative) - General Purpose CD-R and DVD Interchange Profiles.
- 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](#). Does not really exist within the DICOM definition, just a way to minimize storage and have a mapping from [gdc::Tag](#) to the needed information.

- 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](#) 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.

- 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 [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 [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](#) 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.
- class [FileAnonymizer](#)
FileAnonymizer.
- class [FileChangeTransferSyntax](#)
FileChangeTransferSyntax.
- class [FileDecompressLookupTable](#)
[FileDecompressLookupTable](#) class It decompress the segmented LUT into linearized one (only PALETTE_COLOR images) Output will be a [PhotometricInterpretation](#)=RGB image.
- class [FileDerivation](#)
[FileDerivation](#) class See PS 3.16 - 2008 For the list of Code [Value](#) that can be used for in Derivation Code Sequence.
- class [FileExplicitFilter](#)
[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.
- class [FileMetaInformation](#)
Class to represent a [File](#) Meta Information.
- class [Filename](#)
Class to manipulate file name's.
- class [FileNameEvent](#)
[FileNameEvent](#) Special type of event triggered during processing of [FileSet](#).
- class [FilenameGenerator](#)
FilenameGenerator.
- class [FileSet](#)
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.
- class [FileStreamer](#)
[FileStreamer](#) This class let a user create a massive DICOM [DataSet](#) from a template DICOM file, by appending chunks of data.
- class [FileWithName](#)
FileWithName.
- class [FindPatientRootQuery](#)
PatientRootQuery contains: the class which will produce a dataset for c-find with patient root.
- class [FindStudyRootQuery](#)
[FindStudyRootQuery](#) contains: the class which will produce a dataset for C-FIND with study root.
- 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 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.
- class [IconImageGenerator](#)
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:
- struct [ignore_char](#)
- class [Image](#)
Image This is the container for an [Image](#) in the general sense. From this container you should be able to request information like:
- class [ImageApplyLookupTable](#)
ImageApplyLookupTable class It applies the LUT the [PixelData](#) (only PALETTE_COLOR images) Output will be a [PhotometricInterpretation=RGB](#) image.
- class [ImageChangePhotometricInterpretation](#)
ImageChangePhotometricInterpretation class Class to change the Photometric Interpretation of an input DICOM.
- class [ImageChangePlanarConfiguration](#)
ImageChangePlanarConfiguration class Class to change the Planar configuration of an input DICOM By default it will change into the more usual representation: [PlanarConfiguration](#) = 0.
- class [ImageChangeTransferSyntax](#)
ImageChangeTransferSyntax class Class to change the transfer syntax of an input DICOM.
- class [ImageCodec](#)
ImageCodec.
- class [ImageConverter](#)
Image Converter.
- class [ImageFragmentSplitter](#)
ImageFragmentSplitter class For single frame image, DICOM standard allow splitting the frame into multiple fragments.
- class [ImageHelper](#)
ImageHelper (internal class, not intended for user level)
- class [ImageReader](#)
ImageReader.
- class [ImageRegionReader](#)
ImageRegionReader.
- class [ImageToImageFilter](#)
ImageToImageFilter class Super class for all filter taking an image and producing an output image.
- 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](#) 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.
- class [Item](#)
 - 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.
- 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 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.
- class [JPEGLSCodec](#)
 - JPEG-LS.
- class [JSON](#)
- class [KAKADUCodec](#)
 - [KAKADUCodec](#).
- class [LO](#)
 - [LO](#).
- class [LookupTable](#)
 - [LookupTable](#) class.
- class [Macro](#)
 - Class for representing a [Macro](#).
- class [Macros](#)
 - Class for representing a [Modules](#).
- class [MD5](#)
 - Class for [MD5](#).
- class [MediaStorage](#)
 - [MediaStorage](#).
- class [MemberCommand](#)
 - [Command](#) subclass that calls a pointer to a member function.
- class [MeshPrimitive](#)
 - This class defines surface mesh primitives. It is designed from surface mesh primitives macro.
- class [ModalityPerformedProcedureStepCreateQuery](#)
 - [ModalityPerformedProcedureStepCreateQuery](#) contains: the class which will produce a dataset for n-create for Modality Performed Procedure Step sop class.
- class [ModalityPerformedProcedureStepSetQuery](#)

[ModalityPerformedProcedureStepSetQuery](#) contains: the class which will produce a dataset for n-set for Modality Performed Procedure Step sop class.

- 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](#) contains: the class which will produce a dataset for c-move with patient root.

- class [MoveStudyRootQuery](#)

[MoveStudyRootQuery](#) contains: the class which will produce a dataset for C-MOVE with study root.

- class [NestedModuleEntries](#)

Class for representing a [NestedModuleEntries](#).

- class [NoEvent](#)
- class [NormalizedNetworkFunctions](#)

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:

- class [Object](#)

[Object](#).

- class [OpenSSLCryptoFactory](#)
- class [OpenSSLCryptographicMessageSyntax](#)
- class [OpenSSLP7CryptoFactory](#)
- class [OpenSSLP7CryptographicMessageSyntax](#)

Class for [CryptographicMessageSyntax](#) encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities.

- 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 See PGX as used in JPEG 2000 implementation and reference images.*
- class [PhotometricInterpretation](#)
 - Class to represent an [PhotometricInterpretation](#).*
- class [PixelFormat](#)
 - PixelFormat.*
- class [Pixmap](#)
 - 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)
- class [PixmapReader](#)
 - PixmapReader.*
- class [PixmapToPixmapFilter](#)
 - PixmapToPixmapFilter* class Super class for all filter taking an image and producing an output image.
- class [PixmapWriter](#)
 - PixmapWriter* This class will takes two inputs:
- class [PNMCodec](#)
 - Class to do PNM PNM is the Portable anymap file format. The main web page can be found at: <http://netpbm.sourceforge.net/>.*
- class [Preamble](#)
 - DICOM [Preamble](#) (Part 10)*
- class [PresentationContext](#)
 - PresentationContext.*
- class [PresentationContextGenerator](#)
 - 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.
- 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* Special type of event triggered during.
- 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* contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE.
- class [QueryFactory](#)
 - QueryFactory.h.*
- class [QueryImage](#)
 - QueryImage* contains: class to construct an image-based query for C-FIND and C-MOVE.
- class [QueryPatient](#)
 - QueryPatient* contains: class to construct a patient-based query for c-find and c-move.

- class [QuerySeries](#)
QuerySeries contains: class to construct a series-based query for c-find and c-move.
- class [QueryStudy](#)
QueryStudy.h contains: class to construct a study-based query for C-FIND and C-MOVE.
- class [RAWCodec](#)
RAWCodec class.
- class [Reader](#)
Reader ala DOM (Document *Object* Model)
- struct [RealWorldValueMappingContent](#)
- class [Region](#)
Class for manipulation region.
- class [Rescaler](#)
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.
- class [RLECodec](#)
Class to do RLE.
- class [Scanner](#)
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*.
- class [Segment](#)
This class defines a segment. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface.
- class [SegmentedPaletteColorLookupTable](#)
SegmentedPaletteColorLookupTable class.
- class [SegmentReader](#)
This class defines a segment reader. It reads attributes of group 0x0062.
- class [SegmentWriter](#)
This class defines a segment writer. It writes attributes of group 0x0062.
- class [SequenceOfFragments](#)
Class to represent a Sequence Of Fragments.
- class [SequenceOfItems](#)
Class to represent a Sequence Of Items (value representation : SQ)
- 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* This is a typical [Subject](#) Watcher class. It will observe all events.
- class [SmartPointer](#)

Class for Smart Pointer.
 - class [SOPClassUIDToIOD](#)

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

[Sorter](#) General class to do sorting using a custom function You simply need to provide a function of type: [Sorter::Sort←Function](#).
 - class [Spacing](#)

Class for [Spacing](#).
 - class [Spectroscopy](#)

[Spectroscopy](#) class.
 - class [SplitMosaicFilter](#)

[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.
 - class [StartEvent](#)
 - struct [static_assert_test](#)
 - struct [STATIC_ASSERTION_FAILURE](#)
 - struct [STATIC_ASSERTION_FAILURE< true >](#)
 - class [StreamImageReader](#)

[StreamImageReader](#).
 - class [StreamImageWriter](#)

[StreamImageReader](#).
 - class [StrictScanner](#)

[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](#).
 - class [String](#)

[String](#).
 - class [StringFilter](#)

[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.
 - class [Study](#)

[Study](#).
 - class [Subject](#)

[Subject](#).
 - class [Surface](#)

This class defines a SURFACE IE. This members are taken from required surface mesh module attributes.
 - class [SurfaceHelper](#)

[SurfaceHelper](#) Helper class for [Surface](#) object.
 - class [SurfaceReader](#)

This class defines a SURFACE IE reader. It reads surface mesh module attributes.
 - class [SurfaceWriter](#)

This class defines a SURFACE IE writer. It writes surface mesh module attributes.
 - 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](#)). Basically an uint32_t which can also be expressed as two uint16_t (group and 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](#) This class gather two known bugs:*
- class [Unpacker12Bits](#)
 - Pack/Unpack 12 bits pixel into 16bits.*
- class [Usage](#)
 - [Usage](#).
- class [UserEvent](#)
- class [UUIDGenerator](#)
 - Class for generating unique UUID generate DCE 1.1 uid.*
- 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 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.

- 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 contains: the class which will produce a dataset for c-find with patient root.

- class [Writer](#)

Writer ala DOM (Document [Object](#) Model) This class is a non-validating writer, it will only performs well- formedness check only.

- 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](#)<"\", 64, 0 > [UIComp](#)
- 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

- [ignore_char](#) const [backslash](#) ("\\")
- [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 Version &v)`
- `std::ostream & operator<< (std::ostream &_os, const NestedModuleEntries &_val)`
- `std::ostream & operator<< (std::ostream &os, const SwapCode &sc)`
- `std::ostream & operator<< (std::ostream &os, const FileSet &f)`
- `std::ostream & operator<< (std::ostream &os, const Region &r)`
- `std::ostream & operator<< (std::ostream &os, Event &e)`

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

- `std::ostream & operator<< (std::ostream &os, const PDBElement &val)`
- `std::ostream & operator<< (std::ostream &os, const CommandDataSet &val)`
- `std::ostream & operator<< (std::ostream &os, const Orientation &o)`
- `std::ostream & operator<< (std::ostream &_os, const IODs &_val)`
- `std::ostream & operator<< (std::ostream &_os, const Macros &_val)`
- `std::ostream & operator<< (std::ostream &_os, const Modules &_val)`
- `std::ostream & operator<< (std::ostream &_os, const Type &val)`
- `std::ostream & operator<< (std::ostream &_os, const ModuleEntry &_val)`
- `std::ostream & operator<< (std::ostream &_os, const GroupDict &_val)`
- `std::ostream & operator<< (std::ostream &os, const PrivateTag &val)`
- `std::ostream & operator<< (std::ostream &_os, const IOD &_val)`
- `std::ostream & operator<< (std::ostream &os, const File &val)`
- `std::ostream & operator<< (std::ostream &_os, const Usage &val)`
- `std::ostream & operator<< (std::ostream &os, const Sorter &s)`
- `std::ostream & operator<< (std::ostream &os, const CSAHeaderDictEntry &val)`
- `std::ostream & operator<< (std::ostream &os, const Preamble &val)`
- `std::ostream & operator<< (std::ostream &_os, const IODEntry &_val)`
- `std::ostream & operator<< (std::ostream &_os, const Macro &_val)`
- `std::ostream & operator<< (std::ostream &os, const CSAHeaderDict &val)`
- `std::ostream & operator<< (std::ostream &os, const Dicts &d)`
- `std::ostream & operator<< (std::ostream &os, const PDBHeader &d)`
- `std::ostream & operator<< (std::ostream &os, const CodeString &str)`
- `std::ostream & operator<< (std::ostream &os, const Directory &d)`
- `std::ostream & operator<< (std::ostream &_os, const Module &_val)`
- `std::ostream & operator<< (std::ostream &os, const PhotometricInterpretation &val)`
- `std::ostream & operator<< (std::ostream &os, const Global &g)`
- `std::ostream & operator<< (std::ostream &os, const Object &obj)`
- `std::ostream & operator<< (std::ostream &os, const BasicOffsetTable &val)`
- `std::ostream & operator<< (std::ostream &os, const DictEntry &val)`
- `std::ostream & operator<< (std::ostream &os, const VL &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 FileMetaInformation &val)`
- `std::ostream & operator<< (std::ostream &_os, const TransferSyntax &ts)`
- `std::ostream & operator<< (std::ostream &_os, const VM &_val)`
- `std::ostream & operator<< (std::ostream &os, const StrictScanner &s)`
- `std::ostream & operator<< (std::ostream &os, const Scanner &s)`
- `std::ostream & operator<< (std::ostream &os, const Dict &val)`
- `std::ostream & operator<< (std::ostream &_os, const MediaStorage &ms)`
- `std::ostream & operator<< (std::ostream &_os, const VR &val)`
- `std::ostream & operator<< (std::ostream &os, const Fragment &val)`
- `std::ostream & operator<< (std::ostream &os, const PixelFormat &pf)`
- `std::ostream & operator<< (std::ostream &_os, const UI &_val)`

- `std::ostream & operator<< (std::ostream &os, const DataElement &val)`
- `std::ostream & operator<< (std::ostream &_os, const Tag &_val)`
- `std::ostream & operator<< (std::ostream &os, const DataSet &val)`
- `std::ostream & operator<< (std::ostream &os, const Item &val)`
- `std::ostream & operator<< (std::ostream &os, const PrivateDict &val)`
- `std::ostream & operator<< (std::ostream &_os, const UIDs &uid)`
- `bool operator== (const CodeString &ref, const CodeString &cs)`
- `template<char TDelimiter, unsigned int TMaxLength, char TPadChar>
std::istream & operator>> (std::istream &is, String< TDelimiter, TMaxLength, TPadChar > &ms)`
- `std::istream & operator>> (std::istream &in, ignore_char const &ic)`
- `std::istream & operator>> (std::istream &_is, Tag &_val)`
- `template<typename Float >
std::string to_string (Float data)`
- `TYPETOENCODING (SQ, VRBINARY, unsigned char) TYPETOENCODING(UN`

Variables

- static [Global GlobalInstance](#)
- [VRBINAR](#)[Y](#)

26.1.1 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.

26.1.2 Typedef Documentation

26.1.2.1 `typedef String<'\\',16> gdcm::AECComp`

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

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

26.1.2.4 `typedef String<'\\',16> gdcm::CSCComp`

- 26.1.2.5 `typedef String<'\',64> gdcm::DAComp`
- 26.1.2.6 `typedef String<'\',64> gdcm::DTComp`
- 26.1.2.7 `typedef std::vector< SmartPointer<FileWithName> > gdcm::FileList`
- 26.1.2.8 `typedef Bitmap gdcm::IconImage`
- 26.1.2.9 `typedef String<'\',64> gdcm::LOComp`
- 26.1.2.10 `typedef String<'\',64> gdcm::LTComp`
- 26.1.2.11 `typedef ModuleEntry gdcm::MacroEntry`
- 26.1.2.12 `typedef NestedModuleEntries gdcm::NestedMacroEntries`
- 26.1.2.13 `typedef String<'\',64> gdcm::PNComp`
- 26.1.2.14 `typedef String<'\',64> gdcm::SHComp`
- 26.1.2.15 `typedef String<'\',64> gdcm::STComp`
- 26.1.2.16 `typedef String<'\',16> gdcm::TMComp`
- 26.1.2.17 `typedef String<'\',64,0> gdcm::UIComp`
- 26.1.2.18 `typedef String<'\',64> gdcm::UTComp`

26.1.3 Enumeration Type Documentation

- 26.1.3.1 `enum gdcm::CompOperators`

Enumerator

GDCM_EQUAL
GDCM_DIFFERENT
GDCM_GREATER
GDCM_GREATEROREQUAL
GDCM_LESS
GDCM_LESSCOREQUAL

- 26.1.3.2 `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

26.1.3.3 enum gdcm::ENQueryType

Enumerator

eCreateMMPS
eSetMMPS

26.1.3.4 enum gdcm::EQueryLevel

Enumerator

ePatient
eStudy
eSeries
eImage

26.1.3.5 enum gdcm::EQueryType

Enumerator

eFind
eMove
eWLMFind

26.1.3.6 enum gdcm::ERootType

Enumerator

ePatientRootType
eStudyRootType

26.1.3.7 enum gdcM::LodModeType

Enumerator

LD_ALL
LD_NOSEQ
LD_NOSHADOW
LD_NOSHADOWSEQ

26.1.4 Function Documentation

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

Referenced by gdcM::EncodingImplementation< VR::VRASCII >::ReadComputeLength().

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

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

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

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

References gdcM::Version::Print().

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

References gdcM::ModuleEntry::DataElementType, gdcM::ModuleEntry::DescriptionField, and gdcM::ModuleEntry::Name.

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

References gdcM::SwapCode::GetSwapCodeString().

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

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

References gdcM::Region::Print().

26.1.4.10 std::ostream& gdcM::operator<< (std::ostream & os, Event & e) [inline]

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

References gdcM::Event::Print().

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

References gdcM::PDBelement::NameField, and gdcM::PDBelement::ValueField.

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

References `gdcmm::DataSet::Print()`.

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

References `gdcmm::Orientation::Print()`.

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

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

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

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

References `gdcmm::Type::GetTypeString()`.

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

References `gdcmm::ModuleEntry::DataElementType`, `gdcmm::ModuleEntry::DescriptionField`, and `gdcmm::ModuleEntry::↵Name`.

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

References `gdcmm::GroupDict::GetAbbreviation()`, `gdcmm::GroupDict::GetName()`, and `gdcmm::GroupDict::Size()`.

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

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

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

References `gdcmm::File::GetHeader()`.

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

References `gdcmm::Usage::GetUsageString()`.

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

References `gdcmm::Sorter::Print()`.

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

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

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

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

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

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

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

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

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

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

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

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

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

References `gdcm::PhotometricInterpretation::GetPIString()`.

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

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

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

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

References `gdcm::DataElement::GetByteValue()`, `gdcm::DataElement::ValueField`, and `gdcm::DataElement::Value↵LengthField`.

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

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

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

References `gdcm::CSAElement::DataField`, `gdcm::ByteValue::GetLength()`, `gdcm::ByteValue::GetPointer()`, `gdcm::C↵SAElement::KeyField`, `gdcm::CSAElement::NameField`, `gdcm::CSAElement::NoOfItemsField`, `gdcm::CSAElement::↵SyngoDTField`, `gdcm::CSAElement::ValueMultiplicityField`, `gdcm::VM::VM1`, and `gdcm::CSAElement::VRField`.

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

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

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

References `gdcmm::FileMetaInformation::GetPreamble()`, and `gdcmm::DataSet::Print()`.

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

References `gdcmm::TransferSyntax::GetTSSString()`.

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

References `gdcmm::VM::GetVMString()`.

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

References `gdcmm::StrictScanner::Print()`.

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

References `gdcmm::Scanner::Print()`.

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

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

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

References `gdcmm::VR::GetVRString()`.

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

References `gdcmm::DataElement::TagField`, `gdcmm::DataElement::ValueField`, and `gdcmm::DataElement::ValueLengthField`.

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

References `gdcmm::PixelFormat::Print()`.

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

References `gdcmm::UI::Internal`.

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

References `gdcmm::Object::Print()`, `gdcmm::DataElement::TagField`, `gdcmm::DataElement::ValueField`, `gdcmm::DataElement::ValueLengthField`, and `gdcmm::DataElement::VRField`.

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

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

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

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

References `gdcm::DataSet::Print()`, `gdcm::DataElement::TagField`, and `gdcm::DataElement::ValueLengthField`.

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

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

References `gdcm::UIDs::GetName()`, and `gdcm::UIDs::GetString()`.

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

Examples:

[DumpPhilipsECHO.cxx](#).

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

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

References `gdcm::ignore_char::m_char`.

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

References `gdcm::Tag::SetElement()`, and `gdcm::Tag::SetGroup()`.

26.1.4.64 `template<typename Float > std::string gdcm::to_string (Float data)`

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

26.1.4.65 `gdcm::TYPETOENCODING (SQ , VRBINARY , unsigned char)`

26.1.5 Variable Documentation

26.1.5.1 `Global gdcm::GlobalInstance` `[static]`

26.1.5.2 `gdcm::VRBINARY`

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

26.2 gdcmm::network Namespace Reference

Classes

- class [AAbortPDU](#)
AAbortPDU Table 9-26 A-ABORT PDU FIELDS.
- class [AAssociateACPDU](#)
AAssociateACPDU Table 9-17 ASSOCIATE-AC PDU fields.
- class [AAssociateRJPDU](#)
AAssociateRJPDU Table 9-21 ASSOCIATE-RJ PDU FIELDS.
- class [AAssociateRQPDU](#)
AAssociateRQPDU Table 9-11 ASSOCIATE-RQ PDU fields.
- class [AbstractSyntax](#)
AbstractSyntax Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS.
- class [ApplicationContext](#)
ApplicationContext Table 9-12 APPLICATION CONTEXT ITEM FIELDS.
- class [AReleaseRPPDU](#)
AReleaseRPPDU Table 9-25 A-RELEASE-RP PDU fields.
- class [AReleaseRQPDU](#)
AReleaseRQPDU Table 9-24 A-RELEASE-RQ PDU FIELDS.
- class [ARTIMTimer](#)
ARTIMTimer This file contains the code for the ARTIM timer.
- class [AsynchronousOperationsWindowSub](#)
AsynchronousOperationsWindowSub PS 3.7 Table D.3-7 ASYNCHRONOUS OPERATIONS WINDOW SUB-ITEM FIELDS (A-ASSOCIATE-RQ)
- class [BaseCompositeMessage](#)
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.
- class [BaseNormalizedMessage](#)
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.
- class [BasePDU](#)
BasePDU base class for PDUs.
- class [CEchoRQ](#)
CEchoRQ this file defines the messages for the echo action.
- class [CEchoRSP](#)
CEchoRSP this file defines the messages for the echo action.
- class [CFind](#)
- class [CFindCancelRQ](#)
CFindCancelRQ this file defines the messages for the cfind action.
- class [CFindRQ](#)
CFindRQ this file defines the messages for the cfind action.
- class [CFindRSP](#)
CFindRSP this file defines the messages for the cfind action.
- class [CMoveCancelRq](#)
- class [CMoveRQ](#)

- [*CMoveRQ*](#) this file defines the messages for the cmove action.
- class [CMoveRSP](#)
 - [*CMoveRSP*](#) this file defines the messages for the cmove action.
- class [CompositeMessageFactory](#)
 - [*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).
- class [CStoreRQ](#)
 - [*CStoreRQ*](#) this file defines the messages for the cecho action.
- class [CStoreRSP](#)
 - [*CStoreRSP*](#) this file defines the messages for the cecho action.
- class [DIMSE](#)
 - [*DIMSE*](#) PS 3.7 - 2009 Annex E [Command Dictionary \(Normative\)](#) E.1 REGISTRY OF DICOM COMMAND ELEMENTS [Table E.1-1 COMMAND FIELDS \(PART 1\)](#)
- class [ImplementationClassUIDSub](#)
 - [*ImplementationClassUIDSub*](#) PS 3.7 [Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS \(A-ASSOCIATE-RQ\)](#)
- class [ImplementationUIDSub](#)
 - [*ImplementationUIDSub*](#) [Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS \(A-ASSOCIATE-AC\)](#)
- class [ImplementationVersionNameSub](#)
 - [*ImplementationVersionNameSub*](#) [Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS \(A-ASSOCIATE-RQ\)](#)
- class [MaximumLengthSub](#)
 - [*MaximumLengthSub*](#) Annex D [Table D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS \(A-ASSOCIATE-RQ\)](#)
- class [NActionRQ](#)
 - [*NActionRQ*](#) this file defines the messages for the NAction action.
- class [NActionRSP](#)
 - [*NActionRSP*](#) this file defines the messages for the NAction action.
- class [NCreateRQ](#)
 - [*NCreateRQ*](#) this file defines the messages for the ncreate action.
- class [NCreateRSP](#)
 - [*NCreateRSP*](#) this file defines the messages for the ncreate action.
- class [NDeleteRQ](#)
 - [*NDeleteRQ*](#) this file defines the messages for the ndelete action.
- class [NDeleteRSP](#)
 - [*NDeleteRSP*](#) this file defines the messages for the ndelete action.
- class [NEventReportRQ](#)
 - [*NEventReportRQ*](#) this file defines the messages for the neventreport action.
- class [NEventReportRSP](#)
 - [*NEventReportRSP*](#) this file defines the messages for the neventreport action.
- class [NGetRQ](#)
 - [*NGetRQ*](#) this file defines the messages for the nget action.
- class [NGetRSP](#)
 - [*NGetRSP*](#) this file defines the messages for the nget action.
- class [NormalizedMessageFactory](#)
- class [NSetRQ](#)
 - [*NSetRQ*](#) this file defines the messages for the nset action.

- class [NSetRSP](#)
NSetRSP this file defines the messages for the nset action.
- class [PDataTFPDU](#)
PDataTFPDU Table 9-22 P-DATA-TF PDU FIELDS.
- class [PDUFactory](#)
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.
- class [PresentationContextAC](#)
PresentationContextAC Table 9-18 PRESENTATION CONTEXT ITEM FIELDS.
- class [PresentationContextRQ](#)
PresentationContextRQ Table 9-13 PRESENTATION CONTEXT ITEM FIELDS.
- class [PresentationDataValue](#)
PresentationDataValue Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS.
- class [RoleSelectionSub](#)
RoleSelectionSub PS 3.7 Table D.3-9 SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASSOCIATE-RQ)
- class [ServiceClassApplicationInformation](#)
- class [SOPClassExtendedNegociationSub](#)
SOPClassExtendedNegociationSub PS 3.7 Table D.3-11 SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-ASSOCIATE-AC)
- class [TableRow](#)
- class [TransferSyntaxSub](#)
TransferSyntaxSub Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS.
- struct [Transition](#)
- class [ULAction](#)
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](#).
- 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 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](#).
- class [ULConnection](#)

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.
- class [ULConnectionCallback](#)
- class [ULConnectionInfo](#)

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.
- class [ULConnectionManager](#)

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).
- class [ULEvent](#)

ULEvent base class for network events.
- class [ULTransitionTable](#)

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

UserInformation Table 9-16 USER INFORMATION ITEM FIELDS.

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

26.2.1 Enumeration Type Documentation

26.2.1.1 enum [gdcmm::network::EEventID](#)

Enumerator

eAASSOCIATERequestLocalUser
eTransportConnConfirmLocal
eASSOCIATE_ACPDUreceived
eASSOCIATE_RJPDUreceived
eTransportConnIndicLocal
eAASSOCIATE_RQPDUreceived
eAASSOCIATEResponseAccept
eAASSOCIATEResponseReject
ePDATArequest
ePDATATFPDU
eARELEASERequest
eARELEASE_RQPDUReceivedOpen
eARELEASE_RPPDUReceived
eARELEASEResponse
eAABORTRequest
eAABORTPDUReceivedOpen

eTransportConnectionClosed
eARTIMTimerExpired
eUnrecognizedPDURceived
eEventDoesNotExist

26.2.1.2 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

26.2.2 Function Documentation

26.2.2.1 int gdcmm::network::GetStateIndex (EStateID inState) [inline]

References eSta10ReleaseCollisionAc, eSta11ReleaseCollisionRq, eSta12ReleaseCollisionAcLocal, eSta13AwaitingClose, eSta1Idle, eSta2Open, eSta3WaitLocalAssoc, eSta4LocalAssocDone, eSta5WaitRemoteAssoc, eSta6TransferReady, eSta7WaitRelease, eSta8WaitLocalRelease, eSta9ReleaseCollisionRqLocal, and eStaDoesNotExist.

26.2.3 Variable Documentation

26.2.3.1 const int gdcmm::network::cMaxEventID = eEventDoesNotExist

26.2.3.2 const int gdcmm::network::cMaxStateID = 13

Referenced by gdcmm::network::TableRow::TableRow(), and gdcmm::network::TableRow::~~TableRow().

26.3 gdcmm::SegmentHelper Namespace Reference

Classes

- struct [BasicCodedEntry](#)

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

26.4 gdcmm::terminal Namespace Reference

Class for Terminal Allow one to print in color in a shell.

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)

26.4.1 Detailed Description

Class for Terminal Allow one to print in color in a shell.

- support VT100 compatible shell
- win32 console

26.4.2 Enumeration Type Documentation

26.4.2.1 enum gdcmm::terminal::Attribute

Enumerator

reset
bright
dim
underline
blink
reverse
hidden

26.4.2.2 enum gdcmm::terminal::Color

Enumerator

black
red
green
yellow
blue
magenta
cyan
white

26.4.2.3 enum gdcmm::terminal::Mode

Enumerator

CONSOLE
VT100

26.4.3 Function Documentation

26.4.3.1 GDCM_EXPORT std::string gdcmm::terminal::setattribute (Attribute *att*)

26.4.3.2 GDCM_EXPORT std::string gdcmm::terminal::setbgcolor (Color *c*)

26.4.3.3 GDCM_EXPORT std::string gdcmm::terminal::setfgcolor (Color *c*)

26.4.3.4 GDCM_EXPORT void gdcmm::terminal::setmode (Mode *m*)

Chapter 27

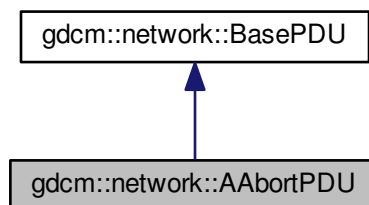
Class Documentation

27.1 gdcm::network::AAbortPDU Class Reference

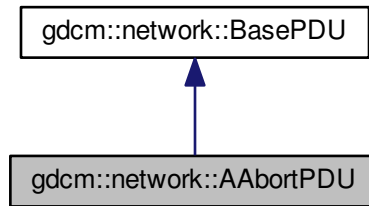
[AAbortPDU](#) [Table 9-26](#) A-ABORT PDU FIELDS.

```
#include <gdcmAAbortPDU.h>
```

Inheritance diagram for `gdcm::network::AAbortPDU`:



Collaboration diagram for `gdcm::network::AAabortPDU`:



Public Member Functions

- [AAabortPDU](#) ()
- `bool` [IsLastFragment](#) () const
- `void` [Print](#) (std::ostream &os) const
- `std::istream &` [Read](#) (std::istream &is)
- `void` [SetReason](#) (const uint8_t r)
- `void` [SetSource](#) (const uint8_t s)
- `size_t` [Size](#) () const
- `const std::ostream &` [Write](#) (std::ostream &os) const

27.1.1 Detailed Description

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

27.1.2 Constructor & Destructor Documentation

27.1.2.1 `gdcm::network::AAabortPDU::AAabortPDU ()`

27.1.3 Member Function Documentation

27.1.3.1 `bool` `gdcm::network::AAabortPDU::IsLastFragment ()` const `[inline], [virtual]`

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

27.1.3.2 `void` `gdcm::network::AAabortPDU::Print (std::ostream & os)` const `[virtual]`

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

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

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

27.1.3.4 void gdcmm::network::AAbortPDU::SetReason (const uint8_t r)

27.1.3.5 void gdcmm::network::AAbortPDU::SetSource (const uint8_t s)

27.1.3.6 size_t gdcmm::network::AAbortPDU::Size () const [virtual]

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

27.1.3.7 const std::ostream& gdcmm::network::AAbortPDU::Write (std::ostream & os) const [virtual]

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

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

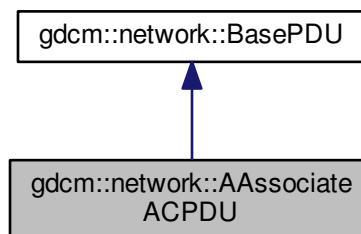
- [gdcmmAbortPDU.h](#)

27.2 gdcmm::network::AAssociateACPDU Class Reference

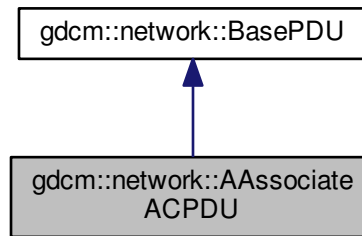
[AAssociateACPDU](#) Table 9-17 ASSOCIATE-AC PDU fields.

```
#include <gdcmmAAssociateACPDU.h>
```

Inheritance diagram for gdcmm::network::AAssociateACPDU:



Collaboration diagram for `gdc::network::AAssociateACPDU`:



Public Types

- typedef `std::vector< PresentationContextAC >::size_type` [SizeType](#)

Public Member Functions

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

Protected Member Functions

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

Friends

- class [AAssociateRQPDU](#)

27.2.1 Detailed Description

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

27.2.2 Member Typedef Documentation

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

27.2.3 Constructor & Destructor Documentation

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

27.2.4 Member Function Documentation

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

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

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

27.2.4.4 `const UserInformation& gdcm::network::AAssociateACPDU::GetUserInformation () const [inline]`

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

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

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

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

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

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

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

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

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

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

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

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

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

27.2.5 Friends And Related Function Documentation

27.2.5.1 friend class **AAssociateRQPDU** [friend]

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

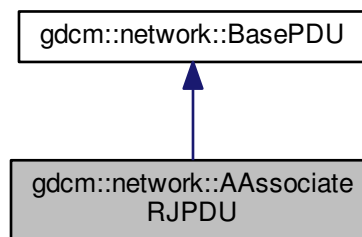
- [gdcmAAssociateACPDU.h](#)

27.3 gdcmm::network::AAssociateRJPDU Class Reference

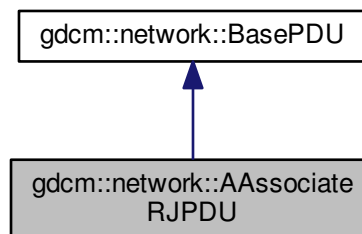
[AAssociateRJPDU](#) Table 9-21 ASSOCIATE-RJ PDU FIELDS.

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

Inheritance diagram for gdcmm::network::AAssociateRJPDU:



Collaboration diagram for gdcmm::network::AAssociateRJPDU:



Public Member Functions

- [AAssociateRJPDU](#) ()
- bool [IsLastFragment](#) () const

- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- size_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

27.3.1 Detailed Description

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

27.3.2 Constructor & Destructor Documentation

27.3.2.1 `gdcm::network::AAssociateRJPDUTable::AAssociateRJPDUTable ()`

27.3.3 Member Function Documentation

27.3.3.1 `bool gdcm::network::AAssociateRJPDUTable::IsLastFragment () const` `[inline]`, `[virtual]`

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

27.3.3.2 `void gdcm::network::AAssociateRJPDUTable::Print (std::ostream & os) const` `[virtual]`

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

27.3.3.3 `std::istream& gdcm::network::AAssociateRJPDUTable::Read (std::istream & is)` `[virtual]`

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

27.3.3.4 `size_t gdcm::network::AAssociateRJPDUTable::Size () const` `[virtual]`

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

27.3.3.5 `const std::ostream& gdcm::network::AAssociateRJPDUTable::Write (std::ostream & os) const` `[virtual]`

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

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

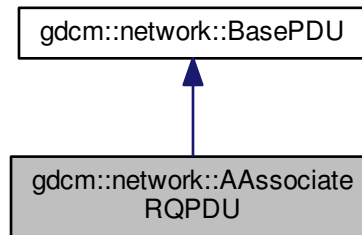
- [gdcmAAssociateRJPDUTable.h](#)

27.4 gdcm::network::AAssociateRQPDU Class Reference

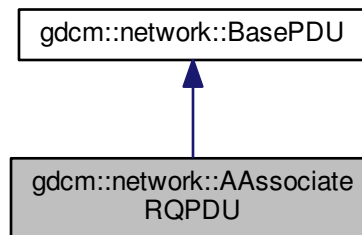
[AAssociateRQPDU Table](#) 9-11 ASSOCIATE-RQ PDU fields.

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

Inheritance diagram for `gdcn::network::AAssociateRQPDU`:



Collaboration diagram for `gdcn::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
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- 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
- const std::ostream & [Write](#) (std::ostream &os) const

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](#)

27.4.1 Detailed Description

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

27.4.2 Member Typedef Documentation

27.4.2.1 typedef std::vector<[PresentationContextRQ](#)> gdcm::network::AAssociateRQPDU::PresentationContextArrayType

27.4.2.2 typedef std::vector<[PresentationContextRQ](#)>::size_type gdcm::network::AAssociateRQPDU::SizeType

27.4.3 Constructor & Destructor Documentation

27.4.3.1 gdcm::network::AAssociateRQPDU::AAssociateRQPDU ()

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

27.4.4 Member Function Documentation

27.4.4.1 void gdcm::network::AAssociateRQPDU::AddPresentationContext ([PresentationContextRQ](#) const & pc)

- 27.4.4.2 `std::string gdcmm::network::AAssociateRQPDU::GetCalledAETitle () const` `[inline]`
- 27.4.4.3 `std::string gdcmm::network::AAssociateRQPDU::GetCallingAETitle () const` `[inline]`
- 27.4.4.4 `SizeType gdcmm::network::AAssociateRQPDU::GetNumberOfPresentationContext () const` `[inline]`
- 27.4.4.5 `PresentationContextRQ const& gdcmm::network::AAssociateRQPDU::GetPresentationContext (SizeType i) const` `[inline]`
- 27.4.4.6 `const PresentationContextRQ* gdcmm::network::AAssociateRQPDU::GetPresentationContextByAbstractSyntax (AbstractSyntax const & absyn) const`
- 27.4.4.7 `const PresentationContextRQ* gdcmm::network::AAssociateRQPDU::GetPresentationContextByID (uint8_t i) const`
- 27.4.4.8 `PresentationContextArrayType const& gdcmm::network::AAssociateRQPDU::GetPresentationContexts ()` `[inline]`
- 27.4.4.9 `std::string gdcmm::network::AAssociateRQPDU::GetReserved43_74 () const` `[protected]`
- 27.4.4.10 `const UserInformation& gdcmm::network::AAssociateRQPDU::GetUserInformation () const` `[inline]`
- 27.4.4.11 `static bool gdcmm::network::AAssociateRQPDU::IsAETitleValid (const char title[16])` `[static]`

Check whether or not the.

Parameters

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

- 27.4.4.12 `bool gdcmm::network::AAssociateRQPDU::IsLastFragment () const` `[inline]`, `[virtual]`

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

- 27.4.4.13 `void gdcmm::network::AAssociateRQPDU::Print (std::ostream & os) const` `[virtual]`

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

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

- 27.4.4.14 `std::istream& gdcmm::network::AAssociateRQPDU::Read (std::istream & is)` `[virtual]`

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

- 27.4.4.15 `void gdcmm::network::AAssociateRQPDU::SetCalledAETitle (const char calledaetitle[16])`

Set the Called AE Title.

- 27.4.4.16 `void gdcmm::network::AAssociateRQPDU::SetCallingAETitle (const char callingaetitle[16])`

Set the Calling AE Title.

27.4.4.17 `void gdcm::network::AAssociateRQPDU::SetUserInfo (UserInfo const & ui)`

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

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

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

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

27.4.5 Friends And Related Function Documentation

27.4.5.1 `friend class AAssociateACPDU` `[friend]`

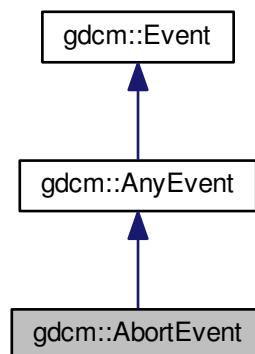
The documentation for this class was generated from the following file:

- [gdcmAAssociateRQPDU.h](#)

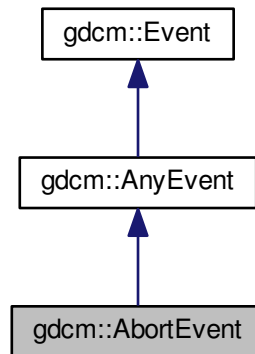
27.5 gdcm::AbortEvent Class Reference

```
#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](#)

27.6 `gdcm::network::AbstractSyntax` Class Reference

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

```
#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

27.6.1 Detailed Description

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

27.6.2 Constructor & Destructor Documentation

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

27.6.3 Member Function Documentation

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

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

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

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

27.6.3.5 `std::istream& gdcm::network::AbstractSyntax::Read (std::istream & is)`

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

27.6.3.7 `void gdcm::network::AbstractSyntax::SetNameFromUID (UIDs::TSName tsname)`

27.6.3.8 `size_t gdcm::network::AbstractSyntax::Size () const`

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

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

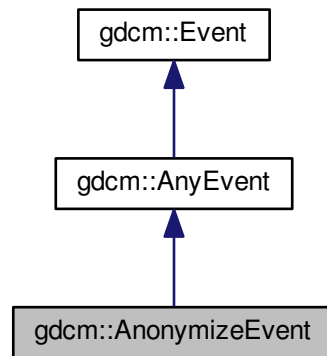
- [gdcmAbstractSyntax.h](#)

27.7 gdcm::AnonymizeEvent Class Reference

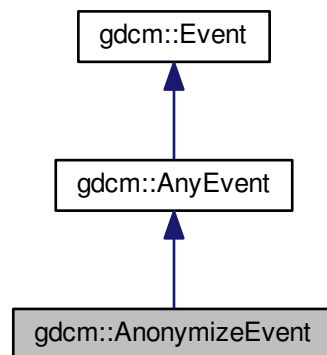
[AnonymizeEvent](#) Special type of event triggered during the Anonymization process.

```
#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](#) ([Tag](#) const &tag=0)
- [AnonymizeEvent](#) (const [Self](#) &s)

- virtual [~AnonymizeEvent](#) ()
- virtual bool [CheckEvent](#) (const [::gdcm::Event](#) *e) const
- virtual const char * [GetEventName](#) () const
- [Tag](#) const & [GetTag](#) () const
- virtual [::gdcm::Event](#) * [MakeObject](#) () const
- void [SetTag](#) (const [Tag](#) &t)

27.7.1 Detailed Description

[AnonymizeEvent](#) Special type of event triggered during the Anonymization process.

See also

[Anonymizer](#)

27.7.2 Member Typedef Documentation

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

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

27.7.3 Constructor & Destructor Documentation

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

27.7.3.2 `virtual gdcm::AnonymizeEvent::~~AnonymizeEvent () [inline],[virtual]`

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

27.7.4 Member Function Documentation

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

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

Return the StringName associated with the event.

Implements [gdcm::Event](#).

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

27.7.4.4 `virtual ::gdcm::Event* gdcm::AnonymizeEvent::MakeObject () const [inline],[virtual]`

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

Implements [gdcm::Event](#).

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

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

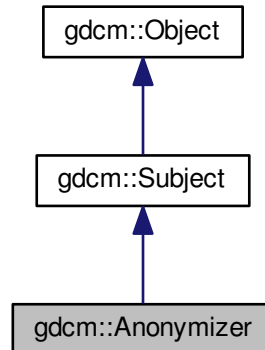
- [gdcmAnonymizeEvent.h](#)

27.8 gdcmm::Anonymizer Class Reference

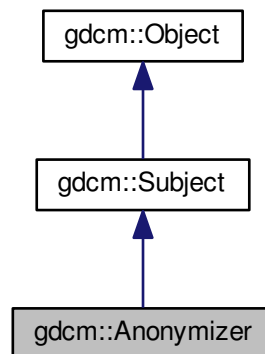
[Anonymizer](#) This class is a multi purpose anonymizer. It can work in 2 mode:

```
#include <gdcmmAnonymizer.h>
```

Inheritance diagram for gdcmm::Anonymizer:



Collaboration diagram for gdcmm::Anonymizer:



Public Member Functions

- [Anonymizer](#) ()
- [~Anonymizer](#) ()

- 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)

27.8.1 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](#).

27.8.2 Constructor & Destructor Documentation

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

27.8.2.2 `gdcm::Anonymizer::~~Anonymizer ()`

27.8.3 Member Function Documentation

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

27.8.3.2 `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

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

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

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

Warning

the mapping is definitely lost

27.8.3.5 `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](#).

27.8.3.6 `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](#).

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

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

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

for wrapped language: instantiate a reference counted object

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

27.8.3.11 `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

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

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

Examples:

[ClinicalTrialAnnotate.cxx](#).

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

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

Examples:

[ClinicalTrialAnnotate.cxx](#).

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

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

27.8.3.15 `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](#).

27.8.3.16 `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

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

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

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

Set/Get [File](#).

Examples:

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

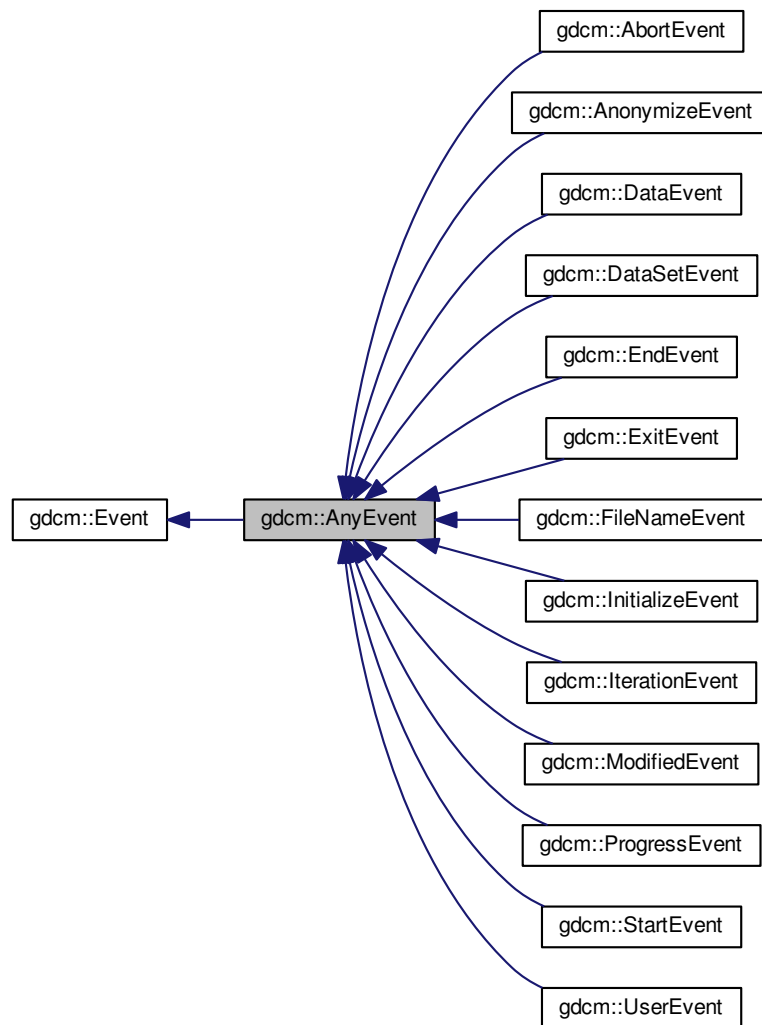
The documentation for this class was generated from the following file:

- [gdcmAnonymizer.h](#)

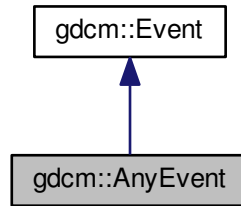
27.9 `gdcm::AnyEvent` Class Reference

```
#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](#)

27.10 gdcm::network::ApplicationContext Class Reference

[ApplicationContext](#) Table 9-12 APPLICATION CONTEXT ITEM FIELDS.

```
#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`

27.10.1 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)

27.10.2 Constructor & Destructor Documentation

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

27.10.3 Member Function Documentation

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

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

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

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

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

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

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

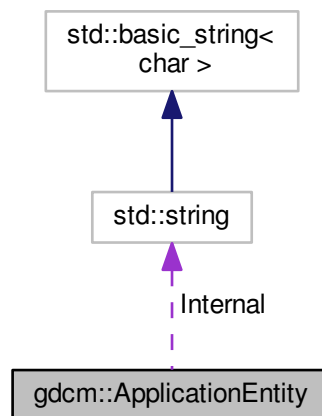
- [gdcmApplicationContext.h](#)

27.11 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](#) = ''

27.11.1 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

27.11.2 Member Function Documentation

27.11.2.1 bool [gdcm::ApplicationEntity::IsValid](#) () const [inline]

27.11.2.2 void [gdcm::ApplicationEntity::Print](#) (std::ostream & os) const [inline]

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

27.11.2.4 void [gdcm::ApplicationEntity::Squeeze](#) () [inline]

27.11.3 Member Data Documentation

27.11.3.1 std::string [gdcm::ApplicationEntity::Internal](#)

27.11.3.2 const unsigned int [gdcm::ApplicationEntity::MaxLength](#) = 16 [static]

27.11.3.3 const unsigned int [gdcm::ApplicationEntity::MaxNumberOfComponents](#) = 1 [static]

27.11.3.4 const char [gdcm::ApplicationEntity::Padding](#) = '' [static]

27.11.3.5 const char [gdcm::ApplicationEntity::Separator](#) = '' [static]

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

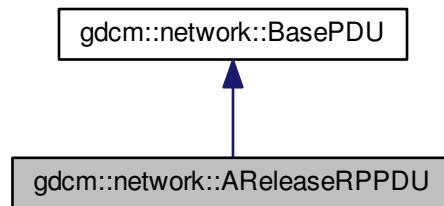
- [gdcmApplicationEntity.h](#)

27.12 gdcmm::network::AReleaseRPPDU Class Reference

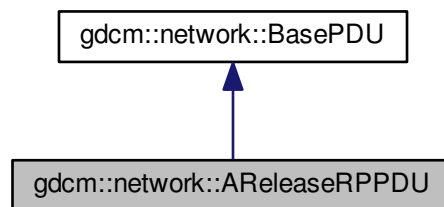
[AReleaseRPPDU](#) Table 9-25 A-RELEASE-RP PDU fields.

```
#include <gdcmAReleaseRPPDU.h>
```

Inheritance diagram for gdcmm::network::AReleaseRPPDU:



Collaboration diagram for gdcmm::network::AReleaseRPPDU:



Public Member Functions

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

27.12.1 Detailed Description

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

27.12.2 Constructor & Destructor Documentation

27.12.2.1 `gdcn::network::AReleaseRPPDU::AReleaseRPPDU ()`

27.12.3 Member Function Documentation

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

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

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

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

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

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

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

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

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

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

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

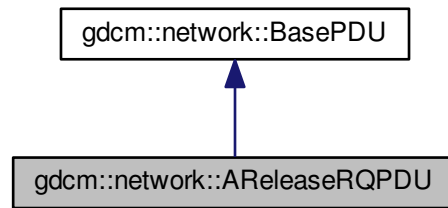
- [gdcnAReleaseRPPDU.h](#)

27.13 gdcn::network::AReleaseRQPDU Class Reference

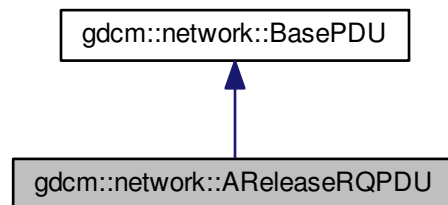
[AReleaseRQPDU Table](#) 9-24 A-RELEASE-RQ PDU FIELDS.

```
#include <gdcnAReleaseRQPDU.h>
```


Inheritance diagram for gdcmm::network::AReleaseRQPDU:



Collaboration diagram for gdcmm::network::AReleaseRQPDU:



Public Member Functions

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

27.13.1 Detailed Description

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

27.13.2 Constructor & Destructor Documentation

27.13.2.1 gdcmm::network::AReleaseRQPDU::AReleaseRQPDU ()

27.13.3 Member Function Documentation

27.13.3.1 `bool gdcn::network::AReleaseRQPDU::IsLastFragment () const [inline],[virtual]`

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

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

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

27.13.3.3 `std::istream& gdcn::network::AReleaseRQPDU::Read (std::istream & is) [virtual]`

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

27.13.3.4 `size_t gdcn::network::AReleaseRQPDU::Size () const [virtual]`

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

27.13.3.5 `const std::ostream& gdcn::network::AReleaseRQPDU::Write (std::ostream & os) const [virtual]`

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

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

- [gdcnAReleaseRQPDU.h](#)

27.14 gdcn::network::ARTIMTimer Class Reference

[ARTIMTimer](#) This file contains the code for the ARTIM timer.

```
#include <gdcnARTIMTimer.h>
```

Public Member Functions

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

27.14.1 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.

27.14.2 Constructor & Destructor Documentation

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

27.14.3 Member Function Documentation

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

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

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

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

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

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

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

- [gdcmARTIMTimer.h](#)

27.15 gdcm::ASN1 Class Reference

Class for [ASN1](#).

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

Public Member Functions

- [ASN1](#) ()
- [~ASN1](#) ()

Static Public Member Functions

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

Protected Member Functions

- int [TestPBKDF2](#) ()

27.15.1 Detailed Description

Class for [ASN1](#).

27.15.2 Constructor & Destructor Documentation

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

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

27.15.3 Member Function Documentation

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

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

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

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

- [gdcmASN1.h](#)

27.16 gdcm::network::AsynchronousOperationsWindowSub Class Reference

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

```
#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

27.16.1 Detailed Description

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

27.16.2 Constructor & Destructor Documentation

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

27.16.3 Member Function Documentation

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

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

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

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

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

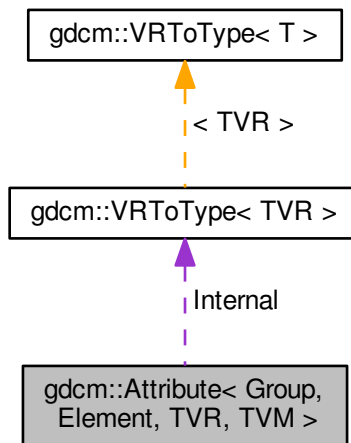
- [gdcmAsynchronousOperationsWindowSub.h](#)

27.17 gdcm::Attribute< Group, Element, TVR, TVM > Class Template Reference

[Attribute](#) class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary.

```
#include <gdcmAttribute.h>
```

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



Public Types

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

Public Member Functions

- [GDCM_STATIC_ASSERT](#) ((([VR::VRType](#)) TVR &([VR::VRType](#))(TagToType< Group, [Element](#) >::VRType)))

- `GDCM_STATIC_ASSERT` (((`VM::VMType`) TVM &(`VM::VMType`)(TagToType< Group, `Element` >::`VMType`)))
- `GDCM_STATIC_ASSERT` (((((`VR::VRType`) TVR &`VR::VR_VM1`)&&((`VM::VMType`) TVM==`VM::VM1`))||!((`VR::VRType`) TVR &`VR::VR_VM1`)))
- `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)

27.17.1 Detailed Description

`template<uint16_t Group, uint16_t Element, int 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](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_←_Image_Writer.cxx](#), [FixOrientation.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GenFakeIdentifyFile.cxx](#), [Get_←_SequenceUltrasound.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [ReadAnd_←_PrintAttributes.cxx](#), [rle2img.cxx](#), [SortImage.cxx](#), [StreamImageReaderTest.cxx](#), and [VolumeSorter.cxx](#).

27.17.2 Member Typedef Documentation

27.17.2.1 `template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, TVM >::ArrayType`

27.17.3 Member Enumeration Documentation

27.17.3.1 `template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> anonymous enum`

Enumerator

VMType

27.17.4 Member Function Documentation

27.17.4.1 `template<uint16_t Group, uint16_t Element, int 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)))`

27.17.4.2 `template<uint16_t Group, uint16_t Element, int 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)))`

27.17.4.3 `template<uint16_t Group, uint16_t Element, int 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)))`

27.17.4.4 `template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> DataElement gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement () const [inline]`

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

27.17.4.5 `template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> static VM gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVM () [inline], [static]`

27.17.4.6 `template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> static VR gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVR () [inline], [static]`

27.17.4.7 `template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> unsigned int gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues () const [inline]`

Referenced by `gdcmm::Attribute< Group, Element, TVR, TVM >::operator<()>`, and `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator<()>`.

27.17.4.8 `template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> static Tag gdcmm::Attribute< Group, Element, TVR, TVM >::GetTag () [inline],[static]`

27.17.4.9 `template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> ArrayType& gdcmm::Attribute< Group, Element, TVR, TVM >::GetValue (unsigned int idx = 0) [inline]`

27.17.4.10 `template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> ArrayType const& gdcmm::Attribute< Group, Element, TVR, TVM >::GetValue (unsigned int idx = 0) const [inline]`

27.17.4.11 `template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> const ArrayType* gdcmm::Attribute< Group, Element, TVR, TVM >::GetValues () const [inline]`

Referenced by `gdcmm::Attribute< Group, Element, TVR, TVM >::operator!=()`, `gdcmm::Attribute< Group, Element, T←VR, VM::VM1 >::operator!=()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::operator<()>`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator<()>`, `gdcmm::Attribute< Group, Element, TVR, TVM >::operator==()`, and `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator==()`.

27.17.4.12 `template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> static VM gdcmm::Attribute< Group, Element, TVR, TVM >::GetVM () [inline],[static]`

27.17.4.13 `template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> static VR gdcmm::Attribute< Group, Element, TVR, TVM >::GetVR () [inline],[static]`

27.17.4.14 `template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> bool gdcmm::Attribute< Group, Element, TVR, TVM >::operator!= (const Attribute< Group, Element, TVR, TVM > & att) const [inline]`

References `gdcmm::Attribute< Group, Element, TVR, TVM >::GetValues()`.

27.17.4.15 `template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> bool gdcmm::Attribute< Group, Element, TVR, TVM >::operator< (const Attribute< Group, Element, TVR, TVM > & att) const [inline]`

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

27.17.4.16 `template<uint16_t Group, uint16_t Element, int 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()`.

27.17.4.17 `template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> ArrayType& gdcm::Attribute< Group, Element, TVR, TVM >::operator[](unsigned int idx) [inline]`

27.17.4.18 `template<uint16_t Group, uint16_t Element, int 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]`

27.17.4.19 `template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> void gdcm::Attribute< Group, Element, TVR, TVM >::Print (std::ostream & os) const [inline]`

27.17.4.20 `template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> void gdcm::Attribute< Group, Element, TVR, TVM >::Set (DataSet const & ds) [inline]`

References `gdcm::DataSet::GetDataElement()`.

27.17.4.21 `template<uint16_t Group, uint16_t Element, int 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()`.

27.17.4.22 `template<uint16_t Group, uint16_t Element, int 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()`.

27.17.4.23 `template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement (DataElement const & de) [inline]`

References `gdcm::DataElement::GetByteValue()`, `gdcm::Tag::GetGroup()`, `gdcm::DataElement::GetTag()`, `gdcm::DataElement::GetVR()`, and `gdcm::DataElement::IsEmpty()`.

27.17.4.24 `template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet (DataSet const & ds) [inline]`

References `gdcm::DataSet::FindDataElement()`, `gdcm::DataSet::GetDataElement()`, and `gdcm::DataElement::IsEmpty()`.

- 27.17.4.25 `template<uint16_t Group, uint16_t Element, int 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]`
- 27.17.4.26 `template<uint16_t Group, uint16_t Element, int 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:

[LargeVRDSExplicit.cxx](#).

27.17.5 Member Data Documentation

- 27.17.5.1 `template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> ArrayType gdcM::Attribute< Group, Element, TVR, TVM >::Internal[VMToLength< TVM >::Length]`

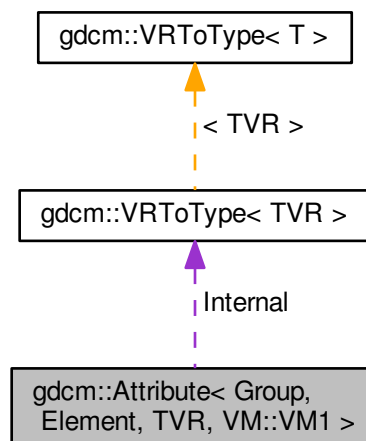
The documentation for this class was generated from the following file:

- [gdcMAttribute.h](#)

27.18 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](#) (VMToLength< [VM::VM1](#) >::Length==1)
- [GDCM_STATIC_ASSERT](#) ((([VR::VRType](#)) TVR &([VR::VRType](#))(TagToType< Group, [Element](#) >::VRType)))
- [GDCM_STATIC_ASSERT](#) ((([VM::VMType](#)) [VM::VM1](#) &([VM::VMType](#))(TagToType< Group, [Element](#) >::VMType)))
- [GDCM_STATIC_ASSERT](#) ((((([VR::VRType](#)) TVR &[VR::VR_VM1](#))&&(([VM::VMType](#)) [VM::VM1](#)==[VM::VM1](#)))||!(([VR::VRType](#)) TVR &[VR::VR_VM1](#))))
- [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)

27.18.1 Member Typedef Documentation

27.18.1.1 `template<uint16_t Group, uint16_t Element, int TVR> typedef VRToType<TVR>::Type gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::ArrayType`

27.18.2 Member Enumeration Documentation

27.18.2.1 `template<uint16_t Group, uint16_t Element, int TVR> anonymous enum`

Enumerator

VMType

27.18.3 Member Function Documentation

27.18.3.1 `template<uint16_t Group, uint16_t Element, int TVR> gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (VMToLength< VM::VM1 >::Length ==1)`

27.18.3.2 `template<uint16_t Group, uint16_t Element, int TVR> gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))`

27.18.3.3 `template<uint16_t Group, uint16_t Element, int TVR> gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (((VM::VMType) VM::VM1 &(VM::VMType)(TagToType< Group, Element >::VMType)))`

27.18.3.4 `template<uint16_t Group, uint16_t Element, int 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)))`

27.18.3.5 `template<uint16_t Group, uint16_t Element, int TVR> DataElement gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement () const [inline]`

References `gdcmm::DataElement::GetVR()`, `gdcmm::DataElement::SetByteValue()`, and `gdcmm::DataElement::SetVR()`.

27.18.3.6 `template<uint16_t Group, uint16_t Element, int TVR> static VM gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetDictVM () [inline],[static]`

27.18.3.7 `template<uint16_t Group, uint16_t Element, int TVR> static VR gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetDictVR () [inline],[static]`

27.18.3.8 `template<uint16_t Group, uint16_t Element, int TVR> unsigned int gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetNumberOfValues () const [inline]`

27.18.3.9 `template<uint16_t Group, uint16_t Element, int TVR> static Tag gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetTag () [inline],[static]`

27.18.3.10 `template<uint16_t Group, uint16_t Element, int TVR> ArrayType& gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetValue () [inline]`

27.18.3.11 `template<uint16_t Group, uint16_t Element, int TVR> ArrayType const& gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetValue () const [inline]`

27.18.3.12 `template<uint16_t Group, uint16_t Element, int TVR> const ArrayType* gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetValues () const [inline]`

27.18.3.13 `template<uint16_t Group, uint16_t Element, int TVR> static VM gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetVM () [inline], [static]`

27.18.3.14 `template<uint16_t Group, uint16_t Element, int TVR> static VR gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetVR () [inline], [static]`

27.18.3.15 `template<uint16_t Group, uint16_t Element, int 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()`.

27.18.3.16 `template<uint16_t Group, uint16_t Element, int 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()`.

27.18.3.17 `template<uint16_t Group, uint16_t Element, int 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()`.

27.18.3.18 `template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Print (std::ostream & os) const [inline]`

27.18.3.19 `template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Set (DataSet const & ds) [inline]`

References `gdcm::DataSet::GetDataElement()`.

27.18.3.20 `template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue (const ByteValue * bv) [inline], [protected]`

References `gdcm::ByteValue::GetLength()`, and `gdcm::ByteValue::GetPointer()`.

27.18.3.21 `template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap (const ByteValue * bv) [inline], [protected]`

References `gdcm::ByteValue::GetLength()`, and `gdcm::ByteValue::GetPointer()`.

27.18.3.22 `template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement (DataElement const & de) [inline]`

References `gdcm::DataElement::GetByteValue()`, `gdcm::Tag::GetGroup()`, `gdcm::DataElement::GetTag()`, `gdcm::DataElement::GetVR()`, and `gdcm::DataElement::IsEmpty()`.

```
27.18.3.23  template<uint16_t Group, uint16_t Element, int TVR> void gdcM::Attribute< Group, Element, TVR, VM::VM1
>::SetFromDataSet ( DataSet const & ds )  [inline]
```

References gdcM::DataSet::FindDataElement(), gdcM::DataSet::GetDataElement(), and gdcM::DataElement::Is←
Empty().

```
27.18.3.24  template<uint16_t Group, uint16_t Element, int TVR> void gdcM::Attribute< Group, Element, TVR, VM::VM1
>::SetValue ( ArrayType v )  [inline]
```

27.18.4 Member Data Documentation

```
27.18.4.1  template<uint16_t Group, uint16_t Element, int TVR> ArrayType gdcM::Attribute< Group, Element, TVR,
VM::VM1 >::Internal
```

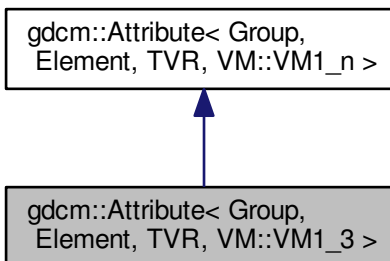
The documentation for this class was generated from the following file:

- [gdcMAttribute.h](#)

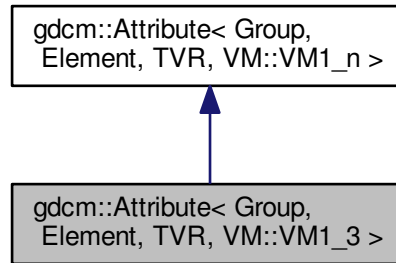
27.19 gdcM::Attribute< Group, Element, TVR, VM::VM1_3 > Class Template Reference

```
#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

27.19.1 Member Function Documentation

27.19.1.1 `template<uint16_t Group, uint16_t Element, int TVR> VM gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >::GetVM () const [inline]`

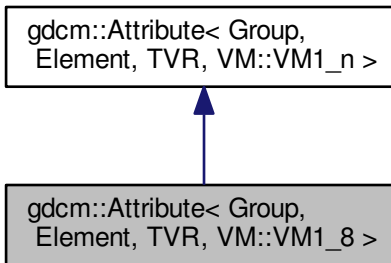
The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

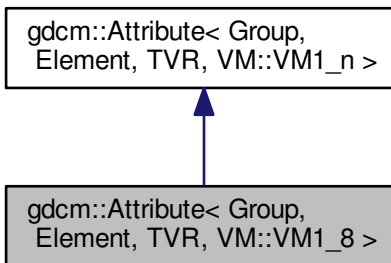
27.20 gdcm::Attribute< Group, Element, TVR, VM::VM1_8 > Class Template Reference

```
#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

27.20.1 Member Function Documentation

27.20.1.1 `template<uint16_t Group, uint16_t Element, int TVR> VM gdcM::Attribute< Group, Element, TVR, VM::VM1_8 >::GetVM () const` [inline]

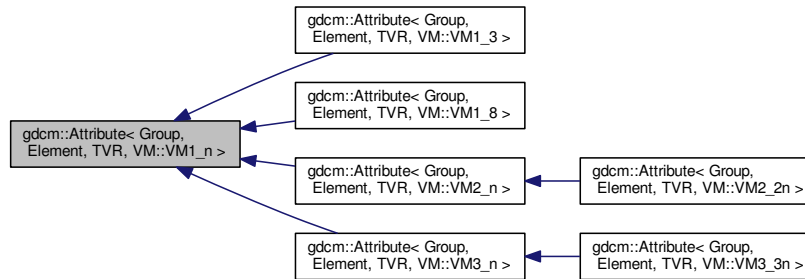
The documentation for this class was generated from the following file:

- [gdcMAttribute.h](#)

27.21 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::VRType)(TagToType< Group, Element >::VRType)))`
- `GDCM_STATIC_ASSERT ((VM::VM1_n & (VM::VMType)(TagToType< Group, Element >::VMType)))`
- `GDCM_STATIC_ASSERT (((((VR::VRType) TVR & VR::VR_VM1) && ((VM::VMType) TagToType< Group, Element >::VMType == VM::VM1)) || ((VR::VRType) TVR & VR::VR_VM1)))`
- `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 (unsigned int idx, ArrayType v)`
- `void SetValue (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)

27.21.1 Member Typedef Documentation

27.21.1.1 `template<uint16_t Group, uint16_t Element, int TVR> typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::ArrayType`

27.21.2 Constructor & Destructor Documentation

27.21.2.1 `template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Attribute () [inline],[explicit]`

27.21.2.2 `template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::~~Attribute () [inline]`

27.21.3 Member Function Documentation

27.21.3.1 `template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))`

27.21.3.2 `template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GDCM_STATIC_ASSERT ((VM::VM1_n &(VM::VMType)(TagToType< Group, Element >::VMType)))`

27.21.3.3 `template<uint16_t Group, uint16_t Element, int 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))||!((VR::VRType) TVR &VR::VR_VM1)))`

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

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

27.21.3.5 `template<uint16_t Group, uint16_t Element, int TVR> static VM gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVM () [inline],[static]`

27.21.3.6 `template<uint16_t Group, uint16_t Element, int TVR> static VR gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVR () [inline],[static]`

27.21.3.7 `template<uint16_t Group, uint16_t Element, int TVR> unsigned int gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetNumberOfValues () const [inline]`

- 27.21.3.8 `template<uint16_t Group, uint16_t Element, int TVR> static Tag gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetTag() [inline], [static]`
- 27.21.3.9 `template<uint16_t Group, uint16_t Element, int TVR> ArrayType& gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue(unsigned int idx = 0) [inline]`
- 27.21.3.10 `template<uint16_t Group, uint16_t Element, int TVR> ArrayType const& gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue(unsigned int idx = 0) const [inline]`
- 27.21.3.11 `template<uint16_t Group, uint16_t Element, int TVR> const ArrayType* gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValues() const [inline]`
- 27.21.3.12 `template<uint16_t Group, uint16_t Element, int TVR> static VM gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetVM() [inline], [static]`
- 27.21.3.13 `template<uint16_t Group, uint16_t Element, int TVR> static VR gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetVR() [inline], [static]`
- 27.21.3.14 `template<uint16_t Group, uint16_t Element, int TVR> ArrayType& gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::operator[](unsigned int idx) [inline]`
- 27.21.3.15 `template<uint16_t Group, uint16_t Element, int TVR> ArrayType const& gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::operator[](unsigned int idx) const [inline]`
- 27.21.3.16 `template<uint16_t Group, uint16_t Element, int TVR> void gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::Print(std::ostream & os) const [inline]`
- 27.21.3.17 `template<uint16_t Group, uint16_t Element, int TVR> void gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::Set(DataSet const & ds) [inline]`

References `gdcmm::DataSet::GetDataElement()`.

- 27.21.3.18 `template<uint16_t Group, uint16_t Element, int TVR> void gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue(const ByteValue * bv) [inline], [protected]`

References `gdcmm::ByteValue::GetLength()`, and `gdcmm::ByteValue::GetPointer()`.

- 27.21.3.19 `template<uint16_t Group, uint16_t Element, int TVR> void gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement(DataElement const & de) [inline]`

References `gdcmm::DataElement::GetByteValue()`, `gdcmm::Tag::GetGroup()`, `gdcmm::DataElement::GetTag()`, `gdcmm::DataElement::GetVR()`, and `gdcmm::DataElement::IsEmpty()`.

- 27.21.3.20 `template<uint16_t Group, uint16_t Element, int TVR> void gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet(DataSet const & ds) [inline]`

References `gdcmm::DataSet::FindDataElement()`, `gdcmm::DataSet::GetDataElement()`, and `gdcmm::DataElement::IsEmpty()`.

27.21.3.21 `template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetNumberOfValues (unsigned int numel) [inline]`

27.21.3.22 `template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue (unsigned int idx, ArrayType v) [inline]`

27.21.3.23 `template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue (ArrayType v) [inline]`

References SetValue().

Referenced by SetValue().

27.21.3.24 `template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValues (const ArrayType * array, unsigned int numel, bool own = false) [inline]`

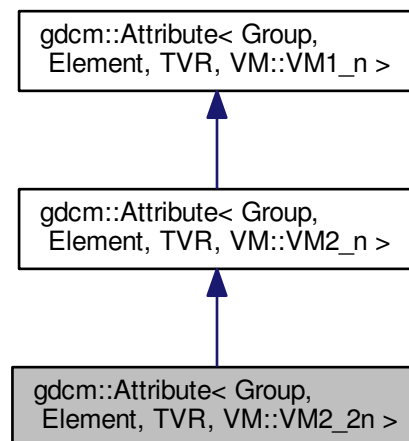
The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

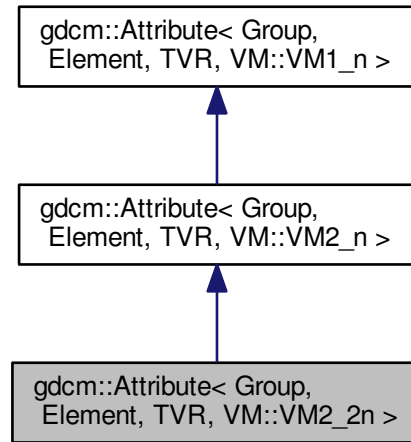
27.22 gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > Class Template Reference

```
#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

27.22.1 Member Function Documentation

27.22.1.1 `template<uint16_t Group, uint16_t Element, int TVR> static VM gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >::GetVM () [inline], [static]`

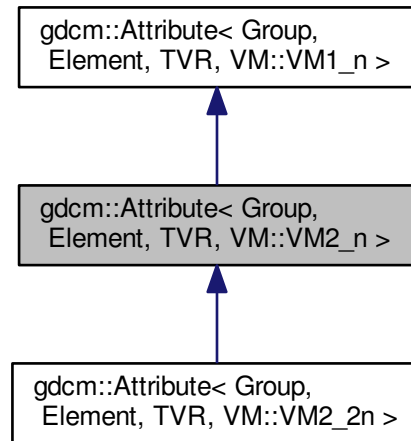
The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

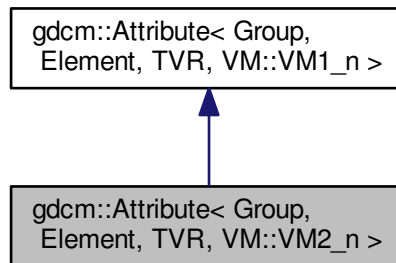
27.23 gdcm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference

```
#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

27.23.1 Member Function Documentation

27.23.1.1 `template<uint16_t Group, uint16_t Element, int TVR> VM gdcM::Attribute< Group, Element, TVR, VM::VM2_n >::GetVM() const [inline]`

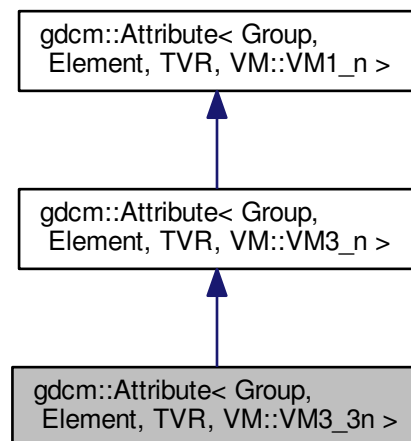
The documentation for this class was generated from the following file:

- [gdcMAttribute.h](#)

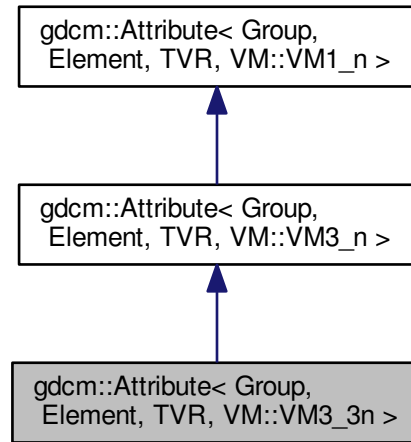
27.24 gdcM::Attribute< Group, Element, TVR, VM::VM3_3n > Class Template Reference

```
#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

27.24.1 Member Function Documentation

27.24.1.1 `template<uint16_t Group, uint16_t Element, int TVR> static VM gdcM::Attribute< Group, Element, TVR, VM::VM3_3n >::GetVM () [inline], [static]`

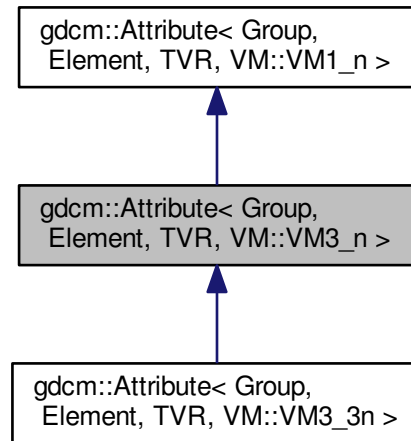
The documentation for this class was generated from the following file:

- [gdcMAttribute.h](#)

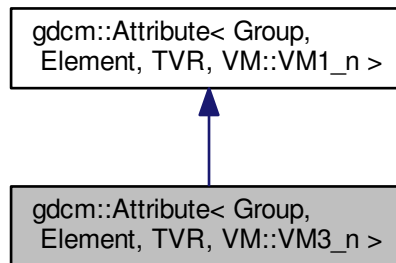
27.25 `gdcM::Attribute< Group, Element, TVR, VM::VM3_n >` Class Template Reference

```
#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

27.25.1 Member Function Documentation

```
27.25.1.1  template<uint16_t Group, uint16_t Element, int 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:

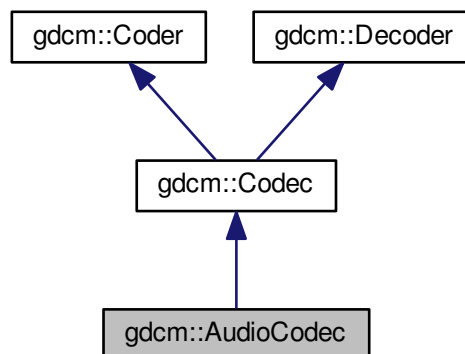
- [gdcMAttribute.h](#)

27.26 gdcM::AudioCodec Class Reference

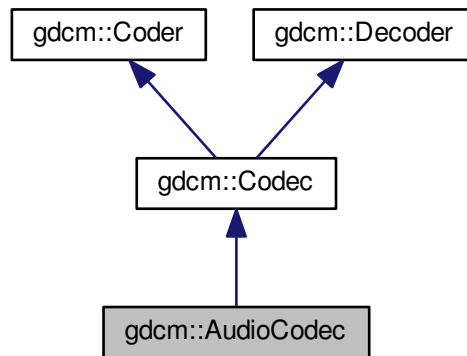
[AudioCodec.](#)

```
#include <gdcMAudioCodec.h>
```

Inheritance diagram for gdcM::AudioCodec:



Collaboration diagram for gdcm::AudioCodec:



Public Member Functions

- [AudioCodec](#) ()
- [~AudioCodec](#) ()
- bool [CanCode](#) ([TransferSyntax](#) const &) const
Return whether this coder support this transfer syntax (can code it)
- bool [CanDecode](#) ([TransferSyntax](#) const &) const
Return whether this decoder support this transfer syntax (can decode it)
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os)
Decode.

Additional Inherited Members

27.26.1 Detailed Description

[AudioCodec](#).

27.26.2 Constructor & Destructor Documentation

27.26.2.1 `gdcm::AudioCodec::AudioCodec ()`

27.26.2.2 `gdcm::AudioCodec::~~AudioCodec ()`

27.26.3 Member Function Documentation

27.26.3.1 `bool gdcm::AudioCodec::CanCode (TransferSyntax const &) const` `[inline], [virtual]`

Return whether this coder support this transfer syntax (can code it)

Implements [gdcm::Coder](#).

27.26.3.2 `bool gdcm::AudioCodec::CanDecode (TransferSyntax const &) const` `[inline], [virtual]`

Return whether this decoder support this transfer syntax (can decode it)

Implements [gdcm::Decoder](#).

27.26.3.3 `bool gdcm::AudioCodec::Decode (DataElement const & , DataElement &)` `[virtual]`

Decode.

Reimplemented from [gdcm::Decoder](#).

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

- [gdcmAudioCodec.h](#)

27.27 gdcm::Base64 Class Reference

Class for [Base64](#).

```
#include <gdcmBase64.h>
```

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`)

27.27.1 Detailed Description

Class for [Base64](#).

27.27.2 Member Function Documentation

27.27.2.1 `static size_t gdcm::Base64::Decode (char * dst, size_t dlen, const char * src, size_t slen)` `[static]`

Decode a base64-formatted buffer.

Parameters

<code>dst</code>	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](#).

27.27.2.2 `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

27.27.2.3 `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](#).

27.27.2.4 `static size_t gdcm::Base64::GetEncodeLength (const char * src, size_t srclen) [static]`

Call this function to obtain the required buffer size

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

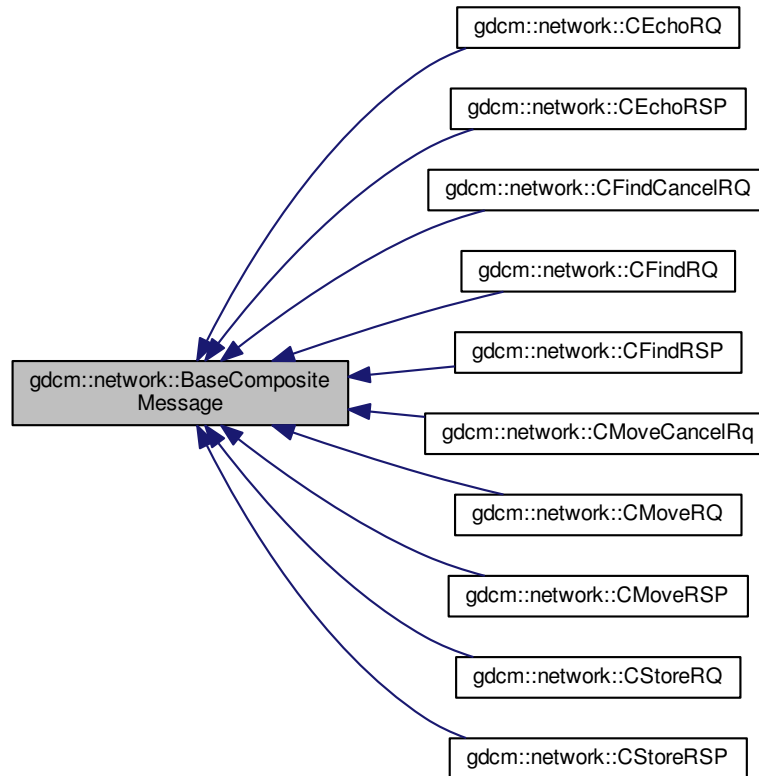
- [gdcmBase64.h](#)

27.28 gdcm::network::BaseCompositeMessage Class Reference

[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.

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

Inheritance diagram for `gdcm::network::BaseCompositeMessage`:



Public Member Functions

- virtual `~BaseCompositeMessage()`
- virtual `std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const Base←RootQuery *inRootQuery)=0`

27.28.1 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.

27.28.2 Constructor & Destructor Documentation

27.28.2.1 `virtual gdcm::network::BaseCompositeMessage::~BaseCompositeMessage () [inline], [virtual]`

27.28.3 Member Function Documentation

27.28.3.1 `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:

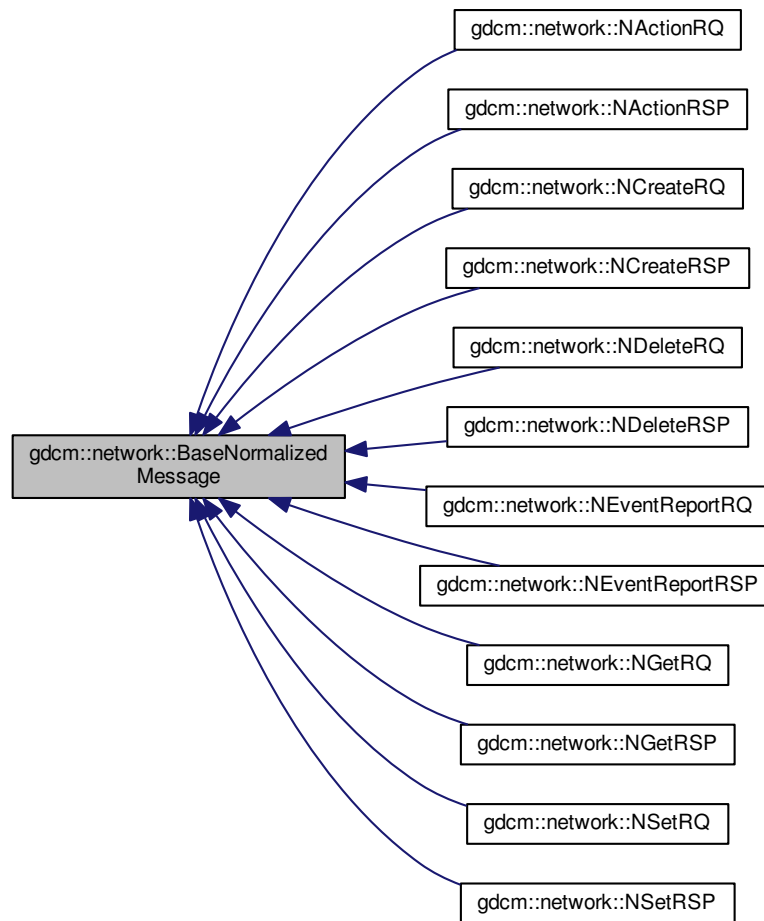
- [gdcmBaseCompositeMessage.h](#)

27.29 gdcm::network::BaseNormalizedMessage Class Reference

[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.

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

Inheritance diagram for `gdcmm::network::BaseNormalizedMessage`:



Public Member Functions

- virtual `~BaseNormalizedMessage()`
- virtual `std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const Base← Query *inQuery)=0`

27.29.1 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 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, [gdcmmNormalizedMessageFactory.h](#).

This is an abstract class. It cannot be instantiated on its own.

27.29.2 Constructor & Destructor Documentation

27.29.2.1 `virtual gdcmm::network::BaseNormalizedMessage::~BaseNormalizedMessage () [inline],[virtual]`

27.29.3 Member Function Documentation

27.29.3.1 `virtual std::vector<PresentationDataValue> gdcmm::network::BaseNormalizedMessage::ConstructPDV (const ULConnection & inConnection, const BaseQuery * inQuery) [pure virtual]`

Implemented in [gdcmm::network::NActionRQ](#), [gdcmm::network::NCreateRQ](#), [gdcmm::network::NDeleteRQ](#), [gdcmm::network::NEventReportRQ](#), [gdcmm::network::NGetRQ](#), and [gdcmm::network::NSetRQ](#).

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

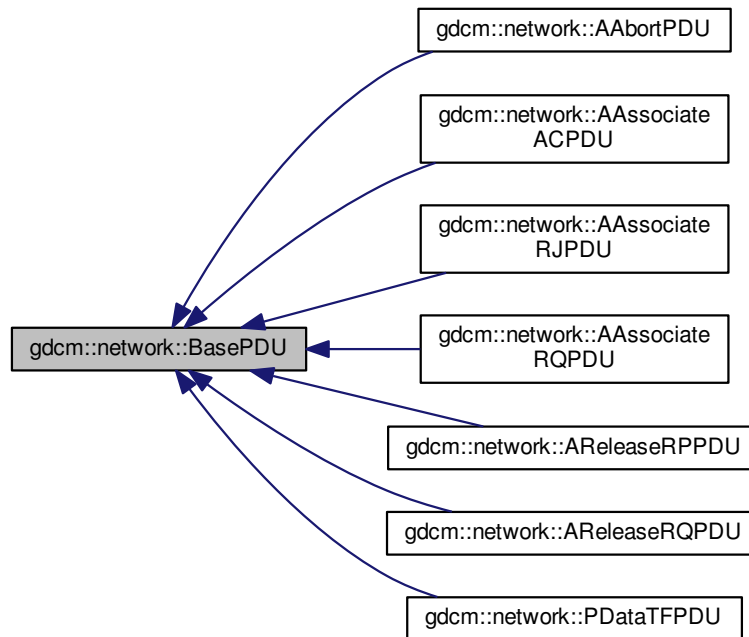
- [gdcmmBaseNormalizedMessage.h](#)

27.30 gdcmm::network::BasePDU Class Reference

[BasePDU](#) base class for PDUs.

```
#include <gdcmmBasePDU.h>
```

Inheritance diagram for `gdc::network::BasePDU`:



Public Member Functions

- virtual `~BasePDU()`
- 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

27.30.1 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

27.30.2 Constructor & Destructor Documentation

27.30.2.1 `virtual gdcm::network::BasePDU::~BasePDU () [inline], [virtual]`

27.30.3 Member Function Documentation

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

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

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

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

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

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

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

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

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

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

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

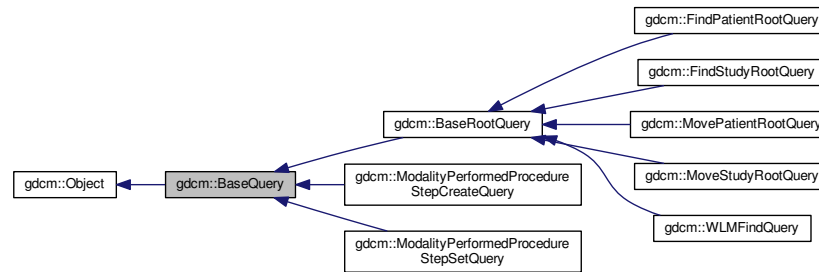
- [gdcmBasePDU.h](#)

27.31 gdcm::BaseQuery Class Reference

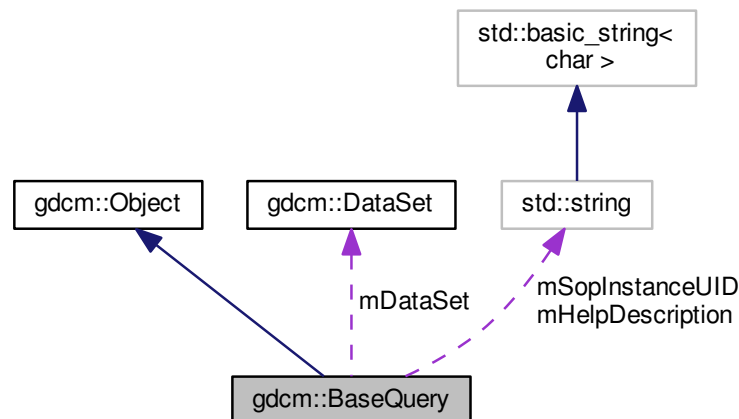
[BaseQuery](#) contains: a baseclass which will produce a dataset for all dimse messages.

```
#include <gdcmBaseQuery.h>
```

Inheritance diagram for `gdcm::BaseQuery`:



Collaboration diagram for `gdcm::BaseQuery`:



Public Member Functions

- virtual `~BaseQuery` ()
- void `AddQueryDataSet` (const `DataSet` &ds)
- virtual `UIDs::TSName GetAbstractSyntaxUID` () const =0
- `DataSet` const & `GetQueryDataSet` () const
Set/Get the internal representation of the query as a `DataSet`.
- `DataSet` & `GetQueryDataSet` ()
- `std::string` `GetSOPInstanceUID` () const
- void `Print` (`std::ostream` &os) const
- void `SetSearchParameter` (const `Tag` &inTag, const `std::string` &inValue)
- void `SetSearchParameter` (const `std::string` &inKeyword, 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 [mHelpDescription](#)
- std::string [mSopInstanceUID](#)

Friends

- class [QueryFactory](#)

27.31.1 Detailed Description

[BaseQuery](#) contains: a baseclass which will produce a dataset for all dimse messages.

27.31.2 Constructor & Destructor Documentation

27.31.2.1 `gdcm::BaseQuery::BaseQuery ()` [protected]

27.31.2.2 `virtual gdcm::BaseQuery::~~BaseQuery ()` [virtual]

27.31.3 Member Function Documentation

27.31.3.1 `void gdcm::BaseQuery::AddQueryDataSet (const DataSet & ds)`

27.31.3.2 `virtual UIDs::TSName gdcm::BaseQuery::GetAbstractSyntaxUID () const` [pure virtual]

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

27.31.3.3 `DataSet const& gdcm::BaseQuery::GetQueryDataSet () const`

Set/Get the internal representation of the query as a [DataSet](#).

27.31.3.4 **DataSet& gdcmm::BaseQuery::GetQueryDataSet ()**

27.31.3.5 **std::string gdcmm::BaseQuery::GetSOPInstanceUID () const** [inline]

27.31.3.6 **void gdcmm::BaseQuery::Print (std::ostream & os) const** [virtual]

Reimplemented from [gdcmm::Object](#).

27.31.3.7 **void gdcmm::BaseQuery::SetSearchParameter (const Tag & inTag, const DictEntry & inDictEntry, const std::string & inValue)** [protected]

27.31.3.8 **void gdcmm::BaseQuery::SetSearchParameter (const Tag & inTag, const std::string & inValue)**

27.31.3.9 **void gdcmm::BaseQuery::SetSearchParameter (const std::string & inKeyword, const std::string & inValue)**

27.31.3.10 **void gdcmm::BaseQuery::SetSOPInstanceUID (const std::string & iSopInstanceUID)** [inline]

27.31.3.11 **virtual bool gdcmm::BaseQuery::ValidateQuery (bool inStrict = true) const** [pure virtual]

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

27.31.3.12 **bool gdcmm::BaseQuery::ValidDataSet (const DataSet & dataSetToValid, const DataSet & dataSetReference) const** [protected]

27.31.3.13 **const std::ostream& gdcmm::BaseQuery::WriteHelpFile (std::ostream & os)**

27.31.3.14 **bool gdcmm::BaseQuery::WriteQuery (const std::string & inFileName)**

27.31.4 Friends And Related Function Documentation

27.31.4.1 **friend class QueryFactory** [friend]

27.31.5 Member Data Documentation

27.31.5.1 **DataSet gdcmm::BaseQuery::mDataSet** [protected]

27.31.5.2 **std::string gdcmm::BaseQuery::mHelpDescription** [protected]

27.31.5.3 **std::string gdcmm::BaseQuery::mSopInstanceUID** [protected]

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

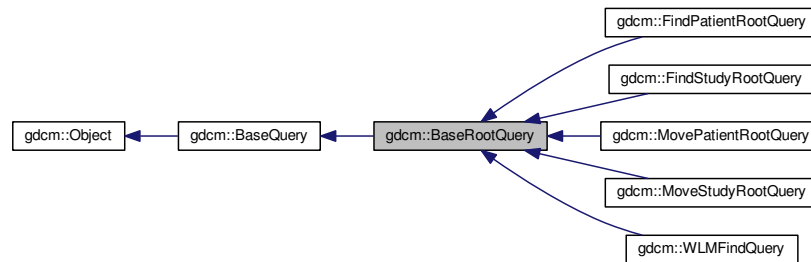
- [gdcmmBaseQuery.h](#)

27.32 gdcmm::BaseRootQuery Class Reference

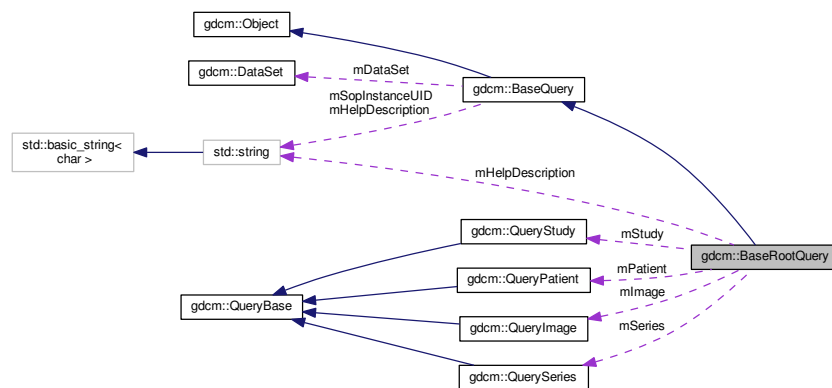
[BaseRootQuery](#) contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root.

```
#include <gdcMBaseRootQuery.h>
```

Inheritance diagram for gdcM::BaseRootQuery:



Collaboration diagram for gdcM::BaseRootQuery:



Public Member Functions

- virtual `~BaseRootQuery()`
- `EQueryLevel` `GetQueryLevelFromQueryRoot` (`ERootType` roottype)
- virtual `std::vector< Tag >` `GetTagListByLevel` (const `EQueryLevel` &inQueryLevel)=0
- virtual void `InitializeDataSet` (const `EQueryLevel` &inQueryLevel)=0
- virtual bool `ValidateQuery` (bool inStrict=true) const =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](#)

27.32.1 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.

27.32.2 Constructor & Destructor Documentation

27.32.2.1 `gdcm::BaseRootQuery::BaseRootQuery ()` [protected]

27.32.2.2 `virtual gdcm::BaseRootQuery::~~BaseRootQuery ()` [virtual]

27.32.3 Member Function Documentation

27.32.3.1 `static QueryBase* gdcm::BaseRootQuery::Construct (ERootType inRootType, EQueryLevel qlevel)` [static]

27.32.3.2 `EQueryLevel gdcm::BaseRootQuery::GetQueryLevelFromQueryRoot (ERootType roottype)`

27.32.3.3 `static int gdcm::BaseRootQuery::GetQueryLevelFromString (const char * str)` [static]

27.32.3.4 `static const char* gdcm::BaseRootQuery::GetQueryLevelString (EQueryLevel ql)` [static]

27.32.3.5 `virtual std::vector<Tag> gdcm::BaseRootQuery::GetTagListByLevel (const EQueryLevel & inQueryLevel)` [pure virtual]

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean `forFind` is true if the query is a find query, or false for a move query.

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

27.32.3.6 `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::FindPatientRootQuery](#), [gdcm::FindStudyRootQuery](#), [gdcm::MovePatientRootQuery](#), and [gdcm::MoveStudyRootQuery](#).

27.32.3.7 `virtual bool gdcm::BaseRootQuery::ValidateQuery (bool inStrict = true) const [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::FindStudyRootQuery](#), [gdcm::MovePatientRootQuery](#), [gdcm::MoveStudyRootQuery](#), [gdcm::WLMFindQuery](#), and [gdcm::FindPatientRootQuery](#).

27.32.4 Friends And Related Function Documentation

27.32.4.1 `friend class QueryFactory [friend]`

27.32.5 Member Data Documentation

27.32.5.1 `std::string gdcm::BaseRootQuery::mHelpDescription [protected]`

27.32.5.2 `QueryImage gdcm::BaseRootQuery::mImage [protected]`

27.32.5.3 `QueryPatient gdcm::BaseRootQuery::mPatient [protected]`

27.32.5.4 `ERootType gdcm::BaseRootQuery::mRootType [protected]`

27.32.5.5 `QuerySeries gdcm::BaseRootQuery::mSeries [protected]`

27.32.5.6 `QueryStudy gdcm::BaseRootQuery::mStudy [protected]`

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

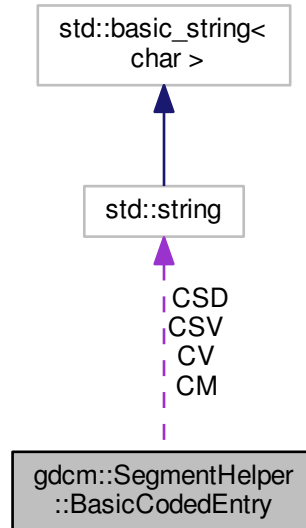
- [gdcmBaseRootQuery.h](#)

27.33 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](#)

27.33.1 Detailed Description

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

See also

PS 3.3 section 8.8.

27.33.2 Constructor & Destructor Documentation

27.33.2.1 `gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry () [inline]`

Constructor.

27.33.2.2 `gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry (const char * a_CV, const char * a_CSD, const char * a_CM) [inline]`

constructor which defines type 1 attributes.

27.33.2.3 `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.

27.33.3 Member Function Documentation

27.33.3.1 `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.
--------------------------------	--------------------------------

27.33.4 Member Data Documentation

27.33.4.1 `std::string gdcm::SegmentHelper::BasicCodedEntry::CM`

Coding Scheme [Version](#) attribute.

27.33.4.2 `std::string gdcm::SegmentHelper::BasicCodedEntry::CSD`

Code [Value](#) attribute.

27.33.4.3 `std::string gdcM::SegmentHelper::BasicCodedEntry::CSV`

Coding Scheme Designator attribute.

27.33.4.4 `std::string gdcM::SegmentHelper::BasicCodedEntry::CV`

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

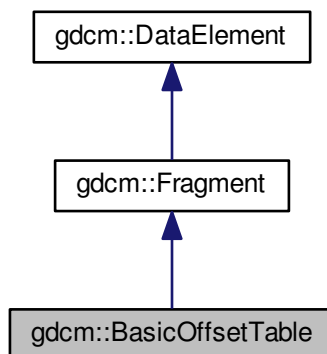
- [gdcMSegmentHelper.h](#)

27.34 `gdcM::BasicOffsetTable` Class Reference

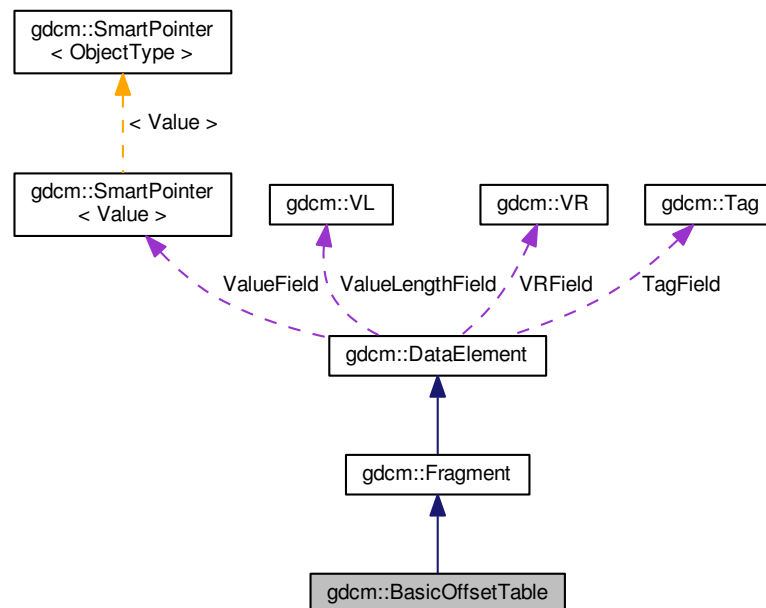
Class to represent a [BasicOffsetTable](#).

```
#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

27.34.1 Detailed Description

Class to represent a [BasicOffsetTable](#).

27.34.2 Constructor & Destructor Documentation

27.34.2.1 `gdcm::BasicOffsetTable::BasicOffsetTable ()` `[inline]`

27.34.3 Member Function Documentation

27.34.3.1 `template<typename TSwap> std::istream& gdcmm::BasicOffsetTable::Read (std::istream & is) [inline]`

References `gdcmmDebugMacro`.

27.34.4 Friends And Related Function Documentation

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

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

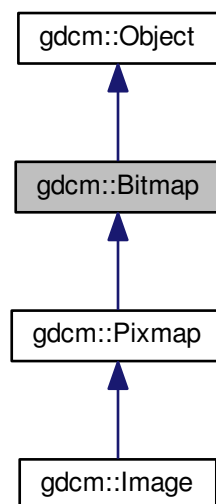
- [gdcmmBasicOffsetTable.h](#)

27.35 gdcmm::Bitmap Class Reference

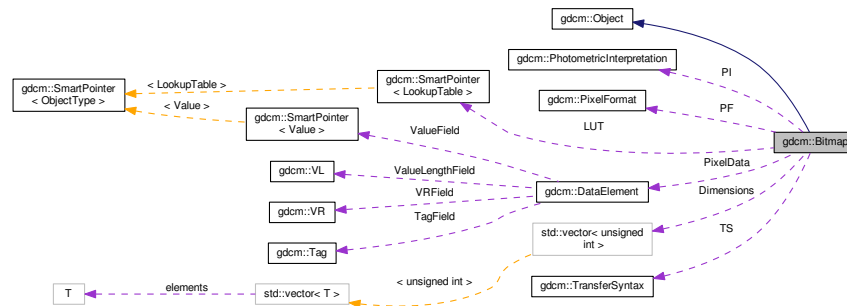
Bitmap class A bitmap based image. Used as parent for both `IconImage` and the main Pixel Data `Image` It does not contains any World Space information (IPP, IOP)

```
#include <gdcmmBitmap.h>
```

Inheritance diagram for `gdcmm::Bitmap`:



Collaboration diagram for gdcm::Bitmap:



Public Member Functions

- [Bitmap](#) ()
- [~Bitmap](#) ()
- virtual bool [AreOverlaysInPixelData](#) () const
- void [Clear](#) ()
- bool [GetBuffer](#) (char *buffer) const
Acces the raw data.
- unsigned long [GetBufferLength](#) () const
- unsigned int [GetColumns](#) () const
- const [DataElement](#) & [GetDataElement](#) () const
- [DataElement](#) & [GetDataElement](#) ()
- 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)...
- const [LookupTable](#) & [GetLUT](#) () const
- [LookupTable](#) & [GetLUT](#) ()
- bool [GetNeedByteSwap](#) () const
- 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
- const [PixelFormat](#) & [GetPixelFormat](#) () const
Get/Set PixelFormat.
- [PixelFormat](#) & [GetPixelFormat](#) ()
- 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

- 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.

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](#)

27.35.1 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](#).

27.35.2 Member Typedef Documentation

27.35.2.1 `typedef SmartPointer<LookupTable> gdcm::Bitmap::LUTPtr` `[protected]`

27.35.3 Constructor & Destructor Documentation

27.35.3.1 `gdcm::Bitmap::Bitmap ()`

27.35.3.2 `gdcm::Bitmap::~~Bitmap ()`

27.35.4 Member Function Documentation

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

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

27.35.4.2 `void gdcm::Bitmap::Clear ()`

27.35.4.3 `bool gdcm::Bitmap::ComputeLossyFlag ()` `[protected]`

27.35.4.4 `bool gdcm::Bitmap::GetBuffer (char * buffer) const`

Acces the raw data.

Examples:

[ConvertToQImage.cxx](#), [ReadMultiTimesException.cxx](#), and [threadgdcm.cxx](#).

27.35.4.5 `bool gdcm::Bitmap::GetBuffer2 (std::ostream & os) const` `[protected]`

27.35.4.6 `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](#).

27.35.4.7 `unsigned int gdcm::Bitmap::GetColumns () const [inline]`

27.35.4.8 `const DataElement& gdcm::Bitmap::GetDataElement () const [inline]`

Examples:

[ExtractIconFromFile.cxx](#).

27.35.4.9 `DataElement& gdcm::Bitmap::GetDataElement () [inline]`

27.35.4.10 `unsigned int gdcm::Bitmap::GetDimension (unsigned int idx) const`

27.35.4.11 `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](#).

27.35.4.12 `const LookupTable& gdcm::Bitmap::GetLUT () const [inline]`

Examples:

[ExtractIconFromFile.cxx](#).

27.35.4.13 `LookupTable& gdcm::Bitmap::GetLUT () [inline]`

27.35.4.14 `bool gdcm::Bitmap::GetNeedByteSwap () const [inline]`

27.35.4.15 `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](#).

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

return the photometric interpretation

Examples:

[ConvertToQImage.cxx](#), [ExtractIconFromFile.cxx](#), and [HelloVizWorld.cxx](#).

27.35.4.17 `const PixelFormat& gdcm::Bitmap::GetPixelFormat () const [inline]`

Get/Set [PixelFormat](#).

Examples:

[ConvertToQImage.cxx](#), [ExtractIconFromFile.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenFakeImage.cxx](#), [GetJPEGSamplePrecision.cxx](#), and [threadgdcm.cxx](#).

27.35.4.18 `PixelFormat& gdcm::Bitmap::GetPixelFormat () [inline]`

27.35.4.19 `unsigned int gdcm::Bitmap::GetPlanarConfiguration () const`

return the planar configuration

27.35.4.20 `unsigned int gdcm::Bitmap::GetRows () const [inline]`

27.35.4.21 `const TransferSyntax& gdcm::Bitmap::GetTransferSyntax () const [inline]`

Examples:

[ExtractIconFromFile.cxx](#).

27.35.4.22 `bool gdcm::Bitmap::IsEmpty () const [inline]`

27.35.4.23 `bool gdcm::Bitmap::IsLossy () const`

Return whether or not the image was compressed using a lossy compressor or not.

27.35.4.24 `bool gdcm::Bitmap::IsTransferSyntaxCompatible (TransferSyntax const & ts) const`

27.35.4.25 `void gdcm::Bitmap::Print (std::ostream &) const [virtual]`

Reimplemented from [gdcm::Object](#).

Reimplemented in [gdcm::Image](#), and [gdcm::Pixmap](#).

Examples:

[ExtractIconFromFile.cxx](#).

27.35.4.26 `void gdcm::Bitmap::SetColumns (unsigned int col) [inline]`

27.35.4.27 `void gdcm::Bitmap::SetDataElement (DataElement const & de) [inline]`

Examples:

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

27.35.4.28 void `gdcmm::Bitmap::SetDimension` (unsigned int *idx*, unsigned int *dim*)

Examples:

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

27.35.4.29 void `gdcmm::Bitmap::SetDimensions` (const unsigned int *dims*[3])

Examples:

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

27.35.4.30 void `gdcmm::Bitmap::SetLossyFlag` (bool *f*) [inline]

Specifically set that the image was compressed using a lossy compression mechanism.

27.35.4.31 void `gdcmm::Bitmap::SetLUT` (`LookupTable` const & *lut*) [inline]

Set/Get LUT.

27.35.4.32 void `gdcmm::Bitmap::SetNeedByteSwap` (bool *b*) [inline]

27.35.4.33 void `gdcmm::Bitmap::SetNumberOfDimensions` (unsigned int *dim*)

Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [GenFakelImage.cxx](#), [GetSubSequenceData.cxx](#), and [iU22tomultisc.cxx](#).

27.35.4.34 void `gdcmm::Bitmap::SetPhotometricInterpretation` (`PhotometricInterpretation` const & *pi*)

Examples:

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

27.35.4.35 void `gdcmm::Bitmap::SetPixelFormat` (`PixelFormat` const & *pf*) [inline]

Examples:

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

References `gdcmm::PixelFormat::Validate()`.

27.35.4.36 void `gdcmm::Bitmap::SetPlanarConfiguration` (unsigned int *pc*)

Warning

you need to call `SetPixelFormat` first (before `SetPlanarConfiguration`) for consistency checking

27.35.4.37 void gdcm::Bitmap::SetRows (unsigned int *rows*) [inline]

27.35.4.38 void gdcm::Bitmap::SetTransferSyntax (TransferSyntax const & *ts*) [inline]

Transfer syntax.

Examples:

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

27.35.4.39 bool gdcm::Bitmap::TryJPEG2000Codec (char * *buffer*, bool & *lossyflag*) const [protected]

27.35.4.40 bool gdcm::Bitmap::TryJPEG2000Codec2 (std::ostream & *os*) const [protected]

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

27.35.4.42 bool gdcm::Bitmap::TryJPEGCodec2 (std::ostream & *os*) const [protected]

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

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

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

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

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

27.35.5 Friends And Related Function Documentation

27.35.5.1 friend class ImageChangeTransferSyntax [friend]

27.35.5.2 friend class PixmapReader [friend]

27.35.6 Member Data Documentation

27.35.6.1 std::vector<unsigned int> gdcm::Bitmap::Dimensions [protected]

27.35.6.2 bool gdcm::Bitmap::LossyFlag [protected]

27.35.6.3 LUTPtr gdcm::Bitmap::LUT [protected]

27.35.6.4 bool gdcm::Bitmap::NeedByteSwap [protected]

27.35.6.5 unsigned int gdcm::Bitmap::NumberOfDimensions [protected]

27.35.6.6 PixelFormat gdcm::Bitmap::PF [protected]

27.35.6.7 PhotometricInterpretation gdcm::Bitmap::PI [protected]

27.35.6.8 DataElement gdcm::Bitmap::PixelData [protected]

27.35.6.9 `unsigned int gdcm::Bitmap::PlanarConfiguration` [protected]

27.35.6.10 `TransferSyntax gdcm::Bitmap::TS` [protected]

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

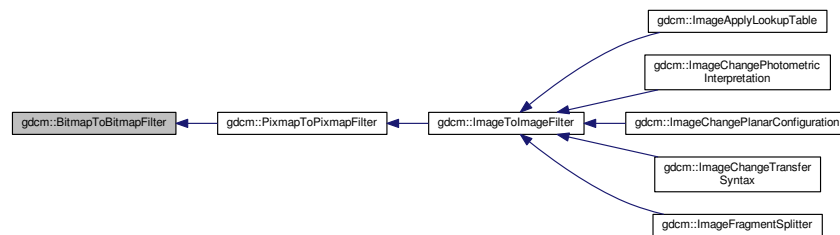
- [gdcmBitmap.h](#)

27.36 gdcm::BitmapToBitmapFilter Class Reference

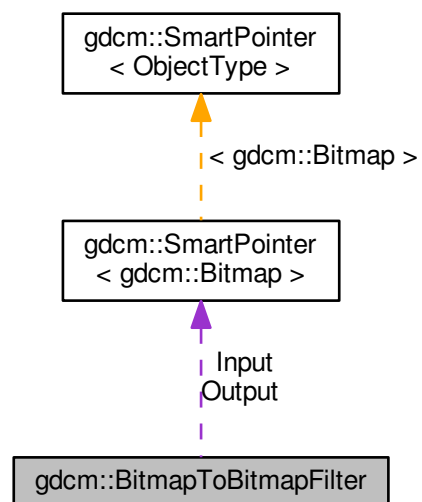
[BitmapToBitmapFilter](#) class Super class for all filter taking an image and producing an output image.

```
#include <gdcmBitmapToBitmapFilter.h>
```

Inheritance diagram for `gdcm::BitmapToBitmapFilter`:



Collaboration diagram for `gdcm::BitmapToBitmapFilter`:



Public Member Functions

- [BitmapToBitmapFilter \(\)](#)
- [~BitmapToBitmapFilter \(\)](#)
- 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](#)

27.36.1 Detailed Description

[BitmapToBitmapFilter](#) class Super class for all filter taking an image and producing an output image.

27.36.2 Constructor & Destructor Documentation

27.36.2.1 `gdcm::BitmapToBitmapFilter::BitmapToBitmapFilter ()`

27.36.2.2 `gdcm::BitmapToBitmapFilter::~~BitmapToBitmapFilter ()` `[inline]`

27.36.3 Member Function Documentation

27.36.3.1 `const Bitmap& gdcm::BitmapToBitmapFilter::GetOutput () const` `[inline]`

Get Output image.

27.36.3.2 `const Bitmap& gdcm::BitmapToBitmapFilter::GetOutputAsBitmap () const`

27.36.3.3 `void gdcm::BitmapToBitmapFilter::SetInput (const Bitmap & image)`

Set input image.

Examples:

[CompressImage.cxx](#).

27.36.4 Member Data Documentation

27.36.4.1 `SmartPointer<Bitmap> gdcm::BitmapToBitmapFilter::Input` `[protected]`

27.36.4.2 `SmartPointer<Bitmap> gdcm::BitmapToBitmapFilter::Output` `[protected]`

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

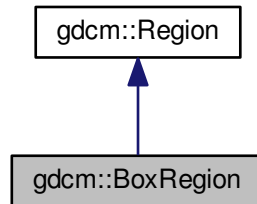
- [gdcmBitmapToBitmapFilter.h](#)

27.37 gdcm::BoxRegion Class Reference

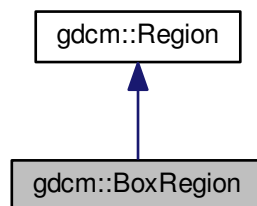
Class for manipulation box region This is a very simple implementation of the [Region](#) class. It only support 3D box type region. It assumes the 3D Box does not have a tilt Origin is as (0,0,0)

```
#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](#) ()
- [size_t Area](#) () const
compute the area
- [Region * Clone](#) () const
- [BoxRegion ComputeBoundingBox](#) ()
Return the Axis-Aligned minimum bounding box for all regions.

- bool [Empty](#) () const
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
return whether this is valid domain
- void [operator=](#) (const [BoxRegion](#) &)
- void [Print](#) (std::ostream &os=std::cout) const
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](#).

27.37.1 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)

27.37.2 Constructor & Destructor Documentation

27.37.2.1 `gdcm::BoxRegion::BoxRegion ()`

27.37.2.2 `gdcm::BoxRegion::~~BoxRegion ()`

27.37.2.3 `gdcm::BoxRegion::BoxRegion (const BoxRegion &)`

copy/cstor and al.

27.37.3 Member Function Documentation

27.37.3.1 `size_t gdcm::BoxRegion::Area () const` `[virtual]`

compute the area

Implements [gdcm::Region](#).

27.37.3.2 `static BoxRegion gdcM::BoxRegion::BoundingBox (BoxRegion const & b1, BoxRegion const & b2)`
`[static]`

Helper class to compute the bounding box of two [BoxRegion](#).

27.37.3.3 `Region* gdcM::BoxRegion::Clone () const` `[virtual]`

Implements [gdcM::Region](#).

27.37.3.4 `BoxRegion gdcM::BoxRegion::ComputeBoundingBox ()` `[virtual]`

Return the Axis-Aligned minimum bounding box for all regions.

Implements [gdcM::Region](#).

27.37.3.5 `bool gdcM::BoxRegion::Empty () const` `[virtual]`

return whether this domain is empty:

Implements [gdcM::Region](#).

27.37.3.6 `unsigned int gdcM::BoxRegion::GetXMax () const`

27.37.3.7 `unsigned int gdcM::BoxRegion::GetXMin () const`

Get domain.

27.37.3.8 `unsigned int gdcM::BoxRegion::GetYMax () const`

27.37.3.9 `unsigned int gdcM::BoxRegion::GetYMin () const`

27.37.3.10 `unsigned int gdcM::BoxRegion::GetZMax () const`

27.37.3.11 `unsigned int gdcM::BoxRegion::GetZMin () const`

27.37.3.12 `bool gdcM::BoxRegion::IsValid () const` `[virtual]`

return whether this is valid domain

Implements [gdcM::Region](#).

27.37.3.13 `void gdcM::BoxRegion::operator= (const BoxRegion &)`

27.37.3.14 `void gdcM::BoxRegion::Print (std::ostream & os = std::cout) const` `[virtual]`

Print.

Reimplemented from [gdcM::Region](#).

27.37.3.15 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](#)

27.38 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](#) ()

27.38.1 Detailed Description

[ByteBuffer](#).

Detailed description here

Note

looks like a std::streambuf or std::filebuf class with the get and peek pointer

27.38.2 Constructor & Destructor Documentation

27.38.2.1 gdcm::ByteBuffer::ByteBuffer () [inline]

27.38.3 Member Function Documentation

27.38.3.1 char* gdcm::ByteBuffer::Get (int *len*) [inline]

27.38.3.2 const char* gdcm::ByteBuffer::GetStart () const [inline]

27.38.3.3 void gdcm::ByteBuffer::ShiftEnd (int *len*) [inline]

27.38.3.4 void gdcm::ByteBuffer::UpdatePosition () [inline]

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

- [gdcmByteBuffer.h](#)

27.39 `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](#) ()

27.39.1 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.](#)

27.39.2 Member Function Documentation

27.39.2.1 `template<class T> static void gdcm::ByteSwap< T >::Swap (T & p) [static]`

27.39.2.2 `template<class T> static void gdcm::ByteSwap< T >::SwapFromSwapCodeIntoSystem (T & p, SwapCode const & sc) [static]`

Examples:

[TestByteSwap.cxx.](#)

27.39.2.3 `template<class T> static void gdcm::ByteSwap< T >::SwapRange (T * p, unsigned int num) [static]`

27.39.2.4 `template<class T> static void gdcm::ByteSwap< T >::SwapRangeFromSwapCodeIntoSystem (T * p, SwapCode const & sc, std::streamoff num) [static]`

Examples:

[TestByteSwap.cxx.](#)

27.39.2.5 `template<class T> static bool gdcm::ByteSwap< T >::SystemIsBigEndian () [static]`

Query the machine Endian-ness.

27.39.2.6 `template<class T> static bool gdcm::ByteSwap<T>::SystemIsLittleEndian () [static]`

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

- [gdcmByteSwap.h](#)

27.40 gdcm::ByteSwapFilter Class Reference

[ByteSwapFilter](#) In place byte-swapping of a dataset FIXME: FL status ??

```
#include <gdcmByteSwapFilter.h>
```

Public Member Functions

- [ByteSwapFilter](#) ([DataSet](#) &ds)
- [~ByteSwapFilter](#) ()
- bool [ByteSwap](#) ()
- void [SetByteSwapTag](#) (bool b)

27.40.1 Detailed Description

[ByteSwapFilter](#) In place byte-swapping of a dataset FIXME: FL status ??

27.40.2 Constructor & Destructor Documentation

27.40.2.1 `gdcm::ByteSwapFilter::ByteSwapFilter (DataSet & ds) [inline]`

27.40.2.2 `gdcm::ByteSwapFilter::~~ByteSwapFilter ()`

27.40.3 Member Function Documentation

27.40.3.1 `bool gdcm::ByteSwapFilter::ByteSwap ()`

27.40.3.2 `void gdcm::ByteSwapFilter::SetByteSwapTag (bool b) [inline]`

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

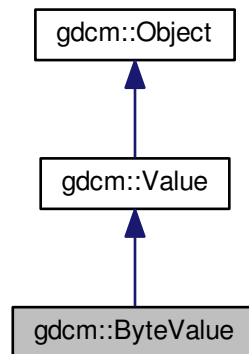
- [gdcmByteSwapFilter.h](#)

27.41 gdcm::ByteValue Class Reference

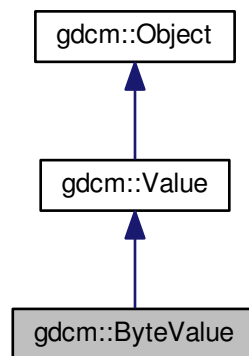
Class to represent binary value (array of bytes)

```
#include <gdcmByteValue.h>
```

Inheritance diagram for `gdcm::ByteValue`:



Collaboration diagram for `gdcm::ByteValue`:



Public Member Functions

- `ByteValue` (`const char *array=0`, `VL const &vl=0`)
- `ByteValue` (`std::vector< char > &v`)
- `~ByteValue` ()
- `void Append` (`ByteValue const &bv`)
- `void Clear` ()
- `VL ComputeLength` () `const`
- `void Fill` (`char c`)

- bool [GetBuffer](#) (char *buffer, unsigned long length) const
- [VL GetLength](#) () const
- const char * [GetPointer](#) () const
- bool [IsEmpty](#) () const
- bool [IsPrintable](#) (VL length) const

Checks whether a 'ByteValue' is printable or not (in order to avoid corrupting the terminal of invocation when printing) / don't think this function is working since it does not handle UNICODE or character set...

- [operator const std::vector< char > &](#) () const
- [ByteValue & operator=](#) (const [ByteValue](#) &val)
- bool [operator==](#) (const [ByteValue](#) &val) const
- bool [operator==](#) (const [Value](#) &val) const
- 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 , typename TType >
std::istream & [Read](#) (std::istream &is, bool readvalues=true)
- template<typename TSwap >
std::istream & [Read](#) (std::istream &is)
- void [SetLength](#) ([VL](#) vl)
- 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
- void [SetLengthOnly](#) ([VL](#) vl)

27.41.1 Detailed Description

Class to represent binary value (array of bytes)

Note

Examples:

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilips←ECHO.cxx](#), [DumpToshibaDTI.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetSubSequenceData.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), and [rle2img.cxx](#).

27.41.2 Constructor & Destructor Documentation

27.41.2.1 `gdcmm::ByteValue::ByteValue (const char * array = 0, VL const & vl = 0) [inline]`

References [gdcmmDebugMacro](#).

27.41.2.2 `gdcmm::ByteValue::ByteValue (std::vector< char > & v) [inline]`

Warning

casting to `uint32_t`

27.41.2.3 `gdcmm::ByteValue::~~ByteValue () [inline]`

27.41.3 Member Function Documentation

27.41.3.1 `void gdcmm::ByteValue::Append (ByteValue const & bv)`

27.41.3.2 `void gdcmm::ByteValue::Clear () [inline],[virtual]`

Implements [gdcmm::Value](#).

27.41.3.3 `VL gdcmm::ByteValue::ComputeLength () const [inline]`

Referenced by `gdcmm::Fragment::Write()`.

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

Examples:

[DuplicatePCDE.cxx](#).

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

Examples:

[FixJAIBugJPEGLS.cxx](#).

27.41.3.6 `VL gdcmm::ByteValue::GetLength () const [inline],[virtual]`

Implements [gdcmm::Value](#).

Examples:

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetSubSequenceData.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), and [rle2img.cxx](#).

Referenced by `gdcmm::operator<<()`, `gdcmm::SequenceOfFragments::ReadValue()`, `gdcmm::Element< VR::OB, VM::VM1_n >::Set()`, `gdcmm::Element< TVR, VM::VM1_n >::Set()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValue()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap()`, `gdcmm::Element< VR::OB, VM::VM1_n >::SetNoSwap()`, `gdcmm::Element< TVR, VM::VM1_n >::SetNoSwap()`, and `gdcmm::Fragment::Write()`.

27.41.3.7 `const char* gdcm::ByteValue::GetPointer () const [inline]`

Examples:

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [GetSubSequenceData.cxx](#), [MrProtocol.cxx](#), [pmsct_rgb1.cxx](#), [ReadExplicitLengthSIVR.cxx](#), [ReadGEMSSDO.cxx](#), and [rle2img.cxx](#).

Referenced by `gdcm::operator<<()`, `gdcm::SequenceOfFragments::ReadValue()`, `gdcm::Element< VR::OB, VM::VM1_n >::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< VR::OB, VM::VM1_n >::SetNoSwap()`, and `gdcm::Element< TVR, VM::VM1_n >::SetNoSwap()`.

27.41.3.8 `bool gdcm::ByteValue::IsEmpty () const [inline]`

27.41.3.9 `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...

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

27.41.3.11 `ByteValue& gdcm::ByteValue::operator= (const ByteValue & val) [inline]`

27.41.3.12 `bool gdcm::ByteValue::operator== (const ByteValue & val) const [inline]`

27.41.3.13 `bool gdcm::ByteValue::operator== (const Value & val) const [inline], [virtual]`

Implements [gdcm::Value](#).

27.41.3.14 `void gdcm::ByteValue::Print (std::ostream & os) const [inline], [protected], [virtual]`

Reimplemented from [gdcm::Object](#).

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

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

27.41.3.17 `void gdcm::ByteValue::PrintGroupLength (std::ostream & os) [inline]`

27.41.3.18 `void gdcm::ByteValue::PrintHex (std::ostream & os, VL maxlength) const`

27.41.3.19 `void gdcm::ByteValue::PrintHexXML (std::ostream & os) const`

27.41.3.20 `void gdcm::ByteValue::PrintPNXML (std::ostream & os) const`

To Print Values in Native DICOM format

27.41.3.21 `template<typename TSwap , typename TType > std::istream& gdcM::ByteValue::Read (std::istream & is, bool readvalues = true) [inline]`

27.41.3.22 `template<typename TSwap > std::istream& gdcM::ByteValue::Read (std::istream & is) [inline]`

27.41.3.23 `void gdcM::ByteValue::SetLength (VL vl) [virtual]`

Implements [gdcM::Value](#).

27.41.3.24 `void gdcM::ByteValue::SetLengthOnly (VL vl) [inline],[protected],[virtual]`

Reimplemented from [gdcM::Value](#).

27.41.3.25 `template<typename TSwap , typename TType > std::ostream const& gdcM::ByteValue::Write (std::ostream & os) const [inline]`

Referenced by `gdcM::Fragment::Write()`.

27.41.3.26 `template<typename TSwap > std::ostream const& gdcM::ByteValue::Write (std::ostream & os) const [inline]`

27.41.3.27 `bool gdcM::ByteValue::WriteBuffer (std::ostream & os) const [inline]`

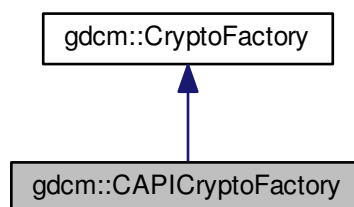
The documentation for this class was generated from the following file:

- [gdcMByteValue.h](#)

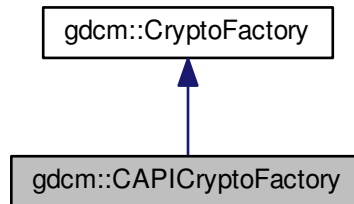
27.42 gdcM::CAPICryptoFactory Class Reference

```
#include <gdcMCAPICryptoFactory.h>
```

Inheritance diagram for `gdcM::CAPICryptoFactory`:



Collaboration diagram for gdcm::CAPICryptoFactory:



Public Member Functions

- [CAPICryptoFactory](#) ([CryptoLib](#) id)
- [CryptographicMessageSyntax](#) * [CreateCMSProvider](#) ()

Additional Inherited Members

27.42.1 Constructor & Destructor Documentation

27.42.1.1 `gdcm::CAPICryptoFactory::CAPICryptoFactory (CryptoLib id)`

27.42.2 Member Function Documentation

27.42.2.1 `CryptographicMessageSyntax* gdcm::CAPICryptoFactory::CreateCMSProvider ()` [[virtual](#)]

Implements [gdcm::CryptoFactory](#).

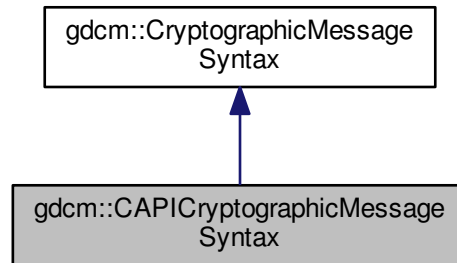
The documentation for this class was generated from the following file:

- [gdcmCAPICryptoFactory.h](#)

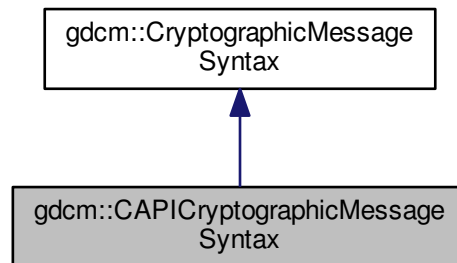
27.43 gdcm::CAPICryptographicMessageSyntax Class Reference

```
#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

27.43.1 Constructor & Destructor Documentation

27.43.1.1 gdcM::CAPICryptographicMessageSyntax::CAPICryptographicMessageSyntax ()

27.43.1.2 gdcM::CAPICryptographicMessageSyntax::~~CAPICryptographicMessageSyntax ()

27.43.2 Member Function Documentation

27.43.2.1 bool gdcM::CAPICryptographicMessageSyntax::Decrypt (char * *output*, size_t & *outlen*, const char * *array*, size_t *len*) const [virtual]

decrypt content from a CMS envelopedData structure

Implements [gdcM::CryptographicMessageSyntax](#).

27.43.2.2 bool gdcM::CAPICryptographicMessageSyntax::Encrypt (char * *output*, size_t & *outlen*, const char * *array*, size_t *len*) const [virtual]

create a CMS envelopedData structure

Implements [gdcM::CryptographicMessageSyntax](#).

27.43.2.3 CipherTypes gdcM::CAPICryptographicMessageSyntax::GetCipherType () const [virtual]

Implements [gdcM::CryptographicMessageSyntax](#).

27.43.2.4 bool gdcM::CAPICryptographicMessageSyntax::GetInitialized () const [inline]

27.43.2.5 bool gdcM::CAPICryptographicMessageSyntax::ParseCertificateFile (const char * *filename*) [virtual]

Implements [gdcM::CryptographicMessageSyntax](#).

27.43.2.6 bool gdcM::CAPICryptographicMessageSyntax::ParseKeyFile (const char * *filename*) [virtual]

Implements [gdcM::CryptographicMessageSyntax](#).

27.43.2.7 void gdcM::CAPICryptographicMessageSyntax::SetCipherType (CipherTypes *type*)

27.43.2.8 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:

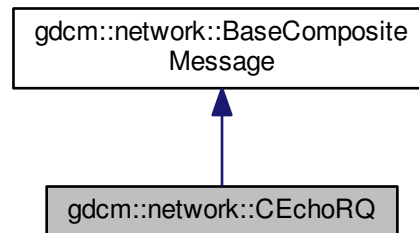
- [gdcMCAPICryptographicMessageSyntax.h](#)

27.44 gdcm::network::CEchoRQ Class Reference

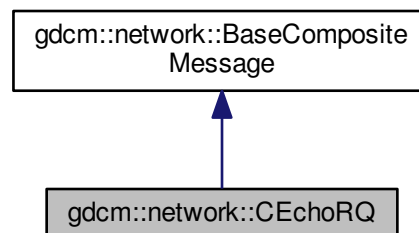
[CEchoRQ](#) this file defines the messages for the cecho action.

```
#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)

Public Attributes

- [UIComp AffectedSOPClassUID](#)
- `uint16_t` [MessageID](#)

27.44.1 Detailed Description

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

27.44.2 Member Function Documentation

27.44.2.1 `std::vector<PresentationDataValue> gdcmm::network::CEchoRQ::ConstructPDV (const ULConnection & inConnection, const BaseRootQuery * inRootQuery) [virtual]`

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

27.44.3 Member Data Documentation

27.44.3.1 `UIComp gdcmm::network::CEchoRQ::AffectedSOPClassUID`

27.44.3.2 `uint16_t gdcmm::network::CEchoRQ::MessageID`

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

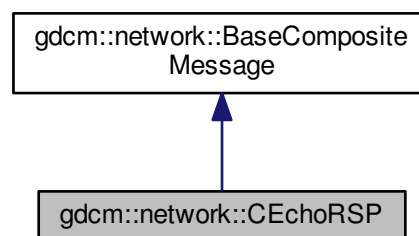
- [gdcmmCEchoMessages.h](#)
- [gdcmmDIMSE.h](#)

27.45 gdcmm::network::CEchoRSP Class Reference

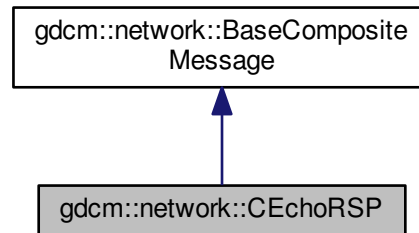
[CEchoRSP](#) this file defines the messages for the cecho action.

```
#include <gdcmmCEchoMessages.h>
```

Inheritance diagram for `gdcmm::network::CEchoRSP`:



Collaboration diagram for `gdcm::network::CEchoRSP`:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)`

27.45.1 Detailed Description

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

27.45.2 Member Function Documentation

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

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

- [gdcmCEchoMessages.h](#)

27.46 gdcm::network::CFind Class Reference

```
#include <gdcmDIMSE.h>
```

27.46.1 Detailed Description

PS 3.4 - 2009 [Table B.2-1](#) C-STORE STATUS

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

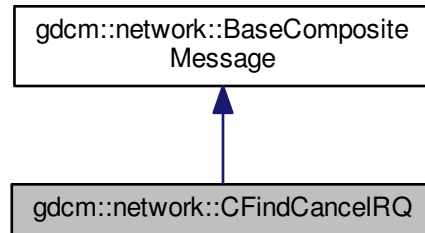
- [gdcmDIMSE.h](#)

27.47 gdcm::network::CFindCancelRQ Class Reference

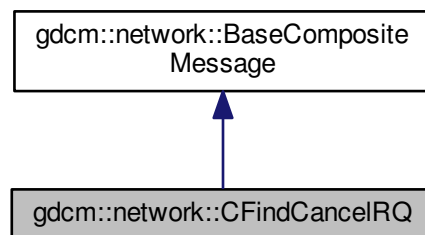
[CFindCancelRQ](#) this file defines the messages for the cfind action.

```
#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)

27.47.1 Detailed Description

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

27.47.2 Member Function Documentation

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

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

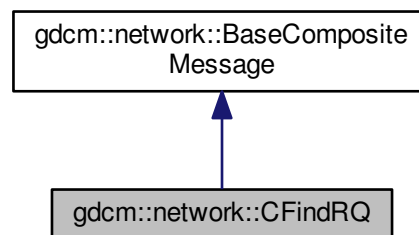
- [gdcmCFindMessages.h](#)

27.48 gdcm::network::CFindRQ Class Reference

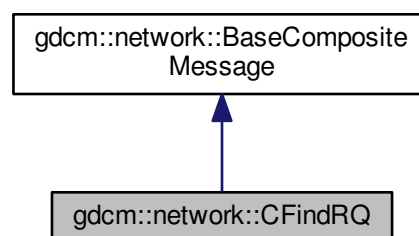
[CFindRQ](#) this file defines the messages for the cfind action.

```
#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)`

27.48.1 Detailed Description

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

27.48.2 Member Function Documentation

27.48.2.1 `std::vector<PresentationDataValue> gdcm::network::CFindRQ::ConstructPDV (const ULConnection & inConnection, const BaseRootQuery * inRootQuery) [virtual]`

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

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

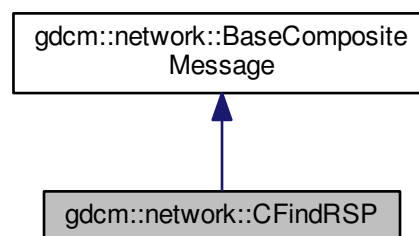
- [gdcmCFindMessages.h](#)

27.49 gdcm::network::CFindRSP Class Reference

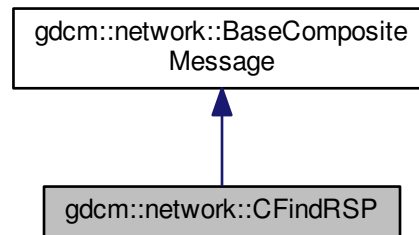
[CFindRSP](#) this file defines the messages for the cfind action.

```
#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)`

27.49.1 Detailed Description

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

27.49.2 Member Function Documentation

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

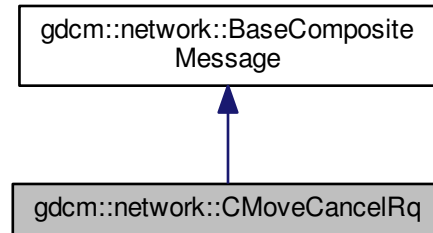
The documentation for this class was generated from the following file:

- [gdcmCFindMessages.h](#)

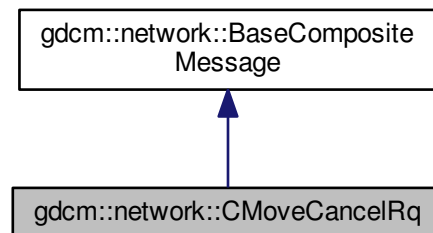
27.50 `gdcm::network::CMoveCancelRq` Class Reference

```
#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)`

27.50.1 Member Function Documentation

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

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

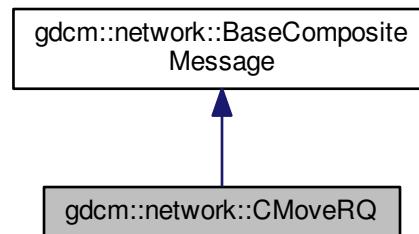
- [gdcmCMoveMessages.h](#)

27.51 gdcmm::network::CMoveRQ Class Reference

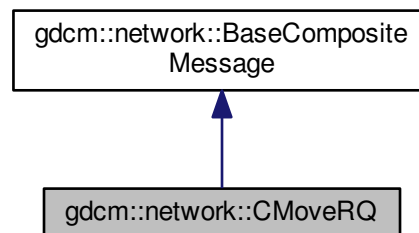
[CMoveRQ](#) this file defines the messages for the cmove action.

```
#include <gdcmmCMoveMessages.h>
```

Inheritance diagram for gdcmm::network::CMoveRQ:



Collaboration diagram for gdcmm::network::CMoveRQ:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseRootQuery](#) *inRootQuery)

27.51.1 Detailed Description

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

27.51.2 Member Function Documentation

27.51.2.1 `std::vector<PresentationDataValue> gdcm::network::CMoveRQ::ConstructPDV (const ULConnection & inConnection, const BaseRootQuery * inRootQuery) [virtual]`

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

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

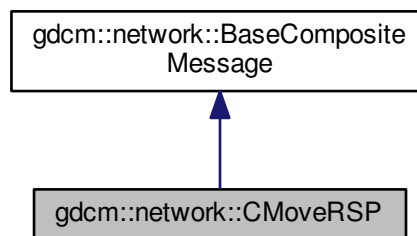
- [gdcmCMoveMessages.h](#)

27.52 gdcm::network::CMoveRSP Class Reference

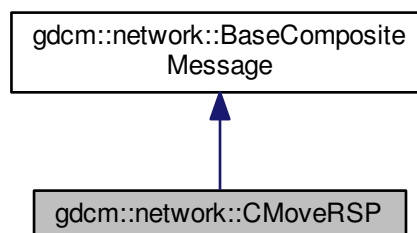
[CMoveRSP](#) this file defines the messages for the cmove action.

```
#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)`

27.52.1 Detailed Description

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

27.52.2 Member Function Documentation

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

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

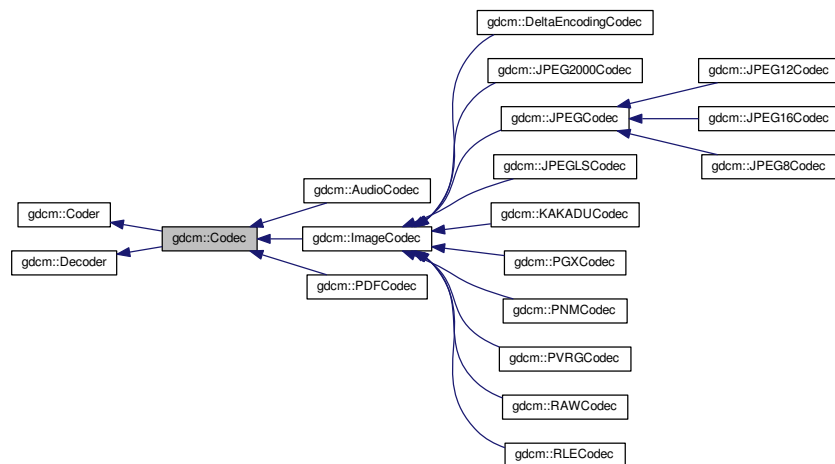
- [gdcmmCMoveMessages.h](#)

27.53 gdcmm::Codec Class Reference

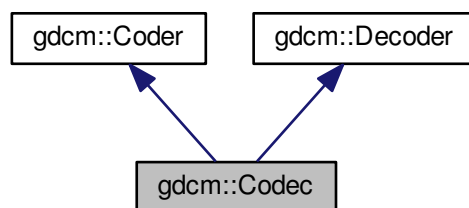
[Codec](#) class.

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

Inheritance diagram for `gdcmm::Codec`:



Collaboration diagram for gdcm::Codec:



Additional Inherited Members

27.53.1 Detailed Description

[Codec](#) class.

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

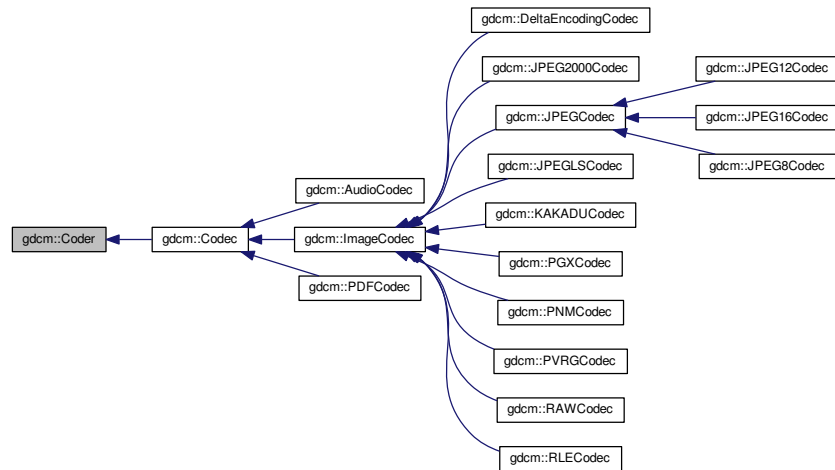
- [gdcmCodec.h](#)

27.54 gdcm::Coder Class Reference

[Coder](#).

```
#include <gdcmCoder.h>
```

Inheritance diagram for `gdcM::Coder`:



Public Member Functions

- virtual `~Coder` ()
- 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)

27.54.1 Detailed Description

`Coder`.

27.54.2 Constructor & Destructor Documentation

27.54.2.1 virtual `gdcM::Coder::~Coder` () [inline], [virtual]

27.54.3 Member Function Documentation

27.54.3.1 virtual bool `gdcM::Coder::CanCode` (`TransferSyntax` const &) const [pure virtual]

Return whether this coder support this transfer syntax (can code it)

Implemented in `gdcM::JPEGCodec`, `gdcM::RLECodec`, `gdcM::PVRGCodec`, `gdcM::JPEG2000Codec`, `gdcM::JPEGLSCodec`, `gdcM::ImageCodec`, `gdcM::PNMCodec`, `gdcM::PGXCodec`, `gdcM::KAKADUCodec`, `gdcM::RAWCodec`, `gdcM::AudioCodec`, and `gdcM::PDFCodec`.

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

Code.

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

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

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

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

- [gdcmCoder.h](#)

27.55 gdcm::CodeString Class Reference

[CodeString](#) This is an implementation of DICOM VR: CS The ctor will properly Trim so that operator== is correct.

```
#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 [value_type](#) *s)
- [CodeString](#) (const [value_type](#) *s, [size_type](#) n)
- [CodeString](#) (const [InternalClass](#) &s, [size_type](#) pos=0, [size_type](#) n=[InternalClass::npos](#))
- `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)`

27.55.1 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.

27.55.2 Member Typedef Documentation

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

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

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

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

27.55.2.5 `typedef InternalClass::iterator gdcm::CodeString::iterator`

27.55.2.6 `typedef InternalClass::pointer gdcm::CodeString::pointer`

27.55.2.7 `typedef InternalClass::reference gdcm::CodeString::reference`

27.55.2.8 `typedef InternalClass::reverse_iterator gdcm::CodeString::reverse_iterator`

27.55.2.9 `typedef InternalClass::size_type gdcm::CodeString::size_type`

27.55.2.10 `typedef InternalClass::value_type gdcm::CodeString::value_type`

27.55.3 Constructor & Destructor Documentation

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

[CodeString](#) constructors.

27.55.3.2 `gdcm::CodeString::CodeString (const value_type * s)` `[inline]`

27.55.3.3 `gdcm::CodeString::CodeString (const value_type * s, size_type n)` `[inline]`

27.55.3.4 `gdcm::CodeString::CodeString (const InternalClass & s, size_type pos = 0, size_type n = InternalClass::npos)` `[inline]`

27.55.4 Member Function Documentation

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

Return the full code string as `std::string`.

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

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

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

Return the size of the string.

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

27.55.5 Friends And Related Function Documentation

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

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

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

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

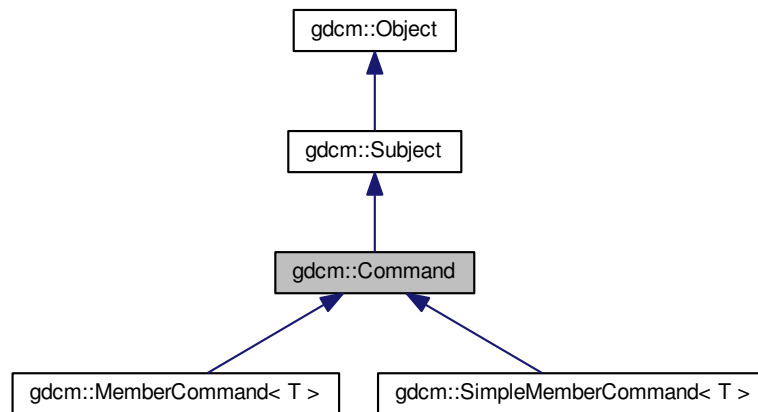
- [gdcmCodeString.h](#)

27.56 gdcm::Command Class Reference

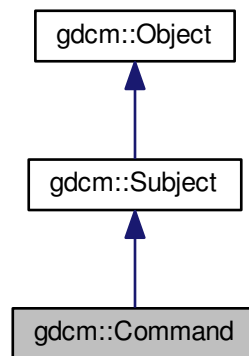
[Command](#) superclass for callback/observer methods.

```
#include <gdcmCommand.h>
```

Inheritance diagram for `gdc::Command`:



Collaboration diagram for `gdc::Command`:



Public Member Functions

- virtual void `Execute` (`Subject` *caller, const `Event` &event)=0
Abstract method that defines the action to be taken by the command.
- virtual void `Execute` (const `Subject` *caller, const `Event` &event)=0

Protected Member Functions

- `Command` ()

- [~Command\(\)](#)

27.56.1 Detailed Description

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

See also

[Subject](#)

27.56.2 Constructor & Destructor Documentation

27.56.2.1 `gdcm::Command::Command()` [protected]

27.56.2.2 `gdcm::Command::~~Command()` [protected]

27.56.3 Member Function Documentation

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

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

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

27.56.3.2 `virtual void gdcm::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 [gdcm::SimpleMemberCommand< T >](#), and [gdcm::MemberCommand< T >](#).

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

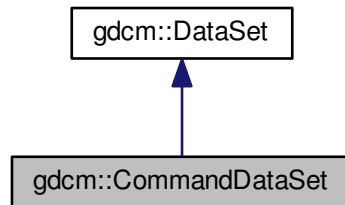
- [gdcmCommand.h](#)

27.57 gdcm::CommandDataSet Class Reference

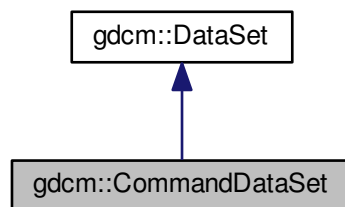
Class to represent a [Command DataSet](#).

```
#include <gdcmCommandDataSet.h>
```

Inheritance diagram for `gdcm::CommandDataSet`:



Collaboration diagram for `gdcm::CommandDataSet`:



Public Member Functions

- [CommandDataSet](#) ()
- [~CommandDataSet](#) ()
- 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

27.57.1 Detailed Description

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

See also

[DataSet](#)

27.57.2 Constructor & Destructor Documentation

27.57.2.1 `gdcm::CommandDataSet::CommandDataSet ()` `[inline]`

27.57.2.2 `gdcm::CommandDataSet::~~CommandDataSet ()` `[inline]`

27.57.3 Member Function Documentation

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

References `gdcmErrorMacro`, `gdcm::Tag::GetGroup()`, and `gdcm::DataElement::GetTag()`.

27.57.3.2 `std::istream& gdcm::CommandDataSet::Read (std::istream & is)`

Read.

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

References `gdcm::DataElement::GetTag()`.

27.57.3.4 `std::ostream& gdcm::CommandDataSet::Write (std::ostream & os) const`

Write.

27.57.4 Friends And Related Function Documentation

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

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

- [gdcmCommandDataSet.h](#)

27.58 gdcm::network::CompositeMessageFactory Class Reference

[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).

```
#include <gdcmCompositeMessageFactory.h>
```

Static Public Member Functions

- static std::vector< [PresentationDataValue](#) > [ConstructCEchoRQ](#) (const [ULConnection](#) &inConnection)
- static std::vector< [PresentationDataValue](#) > [ConstructCFindRQ](#) (const [ULConnection](#) &inConnection, const [BaseRootQuery](#) *inRootQuery)
- static std::vector< [PresentationDataValue](#) > [ConstructCMoveRQ](#) (const [ULConnection](#) &inConnection, const [BaseRootQuery](#) *inRootQuery)
- static std::vector< [PresentationDataValue](#) > [ConstructCStoreRQ](#) (const [ULConnection](#) &inConnection, const [File](#) &file, bool writeDataSet=true)
- static std::vector< [PresentationDataValue](#) > [ConstructCStoreRSP](#) (const [DataSet](#) *inDataSet, const [BasePDU](#) *inPC)

27.58.1 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).

27.58.2 Member Function Documentation

- 27.58.2.1 static std::vector<[PresentationDataValue](#)> gdcm::network::CompositeMessageFactory::ConstructCEchoRQ (const [ULConnection](#) & *inConnection*) [static]
- 27.58.2.2 static std::vector<[PresentationDataValue](#)> gdcm::network::CompositeMessageFactory::ConstructCFindRQ (const [ULConnection](#) & *inConnection*, const [BaseRootQuery](#) * *inRootQuery*) [static]
- 27.58.2.3 static std::vector<[PresentationDataValue](#)> gdcm::network::CompositeMessageFactory::ConstructCMoveRQ (const [ULConnection](#) & *inConnection*, const [BaseRootQuery](#) * *inRootQuery*) [static]
- 27.58.2.4 static std::vector<[PresentationDataValue](#)> gdcm::network::CompositeMessageFactory::ConstructCStoreRQ (const [ULConnection](#) & *inConnection*, const [File](#) & *file*, bool *writeDataSet* =true) [static]
- 27.58.2.5 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](#)

27.59 gdcm::CompositeNetworkFunctions Class Reference

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:

```
#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=NULL, const char *call=NULL)
- static bool [CFind](#) (const char *remote, uint16_t portno, const [BaseRootQuery](#) *query, std::vector< [DataSet](#) > &retDataSets, const char *aetitle=NULL, const char *call=NULL)
- static bool [CMove](#) (const char *remote, uint16_t portno, const [BaseRootQuery](#) *query, uint16_t portscp, const char *aetitle=NULL, const char *call=NULL, const char *outputdir=NULL)
- 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=NULL, const char *call=NULL)

27.59.1 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)

27.59.2 Member Typedef Documentation

27.59.2.1 typedef std::vector< [KeyValuePairType](#) > [gdcm::CompositeNetworkFunctions::KeyValuePairArrayType](#)

27.59.2.2 typedef std::pair< [Tag](#), std::string > [gdcm::CompositeNetworkFunctions::KeyValuePairType](#)

27.59.3 Member Function Documentation

27.59.3.1 `static bool gdcM::CompositeNetworkFunctions::CEcho (const char * remote, uint16_t portno, const char * aetitle = NULL, const char * call = NULL) [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.

27.59.3.2 static bool gdcmm::CompositeNetworkFunctions::CFind (const char * *remote*, uint16_t *portno*, const BaseRootQuery * *query*, std::vector< DataSet > & *retDataSets*, const char * *aetitle* = NULL, const char * *call* = NULL) [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.

27.59.3.3 static bool gdcmm::CompositeNetworkFunctions::CMove (const char * *remote*, uint16_t *portno*, const BaseRootQuery * *query*, uint16_t *portscp*, const char * *aetitle* = NULL, const char * *call* = NULL, const char * *outputdir* = NULL) [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.

27.59.3.4 static **BaseRootQuery*** **gdcmm::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)
------------------	--

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

Deprecated

27.59.3.6 static **bool** **gdcmm::CompositeNetworkFunctions::CStore** (const char * *remote*, **uint16_t** *portno*, const **Directory::FileNamesType** & *filenames*, const char * *aetitle* = **NULL**, const char * *call* = **NULL**) [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:

- [gdcmmCompositeNetworkFunctions.h](#)

27.60 gdcm::ConstCharWrapper Class Reference

Do not use me.

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

Public Member Functions

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

27.60.1 Detailed Description

Do not use me.

27.60.2 Constructor & Destructor Documentation

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

27.60.3 Member Function Documentation

27.60.3.1 [gdcm::ConstCharWrapper::operator const char * \(\)](#) const [inline]

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

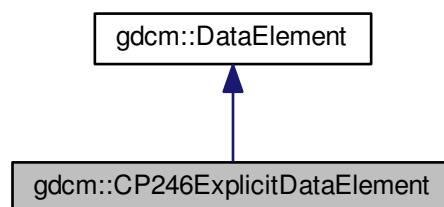
- [gdcmConstCharWrapper.h](#)

27.61 gdcm::CP246ExplicitDataElement Class Reference

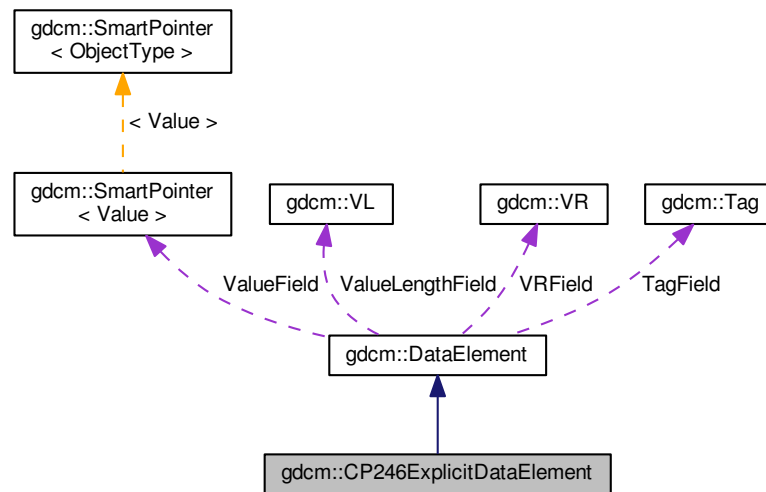
Class to read/write a [DataElement](#) as CP246Explicit Data [Element](#).

```
#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

27.61.1 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

27.61.2 Member Function Documentation

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

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

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

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

27.61.2.5 `template<typename TSwap> std::istream& gdcm::CP246ExplicitDataElement::ReadWithLength (std::istream & is, VL & length)`

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

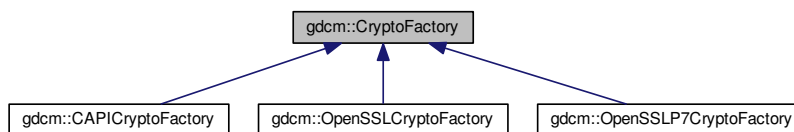
- [gdcmCP246ExplicitDataElement.h](#)

27.62 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 \(CryptoLib id\)](#)
- [CryptoFactory \(\)](#)
- [~CryptoFactory \(\)](#)

27.62.1 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)
- OPENSSLP7 (portable) By default the factory will try: CAPI if on windows OPENSSL if possible OPENSSLP7 when older OpenSSL is used.

27.62.2 Member Enumeration Documentation

27.62.2.1 enum `gdcm::CryptoFactory::CryptoLib`

Enumerator

DEFAULT
OPENSSL
CAPI
OPENSSLP7

27.62.3 Constructor & Destructor Documentation

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

27.62.3.2 `gdcm::CryptoFactory::CryptoFactory ()` `[inline]`, `[protected]`

27.62.3.3 `gdcm::CryptoFactory::~~CryptoFactory ()` `[inline]`, `[protected]`

27.62.4 Member Function Documentation

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

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

27.62.4.2 `static CryptoFactory* gdcm::CryptoFactory::GetFactoryInstance (CryptoLib id = DEFAULT)` `[static]`

Examples:

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

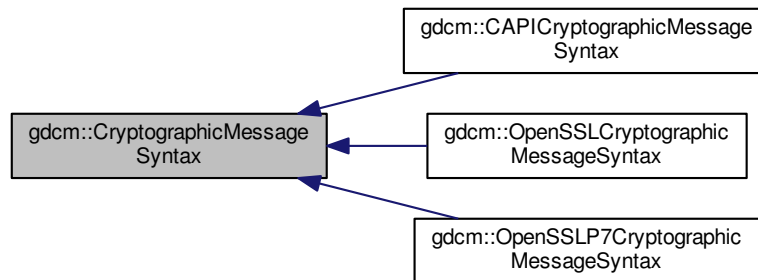
The documentation for this class was generated from the following file:

- [gdcmCryptoFactory.h](#)

27.63 gdcM::CryptographicMessageSyntax Class Reference

```
#include <gdcMCryptographicMessageSyntax.h>
```

Inheritance diagram for gdcM::CryptographicMessageSyntax:



Public Types

- enum [CipherTypes](#) {
[DES3_CIPHER](#),
[AES128_CIPHER](#),
[AES192_CIPHER](#),
[AES256_CIPHER](#) }

Public Member Functions

- [CryptographicMessageSyntax](#) ()
- virtual [~CryptographicMessageSyntax](#) ()
- 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
- 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

27.63.1 Member Enumeration Documentation

27.63.1.1 enum gdcM::CryptographicMessageSyntax::CipherTypes

Enumerator

DES3_CIPHER

AES128_CIPHER***AES192_CIPHER******AES256_CIPHER***

27.63.2 Constructor & Destructor Documentation

27.63.2.1 `gdcmm::CryptographicMessageSyntax::CryptographicMessageSyntax ()` `[inline]`

27.63.2.2 `virtual gdcmm::CryptographicMessageSyntax::~~CryptographicMessageSyntax ()` `[inline],[virtual]`

27.63.3 Member Function Documentation

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

decrypt content from a CMS envelopedData structure

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

27.63.3.2 `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::CAPICryptographicMessageSyntax](#), and [gdcmm::OpenSSLCryptographicMessageSyntax](#).

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

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

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

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

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

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

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

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

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

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

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

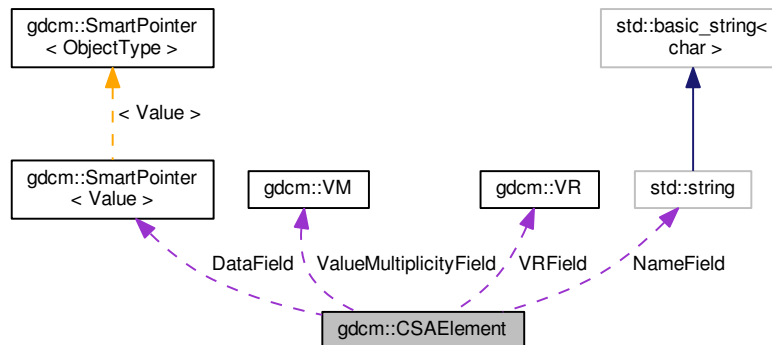
- [gdcM::CryptographicMessageSyntax.h](#)

27.64 gdcM::CSAElement Class Reference

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

```
#include <gdcMCSAElement.h>
```

Collaboration diagram for gdcM::CSAElement:



Public Member Functions

- [CSAElement](#) (unsigned int kf=0)
- [CSAElement](#) (const [CSAElement](#) &_val)
- 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](#) const & [GetValue](#) () const
Set/Get Value (bytes array, SQ of items, SQ of fragments):
- [Value](#) & [GetValue](#) ()

- 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)
- 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)

27.64.1 Detailed Description

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

See also

[CSAHeader](#)

Examples:

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

27.64.2 Member Typedef Documentation

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

27.64.3 Constructor & Destructor Documentation

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

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

27.64.4 Member Function Documentation

27.64.4.1 `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:

[MrProtocol.cxx](#).

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

Set/Get Key.

Referenced by operator<().

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

Set/Get Name.

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

Set/Get NoOfItems.

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

Set/Get SyngoDT.

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

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

Examples:

[csa2img.cxx](#).

27.64.4.7 **Value& gdcmm::CSAElement::GetValue ()** `[inline]`

27.64.4.8 **const VM& gdcmm::CSAElement::GetVM () const** `[inline]`

Set/Get [VM](#).

27.64.4.9 **VR const& gdcmm::CSAElement::GetVR () const** `[inline]`

Set/Get [VR](#).

27.64.4.10 **bool gdcmm::CSAElement::IsEmpty () const** `[inline]`

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

Examples:

[csa2img.cxx](#).

27.64.4.11 **bool gdcmm::CSAElement::operator< (const CSAElement & de) const** `[inline]`

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

27.64.4.12 **CSAElement& gdcmm::CSAElement::operator= (const CSAElement & de)** `[inline]`

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

27.64.4.13 **bool gdcmm::CSAElement::operator== (const CSAElement & de) const** `[inline]`

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

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

Set.

27.64.4.15 **void gdcmm::CSAElement::SetKey (unsigned int key)** `[inline]`

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

27.64.4.17 **void gdcmm::CSAElement::SetNoOfItems (unsigned int items)** `[inline]`

27.64.4.18 **void gdcmm::CSAElement::SetSyngoDT (unsigned int syngodt)** `[inline]`

27.64.4.19 **void gdcmm::CSAElement::SetValue (Value const & vl)** `[inline]`

27.64.4.20 **void gdcmm::CSAElement::SetVM (const VM & vm)** `[inline]`

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

27.64.5 Friends And Related Function Documentation

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

27.64.6 Member Data Documentation

27.64.6.1 `DataPtr gdcm::CSAElement::DataField` [protected]

Referenced by `gdcm::operator<<()`, and `operator=()`.

27.64.6.2 `unsigned int gdcm::CSAElement::KeyField` [protected]

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

27.64.6.3 `std::string gdcm::CSAElement::NameField` [protected]

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

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

Referenced by `gdcm::operator<<()`, and `operator=()`.

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

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

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

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

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

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

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

- [gdcmCSAElement.h](#)

27.65 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](#) ()
- 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 [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)
- template<typename TSwap >
std::istream & [Read](#) (std::istream &is)
- template<typename TSwap >
const std::ostream & [Write](#) (std::ostream &os) const

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)

27.65.1 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

Examples:

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

27.65.2 Member Enumeration Documentation

27.65.2.1 enum gdcm::CSAHeader::CSAHeaderType

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

Enumerator

UNKNOWN
SV10
NOMAGIC
DATASET_FORMAT
INTERFILE
ZEROED_OUT

27.65.3 Constructor & Destructor Documentation

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

27.65.3.2 `gdcm::CSAHeader::~~CSAHeader ()` [[inline](#)]

27.65.4 Member Function Documentation

27.65.4.1 `bool gdcmm::CSAHeader::FindCSAElementByName (const char * name)`

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

Warning

Case Sensitive

Examples:

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

27.65.4.2 `static const PrivateTag& gdcmm::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");

27.65.4.3 `const CSAElement& gdcmm::CSAHeader::GetCSAEnd () const [protected]`

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

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

Warning

Case Sensitive

Examples:

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

27.65.4.5 `static const PrivateTag& gdcmm::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:

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

27.65.4.6 `static const PrivateTag& gdcmm::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](#).

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

Return the [DataSet](#) output (use only if Format == DATASET_FORMAT)

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

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

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

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

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

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

Examples:

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

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

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

Examples:

[csa2img.cxx](#).

Referenced by `gdcm::operator<<()`.

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

27.65.4.13 `template<typename TSwap > const std::ostream& gdcm::CSAHeader::Write (std::ostream & os) const`

27.65.5 Friends And Related Function Documentation

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

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

- [gdcmCSAHeader.h](#)

27.66 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](#) ()
- void [AddCSAHeaderDictEntry](#) (const [CSAHeaderDictEntry](#) &de)
- [ConstIterator](#) [Begin](#) () const
- [ConstIterator](#) [End](#) () const
- const [CSAHeaderDictEntry](#) & [GetCSAHeaderDictEntry](#) (const char *name) const
- bool [IsEmpty](#) () const

Protected Member Functions

- void [LoadDefault](#) ()

Friends

- class [Dicts](#)
- std::ostream & [operator<<](#) (std::ostream &_os, const [CSAHeaderDict](#) &_val)

27.66.1 Detailed Description

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

Examples:

[MrProtocol.cxx](#).

27.66.2 Member Typedef Documentation

27.66.2.1 typedef MapCSAHeaderDictEntry::const_iterator [gdcm::CSAHeaderDict::ConstIterator](#)

27.66.2.2 typedef MapCSAHeaderDictEntry::iterator [gdcm::CSAHeaderDict::Iterator](#)

27.66.2.3 typedef std::set<[CSAHeaderDictEntry](#)> [gdcm::CSAHeaderDict::MapCSAHeaderDictEntry](#)

27.66.3 Constructor & Destructor Documentation

27.66.3.1 [gdcm::CSAHeaderDict::CSAHeaderDict](#) () [\[inline\]](#)

27.66.4 Member Function Documentation

27.66.4.1 void [gdcm::CSAHeaderDict::AddCSAHeaderDictEntry](#) (const [CSAHeaderDictEntry](#) & de) [\[inline\]](#)

27.66.4.2 `ConstIterator gdcm::CSAHeaderDict::Begin () const` `[inline]`

27.66.4.3 `ConstIterator gdcm::CSAHeaderDict::End () const` `[inline]`

27.66.4.4 `const CSAHeaderDictEntry& gdcm::CSAHeaderDict::GetCSAHeaderDictEntry (const char * name) const`
`[inline]`

Examples:

[MrProtocol.cxx](#).

27.66.4.5 `bool gdcm::CSAHeaderDict::IsEmpty () const` `[inline]`

27.66.4.6 `void gdcm::CSAHeaderDict::LoadDefault ()` `[protected]`

27.66.5 Friends And Related Function Documentation

27.66.5.1 `friend class Dicts` `[friend]`

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

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

- [gdcmCSAHeaderDict.h](#)

27.67 gdcm::CSAHeaderDictEntry Class Reference

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.

```
#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)`

27.67.1 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](#).

27.67.2 Constructor & Destructor Documentation

27.67.2.1 `gdcm::CSAHeaderDictEntry::CSAHeaderDictEntry (const char * name = " ", VR const & vr = VR::INVALID, VM const & vm = VM::VMO, const char * desc = " ") [inline]`

27.67.3 Member Function Documentation

27.67.3.1 `const char* gdcm::CSAHeaderDictEntry::GetDescription () const [inline]`

Set/Get Description.

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

Set/Get Name.

Referenced by `operator<()`.

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

Set/Get [VM](#).

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

Set/Get [VR](#).

27.67.3.5 `bool gdcm::CSAHeaderDictEntry::operator< (const CSAHeaderDictEntry & entry) const [inline]`

References `GetName()`.

27.67.3.6 void gdcm::CSAHeaderDictEntry::SetDescription (const char * *desc*) [inline]

27.67.3.7 void gdcm::CSAHeaderDictEntry::SetName (const char * *name*) [inline]

27.67.3.8 void gdcm::CSAHeaderDictEntry::SetVM (VM const & *vm*) [inline]

27.67.3.9 void gdcm::CSAHeaderDictEntry::SetVR (const VR & *vr*) [inline]

27.67.4 Friends And Related Function Documentation

27.67.4.1 std::ostream& operator<< (std::ostream & *_os*, const CSAHeaderDictEntry & *_val*) [friend]

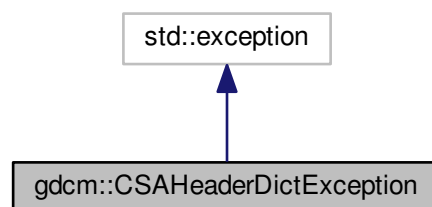
The documentation for this class was generated from the following file:

- [gdcmCSAHeaderDictEntry.h](#)

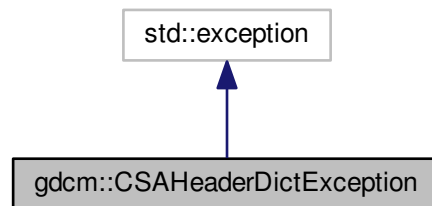
27.68 gdcm::CSAHeaderDictException Class Reference

```
#include <gdcmCSAHeaderDict.h>
```

Inheritance diagram for gdcm::CSAHeaderDictException:



Collaboration diagram for `gdcm::CSAHeaderDictException`:



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

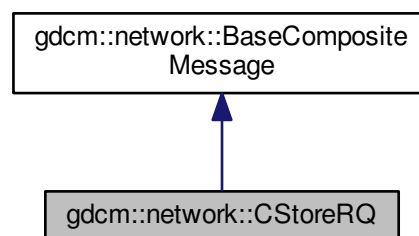
- [gdcmCSAHeaderDict.h](#)

27.69 `gdcm::network::CStoreRQ` Class Reference

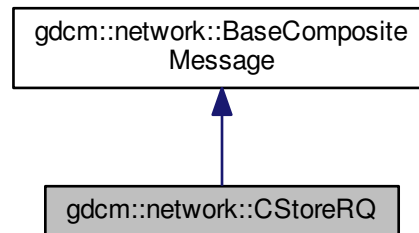
`CStoreRQ` this file defines the messages for the cecho action.

```
#include <gdcmCStoreMessages.h>
```

Inheritance diagram for `gdcm::network::CStoreRQ`:



Collaboration diagram for gdcmm::network::CStoreRQ:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [File](#) &file, bool writeDataSet=true)

27.69.1 Detailed Description

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

27.69.2 Member Function Documentation

27.69.2.1 `std::vector<PresentationDataValue> gdcmm::network::CStoreRQ::ConstructPDV (const ULConnection &inConnection, const File &file, bool writeDataSet=true)`

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

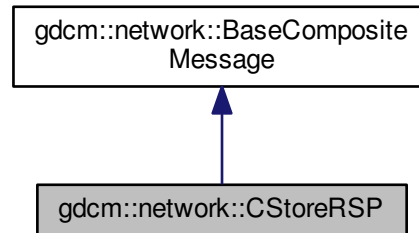
- [gdcmmCStoreMessages.h](#)

27.70 gdcmm::network::CStoreRSP Class Reference

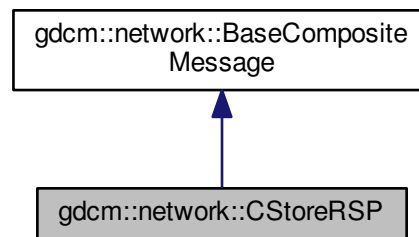
[CStoreRSP](#) this file defines the messages for the cecho action.

```
#include <gdcmmCStoreMessages.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)`

27.70.1 Detailed Description

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

27.70.2 Member Function Documentation

27.70.2.1 `std::vector<PresentationDataValue> gdcm::network::CStoreRSP::ConstructPDV (const DataSet * inDataSet, const BasePDU * inPC)`

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

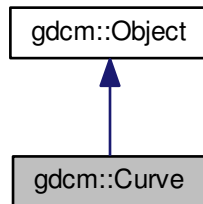
- [gdcmCStoreMessages.h](#)

27.71 gdcm::Curve Class Reference

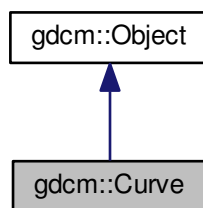
[Curve](#) class to handle element 50xx,3000 [Curve](#) Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004.

```
#include <gdcmCurve.h>
```

Inheritance diagram for gdcm::Curve:



Collaboration diagram for gdcm::Curve:



Public Member Functions

- [Curve](#) ()
- [Curve](#) ([Curve](#) const &ov)
- [~Curve](#) ()
- 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
- 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

27.71.1 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/integriss_HV_5000/xa_integriss.dcm
- TOSHIBA-CurveData[1-3].dcm

27.71.2 Constructor & Destructor Documentation

27.71.2.1 `gdcm::Curve::Curve ()`

27.71.2.2 `gdcm::Curve::~~Curve ()`

27.71.2.3 `gdcm::Curve::Curve (Curve const & ov)`

27.71.3 Member Function Documentation

27.71.3.1 `void gdcm::Curve::Decode (std::istream & is, std::ostream & os)`

27.71.3.2 `void gdcm::Curve::GetAsPoints (float * array) const`

- 27.71.3.3 `std::vector<unsigned short> const& gdcm::Curve::GetCurveDataDescriptor () const`
- 27.71.3.4 `unsigned short gdcm::Curve::GetDataValueRepresentation () const`
- 27.71.3.5 `unsigned short gdcm::Curve::GetDimensions () const`
- 27.71.3.6 `unsigned short gdcm::Curve::GetGroup () const`
- 27.71.3.7 `static unsigned int gdcm::Curve::GetNumberOfCurves (DataSet const & ds) [static]`
- 27.71.3.8 `unsigned short gdcm::Curve::GetNumberOfPoints () const`
- 27.71.3.9 `const char* gdcm::Curve::GetTypeOfData () const`
- 27.71.3.10 `const char* gdcm::Curve::GetTypeOfDataDescription () const`
- 27.71.3.11 `bool gdcm::Curve::IsEmpty () const`
- 27.71.3.12 `void gdcm::Curve::Print (std::ostream &) const [virtual]`

Reimplemented from [gdcm::Object](#).

- 27.71.3.13 `void gdcm::Curve::SetCoordinateStartValue (unsigned short v)`
- 27.71.3.14 `void gdcm::Curve::SetCoordinateStepValue (unsigned short v)`
- 27.71.3.15 `void gdcm::Curve::SetCurve (const char * array, unsigned int length)`
- 27.71.3.16 `void gdcm::Curve::SetCurveDataDescriptor (const uint16_t * values, size_t num)`
- 27.71.3.17 `void gdcm::Curve::SetCurveDescription (const char * curvedescription)`
- 27.71.3.18 `void gdcm::Curve::SetDataValueRepresentation (unsigned short datavaluerepresentation)`
- 27.71.3.19 `void gdcm::Curve::SetDimensions (unsigned short dimensions)`
- 27.71.3.20 `void gdcm::Curve::SetGroup (unsigned short group)`
- 27.71.3.21 `void gdcm::Curve::SetNumberOfPoints (unsigned short numberofpoints)`
- 27.71.3.22 `void gdcm::Curve::SetTypeOfData (const char * typeofdata)`
- 27.71.3.23 `void gdcm::Curve::Update (const DataElement & de)`

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

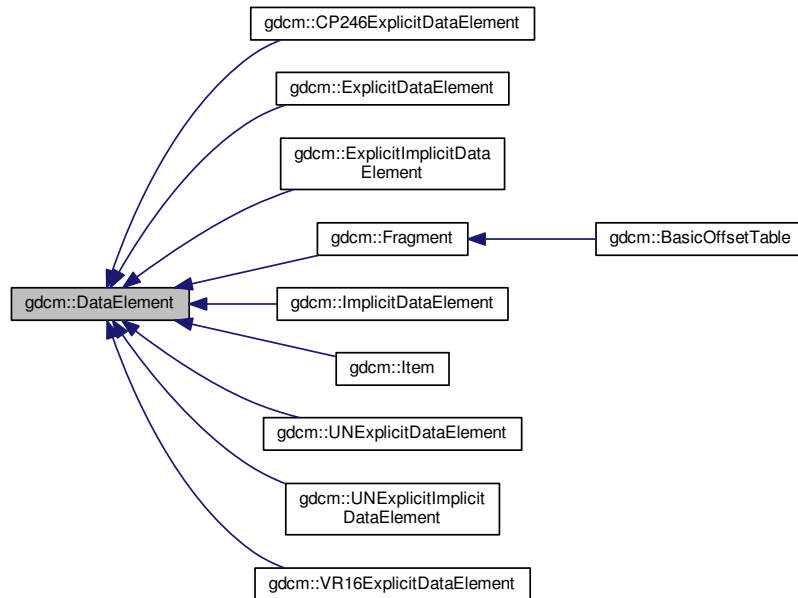
- [gdcmCurve.h](#)

27.72 gdcm::DataElement Class Reference

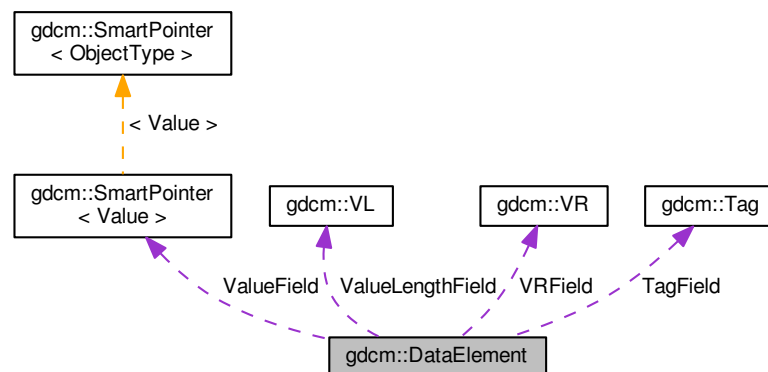
Class to represent a Data [Element](#) either Implicit or Explicit.

```
#include <gdcmDataElement.h>
```

Inheritance diagram for `gdcm::DataElement`:



Collaboration diagram for `gdcm::DataElement`:



Public Member Functions

- `DataElement` (const `Tag` &t=`Tag`(0), const `VL` &vl=0, const `VR` &vr=`VR::INVALID`)
- `DataElement` (const `DataElement` &_val)

- 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
- const [SequenceOfFragments](#) * [GetSequenceOfFragments](#) () const
- [SequenceOfFragments](#) * [GetSequenceOfFragments](#) ()
- const [Tag](#) & [GetTag](#) () const
 - Get [Tag](#).*
- [Tag](#) & [GetTag](#) ()
- [Value](#) const & [GetValue](#) () const
 - Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):*
- [Value](#) & [GetValue](#) ()
- [SmartPointer](#)< [SequenceOfItems](#) > [GetValueAsSQ](#) () const
- const [VL](#) & [GetVL](#) () const
 - Get [VL](#).*
- [VL](#) & [GetVL](#) ()
- [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](#) &de)
- 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)

27.72.1 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 0xffff tags), [Value](#) is NULL

See also

[ExplicitDataElement](#) [ImplicitDataElement](#)

Examples:

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [gdcmrtonplan.cxx](#), [gdcmrtpian.cxx](#), [GenAllIVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [pmsct_rgb1.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [rle2img.cxx](#), and [StreamImageReaderTest.cxx](#).

27.72.2 Member Typedef Documentation

27.72.2.1 `typedef SmartPointer<Value> gdcmm::DataElement::ValuePtr` [protected]

27.72.3 Constructor & Destructor Documentation

27.72.3.1 `gdcm::DataElement::DataElement (const Tag & t = Tag (0), const VL & vl = 0, const VR & vr = VR::INVALID)` `[inline]`

27.72.3.2 `gdcm::DataElement::DataElement (const DataElement & _val)` `[inline]`

27.72.4 Member Function Documentation

27.72.4.1 `void gdcm::DataElement::Clear ()` `[inline]`

Clear Data [Element](#) (make [Value](#) empty and invalidate [Tag](#) & [VR](#))

27.72.4.2 `void gdcm::DataElement::Empty ()` `[inline]`

Make Data [Element](#) empty (no [Value](#))

27.72.4.3 `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](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDICOM.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetSubSequenceData.cxx](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), and [rle2img.cxx](#).

Referenced by `gdcm::operator<<()`, `gdcm::Element< VR::OB, VM::VM1_n >::SetFromDataElement()`, `gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement()`, `gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement()`, `gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement()`, and `gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement()`.

27.72.4.4 `template<typename TDE> VL gdcm::DataElement::GetLength () const` `[inline]`

27.72.4.5 `const SequenceOfFragments* gdcm::DataElement::GetSequenceOfFragments () const`

Return the [Value](#) of [DataElement](#) as a Sequence Of Fragments (if possible)

Warning

: You need to check for NULL return value

Examples:

[FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), and [GetJPEGSamplePrecision.cxx](#).

27.72.4.6 **SequenceOfFragments*** `gdcm::DataElement::GetSequenceOfFragments ()`

27.72.4.7 **const Tag&** `gdcm::DataElement::GetTag () const` `[inline]`

Get [Tag](#).

Examples:

[DumpGEMSMovieGroup.cxx](#), [DuplicatePCDE.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

Referenced by `gdcm::CommandDataSet::Insert()`, `gdcm::FileMetaInformation::Insert()`, `gdcm::DataSet::Insert()`, `operator<()`, `gdcm::SequenceOfItems::Read()`, `gdcm::SequenceOfFragments::ReadValue()`, `gdcm::CommandDataSet::Replace()`, `gdcm::FileMetaInformation::Replace()`, `gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement()`, `gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement()`, and `gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement()`.

27.72.4.8 **Tag&** `gdcm::DataElement::GetTag ()` `[inline]`

27.72.4.9 **Value const&** `gdcm::DataElement::GetValue () const` `[inline]`

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

Examples:

[ReadAndDumpDICOMDIR.cxx](#).

Referenced by `gdcm::DataSet::InsertDataElement()`, `gdcm::Element< VR::OB, VM::VM1_n >::SetFromDataElement()`, and `gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement()`.

27.72.4.10 **Value&** `gdcm::DataElement::GetValue ()` `[inline]`

27.72.4.11 **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](#), [ExtractEncryptedContent.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), and [ReadAndDumpDICOMDIR.cxx](#).

27.72.4.12 **const VL&** `gdcm::DataElement::GetVL () const` `[inline]`

Get [VL](#).

Referenced by `gdcm::DataSet::InsertDataElement()`, `gdcm::SequenceOfItems::Read()`, and `gdcm::SequenceOfFragments::ReadValue()`.

27.72.4.13 `VL& gdcm::DataElement::GetVL () [inline]`

27.72.4.14 `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::Element< VR::OB, VM::VM1_n >::GetAsDataElement()`, `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, VM::VM1_n >::GetAsDataElement()`, `gdcm::Element< VR::OB, VM::VM1_n >::SetFromDataElement()`, `gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement()`, `gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement()`, `gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement()`, and `gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement()`.

27.72.4.15 `bool gdcm::DataElement::IsEmpty () const [inline]`

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

Examples:

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpToshibaDTI.cxx](#), [ELSCINT1WaveToText.cxx](#), [FixJAIBugJPEGLS.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

Referenced by `gdcm::DataSet::InsertDataElement()`, `gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement()`, `gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement()`, `gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement()`, `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()`.

27.72.4.16 `bool gdcm::DataElement::IsUndefinedLength () const [inline]`

return if [Value](#) Length if of undefined length

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

References `GetTag()`.

27.72.4.18 `DataElement& gdcm::DataElement::operator= (const DataElement & de) [inline]`

References `TagField`, `ValueField`, `ValueLengthField`, and `VRField`.

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

References `TagField`, `ValueField`, `ValueLengthField`, and `VRField`.

- 27.72.4.20 `template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::Read (std::istream & is)`
`[inline]`
- 27.72.4.21 `template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadOrSkip (std::istream & is,`
`std::set< Tag > const & skiptags) [inline]`
- 27.72.4.22 `template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadPreValue (std::istream & is,`
`std::set< Tag > const & skiptags) [inline]`
- 27.72.4.23 `template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadValue (std::istream & is,`
`std::set< Tag > const & skiptags) [inline]`
- 27.72.4.24 `template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadValueWithLength (std::istream &`
`is, VL & length, std::set< Tag > const & skiptags) [inline]`
- 27.72.4.25 `template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadWithLength (std::istream & is,`
`VL & length) [inline]`
- 27.72.4.26 `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](#), [FixJAIBugJPEGLS.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), [NewSequence.cs](#), and [StreamImageReaderTest.cxx](#).

Referenced by `gdcm::Element< VR::OB, VM::VM1_n >::GetAsDataElement()`, `gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement()`, `gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement()`, `gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement()`, and `gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement()`.

- 27.72.4.27 `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](#).

- 27.72.4.28 `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\(\)](#).

27.72.4.29 void [gdcm::DataElement::SetValueFieldLength](#) ([VL vl](#), [bool readvalues](#)) [protected]

27.72.4.30 void [gdcm::DataElement::SetVL](#) ([const VL & vl](#)) [inline]

Set [VL](#) Use with cautious (need to match Part 6), advanced user only

See also

[SetByteValue](#)

27.72.4.31 void [gdcm::DataElement::SetVLToUndefined](#) ()

Examples:

[Fake_Image_Using_Stream_Image_Writer.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), and [NewSequence.cs](#).

27.72.4.32 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](#), [FixJAI BugJPEGLS.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), [NewSequence.cs](#), and [StreamImageReaderTest.cxx](#).

References [gdcm::VR::IsVRFile\(\)](#).

Referenced by [gdcm::Element< VR::OB, VM::VM1_n >::GetAsDataElement\(\)](#), [gdcm::Attribute< Group, Element, T < VR, TVM >::GetAsDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement\(\)](#), and [gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement\(\)](#).

27.72.4.33 `template<typename TDE , typename TSwap > const std::ostream& gdcm::DataElement::Write (std::ostream & os)
const [inline]`

27.72.5 Friends And Related Function Documentation

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

27.72.6 Member Data Documentation

27.72.6.1 `Tag gdcm::DataElement::TagField [protected]`

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

27.72.6.2 `ValuePtr gdcm::DataElement::ValueField [protected]`

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

27.72.6.3 `VL gdcm::DataElement::ValueLengthField [protected]`

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

27.72.6.4 `VR gdcm::DataElement::VRField [protected]`

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

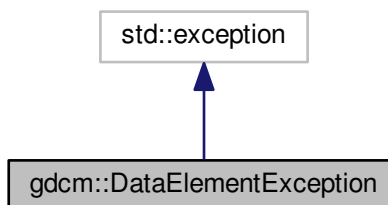
The documentation for this class was generated from the following file:

- [gdcmDataElement.h](#)

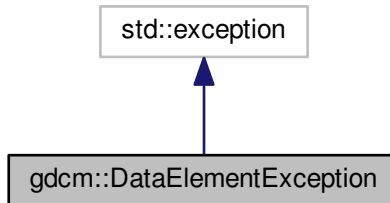
27.73 gdcm::DataElementException Class Reference

```
#include <gdcmDataSet.h>
```

Inheritance diagram for `gdcm::DataElementException`:



Collaboration diagram for gdcm::DataElementException:



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

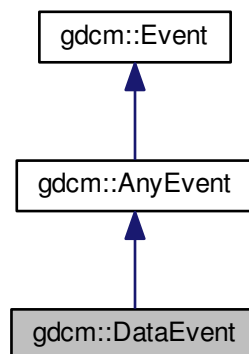
- [gdcmDataSet.h](#)

27.74 gdcm::DataEvent Class Reference

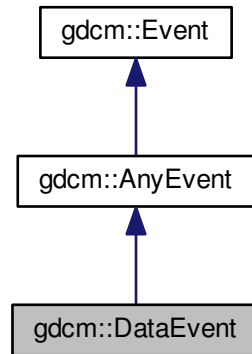
[DataEvent](#).

```
#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=0, size_t len=0)
- [DataEvent](#) (const [Self](#) &s)
- virtual [~DataEvent](#) ()
- virtual bool [CheckEvent](#) (const ::[gdcm::Event](#) *e) const
- const char * [GetData](#) () const
- size_t [GetDataLength](#) () const
- virtual const char * [GetEventName](#) () const
- virtual ::[gdcm::Event](#) * [MakeObject](#) () const
- void [SetData](#) (const char *bytes, size_t len)

27.74.1 Detailed Description

[DataEvent](#).

27.74.2 Member Typedef Documentation

27.74.2.1 typedef `DataEvent` `gdcm::DataEvent::Self`

27.74.2.2 typedef `AnyEvent` `gdcm::DataEvent::Superclass`

27.74.3 Constructor & Destructor Documentation

27.74.3.1 `gdcm::DataEvent::DataEvent (const char * bytes = 0, size_t len = 0) [inline]`

27.74.3.2 `virtual gdcm::DataEvent::~~DataEvent () [inline],[virtual]`

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

27.74.4 Member Function Documentation

27.74.4.1 `virtual bool gdcm::DataEvent::CheckEvent (const ::gdcm::Event * e) const [inline],[virtual]`

27.74.4.2 `const char* gdcm::DataEvent::GetData () const [inline]`

27.74.4.3 `size_t gdcm::DataEvent::GetDataLength () const [inline]`

27.74.4.4 `virtual const char* gdcm::DataEvent::GetEventName () const [inline],[virtual]`

Return the StringName associated with the event.

Implements [gdcm::Event](#).

27.74.4.5 `virtual ::gdcm::Event* gdcm::DataEvent::MakeObject () const [inline],[virtual]`

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

Implements [gdcm::Event](#).

27.74.4.6 `void gdcm::DataEvent::SetData (const char * bytes, size_t len) [inline]`

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

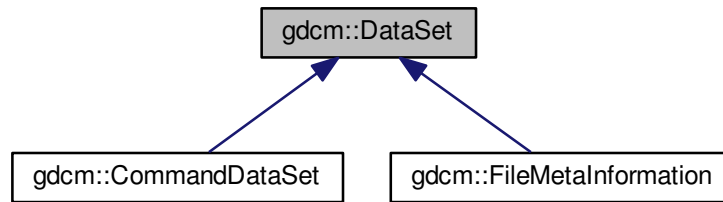
- [gdcmDataEvent.h](#)

27.75 gdcm::DataSet Class Reference

Class to represent a Data Set (which contains Data Elements) A Data Set represents an instance of a real world Information [Object](#).

```
#include <gdcmDataSet.h>
```

Inheritance diagram for `gdcm::DataSet`:



Public Types

- typedef `DataSet::const_iterator` [ConstIterator](#)
- typedef `std::set< DataElement >` [DataSet](#)
- typedef `DataSet::iterator` [Iterator](#)
- typedef `DataSet::size_type` [SizeType](#)

Public Member Functions

- [ConstIterator Begin](#) () const
- [Iterator Begin](#) ()
- void [Clear](#) ()
- template<typename TDE >
unsigned int [ComputeGroupLength](#) (Tag const &tag) const
- [ConstIterator End](#) () const
- [Iterator End](#) ()
- 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 Tag &t) const
- const DataElement & [GetDataElement](#) (const PrivateTag &t) const
Return the dataelement.
- const DataSet & [GetDES](#) () const
- DataSet & [GetDES](#) ()
- 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 &val)
- 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](#) &val)

27.75.1 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](#), [csa2img.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [FixOrientation.cxx](#), [gdcmrtonplan.cxx](#), [gdcmrtpian.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [rle2img.cxx](#), [SortImage.cxx](#), [StreamImageReaderTest.cxx](#), and [VolumeSorter.cxx](#).

27.75.2 Member Typedef Documentation

27.75.2.1 `typedef DataElementSet::const_iterator gdcm::DataSet::ConstIterator`

27.75.2.2 `typedef std::set<DataElement> gdcm::DataSet::DataElementSet`

27.75.2.3 `typedef DataElementSet::iterator gdcm::DataSet::Iterator`

27.75.2.4 `typedef DataElementSet::size_type gdcm::DataSet::SizeType`

27.75.3 Member Function Documentation

27.75.3.1 `ConstIterator gdcm::DataSet::Begin () const` `[inline]`

Examples:

[DiffFile.cxx](#), [DumpGEMSMovieGroup.cxx](#), and [DuplicatePCDE.cxx](#).

27.75.3.2 `Iterator gdcm::DataSet::Begin ()` `[inline]`

27.75.3.3 `void gdcm::DataSet::Clear ()` `[inline]`

Referenced by `gdcm::Item::Read()`.

27.75.3.4 **Tag** gdcm::DataSet::ComputeDataElement (const PrivateTag & t) const [protected]

27.75.3.5 **template**<typename TDE > unsigned int gdcm::DataSet::ComputeGroupLength (Tag const & tag) const [inline]

References gdcm::Tag::GetElement(), and gdcm::Tag::GetGroup().

27.75.3.6 **ConstIterator** gdcm::DataSet::End () const [inline]

Examples:

[DiffFile.cxx](#), [DumpGEMSMovieGroup.cxx](#), and [DuplicatePCDE.cxx](#).

27.75.3.7 **Iterator** gdcm::DataSet::End () [inline]

27.75.3.8 **bool** gdcm::DataSet::FindDataElement (const PrivateTag & t) const

Look up if private tag 't' is present in the dataset:

Examples:

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [csa2img.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [gdcmrtonplan.cxx](#), [gdcmrtpplan.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [MrProtocol.cxx](#), [pmsct_rgb1.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadGEMSSDO.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().

27.75.3.9 **bool** gdcm::DataSet::FindDataElement (const Tag & t) const [inline]

27.75.3.10 **const DataElement&** gdcm::DataSet::FindNextDataElement (const Tag & t) const [inline]

Examples:

[DuplicatePCDE.cxx](#).

27.75.3.11 **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](#), [csa2img.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [gdcmrtonplan.cxx](#), [gdcmrtpplan.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), and [rle2img.cxx](#).

Referenced by `gdcmm::Attribute< Group, Element, TVR, TVM >::Set()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::Set()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::Set()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet()`, and `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet()`.

27.75.3.12 `const DataElement& gdcmm::DataSet::GetDataElement (const PrivateTag & t) const`

Return the dataelement.

27.75.3.13 `const DataElement& gdcmm::DataSet::GetDEEnd () const` `[protected]`

27.75.3.14 `const DataElementSet& gdcmm::DataSet::GetDES () const` `[inline]`

Examples:

[ReadAndDumpDICOMDIR.cxx](#).

27.75.3.15 `DataElementSet& gdcmm::DataSet::GetDES ()` `[inline]`

27.75.3.16 `template<typename TDE > VL gdcmm::DataSet::GetLength () const` `[inline]`

27.75.3.17 `MediaStorage gdcmm::DataSet::GetMediaStorage () const`

27.75.3.18 `std::string gdcmm::DataSet::GetPrivateCreator (const Tag & t) const`

Return the private creator of the private tag 't':

Examples:

[DuplicatePCDE.cxx](#).

27.75.3.19 `void gdcmm::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:

[CreateJIPIDataSet.cxx](#), [DuplicatePCDE.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [NewSequence.cs](#), and [StreamImageReaderTest.cxx](#).

References `gdcmmErrorMacro`, `gdcmm::Tag::GetGroup()`, and `gdcmm::DataElement::GetTag()`.

27.75.3.20 `void gdcmm::DataSet::InsertDataElement (const DataElement & de)` `[inline]`, `[protected]`

References `gdcmmWarningMacro`, `gdcmm::Value::GetLength()`, `gdcmm::DataElement::GetValue()`, `gdcmm::DataElement::GetVL()`, and `gdcmm::DataElement::IsEmpty()`.

27.75.3.21 `bool gdcm::DataSet::IsEmpty () const [inline]`

Returns if the dataset is empty.

Referenced by `gdcm::Item::Read()`.

27.75.3.22 `const DataElement& gdcm::DataSet::operator() (uint16_t group, uint16_t element) const [inline]`

27.75.3.23 `DataSet& gdcm::DataSet::operator= (DataSet const & val) [inline]`

27.75.3.24 `const DataElement& gdcm::DataSet::operator[] (const Tag & t) const [inline]`

27.75.3.25 `void gdcm::DataSet::Print (std::ostream & os, std::string const & indent = " ") const [inline]`

Referenced by `gdcm::operator<<()`.

27.75.3.26 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::Read (std::istream & is)`

27.75.3.27 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadNested (std::istream & is)`

27.75.3.28 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadSelectedPrivateTags (std::istream & is, const std::set< PrivateTag > & tags, bool readvalues = true)`

27.75.3.29 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadSelectedPrivateTagsWithLength (std::istream & is, const std::set< PrivateTag > & tags, VL & length, bool readvalues = true)`

27.75.3.30 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadSelectedTags (std::istream & is, const std::set< Tag > & tags, bool readvalues = true)`

27.75.3.31 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadSelectedTagsWithLength (std::istream & is, const std::set< Tag > & tags, VL & length, bool readvalues = true)`

27.75.3.32 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadUpToTag (std::istream & is, const Tag & t, std::set< Tag > const & skiptags)`

27.75.3.33 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadUpToTagWithLength (std::istream & is, const Tag & t, std::set< Tag > const & skiptags, VL & length)`

27.75.3.34 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadWithLength (std::istream & is, VL & length)`

27.75.3.35 `SizeType gdcm::DataSet::Remove (const Tag & tag) [inline]`

Completely remove a dataelement from the dataset.

Examples:

[ClinicalTrialIdentificationWorkflow.cs](#), [GenFakeIdentifyFile.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [pmsct_rgb1.cxx](#), [ReformatFile.cs](#), [rle2img.cxx](#), and [StandardizeFiles.cs](#).

27.75.3.36 `void gdcm::DataSet::Replace (const DataElement & de) [inline]`

Replace a dataelement with another one.

Examples:

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CreateFakeRTDOSE.cxx](#), [FixBrokenJ2K.cxx](#), [FixJ↔AIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenFakeIdentifyFile.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [i↔U22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

27.75.3.37 `void gdcm::DataSet::ReplaceEmpty (const DataElement & de) [inline]`

Only replace a DICOM attribute when it is missing or empty.

27.75.3.38 `SizeType gdcm::DataSet::Size () const [inline]`

Examples:

[DumpGEMSMovieGroup.cxx](#).

Referenced by `gdcm::SequenceOfItems::Read()`.

27.75.3.39 `template<typename TDE , typename TSwap > std::ostream const& gdcm::DataSet::Write (std::ostream & os) const`

27.75.4 Friends And Related Function Documentation

27.75.4.1 `friend class CSAHeader [friend]`

27.75.4.2 `std::ostream& operator<< (std::ostream & _os, const DataSet & val) [friend]`

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

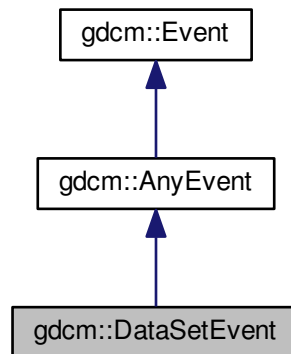
- [gdcmDataSet.h](#)

27.76 gdcm::DataSetEvent Class Reference

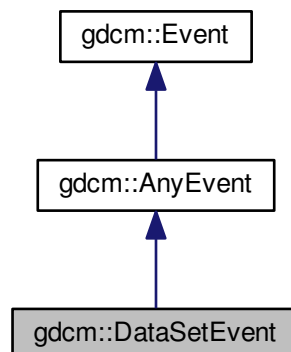
[DataSetEvent](#) Special type of event triggered during the [DataSet](#) store/move process.

```
#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](#) ([DataSet](#) const *ds=NULL)
- [DataSetEvent](#) (const [Self](#) &s)

- virtual [~DataSetEvent](#) ()
- virtual bool [CheckEvent](#) (const [::gdcm::Event](#) *e) const
- [DataSet](#) const & [GetDataSet](#) () const
- virtual const char * [GetEventName](#) () const
- virtual [::gdcm::Event](#) * [MakeObject](#) () const

27.76.1 Detailed Description

[DataSetEvent](#) Special type of event triggered during the [DataSet](#) store/move process.

See also

27.76.2 Member Typedef Documentation

27.76.2.1 typedef [DataSetEvent](#) [gdcm::DataSetEvent::Self](#)

27.76.2.2 typedef [AnyEvent](#) [gdcm::DataSetEvent::Superclass](#)

27.76.3 Constructor & Destructor Documentation

27.76.3.1 [gdcm::DataSetEvent::DataSetEvent](#) ([DataSet](#) const * *ds* = NULL) [inline]

27.76.3.2 virtual [gdcm::DataSetEvent::~~DataSetEvent](#) () [inline],[virtual]

27.76.3.3 [gdcm::DataSetEvent::DataSetEvent](#) (const [Self](#) & *s*) [inline]

27.76.4 Member Function Documentation

27.76.4.1 virtual bool [gdcm::DataSetEvent::CheckEvent](#) (const [::gdcm::Event](#) * *e*) const [inline],[virtual]

27.76.4.2 [DataSet](#) const& [gdcm::DataSetEvent::GetDataSet](#) () const [inline]

27.76.4.3 virtual const char* [gdcm::DataSetEvent::GetEventName](#) () const [inline],[virtual]

Return the StringName associated with the event.

Implements [gdcm::Event](#).

27.76.4.4 virtual [::gdcm::Event*](#) [gdcm::DataSetEvent::MakeObject](#) () const [inline],[virtual]

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

Implements [gdcm::Event](#).

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

- [gdcmDataSetEvent.h](#)

27.77 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)

27.77.1 Detailed Description

DataSetHelper (internal class, not intended for user level)

27.77.2 Member Function Documentation

27.77.2.1 `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:

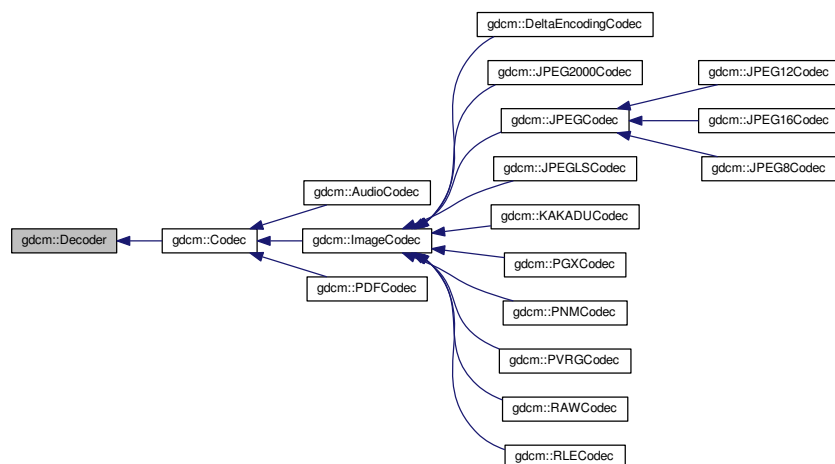
- `gdcmDataSetHelper.h`

27.78 gdcm::Decoder Class Reference

Decoder.

```
#include <gdcmDecoder.h>
```

Inheritance diagram for `gdc::Decoder`:



Public Member Functions

- virtual [~Decoder](#) ()
- virtual bool [CanDecode](#) ([TransferSyntax](#) const &) const =0
Return whether this decoder support this transfer syntax (can decode it)
- virtual bool [Decode](#) ([DataElement](#) const &, [DataElement](#) &)
Decode.

Protected Member Functions

- virtual bool [DecodeByStreams](#) (std::istream &, std::ostream &)

27.78.1 Detailed Description

[Decoder](#).

27.78.2 Constructor & Destructor Documentation

27.78.2.1 virtual [gdcmm::Decoder::~Decoder](#) () [inline],[virtual]

27.78.3 Member Function Documentation

27.78.3.1 virtual bool [gdcmm::Decoder::CanDecode](#) ([TransferSyntax](#) const &) const [pure virtual]

Return whether this decoder support this transfer syntax (can decode it)

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

27.78.3.2 virtual bool [gdcmm::Decoder::Decode](#) ([DataElement](#) const &, [DataElement](#) &) [inline],[virtual]

Decode.

Reimplemented in [gdcmm::JPEGCodec](#), [gdcmm::RLECodec](#), [gdcmm::JPEGLSCodec](#), [gdcmm::PVRGCodec](#), [gdcmm::JPEG2000Codec](#), [gdcmm::ImageCodec](#), [gdcmm::DeltaEncodingCodec](#), [gdcmm::KAKADUCodec](#), [gdcmm::RAWCodec](#), [gdcmm::AudioCodec](#), and [gdcmm::PDFCodec](#).

27.78.3.3 virtual bool [gdcmm::Decoder::DecodeByStreams](#) (std::istream &, std::ostream &) [inline],[protected],[virtual]

Reimplemented in [gdcmm::JPEGCodec](#), [gdcmm::JPEG2000Codec](#), [gdcmm::RLECodec](#), [gdcmm::ImageCodec](#), [gdcmm::RAWCodec](#), [gdcmm::JPEG12Codec](#), [gdcmm::JPEG16Codec](#), and [gdcmm::JPEG8Codec](#).

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

- [gdcmmDecoder.h](#)

27.79 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](#) ()

27.79.1 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.

27.79.2 Constructor & Destructor Documentation

27.79.2.1 `gdcm::DefinedTerms::DefinedTerms ()` `[inline]`

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

- [gdcmDefinedTerms.h](#)

27.80 gdcm::Defs Class Reference

FIXME I do not like the name '[Defs](#)'.

```
#include <gdcmDefs.h>
```

Public Member Functions

- [Defs](#) ()
- [~Defs](#) ()
- const [IOD](#) & [GetIODFromFile](#) (const [File](#) &file) const
- const [IODs](#) & [GetIODs](#) () const
- [IODs](#) & [GetIODs](#) ()
- const [Macros](#) & [GetMacros](#) () const
- [Macros](#) & [GetMacros](#) ()

- const [Modules](#) & [GetModules](#) () const
- [Modules](#) & [GetModules](#) ()
- [Type](#) [GetTypeFromTag](#) (const [File](#) &file, const [Tag](#) &tag) const
- bool [IsEmpty](#) () const
- bool [Verify](#) (const [File](#) &file) const
- bool [Verify](#) (const [DataSet](#) &ds) 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](#)

27.80.1 Detailed Description

FIXME I do not like the name '[Defs](#)'.

Note

bla

Examples:

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

27.80.2 Constructor & Destructor Documentation

27.80.2.1 [gdcm::Defs::Defs](#) ()

27.80.2.2 [gdcm::Defs::~~Defs](#) ()

27.80.3 Member Function Documentation

27.80.3.1 const IOD& [gdcm::Defs::GetIODFromFile](#) (const [File](#) & *file*) const

27.80.3.2 static const char* [gdcm::Defs::GetIODNameFromMediaStorage](#) ([MediaStorage](#) const & *ms*) [static]

Examples:

[GenerateStandardSOPClasses.cxx](#).

27.80.3.3 `const IODs& gdcmm::Defs::GetIODs () const [inline]`

Examples:

[TraverseModules.cxx](#).

27.80.3.4 `IODs& gdcmm::Defs::GetIODs () [inline]`

27.80.3.5 `const Macros& gdcmm::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 [gdcmm::Module](#) API directly

Examples:

[TraverseModules.cxx](#).

27.80.3.6 `Macros& gdcmm::Defs::GetMacros () [inline]`

27.80.3.7 `const Modules& gdcmm::Defs::GetModules () const [inline]`

Examples:

[TraverseModules.cxx](#).

27.80.3.8 `Modules& gdcmm::Defs::GetModules () [inline]`

27.80.3.9 `Type gdcmm::Defs::GetTypeFromTag (const File & file, const Tag & tag) const`

27.80.3.10 `bool gdcmm::Defs::IsEmpty () const [inline]`

27.80.3.11 `void gdcmm::Defs::LoadDefaults () [protected]`

27.80.3.12 `void gdcmm::Defs::LoadFromFile (const char * filename) [protected]`

27.80.3.13 `bool gdcmm::Defs::Verify (const File & file) const`

27.80.3.14 `bool gdcmm::Defs::Verify (const DataSet & ds) const`

27.80.4 Friends And Related Function Documentation

27.80.4.1 `friend class Global [friend]`

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

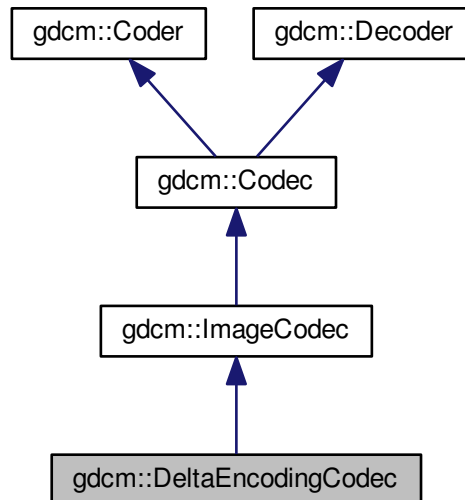
- [gdcmmDefs.h](#)

27.81 gdcm::DeltaEncodingCodec Class Reference

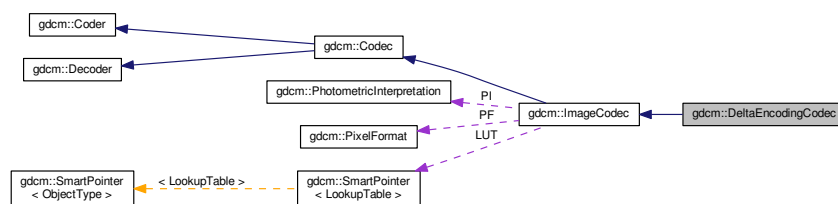
[DeltaEncodingCodec](#) compression used by some private vendor.

```
#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

27.81.1 Detailed Description

[DeltaEncodingCodec](#) compression used by some private vendor.

27.81.2 Constructor & Destructor Documentation

27.81.2.1 `gdcm::DeltaEncodingCodec::DeltaEncodingCodec ()`

27.81.2.2 `gdcm::DeltaEncodingCodec::~~DeltaEncodingCodec ()`

27.81.3 Member Function Documentation

27.81.3.1 `bool gdcm::DeltaEncodingCodec::CanDecode (TransferSyntax const & ts)`

27.81.3.2 `bool gdcm::DeltaEncodingCodec::Decode (DataElement const &, DataElement &)` [virtual]

Decode.

Reimplemented from [gdcm::Decoder](#).

27.81.3.3 `bool gdcm::DeltaEncodingCodec::Decode (std::istream & is, std::ostream & os)` [protected]

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

- [gdcmDeltaEncodingCodec.h](#)

27.82 gdcm::DICOMDIR Class Reference

[DICOMDIR](#) class.

```
#include <gdcmDICOMDIR.h>
```

Public Member Functions

- [DICOMDIR](#) ()
- [DICOMDIR](#) (const [FileSet](#) &fs)

27.82.1 Detailed Description

[DICOMDIR](#) class.

Structured for handling [DICOMDIR](#)

27.82.2 Constructor & Destructor Documentation

27.82.2.1 `gdcm::DICOMDIR::DICOMDIR ()` `[inline]`

27.82.2.2 `gdcm::DICOMDIR::DICOMDIR (const FileSet & fs)` `[inline]`

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

- [gdcmDICOMDIR.h](#)

27.83 gdcm::DICOMDIRGenerator Class Reference

[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.

```
#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](#) ()

27.83.1 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

27.83.2 Member Typedef Documentation

27.83.2.1 `typedef Directory::FilenameType gdcm::DICOMDIRGenerator::FilenameType`

27.83.2.2 `typedef Directory::FilenameType gdcm::DICOMDIRGenerator::FilenameType`

27.83.3 Constructor & Destructor Documentation

27.83.3.1 `gdcm::DICOMDIRGenerator::DICOMDIRGenerator ()`

27.83.3.2 `gdcm::DICOMDIRGenerator::~~DICOMDIRGenerator ()`

27.83.4 Member Function Documentation

27.83.4.1 `bool gdcm::DICOMDIRGenerator::AddImageDirectoryRecord ()` [protected]

27.83.4.2 `bool gdcm::DICOMDIRGenerator::AddPatientDirectoryRecord ()` [protected]

27.83.4.3 `bool gdcm::DICOMDIRGenerator::AddSeriesDirectoryRecord ()` [protected]

27.83.4.4 `bool gdcm::DICOMDIRGenerator::AddStudyDirectoryRecord ()` [protected]

27.83.4.5 `bool gdcm::DICOMDIRGenerator::Generate ()`

Main function to generate the [DICOMDIR](#).

27.83.4.6 **File**& **gdcm::DICOMDIRGenerator::GetFile** ()

27.83.4.7 **Scanner**& **gdcm::DICOMDIRGenerator::GetScanner** () [protected]

27.83.4.8 void **gdcm::DICOMDIRGenerator::SetDescriptor** (const char * *d*)

Set the [File](#) Set ID.

Warning

this need to be a valid [VR::CS](#) value

27.83.4.9 void **gdcm::DICOMDIRGenerator::SetFile** (const **File** & *f*)

Set/Get file. The [DICOMDIR](#) file will be valid once a call to Generate has been done.

27.83.4.10 void **gdcm::DICOMDIRGenerator::SetFilenames** (**FilenamesType** const & *fns*)

Set the list of filenames from which the [DICOMDIR](#) should be generated from.

27.83.4.11 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](#)

27.84 **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](#) ()
- 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

Protected Member Functions

- void [LoadDefault](#) ()

Friends

- class [Dicts](#)
- std::ostream & [operator<<](#) (std::ostream &_os, const [Dict](#) &_val)

27.84.1 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](#).

27.84.2 Member Typedef Documentation

27.84.2.1 typedef MapDictEntry::const_iterator gdcm::Dict::ConstIterator

27.84.2.2 typedef MapDictEntry::iterator gdcm::Dict::Iterator

27.84.2.3 typedef std::map<[Tag](#), [DictEntry](#)> gdcm::Dict::MapDictEntry

27.84.3 Constructor & Destructor Documentation

27.84.3.1 gdcm::Dict::Dict () [\[inline\]](#)

27.84.4 Member Function Documentation

27.84.4.1 void gdcm::Dict::AddDictEntry (const [Tag](#) & tag, const [DictEntry](#) & de) [\[inline\]](#)

27.84.4.2 ConstIterator gdcm::Dict::Begin () const [\[inline\]](#)

Examples:

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

27.84.4.3 `ConstIterator gdcmm::Dict::End () const` `[inline]`

Examples:

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

27.84.4.4 `const DictEntry& gdcmm::Dict::GetDictEntry (const Tag & tag) const` `[inline]`

Examples:

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

27.84.4.5 `const DictEntry& gdcmm::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](#).

27.84.4.6 `const DictEntry& gdcmm::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 unique and can be uniquely link to a tag

Examples:

[ReadAndPrintAttributes.cxx](#).

27.84.4.7 `const char* gdcmm::Dict::GetKeywordFromTag (Tag const & tag) const` `[inline]`

Function to return the Keyword from a [Tag](#).

27.84.4.8 `bool gdcmm::Dict::IsEmpty () const` `[inline]`

Referenced by `gdcmm::Dicts::IsEmpty()`.

27.84.4.9 `void gdcmm::Dict::LoadDefault ()` `[protected]`

27.84.5 Friends And Related Function Documentation

27.84.5.1 `friend class Dicts` `[friend]`

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

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

- [gdcmmDict.h](#)

27.85 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](#) ()

27.85.1 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

27.85.2 Member Enumeration Documentation

27.85.2.1 enum gdcm::DictConverter::OutputTypes

Enumerator

DICT_DEFAULT

DICT_DEBUG

DICT_XML

27.85.3 Constructor & Destructor Documentation

27.85.3.1 gdcm::DictConverter::DictConverter ()

27.85.3.2 gdcm::DictConverter::~~DictConverter ()

27.85.4 Member Function Documentation

27.85.4.1 void gdcm::DictConverter::AddGroupLength () [protected]

27.85.4.2 void gdcm::DictConverter::Convert ()

27.85.4.3 bool gdcm::DictConverter::ConvertToCXX (const char * *raw*, std::string & *cxx*) [protected]

27.85.4.4 bool gdcm::DictConverter::ConvertToXML (const char * *raw*, std::string & *cxx*) [protected]

27.85.4.5 const std::string& gdcm::DictConverter::GetDictName () const

27.85.4.6 const std::string& gdcm::DictConverter::GetInputFilename () const

27.85.4.7 const std::string& gdcm::DictConverter::GetOutputFilename () const

27.85.4.8 int gdcm::DictConverter::GetOutputType () const [inline]

27.85.4.9 static bool gdcm::DictConverter::Readuint16 (const char * *raw*, uint16_t & *ov*) [static]

27.85.4.10 static bool gdcm::DictConverter::ReadVM (const char * *raw*, VM::VMType & *type*) [static]

27.85.4.11 static bool gdcm::DictConverter::ReadVR (const char * *raw*, VR::VRType & *type*) [static]

27.85.4.12 void gdcm::DictConverter::SetDictName (const char * *name*)

27.85.4.13 void gdcm::DictConverter::SetInputFileName (const char * *filename*)

27.85.4.14 void gdcm::DictConverter::SetOutputFileName (const char * *filename*)

27.85.4.15 void gdcm::DictConverter::SetOutputType (int *type*) [inline]

27.85.4.16 void gdcmmDictConverter::WriteFooter () [protected]

27.85.4.17 void gdcmmDictConverter::WriteHeader () [protected]

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

- [gdcmmDictConverter.h](#)

27.86 gdcmmDictEntry Class Reference

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 [gdcmmTag](#) to the needed information.

```
#include <gdcmmDictEntry.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)

27.86.1 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](#), [GenFakeldentifyFile.cxx](#), [PublicDict.cxx](#), and [TraverseModules.cxx](#).

27.86.2 Constructor & Destructor Documentation

27.86.2.1 `gdcm::DictEntry::DictEntry (const char * name = " ", const char * keyword = " ", VR const & vr = VR::INVALID, VM const & vm = VM::VMO, bool ret = false) [inline]`

27.86.3 Member Function Documentation

27.86.3.1 `const char* gdcm::DictEntry::GetKeyword () const [inline]`

same as GetName but without spaces...

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

Set/Get Name.

Referenced by `gdcm::PrivateDict::PrintXML()`.

27.86.3.3 `bool gdcm::DictEntry::GetRetired () const [inline]`

Set/Get Retired flag.

Examples:

[GenAllVR.cxx](#).

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

Set/Get [VM](#).

Referenced by `gdcm::PrivateDict::AddDictEntry()`, and `gdcm::PrivateDict::PrintXML()`.

27.86.3.5 `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()`.

27.86.3.6 `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)

27.86.3.7 `void gdcm::DictEntry::SetElementXX (bool v)` `[inline]`

Set whether element is shared in multiple elements (Source [Image](#) IDs typically)

27.86.3.8 `void gdcm::DictEntry::SetGroupXX (bool v)` `[inline]`

Set whether element is shared in multiple groups (Curve/Overlay typically)

27.86.3.9 `void gdcm::DictEntry::SetKeyword (const char * keyword)` `[inline]`

27.86.3.10 `void gdcm::DictEntry::SetName (const char * name)` `[inline]`

27.86.3.11 `void gdcm::DictEntry::SetRetired (bool retired)` `[inline]`

27.86.3.12 `void gdcm::DictEntry::SetVM (VM const & vm)` `[inline]`

27.86.3.13 `void gdcm::DictEntry::SetVR (const VR & vr)` `[inline]`

Referenced by `gdcm::PrivateDict::AddDictEntry()`.

27.86.4 Friends And Related Function Documentation

27.86.4.1 `friend class Dict` `[friend]`

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

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

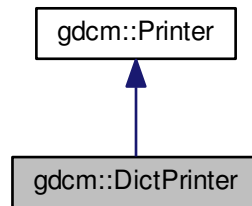
- [gdcmDictEntry.h](#)

27.87 gdcm::DictPrinter Class Reference

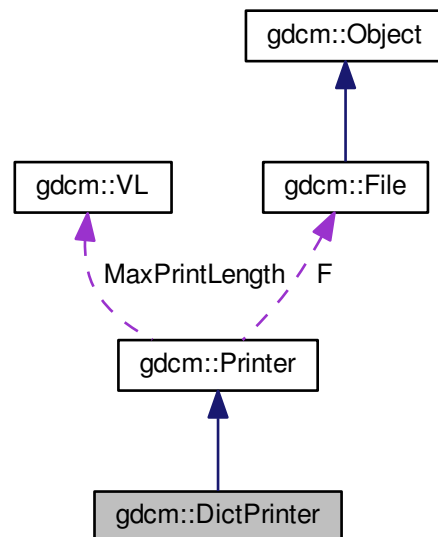
[DictPrinter](#) class.

```
#include <gdcDictPrinter.h>
```

Inheritance diagram for `gdc::DictPrinter`:



Collaboration diagram for `gdc::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

27.87.1 Detailed Description

[DictPrinter](#) class.

27.87.2 Constructor & Destructor Documentation

27.87.2.1 `gdcm::DictPrinter::DictPrinter ()`

27.87.2.2 `gdcm::DictPrinter::~~DictPrinter ()`

27.87.3 Member Function Documentation

27.87.3.1 `void gdcm::DictPrinter::Print (std::ostream & os)`

27.87.3.2 `void gdcm::DictPrinter::PrintDataElement2 (std::ostream & os, const DataSet & ds, const DataElement & ide)`
[protected]

27.87.3.3 `void gdcm::DictPrinter::PrintDataSet2 (std::ostream & os, const DataSet & ds)` [protected]

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

- [gdcmDictPrinter.h](#)

27.88 gdcm::Dicts Class Reference

Class to manipulate the sum of knowledge (all the dict user load)

```
#include <gdcmDicts.h>
```

Public Member Functions

- [Dicts](#) ()
- [~Dicts](#) ()
- const [CSAHeaderDict](#) & [GetCSAHeaderDict](#) () const
- const [DictEntry](#) & [GetDictEntry](#) (const [Tag](#) &tag, const char *owner=NULL) const
NOT THREAD SAFE.
- const [DictEntry](#) & [GetDictEntry](#) (const [PrivateTag](#) &tag) const
- const [PrivateDict](#) & [GetPrivateDict](#) () const
- [PrivateDict](#) & [GetPrivateDict](#) ()
- const [Dict](#) & [GetPublicDict](#) () const
- bool [IsEmpty](#) () const

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)

27.88.1 Detailed Description

Class to manipulate the sum of knowledge (all the dict user load)

Note

bla

Examples:

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [PublicDict.cxx](#), [ReadAndPrintAttributes.cxx](#), and [TraverseModules.cxx](#).

27.88.2 Member Enumeration Documentation

27.88.2.1 enum `gdcm::Dicts::ConstructorType` [protected]

Enumerator

PHILIPS

GEMS

SIEMENS

27.88.3 Constructor & Destructor Documentation

27.88.3.1 `gdcm::Dicts::Dicts ()`

27.88.3.2 `gdcm::Dicts::~~Dicts ()`

27.88.4 Member Function Documentation

27.88.4.1 `static const char* gdcmm::Dicts::GetConstructorString (ConstructorType type)` `[static], [protected]`

27.88.4.2 `const CSAHeaderDict& gdcmm::Dicts::GetCSAHeaderDict () const`

Examples:

[MrProtocol.cxx](#).

27.88.4.3 `const DictEntry& gdcmm::Dicts::GetDictEntry (const Tag & tag, const char * owner = NULL) 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](#), and [TraverseModules.cxx](#).

27.88.4.4 `const DictEntry& gdcmm::Dicts::GetDictEntry (const PrivateTag & tag) const`

27.88.4.5 `const PrivateDict& gdcmm::Dicts::GetPrivateDict () const`

27.88.4.6 `PrivateDict& gdcmm::Dicts::GetPrivateDict ()`

27.88.4.7 `const Dict& gdcmm::Dicts::GetPublicDict () const`

Examples:

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [PublicDict.cxx](#), and [ReadAndPrintAttributes.cxx](#).

27.88.4.8 `bool gdcmm::Dicts::IsEmpty () const` `[inline]`

References `gdcmm::Dict::IsEmpty()`.

27.88.4.9 `void gdcmm::Dicts::LoadDefaults ()` `[protected]`

27.88.5 Friends And Related Function Documentation

27.88.5.1 `friend class Global` `[friend]`

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

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

- [gdcmmDicts.h](#)

27.89 gdcm::network::DIMSE Class Reference

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)

```
#include <gdcmDIMSE.h>
```

Public Types

- enum [CommandTypes](#) {
 - [C_STORE_RQ](#) = 0x0001,
 - [C_STORE_RSP](#) = 0x8001,
 - [C_GET_RQ](#) = 0x0010,
 - [C_GET_RSP](#) = 0x8010,
 - [C_FIND_RQ](#) = 0x0020,
 - [C_FIND_RSP](#) = 0x8020,
 - [C_MOVE_RQ](#) = 0x0021,
 - [C_MOVE_RSP](#) = 0x8021,
 - [C_ECHO_RQ](#) = 0x0030,
 - [C_ECHO_RSP](#) = 0x8030,
 - [N_EVENT_REPORT_RQ](#) = 0x0100,
 - [N_EVENT_REPORT_RSP](#) = 0x8100,
 - [N_GET_RQ](#) = 0x0110,
 - [N_GET_RSP](#) = 0x8110,
 - [N_SET_RQ](#) = 0x0120,
 - [N_SET_RSP](#) = 0x8120,
 - [N_ACTION_RQ](#) = 0x0130,
 - [N_ACTION_RSP](#) = 0x8130,
 - [N_CREATE_RQ](#) = 0x0140,
 - [N_CREATE_RSP](#) = 0x8140,
 - [N_DELETE_RQ](#) = 0x0150,
 - [N_DELETE_RSP](#) = 0x8150,
 - [C_CANCEL_RQ](#) = 0x0FFF }

27.89.1 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)

27.89.2 Member Enumeration Documentation

27.89.2.1 enum gdcm::network::DIMSE::CommandTypes

Enumerator

C_STORE_RQ
C_STORE_RSP
C_GET_RQ
C_GET_RSP
C_FIND_RQ
C_FIND_RSP

C_MOVE_RQ
C_MOVE_RSP
C_ECHO_RQ
C_ECHO_RSP
N_EVENT_REPORT_RQ
N_EVENT_REPORT_RSP
N_GET_RQ
N_GET_RSP
N_SET_RQ
N_SET_RSP
N_ACTION_RQ
N_ACTION_RSP
N_CREATE_RQ
N_CREATE_RSP
N_DELETE_RQ
N_DELETE_RSP
C_CANCEL_RQ

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

- [gdcmmDIMSE.h](#)

27.90 gdcmm::DirectionCosines Class Reference

class to handle [DirectionCosines](#)

```
#include <gdcmmDirectionCosines.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)`

27.90.1 Detailed Description

class to handle [DirectionCosines](#)

Examples:

[DiscriminateVolume.cxx](#).

27.90.2 Constructor & Destructor Documentation

27.90.2.1 `gdc::DirectionCosines::DirectionCosines ()`

27.90.2.2 `gdc::DirectionCosines::DirectionCosines (const double dircos[6])`

27.90.2.3 `gdc::DirectionCosines::~~DirectionCosines ()`

27.90.3 Member Function Documentation

27.90.3.1 `double gdc::DirectionCosines::ComputeDistAlongNormal (const double ipp[3]) const`

Compute the distance along the normal.

27.90.3.2 `void gdc::DirectionCosines::Cross (double z[3]) const`

Compute Cross product.

27.90.3.3 `double gdc::DirectionCosines::CrossDot (DirectionCosines const & dc) const`

Compute the Dot product of the two cross vector of both [DirectionCosines](#) object.

Examples:

[DiscriminateVolume.cxx](#).

27.90.3.4 `double gdc::DirectionCosines::Dot () const`

Compute Dot.

27.90.3.5 `bool gdc::DirectionCosines::IsValid () const`

Return whether or not this is a valid direction cosines.

27.90.3.6 void gdcm::DirectionCosines::Normalize ()

Normalize in-place.

27.90.3.7 gdcm::DirectionCosines::operator const double * () const [inline]

Make the class behave like a const double *.

27.90.3.8 void gdcm::DirectionCosines::Print (std::ostream &) const

Print.

27.90.3.9 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](#)

27.91 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](#) ()
- [~Directory](#) ()
- [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)

27.91.1 Detailed Description

Class for manipulation directories.

Note

This implementation provide a cross platform implementation for manipulating diretores: 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 [gdcmm::Directory](#) does not make much sense.

Examples:

[DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [gdcmmorthoplanes.cxx](#), [GenerateRTSTRUCT.cxx](#), [ReadUTF8Qt←Dir.cxx](#), [reslicesphere.cxx](#), [SortImage.cxx](#), [threadgdcmm.cxx](#), and [VolumeSorter.cxx](#).

27.91.2 Member Typedef Documentation

27.91.2.1 `typedef std::vector<FilenameType> gdcmm::Directory::FilenamesType`

Examples:

[DiscriminateVolume.cxx](#).

27.91.2.2 `typedef std::string gdcmm::Directory::FilenameType`

27.91.3 Constructor & Destructor Documentation

27.91.3.1 `gdcmm::Directory::Directory () [inline]`

27.91.3.2 `gdcmm::Directory::~~Directory () [inline]`

27.91.4 Member Function Documentation

27.91.4.1 `unsigned int gdcmm::Directory::Explore (FilenameType const & name, bool recursive) [protected]`

Return number of file found when 'recursive'ly exploring directory `name`

27.91.4.2 `FilenameType const& gdcm::Directory::GetDirectories () const` `[inline]`

Return the Directories traversed.

27.91.4.3 `FilenameType const& gdcm::Directory::GetFileNames () const` `[inline]`

Set/Get the file names within the directory.

Examples:

[DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [gdcmorthoplanes.cxx](#), [GenerateRTSTRUCT.cxx](#), [ReadUTF8QtDir.cxx](#), [reslicesphere.cxx](#), [SortImage.cxx](#), [threadgdcm.cxx](#), and [VolumeSorter.cxx](#).

27.91.4.4 `FilenameType const& gdcm::Directory::GetToplevel () const` `[inline]`

Get the name of the toplevel directory.

27.91.4.5 `unsigned int gdcm::Directory::Load (FilenameType const & name, bool recursive = false)`

construct a list of filenames and subdirectory beneath directory: name

Warning

: hidden file and hidden directory are not loaded.

Examples:

[DecompressImageMultiframe.cs](#), [DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [gdcmorthoplanes.cxx](#), [GenerateRTSTRUCT.cxx](#), [ReadUTF8QtDir.cxx](#), [reslicesphere.cxx](#), [SortImage.cxx](#), [threadgdcm.cxx](#), and [VolumeSorter.cxx](#).

27.91.4.6 `void gdcm::Directory::Print (std::ostream & os = std::cout) const`

Print.

Examples:

[SortImage.cxx](#).

Referenced by `gdcm::operator<<()`.

27.91.5 Friends And Related Function Documentation

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

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

- [gdcmDirectory.h](#)

27.92 gdcm::DirectoryHelper Class Reference

[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.

```
#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 GetMRIImageSeriesUIDs](#) (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)

27.92.1 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.

27.92.2 Member Function Documentation

27.92.2.1 static [Directory::FilenamesType](#) `gdcm::DirectoryHelper::GetCTImageSeriesUIDs (const std::string & inDirectory)` `[static]`

27.92.2.2 static [Directory::FilenamesType](#) `gdcm::DirectoryHelper::GetFilenamesFromSeriesUIDs (const std::string & inDirectory, const std::string & inSeriesUID)` `[static]`

Examples:

[GenerateRTSTRUCT.cxx](#).

27.92.2.3 static std::string `gdcm::DirectoryHelper::GetFrameOfReference (const std::vector< DataSet > & inDS)` `[static]`

27.92.2.4 static [Directory::FilenamesType](#) `gdcm::DirectoryHelper::GetMRIImageSeriesUIDs (const std::string & inDirectory)` `[static]`

27.92.2.5 static `Directory::FilenameType` `gdcm::DirectoryHelper::GetRTStructSeriesUIDs` (`const std::string & inDirectory`)
[static]

Examples:

[GenerateRTSTRUCT.cxx](#).

27.92.2.6 static `Directory::FilenameType` `gdcm::DirectoryHelper::GetSeriesUIDsBySOPClassUID` (`const std::string & inDirectory`, `const std::string & inSOPClassUID`) [static]

27.92.2.7 static `std::string` `gdcm::DirectoryHelper::GetSOPClassUID` (`const std::vector< DataSet > & inDS`) [static]

27.92.2.8 static `std::string` `gdcm::DirectoryHelper::GetStringValueFromTag` (`const Tag & t`, `const DataSet & ds`) [static]

27.92.2.9 static `std::vector<DataSet>` `gdcm::DirectoryHelper::LoadImageFromFiles` (`const std::string & inDirectory`, `const std::string & inSeriesUID`) [static]

27.92.2.10 static `std::string` `gdcm::DirectoryHelper::RetrieveSOPInstanceUIDFromIndex` (`int inIndex`, `const std::vector< DataSet > & inDS`) [static]

27.92.2.11 static `std::string` `gdcm::DirectoryHelper::RetrieveSOPInstanceUIDFromZPosition` (`double inZPos`, `const std::vector< DataSet > & inDS`) [static]

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

- [gdcmDirectoryHelper.h](#)

27.93 gdcm::DummyValueGenerator Class Reference

Class for generating dummy value.

```
#include <gdcmDummyValueGenerator.h>
```

Static Public Member Functions

- static `const char *` [Generate](#) (`const char *input`)

27.93.1 Detailed Description

Class for generating dummy value.

See also

[Anonymizer](#)

27.93.2 Member Function Documentation

27.93.2.1 `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:

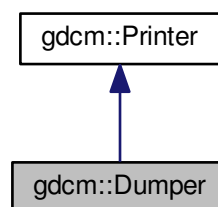
- [gdcmDummyValueGenerator.h](#)

27.94 gdcm::Dumper Class Reference

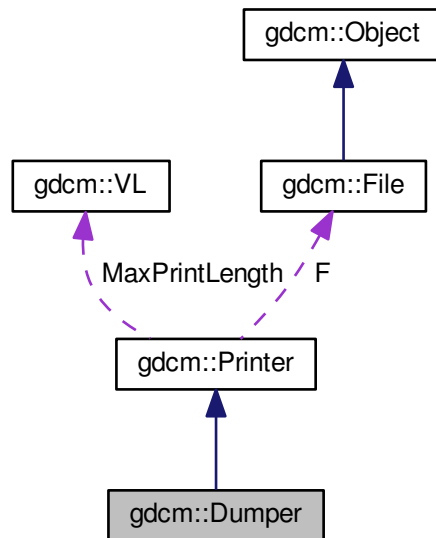
[Codec](#) class.

```
#include <gdcmDumper.h>
```

Inheritance diagram for `gdcm::Dumper`:



Collaboration diagram for gdcmm::Dumper:



Public Member Functions

- [Dumper](#) ()
- [~Dumper](#) ()

Additional Inherited Members

27.94.1 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

27.94.2 Constructor & Destructor Documentation

27.94.2.1 `gdcmm::Dumper::Dumper ()` `[inline]`

27.94.2.2 `gdcmm::Dumper::~~Dumper ()` `[inline]`

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

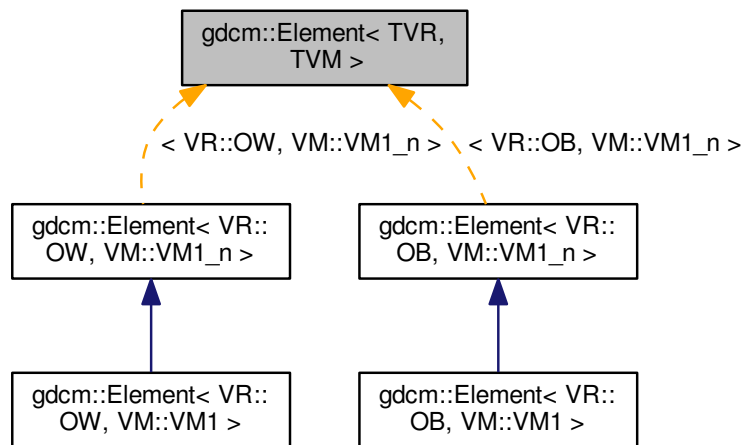
- [gdcmmDumper.h](#)

27.95 gdcElement< TVR, TVM > Class Template Reference

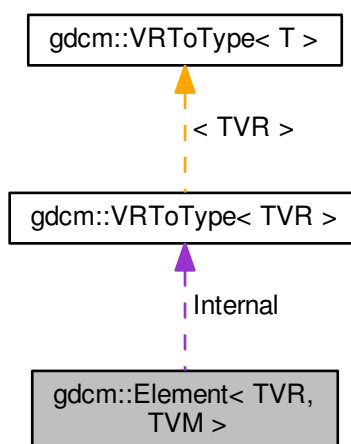
[Element](#) class.

```
#include <gdcElement.h>
```

Inheritance diagram for gdcElement< TVR, TVM >:



Collaboration diagram for gdcElement< TVR, TVM >:



Public Types

- typedef [VRToType](#)< TVR >::Type Type

Public Member Functions

- [DataElement](#) [GetAsDataElement](#) () const
- unsigned long [GetLength](#) () const
- const [VRToType](#)< TVR >::Type & [GetValue](#) (unsigned int idx=0) const
- [VRToType](#)< TVR >::Type & [GetValue](#) (unsigned int idx=0)
- const [VRToType](#)< TVR >::Type * [GetValues](#) () const
- [VRToType](#)< TVR >::Type [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &_os) const
- void [Read](#) (std::istream &_is)
- void [Set](#) ([Value](#) const &v)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetValue](#) (typename [VRToType](#)< TVR >::Type v, unsigned int idx=0)
- void [Write](#) (std::ostream &_os) const

Static Public Member Functions

- static [VM](#) [GetVM](#) ()
- static [VR](#) [GetVR](#) ()

Public Attributes

- [VRToType](#)< TVR >::Type [Internal](#) [[VMToLength](#)< TVM >::Length]

Protected Member Functions

- void [SetNoSwap](#) ([Value](#) const &v)

27.95.1 Detailed Description

template<int TVR, int TVM>class gdcmm::Element< TVR, TVM >

[Element](#) class.

Note

TODO

Examples:

[csa2img.cxx](#), [DumpADAC.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [GetSubSequenceData.cxx](#), and [iU22tomultisc.cxx](#).

27.95.2 Member Typedef Documentation

27.95.2.1 `template<int TVR, int TVM> typedef VRToType<TVR>::Type gdcM::Element< TVR, TVM >::Type`

27.95.3 Member Function Documentation

27.95.3.1 `template<int TVR, int TVM> DataElement gdcM::Element< TVR, TVM >::GetAsDataElement () const`
[inline]

27.95.3.2 `template<int TVR, int TVM> unsigned long gdcM::Element< TVR, TVM >::GetLength () const` [inline]

27.95.3.3 `template<int TVR, int TVM> const VRToType<TVR>::Type& gdcM::Element< TVR, TVM >::GetValue (unsigned int idx = 0) const` [inline]

27.95.3.4 `template<int TVR, int TVM> VRToType<TVR>::Type& gdcM::Element< TVR, TVM >::GetValue (unsigned int idx = 0)` [inline]

27.95.3.5 `template<int TVR, int TVM> const VRToType<TVR>::Type* gdcM::Element< TVR, TVM >::GetValues () const`
[inline]

27.95.3.6 `template<int TVR, int TVM> static VM gdcM::Element< TVR, TVM >::GetVM ()` [inline],[static]

27.95.3.7 `template<int TVR, int TVM> static VR gdcM::Element< TVR, TVM >::GetVR ()` [inline],[static]

27.95.3.8 `template<int TVR, int TVM> VRToType<TVR>::Type gdcM::Element< TVR, TVM >::operator[] (unsigned int idx) const` [inline]

27.95.3.9 `template<int TVR, int TVM> void gdcM::Element< TVR, TVM >::Print (std::ostream &_os) const` [inline]

27.95.3.10 `template<int TVR, int TVM> void gdcM::Element< TVR, TVM >::Read (std::istream &_is)` [inline]

27.95.3.11 `template<int TVR, int TVM> void gdcM::Element< TVR, TVM >::Set (Value const & v)` [inline]

27.95.3.12 `template<int TVR, int TVM> void gdcM::Element< TVR, TVM >::SetFromDataElement (DataElement< TVR, TVM > const & de)` [inline]

27.95.3.13 `template<int TVR, int TVM> void gdcM::Element< TVR, TVM >::SetNoSwap (Value const & v)` [inline],[protected]

27.95.3.14 `template<int TVR, int TVM> void gdcM::Element< TVR, TVM >::SetValue (typename VRToType< TVR >::Type v, unsigned int idx = 0)` [inline]

27.95.3.15 `template<int TVR, int TVM> void gdcM::Element< TVR, TVM >::Write (std::ostream &_os) const` [inline]

27.95.4 Member Data Documentation

27.95.4.1 `template<int TVR, int TVM> VRToType<TVR>::Type gdcM::Element< TVR, TVM >::Internal[VMToLength< TVM >::Length]`

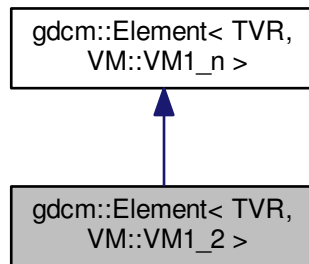
The documentation for this class was generated from the following file:

- [gdcMElement.h](#)

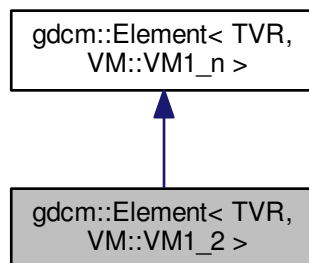
27.96 gdcM::Element< TVR, VM::VM1_2 > Class Template Reference

```
#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

27.96.1 Member Typedef Documentation

27.96.1.1 `template<int TVR> typedef Element<TVR, VM::VM1_n> gdcmm::Element< TVR, VM::VM1_2 >::Parent`

27.96.2 Member Function Documentation

27.96.2.1 `template<int TVR> void gdcmm::Element< TVR, VM::VM1_2 >::SetLength (int len) [inline]`

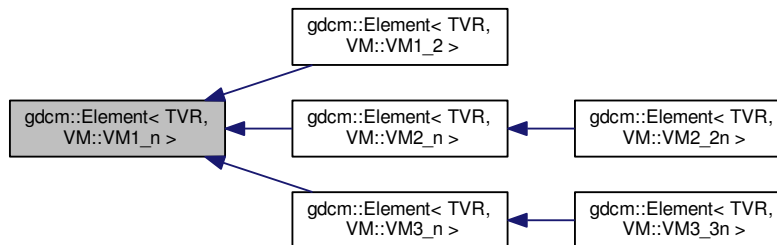
The documentation for this class was generated from the following file:

- [gdcmmElement.h](#)

27.97 gdcmm::Element< TVR, VM::VM1_n > Class Template Reference

```
#include <gdcmmElement.h>
```

Inheritance diagram for gdcmm::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`
- `const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const`
- `VRToType< TVR >::Type & GetValue (unsigned int idx=0)`
- `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)

27.97.1 Member Typedef Documentation

27.97.1.1 `template<int TVR> typedef VRToType<TVR>::Type gdcm::Element< TVR, VM::VM1_n >::Type`

27.97.2 Constructor & Destructor Documentation

27.97.2.1 `template<int TVR> gdcm::Element< TVR, VM::VM1_n >::Element () [inline],[explicit]`

27.97.2.2 `template<int TVR> gdcm::Element< TVR, VM::VM1_n >::~~Element () [inline]`

27.97.2.3 `template<int TVR> gdcm::Element< TVR, VM::VM1_n >::Element (const Element< TVR, VM::VM1_n > &_val) [inline]`

27.97.3 Member Function Documentation

27.97.3.1 `template<int TVR> DataElement gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement () const [inline]`

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

27.97.3.2 `template<int TVR> unsigned long gdcm::Element< TVR, VM::VM1_n >::GetLength () const [inline]`

27.97.3.3 `template<int TVR> const VRToType<TVR>::Type& gdcm::Element< TVR, VM::VM1_n >::GetValue (unsigned int idx=0) const [inline]`

27.97.3.4 `template<int TVR> VRToType<TVR>::Type& gdcm::Element< TVR, VM::VM1_n >::GetValue (unsigned int idx=0) [inline]`

27.97.3.5 `template<int TVR> static VM gdcm::Element< TVR, VM::VM1_n >::GetVM () [inline],[static]`

27.97.3.6 `template<int TVR> static VR gdcm::Element< TVR, VM::VM1_n >::GetVR () [inline],[static]`

27.97.3.7 `template<int TVR> Element& gdcM::Element< TVR, VM::VM1_n >::operator=(const Element< TVR, VM::VM1_n > &_val) [inline]`

27.97.3.8 `template<int TVR> VRToType<TVR>::Type gdcM::Element< TVR, VM::VM1_n >::operator[] (unsigned int idx) const [inline]`

27.97.3.9 `template<int TVR> void gdcM::Element< TVR, VM::VM1_n >::Print (std::ostream &_os) const [inline]`

27.97.3.10 `template<int TVR> void gdcM::Element< TVR, VM::VM1_n >::Read (std::istream &_is) [inline]`

27.97.3.11 `template<int TVR> void gdcM::Element< TVR, VM::VM1_n >::Set (Value const & v) [inline]`

References `gdcM::ByteValue::GetLength()`, `gdcM::ByteValue::GetPointer()`, and `gdcM::VRBINARY`.

27.97.3.12 `template<int TVR> void gdcM::Element< TVR, VM::VM1_n >::SetArray (const Type * array, unsigned long len, bool save = false) [inline]`

27.97.3.13 `template<int 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()`.

27.97.3.14 `template<int TVR> void gdcM::Element< TVR, VM::VM1_n >::SetLength (unsigned long len) [inline]`

27.97.3.15 `template<int TVR> void gdcM::Element< TVR, VM::VM1_n >::SetNoSwap (Value const & v) [inline], [protected]`

References `gdcM::ByteValue::GetLength()`, `gdcM::ByteValue::GetPointer()`, and `gdcM::VRBINARY`.

27.97.3.16 `template<int TVR> void gdcM::Element< TVR, VM::VM1_n >::SetValue (typename VRToType< TVR >::Type v, unsigned int idx = 0) [inline]`

27.97.3.17 `template<int TVR> void gdcM::Element< TVR, VM::VM1_n >::Write (std::ostream &_os) const [inline]`

27.97.3.18 `template<int TVR> void gdcM::Element< TVR, VM::VM1_n >::WriteASCII (std::ostream & os) const [inline]`

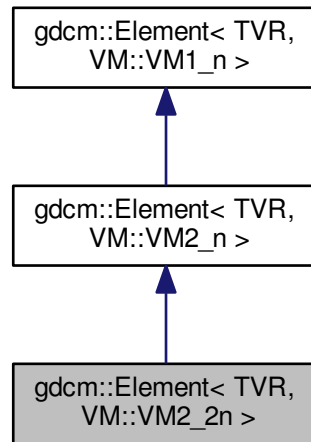
The documentation for this class was generated from the following file:

- [gdcMElement.h](#)

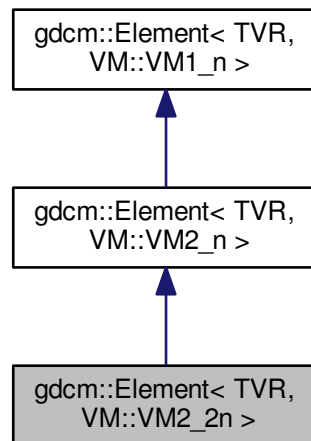
27.98 `gdcM::Element< TVR, VM::VM2_2n >` Class Template Reference

```
#include <gdcMElement.h>
```

Inheritance diagram for gdcM::Element< TVR, VM::VM2_2n >:



Collaboration diagram for gdcM::Element< TVR, VM::VM2_2n >:



Public Types

- typedef [Element](#)< TVR, [VM::VM2_n](#) > [Parent](#)

Public Member Functions

- void [SetLength](#) (int len)

Additional Inherited Members

27.98.1 Member Typedef Documentation

27.98.1.1 `template<int TVR> typedef Element<TVR, VM::VM2_n> gdcm::Element< TVR, VM::VM2_2n >::Parent`

27.98.2 Member Function Documentation

27.98.2.1 `template<int TVR> void gdcm::Element< TVR, VM::VM2_2n >::SetLength (int len) [inline]`

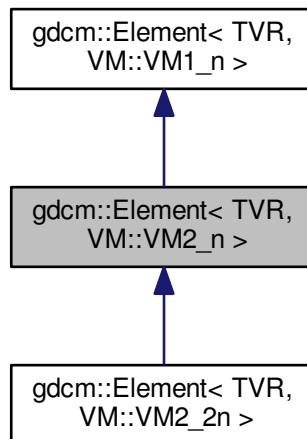
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

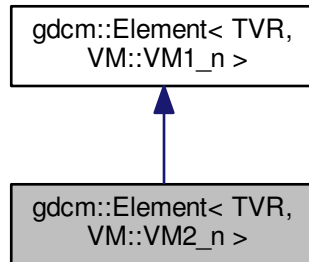
27.99 gdcm::Element< TVR, VM::VM2_n > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for `gdcm::Element< TVR, VM::VM2_n >`:



Collaboration diagram for gdcmm::Element< TVR, VM::VM2_n >:



Public Types

- typedef [Element](#)< TVR, [VM::VM1_n](#) > [Parent](#)

Public Member Functions

- void [SetLength](#) (int len)

Additional Inherited Members

27.99.1 Member Typedef Documentation

27.99.1.1 `template<int TVR> typedef Element<TVR, VM::VM1_n> gdcmm::Element< TVR, VM::VM2_n >::Parent`

27.99.2 Member Function Documentation

27.99.2.1 `template<int TVR> void gdcmm::Element< TVR, VM::VM2_n >::SetLength (int len) [inline]`

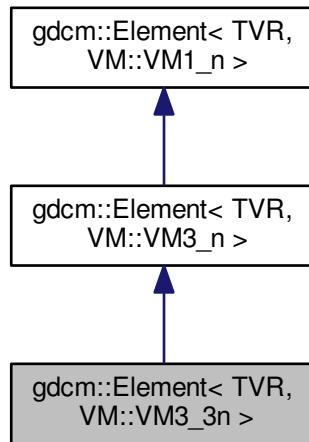
The documentation for this class was generated from the following file:

- [gdcmmElement.h](#)

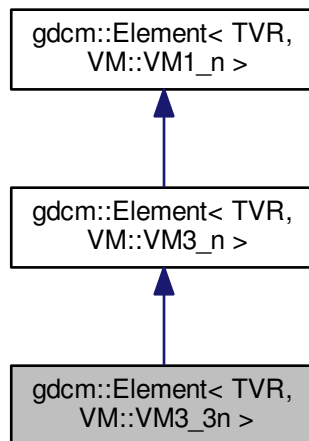
27.100 gdcmm::Element< TVR, VM::VM3_3n > Class Template Reference

```
#include <gdcmmElement.h>
```

Inheritance diagram for `gdc::Element< TVR, VM::VM3_3n >`:



Collaboration diagram for `gdc::Element< TVR, VM::VM3_3n >`:



Public Types

- typedef `Element< TVR, VM::VM3_3n >` `Parent`

Public Member Functions

- void [SetLength](#) (int len)

Additional Inherited Members

27.100.1 Member Typedef Documentation

27.100.1.1 `template<int TVR> typedef Element<TVR, VM::VM3_n> gdcmm::Element< TVR, VM::VM3_3n >::Parent`

27.100.2 Member Function Documentation

27.100.2.1 `template<int TVR> void gdcmm::Element< TVR, VM::VM3_3n >::SetLength (int len) [inline]`

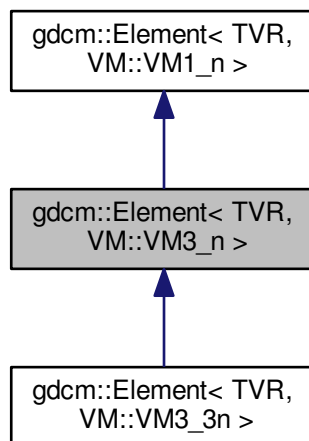
The documentation for this class was generated from the following file:

- [gdcmmElement.h](#)

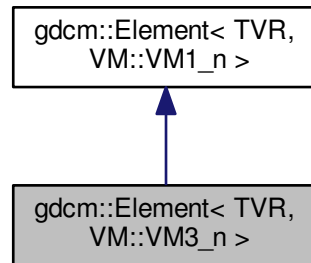
27.101 gdcmm::Element< TVR, VM::VM3_n > Class Template Reference

```
#include <gdcmmElement.h>
```

Inheritance diagram for gdcmm::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

27.101.1 Member Typedef Documentation

27.101.1.1 `template<int TVR> typedef Element<TVR, VM::VM1_n> gdcM::Element< TVR, VM::VM3_n >::Parent`

27.101.2 Member Function Documentation

27.101.2.1 `template<int 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`

27.102 `gdcM::Element< VR::AS, VM::VM5 >` Class Template Reference

```
#include <gdcMElement.h>
```

Public Member Functions

- unsigned long `GetLength` () const
- void `Print` (std::ostream &_os) const

Public Attributes

- char [Internal](#) [[VMToLength](#)< VM::VM5 >::Length *sizeof([VRToType](#)< VR::AS >::Type)]

27.102.1 Member Function Documentation

27.102.1.1 unsigned long [gdcm::Element< VR::AS, VM::VM5 >::GetLength](#) () const [\[inline\]](#)

27.102.1.2 void [gdcm::Element< VR::AS, VM::VM5 >::Print](#) (std::ostream &_os) const [\[inline\]](#)

27.102.2 Member Data Documentation

27.102.2.1 char [gdcm::Element< VR::AS, VM::VM5 >::Internal](#)[[VMToLength](#)< VM::VM5 >::Length *sizeof([VRToType](#)< VR::AS >::Type)]

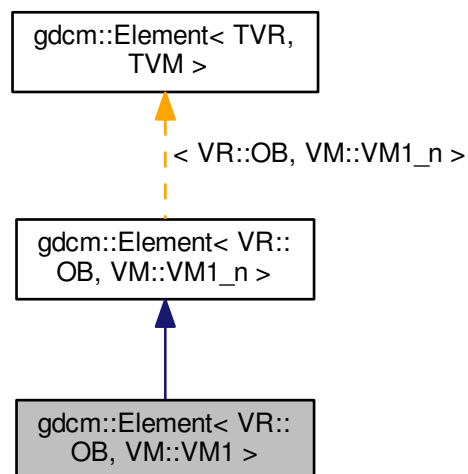
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

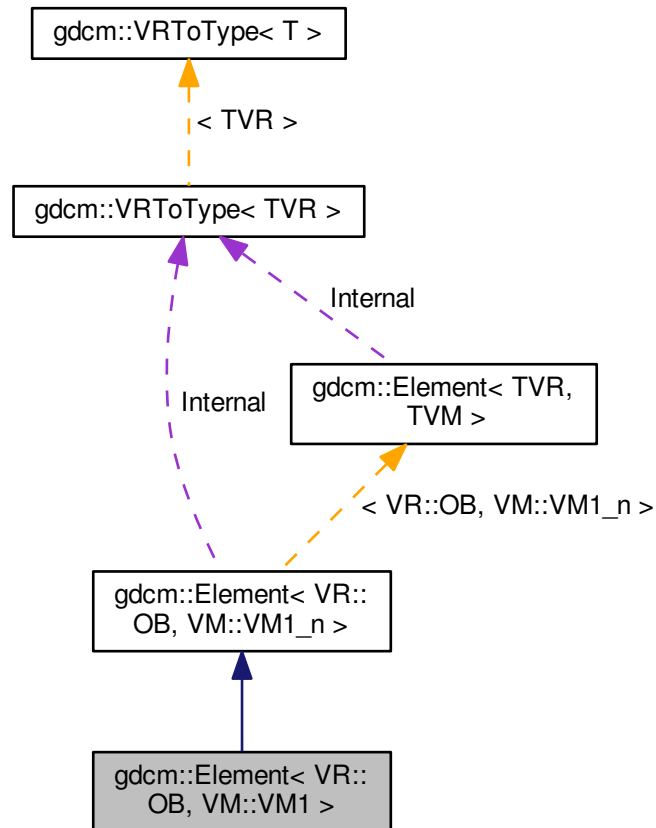
27.103 gdcm::Element< VR::OB, VM::VM1 > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for `gdcm::Element< VR::OB, VM::VM1 >`:



Collaboration diagram for `gdcM::Element< VR::OB, VM::VM1 >`:



Additional Inherited Members

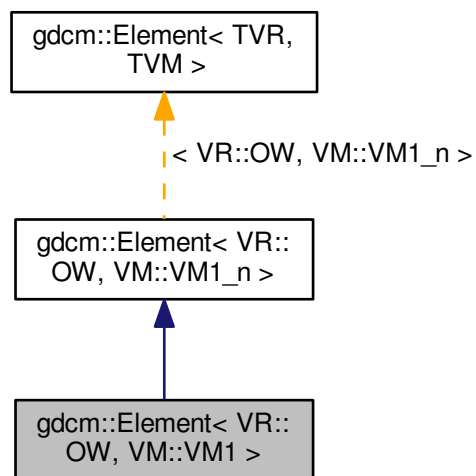
The documentation for this class was generated from the following file:

- [gdcMElement.h](#)

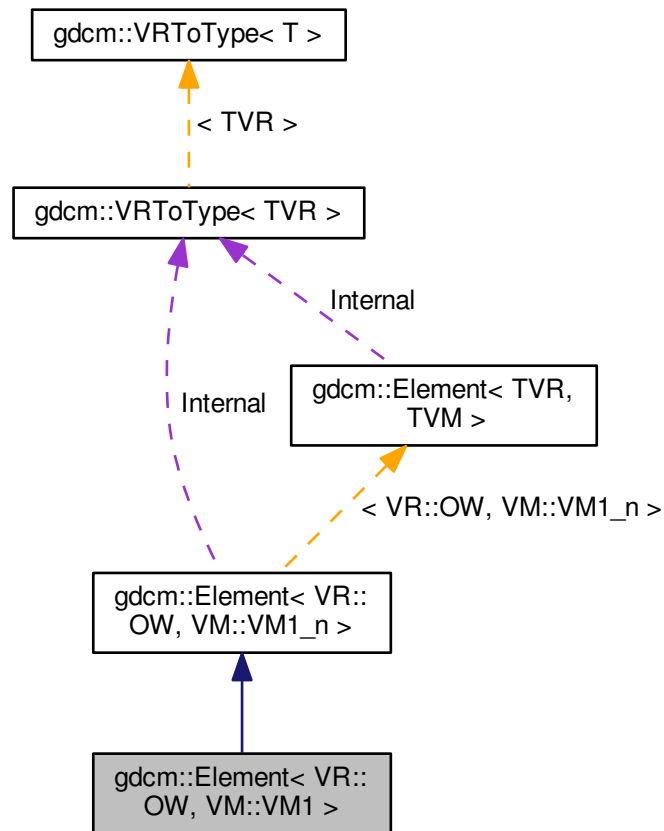
27.104 `gdcM::Element< VR::OW, VM::VM1 >` Class Template Reference

```
#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](#)

27.105 `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>
```

27.105.1 Detailed Description

```
template<int 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](#)

27.106 `gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n >` Class Template Reference

```
#include <gdcmElement.h>
```

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

- [gdcmElement.h](#)

27.107 `gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n >` Class Template Reference

```
#include <gdcmElement.h>
```

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

- [gdcmElement.h](#)

27.108 `gdcm::EncapsulatedDocument` Class Reference

[EncapsulatedDocument.](#)

```
#include <gdcmEncapsulatedDocument.h>
```

Public Member Functions

- [EncapsulatedDocument\(\)](#)

27.108.1 Detailed Description

[EncapsulatedDocument.](#)

27.108.2 Constructor & Destructor Documentation

27.108.2.1 `gdcm::EncapsulatedDocument::EncapsulatedDocument()` `[inline]`

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

- [gdcmEncapsulatedDocument.h](#)

27.109 gdcm::EncodingImplementation< T > Class Template Reference

[EncodingImplementation](#).

```
#include <gdcmElement.h>
```

27.109.1 Detailed Description

```
template<int T>class gdcm::EncodingImplementation< T >
```

[EncodingImplementation](#).

Note

TODO

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

- [gdcmElement.h](#)

27.110 gdcm::EncodingImplementation< VR::VRASCII > Class Template Reference

```
#include <gdcmElement.h>
```

Public Member Functions

- `template<>`
void [Write](#) (const float *data, unsigned long length, std::ostream &_os)
- `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)

27.110.1 Member Function Documentation

27.110.1.1 `template<typename T > static void gdcm::EncodingImplementation< VR::VRASCII >::Read (T * data, unsigned long length, std::istream &_is) [inline], [static]`

27.110.1.2 `template<typename T> static void gdcm::EncodingImplementation< VR::VRASCII >::ReadComputeLength (T * data, unsigned int & length, std::istream & _is) [inline], [static]`

References `gdcm::backslash()`.

27.110.1.3 `template<typename T> static void gdcm::EncodingImplementation< VR::VRASCII >::ReadNoSwap (T * data, unsigned long length, std::istream & _is) [inline], [static]`

27.110.1.4 `template<typename T> static void gdcm::EncodingImplementation< VR::VRASCII >::Write (const T * data, unsigned long length, std::ostream & _os) [inline], [static]`

27.110.1.5 `template<> void gdcm::EncodingImplementation< VR::VRASCII >::Write (const float * data, unsigned long length, std::ostream & _os) [inline]`

References `gdcm::to_string()`.

27.110.1.6 `template<> void gdcm::EncodingImplementation< VR::VRASCII >::Write (const double * data, unsigned long length, std::ostream & _os) [inline]`

References `gdcm::to_string()`.

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

- [gdcmElement.h](#)

27.111 gdcm::EncodingImplementation< VR::VRBINARY > Class Template 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)`

27.111.1 Member Function Documentation

27.111.1.1 `template<typename T> static void gdcm::EncodingImplementation< VR::VRBINARY >::Read (T * data, unsigned long length, std::istream & _is) [inline], [static]`

27.111.1.2 `template<typename T> static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadComputeLength (T * data, unsigned int & length, std::istream & _is) [inline], [static]`

27.111.1.3 `template<typename T> static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadNoSwap (T *
data, unsigned long length, std::istream &_is) [inline],[static]`

27.111.1.4 `template<typename T> static void gdcm::EncodingImplementation< VR::VRBINARY >::Write (const T *
data, unsigned long length, std::ostream &_os) [inline],[static]`

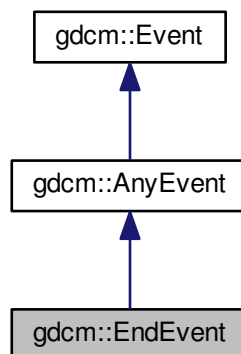
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

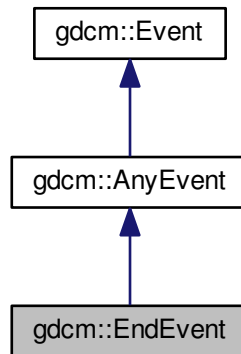
27.112 gdcm::EndEvent Class Reference

```
#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](#)

27.113 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](#) ()

27.113.1 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](#).

27.113.2 Constructor & Destructor Documentation

27.113.2.1 `gdcm::EnumeratedValues::EnumeratedValues ()` `[inline]`

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

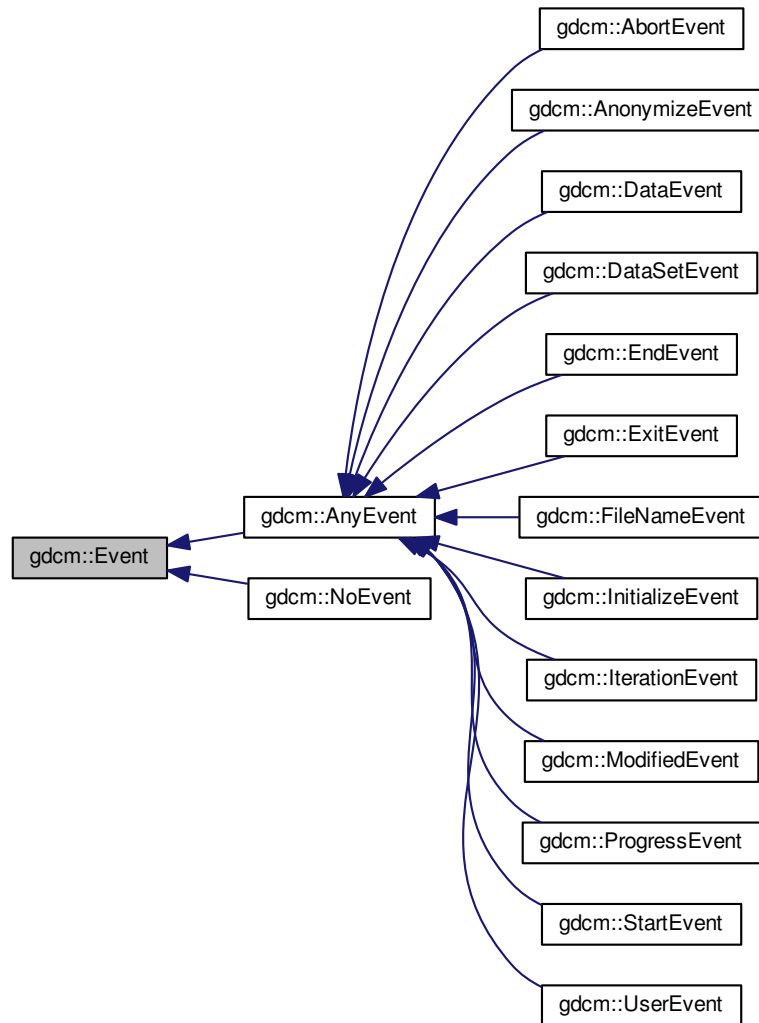
- [gdcmEnumeratedValues.h](#)

27.114 `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](#) (void) const =0
- virtual [Event](#) * [MakeObject](#) () const =0
- virtual void [Print](#) (std::ostream &os) const

27.114.1 Detailed Description

superclass for callback/observer methods

See also

[Command Subject](#)

Examples:

[SimpleScanner.cxx](#).

27.114.2 Constructor & Destructor Documentation

27.114.2.1 `gdcm::Event::Event ()`

27.114.2.2 `gdcm::Event::Event (const Event &)`

27.114.2.3 `virtual gdcm::Event::~~Event ()` `[virtual]`

27.114.3 Member Function Documentation

27.114.3.1 `virtual bool gdcm::Event::CheckEvent (const Event *) const` `[pure virtual]`

Check if given event matches or derives from this event.

27.114.3.2 `virtual const char* gdcm::Event::GetEventName (void) const` `[pure virtual]`

Return the StringName associated with the event.

Implemented in [gdcm::FileNameEvent](#), [gdcm::ProgressEvent](#), [gdcm::DataSetEvent](#), [gdcm::AnonymizeEvent](#), and [gdcm::DataEvent](#).

27.114.3.3 `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::FileNameEvent](#), [gdcm::ProgressEvent](#), [gdcm::DataSetEvent](#), [gdcm::AnonymizeEvent](#), and [gdcm::DataEvent](#).

27.114.3.4 `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:

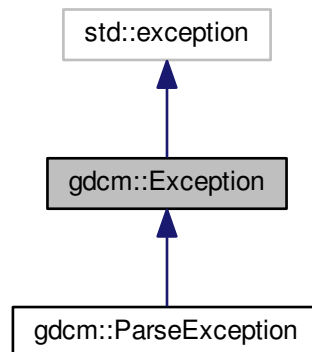
- [gdcmEvent.h](#)

27.115 gdcm::Exception Class Reference

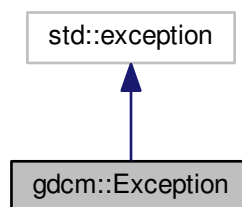
[Exception.](#)

```
#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="")
- virtual [~Exception](#) () throw ()
- const char * [GetDescription](#) () const
Return the Description.
- const char * [what](#) () const throw ()
what implementation

27.115.1 Detailed Description

[Exception.](#)

Standard exception handling object.

Note

Its copy-constructor and assignment operator are generated by the compiler.

27.115.2 Constructor & Destructor Documentation

27.115.2.1 `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.

27.115.2.2 `virtual gdcm::Exception::~~Exception () throw) [inline],[virtual]`

27.115.3 Member Function Documentation

27.115.3.1 `const char* gdcm::Exception::GetDescription () const [inline]`

Return the Description.

Referenced by `gdcm::SequenceOfItems::Read()`.

27.115.3.2 `const char* gdcm::Exception::what () const throw) [inline]`

what implementation

Referenced by `gdcm::SequenceOfFragments::ReadValue()`.

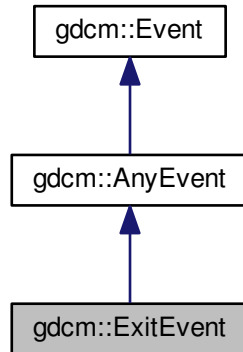
The documentation for this class was generated from the following file:

- [gdcmException.h](#)

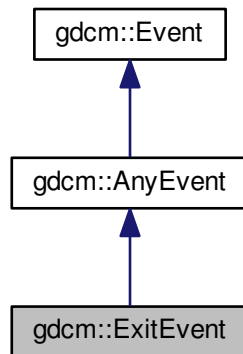
27.116 gdcm::ExitEvent Class Reference

```
#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:

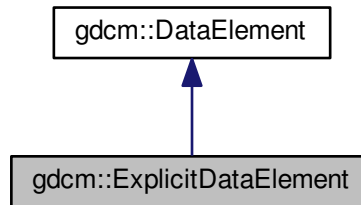
- [gdcmEvent.h](#)

27.117 gdcm::ExplicitDataElement Class Reference

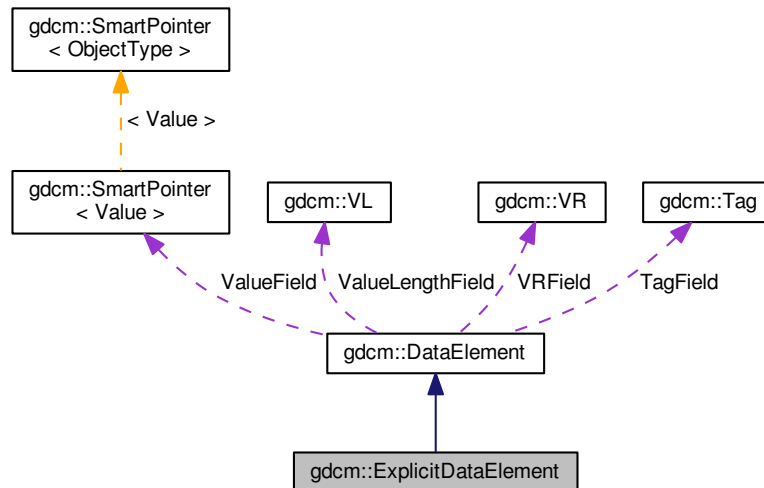
Class to read/write a [DataElement](#) as Explicit Data [Element](#).

```
#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

27.117.1 Detailed Description

Class to read/write a [DataElement](#) as Explicit Data [Element](#).

Note

bla

27.117.2 Member Function Documentation

27.117.2.1 `VL gdcm::ExplicitDataElement::GetLength () const`

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

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

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

27.117.2.5 `template<typename TSwap > std::istream& gdcm::ExplicitDataElement::ReadWithLength (std::istream & is, VL & length)`

27.117.2.6 `template<typename TSwap > const std::ostream& gdcm::ExplicitDataElement::Write (std::ostream & os) const`

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

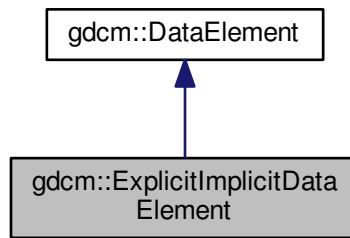
- [gdcmExplicitDataElement.h](#)

27.118 gdcm::ExplicitImplicitDataElement Class Reference

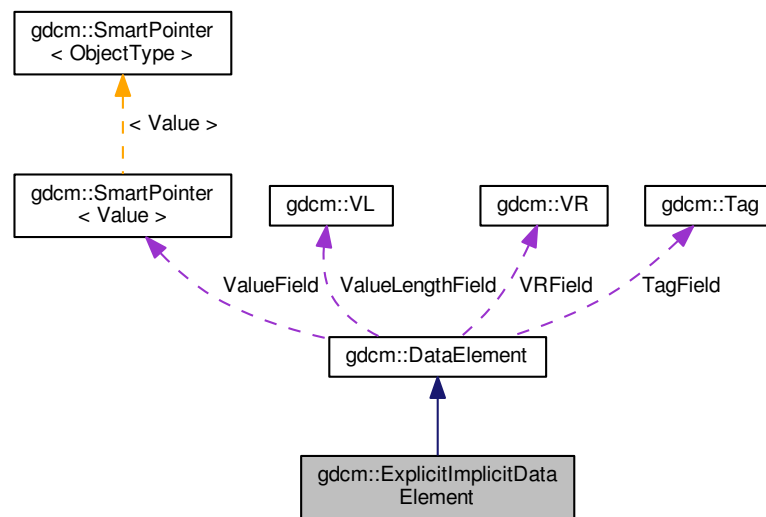
Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#).

```
#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

27.118.1 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.

27.118.2 Member Function Documentation

27.118.2.1 `VL gdcm::ExplicitImplicitDataElement::GetLength () const`

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

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

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

27.118.2.5 `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](#)

27.119 gdcm::Fiducials Class Reference

[Fiducials](#).

```
#include <gdcmFiducials.h>
```

Public Member Functions

- [Fiducials](#) ()

27.119.1 Detailed Description

[Fiducials](#).

27.119.2 Constructor & Destructor Documentation

27.119.2.1 `gdcm::Fiducials::Fiducials ()` `[inline]`

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

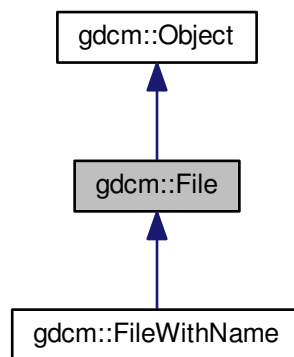
- [gdcmFiducials.h](#)

27.120 `gdcm::File` Class Reference

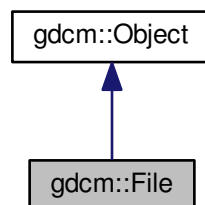
a DICOM [File](#) See PS 3.10 [File](#): A [File](#) is an ordered string of zero or more bytes, where the first byte is at the beginning of the file and the last byte at the end of the [File](#). Files are identified by a unique [File](#) ID and may be written, read and/or deleted.

```
#include <gdcmFile.h>
```

Inheritance diagram for `gdcm::File`:



Collaboration diagram for `gdcm::File`:



Public Member Functions

- [File](#) ()
- [~File](#) ()
- const [DataSet](#) & [GetDataSet](#) () const
Get Data Set.
- [DataSet](#) & [GetDataSet](#) ()
Get Data Set.
- const [FileMetaInformation](#) & [GetHeader](#) () const
Get File Meta Information.
- [FileMetaInformation](#) & [GetHeader](#) ()
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

27.120.1 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](#), [DiffFile.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.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](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadGEMSSDO.cxx](#), and [StreamImageReaderTest.cxx](#).

27.120.2 Constructor & Destructor Documentation

27.120.2.1 `gdcm::File::File ()`

27.120.2.2 `gdcm::File::~~File ()`

27.120.3 Member Function Documentation

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

Get Data Set.

Examples:

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [csa2img.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [rle2img.cxx](#), and [StreamImageReaderTest.cxx](#).

27.120.3.2 `DataSet& gdcm::File::GetDataSet ()` `[inline]`

Get Data Set.

27.120.3.3 `const FileMetaInformation& gdcm::File::GetHeader () const` `[inline]`

Get [File](#) Meta Information.

Examples:

[CreateJPIPDataSet.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](#), [pmsct_rgb1.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [rle2img.cxx](#), and [StreamImageReaderTest.cxx](#).

Referenced by `gdcm::operator<<()`.

27.120.3.4 `FileMetaInformation& gdcm::File::GetHeader ()` `[inline]`

Get [File](#) Meta Information.

27.120.3.5 `std::istream& gdcm::File::Read (std::istream & is)`

Read.

27.120.3.6 void gdcm::File::SetDataSet (const DataSet & ds) [inline]

Set Data Set.

27.120.3.7 void gdcm::File::SetHeader (const FileMetaInformation & fmi) [inline]

Set [File](#) Meta Information.

27.120.3.8 std::ostream const& gdcm::File::Write (std::ostream & os) const

Write.

27.120.4 Friends And Related Function Documentation

27.120.4.1 std::ostream& operator<< (std::ostream & os, const File & val) [friend]

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

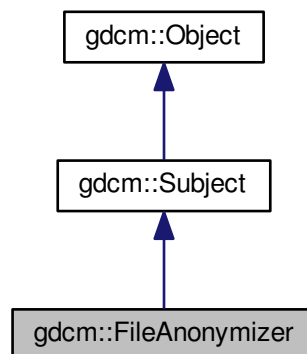
- [gdcmFile.h](#)

27.121 gdcm::FileAnonymizer Class Reference

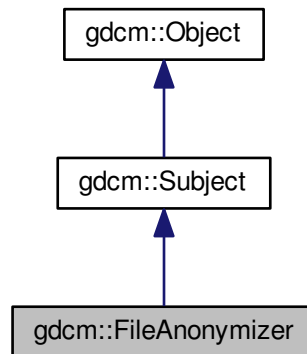
[FileAnonymizer](#).

```
#include <gdcmFileAnonymizer.h>
```

Inheritance diagram for gdcm::FileAnonymizer:



Collaboration diagram for `gdcm::FileAnonymizer`:



Public Member Functions

- [FileAnonymizer](#) ()
- [~FileAnonymizer](#) ()
- 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_str)
- void [Replace](#) ([Tag](#) const &t, const char *value_data, [VL](#) const &vl)
- 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

27.121.1 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](#).

27.121.2 Constructor & Destructor Documentation

27.121.2.1 `gdcmm::FileAnonymizer::FileAnonymizer ()`

27.121.2.2 `gdcmm::FileAnonymizer::~~FileAnonymizer ()`

27.121.3 Member Function Documentation

27.121.3.1 `void gdcmm::FileAnonymizer::Empty (Tag const & t)`

Make [Tag](#) t empty **Warning:** does not handle SQ element

Examples:

[MakeTemplate.cxx](#).

27.121.3.2 `void gdcmm::FileAnonymizer::Remove (Tag const & t)`

remove a tag (even a SQ can be removed)

27.121.3.3 `void gdcmm::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](#) !

27.121.3.4 `void gdcmm::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

27.121.3.5 void `gdcm::FileAnonymizer::SetInputFileName (const char * filename_native)`

Set input filename.

Examples:

[FileAnonymize.cs](#), and [MakeTemplate.cxx](#).

27.121.3.6 void `gdcm::FileAnonymizer::SetOutputFileName (const char * filename_native)`

Set output filename.

Examples:

[MakeTemplate.cxx](#).

27.121.3.7 bool `gdcm::FileAnonymizer::Write ()`

Write the output file.

Examples:

[MakeTemplate.cxx](#).

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

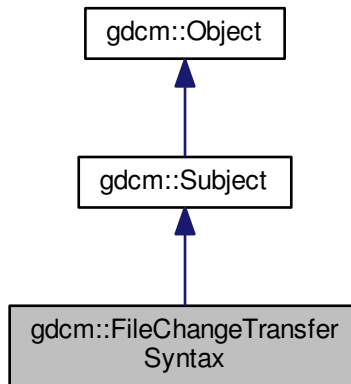
- [gdcmFileAnonymizer.h](#)

27.122 `gdcm::FileChangeTransferSyntax` Class Reference

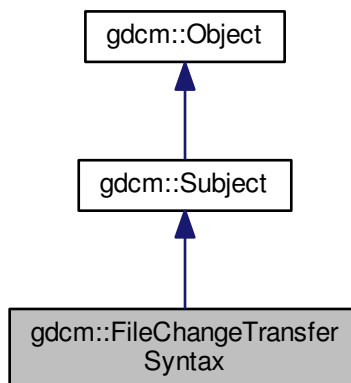
[FileChangeTransferSyntax](#).

```
#include <gdcmFileChangeTransferSyntax.h>
```

Inheritance diagram for gdcm::FileChangeTransferSyntax:



Collaboration diagram for gdcm::FileChangeTransferSyntax:



Public Member Functions

- [FileChangeTransferSyntax](#) ()
- [~FileChangeTransferSyntax](#) ()
- 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

27.122.1 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

27.122.2 Constructor & Destructor Documentation

27.122.2.1 `gdcm::FileChangeTransferSyntax::FileChangeTransferSyntax ()`

27.122.2.2 `gdcm::FileChangeTransferSyntax::~~FileChangeTransferSyntax ()`

27.122.3 Member Function Documentation

27.122.3.1 `bool gdcm::FileChangeTransferSyntax::Change ()`

Change the transfer syntax.

27.122.3.2 `ImageCodec* gdcm::FileChangeTransferSyntax::GetCodec ()`

Retrieve the actual codec (valid after calling SetTransferSyntax) Only advanced users should call this function.

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

for wrapped language: instantiate a reference counted object

27.122.3.4 void gdcm::FileChangeTransferSyntax::SetInputFileName (const char * *filename_native*)

Set input filename (raw DICOM)

27.122.3.5 void gdcm::FileChangeTransferSyntax::SetOutputFileName (const char * *filename_native*)

Set output filename (target compressed DICOM)

27.122.3.6 void gdcm::FileChangeTransferSyntax::SetTransferSyntax (TransferSyntax const & *ts*)

Specify the Target Transfer Syntax.

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

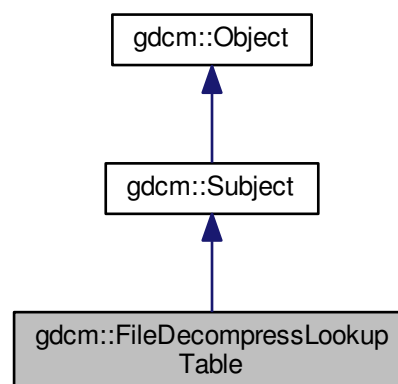
- [gdcmFileChangeTransferSyntax.h](#)

27.123 gdcm::FileDecompressLookupTable Class Reference

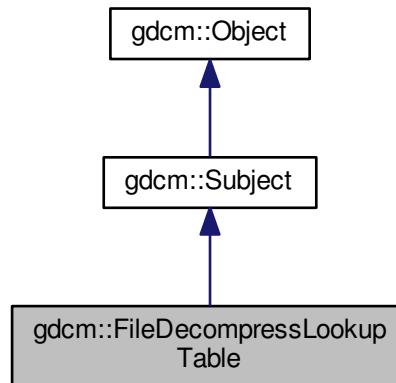
[FileDecompressLookupTable](#) class It decompress the segmented LUT into linearized one (only PALETTE_COLOR images) Output will be a [PhotometricInterpretation](#)=RGB image.

```
#include <gdcmFileDecompressLookupTable.h>
```

Inheritance diagram for gdcm::FileDecompressLookupTable:



Collaboration diagram for `gdcm::FileDecompressLookupTable`:



Public Member Functions

- [FileDecompressLookupTable](#) ()
- [~FileDecompressLookupTable](#) ()
- `bool` [Change](#) ()
Decompress.
- [File](#) & [GetFile](#) ()
- `const` [Pixmap](#) & [GetPixmap](#) () `const`
- [Pixmap](#) & [GetPixmap](#) ()
- `void` [SetFile](#) (`const` [File](#) &`f`)
Set/Get File.
- `void` [SetPixmap](#) ([Pixmap](#) `const` &`img`)

Additional Inherited Members

27.123.1 Detailed Description

[FileDecompressLookupTable](#) class It decompress the segmented LUT into linearized one (only PALETTE_COLOR images) Output will be a [PhotometricInterpretation](#)=RGB image.

27.123.2 Constructor & Destructor Documentation

27.123.2.1 `gdcm::FileDecompressLookupTable::FileDecompressLookupTable ()` `[inline]`

27.123.2.2 `gdcm::FileDecompressLookupTable::~~FileDecompressLookupTable ()` `[inline]`

27.123.3 Member Function Documentation

27.123.3.1 `bool gdcm::FileDecompressLookupTable::Change ()`

Decompress.

27.123.3.2 `File& gdcm::FileDecompressLookupTable::GetFile () [inline]`

27.123.3.3 `const Pixmap& gdcm::FileDecompressLookupTable::GetPixmap () const [inline]`

27.123.3.4 `Pixmap& gdcm::FileDecompressLookupTable::GetPixmap () [inline]`

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

Set/Get [File](#).

27.123.3.6 `void gdcm::FileDecompressLookupTable::SetPixmap (Pixmap const & img) [inline]`

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

- [gdcmFileDecompressLookupTable.h](#)

27.124 gdcm::FileDerivation Class Reference

[FileDerivation](#) class See PS 3.16 - 2008 For the list of Code [Value](#) that can be used for in Derivation Code Sequence.

```
#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 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 ()`

27.124.1 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

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:

[GenFakelImage.cxx](#).

27.124.2 Constructor & Destructor Documentation

27.124.2.1 `gdcm::FileDerivation::FileDerivation ()`

27.124.2.2 `gdcm::FileDerivation::~~FileDerivation ()`

27.124.3 Member Function Documentation

27.124.3.1 `bool gdcm::FileDerivation::AddDerivationDescription ()` [protected]

27.124.3.2 `bool gdcm::FileDerivation::AddPurposeOfReferenceCodeSequence (DataSet & ds)` [protected]

27.124.3.3 `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:

[GenFakelImage.cxx](#).

27.124.3.4 `bool gdcm::FileDerivation::AddSourceImageSequence ()` [protected]

27.124.3.5 `bool gdcm::FileDerivation::Derive ()`

Change.

Examples:

[GenFakelImage.cxx](#).

27.124.3.6 `File& gdcm::FileDerivation::GetFile () [inline]`

Examples:

[GenFakelImage.cxx](#).

27.124.3.7 `const File& gdcm::FileDerivation::GetFile () const [inline]`

27.124.3.8 `void gdcm::FileDerivation::SetDerivationCodeSequenceCodeValue (unsigned int codevalue)`

Specify the Derivation Code Sequence Code [Value](#). Eg 113040.

Examples:

[GenFakelImage.cxx](#).

27.124.3.9 `void gdcm::FileDerivation::SetDerivationDescription (const char * dd)`

Specify the Derivation Description. Eg "lossy conversion".

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

Set/Get [File](#).

Examples:

[GenFakelImage.cxx](#).

27.124.3.11 `void gdcm::FileDerivation::SetPurposeOfReferenceCodeSequenceCodeValue (unsigned int codevalue)`

Specify the Purpose Of Reference Code [Value](#). Eg. 121320.

Examples:

[GenFakelImage.cxx](#).

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

- [gdcmFileDerivation.h](#)

27.125 gdcm::FileExplicitFilter Class Reference

[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.

```
#include <gdcmFileExplicitFilter.h>
```

Public Member Functions

- [FileExplicitFilter](#) ()
- [~FileExplicitFilter](#) ()
- 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)

27.125.1 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](#).

27.125.2 Constructor & Destructor Documentation

27.125.2.1 `gdcm::FileExplicitFilter::FileExplicitFilter () [inline]`

27.125.2.2 `gdcm::FileExplicitFilter::~~FileExplicitFilter () [inline]`

27.125.3 Member Function Documentation

27.125.3.1 `bool gdcm::FileExplicitFilter::Change ()`

Set FMI Transfer Syntax.

Change

Examples:

[GenAllVR.cxx](#), and [LargeVRDSExplicit.cxx](#).

27.125.3.2 `bool gdcm::FileExplicitFilter::ChangeFMI ()` `[protected]`

27.125.3.3 `File& gdcm::FileExplicitFilter::GetFile ()` `[inline]`

27.125.3.4 `bool gdcm::FileExplicitFilter::ProcessDataSet (DataSet & ds, Dicts const & dicts)` `[protected]`

27.125.3.5 `void gdcm::FileExplicitFilter::SetChangePrivateTags (bool b)` `[inline]`

Decide whether or not to [VR](#)ify private tags.

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

Set/Get [File](#).

Examples:

[GenAllVR.cxx](#), and [LargeVRDSExplicit.cxx](#).

27.125.3.7 `void gdcm::FileExplicitFilter::SetRecomputeItemLength (bool b)`

By default set Sequence & [Item](#) length to Undefined to avoid recomputing length:

27.125.3.8 `void gdcm::FileExplicitFilter::SetRecomputeSequenceLength (bool b)`

27.125.3.9 `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:

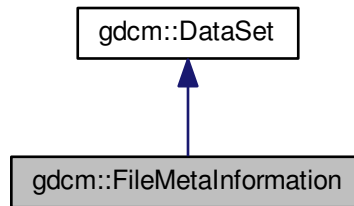
- [gdcmFileExplicitFilter.h](#)

27.126 gdcm::FileMetaInformation Class Reference

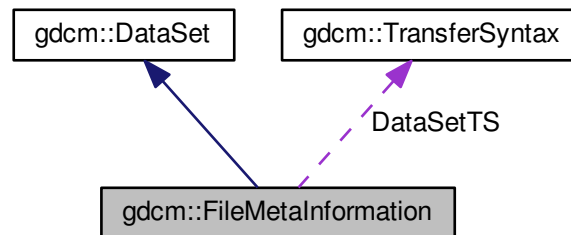
Class to represent a [File](#) Meta Information.

```
#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::NegotiatedType](#) [GetMetaInformationTS](#) () const
- const [Preamble](#) & [GetPreamble](#) () const
 - Get [Preamble](#).*
- [Preamble](#) & [GetPreamble](#) ()
- 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

27.126.1 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:

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), and [ReadAndDumpDICOM↵DIR.cxx](#).

27.126.2 Constructor & Destructor Documentation

27.126.2.1 `gdcm::FileMetaInformation::FileMetaInformation ()`

27.126.2.2 `gdcm::FileMetaInformation::~~FileMetaInformation ()`

27.126.2.3 `gdcm::FileMetaInformation::FileMetaInformation (FileMetaInformation const & fmi)` `[inline]`

References [DataSetMS](#), [DataSetTS](#), and [MetaInformationTS](#).

27.126.3 Member Function Documentation

27.126.3.1 `static void gdcm::FileMetaInformation::AppendImplementationClassUID (const char * imp)` `[static]`

27.126.3.2 `void gdcm::FileMetaInformation::ComputeDataSetMediaStorageSOPClass ()` `[protected]`

27.126.3.3 `void gdcm::FileMetaInformation::ComputeDataSetTransferSyntax ()` `[protected]`

27.126.3.4 `void gdcm::FileMetaInformation::Default ()` `[protected]`

27.126.3.5 `void gdcm::FileMetaInformation::FillFromDataSet (DataSet const & ds)`

Construct a [FileMetaInformation](#) from an already existing [DataSet](#):

27.126.3.6 `const TransferSyntax& gdcm::FileMetaInformation::GetDataSetTransferSyntax () const` `[inline]`

Examples:

[GetJPEGSamplePrecision.cxx](#), and [MergeTwoFiles.cxx](#).

27.126.3.7 `static const char* gdcm::FileMetaInformation::GetFileMetaInformationVersion () [static],[protected]`

27.126.3.8 `VL gdcm::FileMetaInformation::GetFullLength () const [inline]`

References `gdcm::VL::GetLength()`.

27.126.3.9 `static const char* gdcm::FileMetaInformation::GetGDCMImplementationClassUID () [static],[protected]`

27.126.3.10 `static const char* gdcm::FileMetaInformation::GetGDCMImplementationVersionName () [static],[protected]`

27.126.3.11 `static const char* gdcm::FileMetaInformation::GetGDCMSourceApplicationEntityTitle () [static],[protected]`

27.126.3.12 `static const char* gdcm::FileMetaInformation::GetImplementationClassUID () [static]`

27.126.3.13 `static const char* gdcm::FileMetaInformation::GetImplementationVersionName () [static]`

27.126.3.14 `MediaStorage gdcm::FileMetaInformation::GetMediaStorage () const`

27.126.3.15 `std::string gdcm::FileMetaInformation::GetMediaStorageAsString () const`

27.126.3.16 `TransferSyntax::NegociatedType gdcm::FileMetaInformation::GetMetaInformationTS () const [inline]`

27.126.3.17 `const Preamble& gdcm::FileMetaInformation::GetPreamble () const [inline]`

Get [Preamble](#).

Referenced by `gdcm::operator<<()`.

27.126.3.18 `Preamble& gdcm::FileMetaInformation::GetPreamble () [inline]`

27.126.3.19 `static const char* gdcm::FileMetaInformation::GetSourceApplicationEntityTitle () [static]`

27.126.3.20 `void gdcm::FileMetaInformation::Insert (const DataElement & de) [inline]`

References `gdcmErrorMacro`, `gdcm::Tag::GetGroup()`, and `gdcm::DataElement::GetTag()`.

27.126.3.21 `bool gdcm::FileMetaInformation::IsValid () const [inline]`

27.126.3.22 `std::istream& gdcm::FileMetaInformation::Read (std::istream & is)`

Read.

27.126.3.23 `std::istream& gdcm::FileMetaInformation::ReadCompat (std::istream & is)`

27.126.3.24 `template<typename TSwap > std::istream& gdcm::FileMetaInformation::ReadCompatInternal (std::istream & is) [protected]`

27.126.3.25 `void gdcm::FileMetaInformation::Replace (const DataElement & de) [inline]`

Examples:

[LargeVRDSExplicit.cxx](#).

References `gdcm::DataElement::GetTag()`.

27.126.3.26 `void gdcm::FileMetaInformation::SetDataSetTransferSyntax (const TransferSyntax & ts)`

Examples:

[CreateJPIPDataSet.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](#), [pmsct_rgb1.cxx](#), [QIDO-RS.cxx](#), [rle2img.cxx](#), and [StreamImageReaderTest.cxx](#).

27.126.3.27 `static void gdcm::FileMetaInformation::SetImplementationClassUID (const char * imp) [static]`

Override the GDCM default values:

27.126.3.28 `static void gdcm::FileMetaInformation::SetImplementationVersionName (const char * version) [static]`

27.126.3.29 `void gdcm::FileMetaInformation::SetPreamble (const Preamble & p) [inline]`

27.126.3.30 `static void gdcm::FileMetaInformation::SetSourceApplicationEntityTitle (const char * title) [static]`

Examples:

[ClinicalTrialIdentificationWorkflow.cs](#), [FixJAIBugJPEGLS.cxx](#), [GenerateDICOMDIR.cs](#), [ReformatFile.cs](#), and [StandardizeFiles.cs](#).

27.126.3.31 `std::ostream& gdcm::FileMetaInformation::Write (std::ostream & os) const`

Write.

27.126.4 Friends And Related Function Documentation

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

27.126.5 Member Data Documentation

27.126.5.1 `MediaStorage::MSType gdcm::FileMetaInformation::DataSetMS [protected]`

Referenced by `FileMetaInformation()`.

27.126.5.2 `TransferSyntax gdcm::FileMetaInformation::DataSetTS [protected]`

Referenced by `FileMetaInformation()`.

27.126.5.3 TransferSyntax::NegotiatedType gdcm::FileMetaInformation::MetaInformationTS [protected]

Referenced by FileMetaInformation().

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

- [gdcmFileMetaInformation.h](#)

27.127 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 toward slash (UNIX style) to windows style slash.

Static Public Member Functions

- static const char * [Join](#) (const char *path, const char *filename)

27.127.1 Detailed Description

Class to manipulate file name's.

Note

OS independant representation of a filename (to query path, name and extension from a filename)

27.127.2 Constructor & Destructor Documentation

27.127.2.1 `gdcm::Filename::Filename (const char * filename = " ") [inline]`

27.127.3 Member Function Documentation

27.127.3.1 `bool gdcm::Filename::EndWith (const char ending[]) const`

Does the filename ends with a particular string ?

27.127.3.2 `const char* gdcm::Filename::GetExtension ()`

return only the extension part of a filename

27.127.3.3 `const char* gdcm::Filename::GetFileName () const [inline]`

Return the full filename.

27.127.3.4 `const char* gdcm::Filename::GetName ()`

return only the name part of a filename

27.127.3.5 `const char* gdcm::Filename::GetPath ()`

Return only the path component of a filename.

Examples:

[ClinicalTrialIdentificationWorkflow.cs](#).

27.127.3.6 `bool gdcm::Filename::IsEmpty () const [inline]`

return whether the filename is empty

27.127.3.7 `bool gdcm::Filename::IsIdentical (Filename const & fn) const`

27.127.3.8 `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](#).

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

Simple operator to allow [Filename](#) myfilename("...") ; const char * s = myfilename;

27.127.3.10 `const char* gdcm::Filename::ToUnixSlashes ()`

Convert backslash (windows style) to UNIX style slash.

27.127.3.11 `const char* gdcm::Filename::ToWindowsSlashes ()`

Convert foward slash (UNIX style) to windows style slash.

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

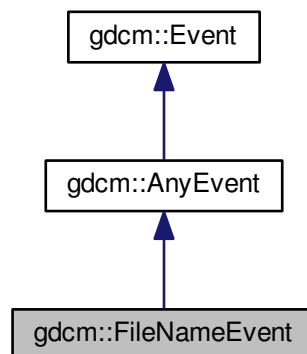
- [gdcmFilename.h](#)

27.128 gdcm::FileNameEvent Class Reference

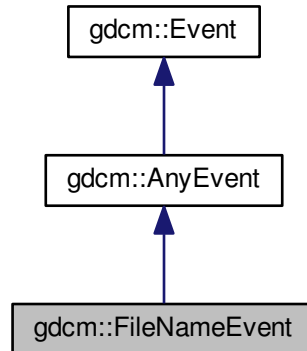
FileNameEvent Special type of event triggered during processing of **FileSet**.

```
#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`)
- virtual [~FileNameEvent](#) (`()`)
- virtual `bool` [CheckEvent](#) (`const ::gdcm::Event *e`) `const`
- virtual `const char *` [GetEventName](#) (`() const`)
- `const char *` [GetFileName](#) (`() const`)
- virtual `::gdcm::Event *` [MakeObject](#) (`() const`)
- void [SetFileName](#) (`const char *f`)

27.128.1 Detailed Description

[FileNameEvent](#) Special type of event triggered during processing of [FileSet](#).

See also

[AnyEvent](#)

Examples:

[SimpleScanner.cxx](#).

27.128.2 Member Typedef Documentation

27.128.2.1 `typedef FileNameEvent gdcm::FileNameEvent::Self`

27.128.2.2 `typedef AnyEvent gdcm::FileNameEvent::Superclass`

27.128.3 Constructor & Destructor Documentation

27.128.3.1 `gdcm::FileNameEvent::FileNameEvent (const char * s = " ") [inline]`

27.128.3.2 `virtual gdcm::FileNameEvent::~~FileNameEvent () [inline],[virtual]`

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

27.128.4 Member Function Documentation

27.128.4.1 `virtual bool gdcm::FileNameEvent::CheckEvent (const ::gdcm::Event * e) const [inline],[virtual]`

27.128.4.2 `virtual const char* gdcm::FileNameEvent::GetEventName () const [inline],[virtual]`

Return the StringName associated with the event.

Implements [gdcm::Event](#).

27.128.4.3 `const char* gdcm::FileNameEvent::GetFileName () const [inline]`

Examples:

[SimpleScanner.cxx](#).

27.128.4.4 `virtual ::gdcm::Event* gdcm::FileNameEvent::MakeObject () const [inline],[virtual]`

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

Implements [gdcm::Event](#).

27.128.4.5 `void gdcm::FileNameEvent::SetFileName (const char * f) [inline]`

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

- [gdcmFileNameEvent.h](#)

27.129 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](#) ()
- 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.

27.129.1 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](#).

27.129.2 Member Typedef Documentation

27.129.2.1 typedef std::vector<[FilenameType](#)> [gdcm::FilenameGenerator::FileNamesType](#)

27.129.2.2 typedef std::string [gdcm::FilenameGenerator::FilenameType](#)

27.129.2.3 typedef [FileNamesType](#)::size_type [gdcm::FilenameGenerator::SizeType](#)

27.129.3 Constructor & Destructor Documentation

27.129.3.1 `gdcm::FilenameGenerator::FilenameGenerator ()` `[inline]`

27.129.3.2 `gdcm::FilenameGenerator::~~FilenameGenerator ()` `[inline]`

27.129.4 Member Function Documentation

27.129.4.1 `bool gdcm::FilenameGenerator::Generate ()`

Generate (return success)

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

27.129.4.2 `const char* gdcm::FilenameGenerator::GetFilename (SizeType n) const`

Get a particular filename (call after Generate)

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

27.129.4.3 `FilenameType const& gdcm::FilenameGenerator::GetFilenames () const` `[inline]`

Return all filenames.

27.129.4.4 `SizeType gdcm::FilenameGenerator::GetNumberOfFilenames () const`

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

27.129.4.5 `const char* gdcm::FilenameGenerator::GetPattern () const` `[inline]`

27.129.4.6 `const char* gdcm::FilenameGenerator::GetPrefix () const` `[inline]`

27.129.4.7 `void gdcm::FilenameGenerator::SetNumberOfFilenames (SizeType nfiles)`

Set/Get the number of filenames to generate.

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

27.129.4.8 `void gdcm::FilenameGenerator::SetPattern (const char * pattern)` `[inline]`

Set/Get pattern.

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

27.129.4.9 void `gdcm::FilenameGenerator::SetPrefix (const char * prefix)` `[inline]`

Set/Get prefix.

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

- [gdcmFilenameGenerator.h](#)

27.130 gdcm::FileSet Class Reference

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.

```
#include <gdcmFileSet.h>
```

Public Types

- typedef std::vector< [FileType](#) > [FilesType](#)
- typedef std::string [FileType](#)

Public Member Functions

- [FileSet](#) ()
- void [AddFile](#) ([File](#) const &)
- bool [AddFile](#) (const char *filename)
- [FilesType](#) const & [GetFiles](#) () const
- void [SetFiles](#) ([FilesType](#) const &files)

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [FileSet](#) &d)

27.130.1 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.

27.130.2 Member Typedef Documentation

27.130.2.1 typedef std::vector<[FileType](#)> [gdcm::FileSet::FilesType](#)

27.130.2.2 typedef std::string [gdcm::FileSet::FileType](#)

27.130.3 Constructor & Destructor Documentation

27.130.3.1 [gdcm::FileSet::FileSet \(\)](#) `[inline]`

27.130.4 Member Function Documentation

27.130.4.1 void gdcm::FileSet::AddFile (File const &) [inline]

Deprecated . Does nothing

27.130.4.2 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.

27.130.4.3 FileType const& gdcm::FileSet::GetFiles () const [inline]

27.130.4.4 void gdcm::FileSet::SetFiles (FileType const & files)

27.130.5 Friends And Related Function Documentation

27.130.5.1 std::ostream& operator<< (std::ostream &_os, const FileSet &d) [friend]

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

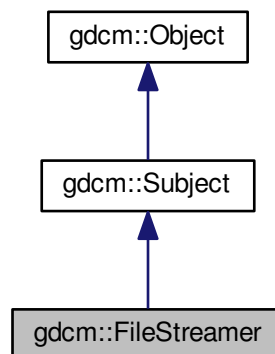
- [gdcmFileSet.h](#)

27.131 gdcm::FileStreamer Class Reference

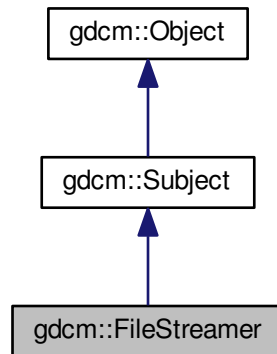
FileStreamer This class let a user create a massive DICOM [DataSet](#) from a template DICOM file, by appending chunks of data.

```
#include <gdcmFileStreamer.h>
```

Inheritance diagram for gdcm::FileStreamer:



Collaboration diagram for `gdcm::FileStreamer`:



Public Member Functions

- [FileStreamer](#) ()
- [~FileStreamer](#) ()
- 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

27.131.1 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.

27.131.2 Constructor & Destructor Documentation

27.131.2.1 `gdcm::FileStreamer::FileStreamer ()`

27.131.2.2 `gdcm::FileStreamer::~~FileStreamer ()`

27.131.3 Member Function Documentation

27.131.3.1 `bool gdcm::FileStreamer::AppendToDataElement (const Tag & t, const char * array, size_t len)`

Append to previously started [Tag](#) t.

27.131.3.2 `bool gdcm::FileStreamer::AppendToGroupDataElement (const PrivateTag & pt, const char * array, size_t len)`

Append to previously started private creator.

27.131.3.3 `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

27.131.3.4 `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.

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

for wrapped language: instantiate a reference counted object

27.131.3.6 `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

27.131.3.7 `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

27.131.3.8 `void gdcM::FileStreamer::SetOutputFileName (const char * filename_native)`

Set output filename (target file)

27.131.3.9 `void gdcM::FileStreamer::SetTemplateFileName (const char * filename_native)`

Set input DICOM template filename.

Examples:

[FileStreaming.cs](#).

27.131.3.10 `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.

27.131.3.11 `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). Value should be in [0x0, 0xff] This will find the first available private creator.
----------------	--

27.131.3.12 `bool gdcM::FileStreamer::StopDataElement (const Tag & t)`

Stop appending to tag t. This will compute the proper attribute length.

27.131.3.13 `bool gdcM::FileStreamer::StopGroupDataElement (const PrivateTag & pt)`

Stop appending to private creator.

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

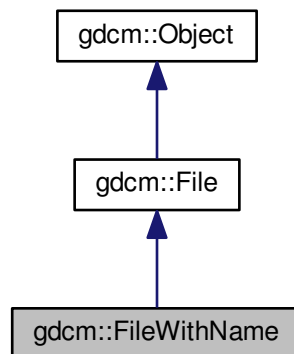
- [gdcMFileStreamer.h](#)

27.132 gdcm::FileWithName Class Reference

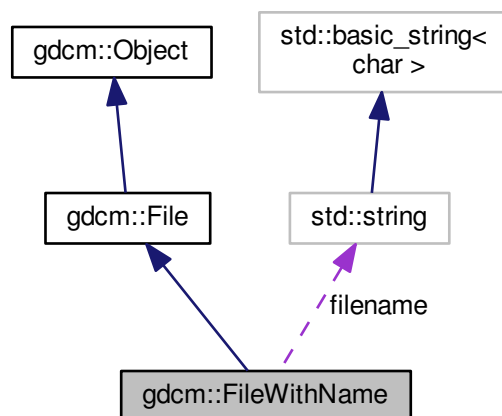
[FileWithName](#).

```
#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

27.132.1 Detailed Description

[FileWithName.](#)

Backward only class do not use in newer code

27.132.2 Constructor & Destructor Documentation

27.132.2.1 `gdcm::FileWithName::FileWithName (File & f)` `[inline]`

27.132.3 Member Data Documentation

27.132.3.1 `std::string gdcm::FileWithName::filename`

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

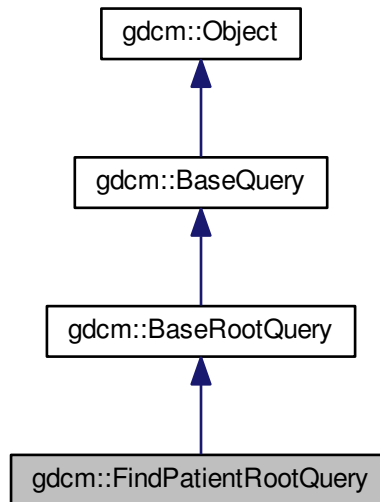
- [gdcmSerieHelper.h](#)

27.133 gdcm::FindPatientRootQuery Class Reference

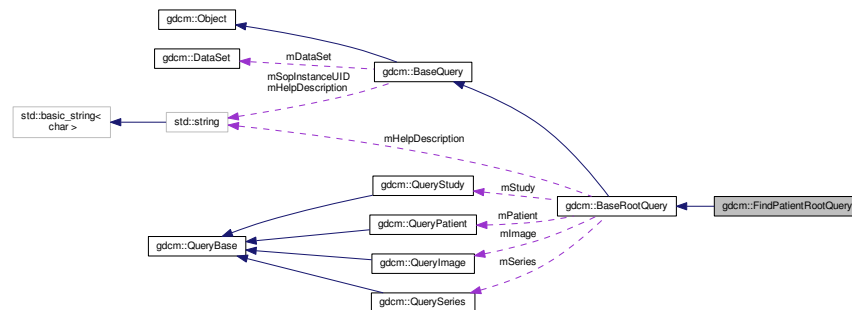
PatientRootQuery contains: the class which will produce a dataset for c-find with patient root.

```
#include <gdcmFindPatientRootQuery.h>
```

Inheritance diagram for gdcm::FindPatientRootQuery:



Collaboration diagram for gdcm::FindPatientRootQuery:



Public Member Functions

- [FindPatientRootQuery](#) ()
- [UIDs::TSName GetAbstractSyntaxUID](#) () const
- [std::vector< Tag > GetTagListByLevel](#) (const [EQueryLevel](#) &inQueryLevel)
- void [InitializeDataSet](#) (const [EQueryLevel](#) &inQueryLevel)
- bool [ValidateQuery](#) (bool inStrict=true) const

Friends

- class [QueryFactory](#)

Additional Inherited Members

27.133.1 Detailed Description

PatientRootQuery contains: the class which will produce a dataset for c-find with patient root.

27.133.2 Constructor & Destructor Documentation

27.133.2.1 `gdcm::FindPatientRootQuery::FindPatientRootQuery ()`

27.133.3 Member Function Documentation

27.133.3.1 `UIDs::TSName gdcm::FindPatientRootQuery::GetAbstractSyntaxUID () const [virtual]`

Implements [gdcm::BaseQuery](#).

27.133.3.2 `std::vector<Tag> gdcm::FindPatientRootQuery::GetTagListByLevel (const EQueryLevel & inQueryLevel) [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](#).

27.133.3.3 `void gdcm::FindPatientRootQuery::InitializeDataSet (const EQueryLevel & inQueryLevel) [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

Implements [gdcm::BaseRootQuery](#).

27.133.3.4 `bool gdcm::FindPatientRootQuery::ValidateQuery (bool inStrict = true) const [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](#).

27.133.4 Friends And Related Function Documentation

27.133.4.1 friend class QueryFactory [friend]

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

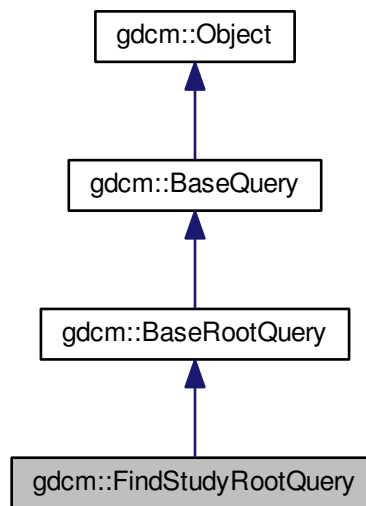
- [gdcmFindPatientRootQuery.h](#)

27.134 gdcm::FindStudyRootQuery Class Reference

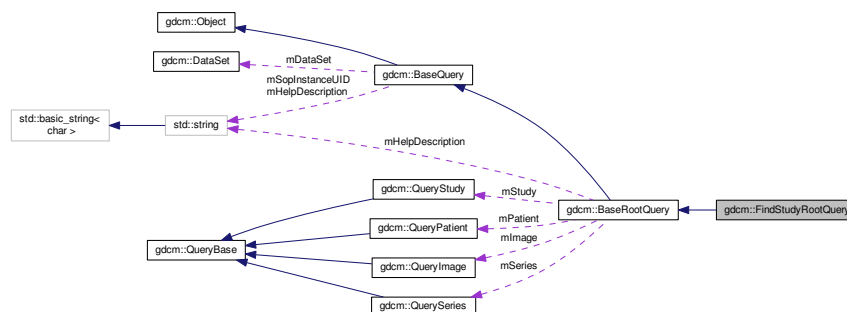
[FindStudyRootQuery](#) contains: the class which will produce a dataset for C-FIND with study root.

```
#include <gdcmFindStudyRootQuery.h>
```

Inheritance diagram for gdcm::FindStudyRootQuery:



Collaboration diagram for gdcm::FindStudyRootQuery:



Public Member Functions

- [FindStudyRootQuery](#) ()
- [UIDs::TSName GetAbstractSyntaxUID](#) () const
- `std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)`
- `void InitializeDataSet (const EQueryLevel &inQueryLevel)`
- `bool ValidateQuery (bool inStrict=true) const`

Friends

- class [QueryFactory](#)

Additional Inherited Members

27.134.1 Detailed Description

[FindStudyRootQuery](#) contains: the class which will produce a dataset for C-FIND with study root.

27.134.2 Constructor & Destructor Documentation

27.134.2.1 `gdcm::FindStudyRootQuery::FindStudyRootQuery ()`

27.134.3 Member Function Documentation

27.134.3.1 `UIDs::TSName gdcm::FindStudyRootQuery::GetAbstractSyntaxUID () const` `[virtual]`

Implements [gdcm::BaseQuery](#).

27.134.3.2 `std::vector<Tag> gdcm::FindStudyRootQuery::GetTagListByLevel (const EQueryLevel & inQueryLevel)`
`[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](#).

27.134.3.3 `void gdcm::FindStudyRootQuery::InitializeDataSet (const EQueryLevel & inQueryLevel)` `[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](#).

27.134.3.4 `bool gdcm::FindStudyRootQuery::ValidateQuery (bool inStrict =true) const` `[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](#).

27.134.4 Friends And Related Function Documentation

27.134.4.1 friend class QueryFactory [friend]

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

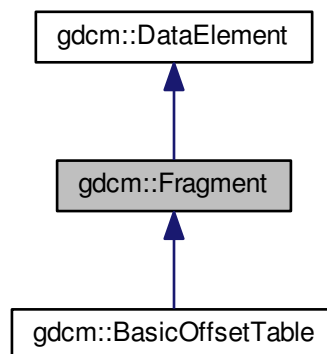
- [gdcmFindStudyRootQuery.h](#)

27.135 gdcm::Fragment Class Reference

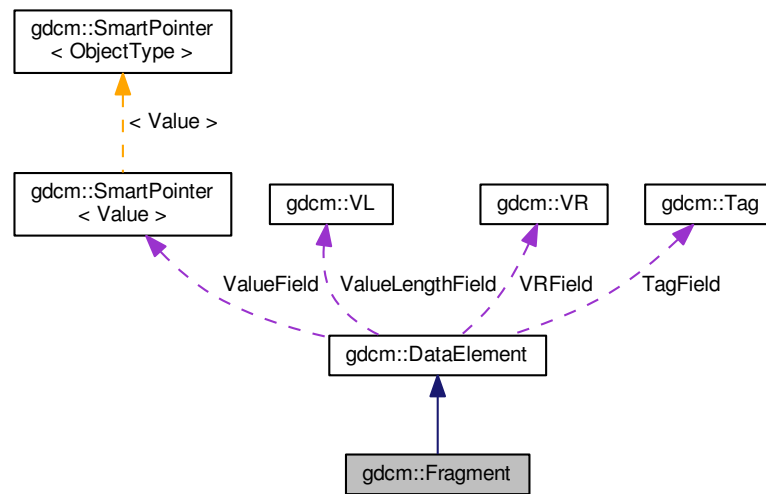
Class to represent a [Fragment](#).

```
#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

27.135.1 Detailed Description

Class to represent a [Fragment](#).

Examples:

[FixBrokenJ2K.cxx](#), and [FixJAIBugJPEGLS.cxx](#).

27.135.2 Constructor & Destructor Documentation

27.135.2.1 `gdcm::Fragment::Fragment ()` `[inline]`

27.135.3 Member Function Documentation

27.135.3.1 `VL gdcm::Fragment::ComputeLength ()` `const`

27.135.3.2 `VL gdcm::Fragment::GetLength ()` `const`

27.135.3.3 `template<typename TSwap> std::istream& gdcm::Fragment::Read (std::istream & is)` `[inline]`

Referenced by `gdcm::SequenceOfFragments::ReadValue()`.

27.135.3.4 `template<typename TSwap> std::istream& gdcm::Fragment::ReadBacktrack (std::istream & is)` `[inline]`

References `gdcmErrorMacro`, `gdcmWarningMacro`, and `gdcm::ParseException::SetLastElement()`.

Referenced by `gdcm::SequenceOfFragments::ReadValue()`.

27.135.3.5 `template<typename TSwap> std::istream& gdcm::Fragment::ReadPreValue (std::istream & is)` `[inline]`

27.135.3.6 `template<typename TSwap> std::istream& gdcm::Fragment::ReadValue (std::istream & is)` `[inline]`

References `gdcmWarningMacro`, and `gdcm::ParseException::SetLastElement()`.

27.135.3.7 `template<typename TSwap> std::ostream& gdcm::Fragment::Write (std::ostream & os)` `const` `[inline]`

References `gdcm::ByteValue::ComputeLength()`, `gdcm::ByteValue::GetLength()`, `gdcm::VL::Write()`, and `gdcm::ByteValue::Write()`.

27.135.4 Friends And Related Function Documentation

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

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

- [gdcmFragment.h](#)

27.136 gdcm::Global Class Reference

[Global](#).

```
#include <gdcmGlobal.h>
```

Public Member Functions

- [Global](#) ()
- [~Global](#) ()
- bool [Append](#) (const char *path)
- [Defs](#) const & [GetDefs](#) () const
- [Dicts](#) const & [GetDicts](#) () const
- [Dicts](#) & [GetDicts](#) ()
- bool [LoadResourcesFiles](#) ()
- 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)

27.136.1 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](#), [GenerateStandardSOPClasses.cxx](#), [GenFakeldentifyFile.cxx](#), [PublicDict.cxx](#), [ReadAndPrintAttributes.cxx](#), and [TraverseModules.cxx](#).

27.136.2 Constructor & Destructor Documentation

27.136.2.1 `gdcmm::Global::Global ()`

27.136.2.2 `gdcmm::Global::~~Global ()`

27.136.3 Member Function Documentation

27.136.3.1 `bool gdcmm::Global::Append (const char * path)`

Append path at the end of the path list

Warning

not thread safe !

27.136.3.2 Defs const& gdcM::Global::GetDefs () const

retrieve the default/internal (Part 3) You need to explicitly call LoadResourcesFiles before

Examples:

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

27.136.3.3 Dicts const& gdcM::Global::GetDicts () const

retrieve the default/internal dicts (Part 6) This dict is filled up at load time

Examples:

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [MrProtocol.cxx](#), [PublicDict.cxx](#), [ReadAndPrintAttributes.cxx](#), and [TraverseModules.cxx](#).

27.136.3.4 Dicts& gdcM::Global::GetDicts ()**27.136.3.5 static Global& gdcM::Global::GetInstance () [static]**

return the singleton instance

Examples:

[BasicAnonymizer.cs](#), [ClinicalTrialIdentificationWorkflow.cs](#), [GenAllVR.cxx](#), [GenerateStandardSOPClasses.cxx](#), [GenFakeIdentifyFile.cxx](#), [MrProtocol.cxx](#), [PublicDict.cxx](#), [ReadAndPrintAttributes.cxx](#), and [TraverseModules.cxx](#).

27.136.3.6 bool gdcM::Global::LoadResourcesFiles ()

Load all internal XML files, resource path need to have been set before calling this member function (see Appendix/Prepend members func)

Warning

not thread safe !

Examples:

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

27.136.3.7 const char* gdcM::Global::Locate (const char * resfile) const [protected]

Locate a resource file.

27.136.3.8 `bool gdcmm::Global::Prepend (const char * path)`

Prepend path at the beginning of the path list

Warning

not thread safe !

27.136.4 Friends And Related Function Documentation

27.136.4.1 `std::ostream& operator<< (std::ostream &_os, const Global &g)` [*friend*]

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

- [gdcmmGlobal.h](#)

27.137 gdcmm::GroupDict Class Reference

Class to represent the mapping from group number to its abbreviation and name.

```
#include <gdcmmGroupDict.h>
```

Public Types

- typedef std::vector< std::string > [GroupStringVector](#)

Public Member Functions

- [GroupDict](#) ()
- [~GroupDict](#) ()
- 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)

27.137.1 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 ?

27.137.2 Member Typedef Documentation

27.137.2.1 `typedef std::vector<std::string> gdcm::GroupDict::GroupStringVector`

27.137.3 Constructor & Destructor Documentation

27.137.3.1 `gdcm::GroupDict::GroupDict ()` `[inline]`

27.137.3.2 `gdcm::GroupDict::~~GroupDict ()` `[inline]`

27.137.4 Member Function Documentation

27.137.4.1 `void gdcm::GroupDict::Add (std::string const & abbreviation, std::string const & name)` `[protected]`

27.137.4.2 `std::string const& gdcm::GroupDict::GetAbbreviation (uint16_t num) const`

Referenced by `gdcm::operator<<()`.

27.137.4.3 `std::string const& gdcm::GroupDict::GetName (uint16_t num) const`

Referenced by `gdcm::operator<<()`.

27.137.4.4 `void gdcm::GroupDict::Insert (uint16_t num, std::string const & abbreviation, std::string const & name)`
`[protected]`

27.137.4.5 `size_t gdcm::GroupDict::Size () const` `[inline]`

Referenced by `gdcm::operator<<()`.

27.137.5 Friends And Related Function Documentation

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

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

- [gdcmGroupDict.h](#)

27.138 gdcm::IconImageFilter Class Reference

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.

```
#include <gdcmIconImageFilter.h>
```

Public Member Functions

- [IconImageFilter](#) ()
- [~IconImageFilter](#) ()
- bool [Extract](#) ()
 - Extract all Icon found in [File](#).*
- [File](#) & [GetFile](#) ()
- const [File](#) & [GetFile](#) () const
- [IconImage](#) & [GetIconImage](#) (unsigned int i) const
- unsigned int [GetNumberOfIconImages](#) () const
 - Retrieve extract IconImage (need to call Extract first)*
- void [SetFile](#) (const [File](#) &f)
 - Set/Get [File](#).*

Protected Member Functions

- void [ExtractIconImages](#) ()
- void [ExtractVeprolIconImages](#) ()

27.138.1 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](#).

27.138.2 Constructor & Destructor Documentation

27.138.2.1 `gdcm::IconImageFilter::IconImageFilter ()`

27.138.2.2 `gdcm::IconImageFilter::~~IconImageFilter ()`

27.138.3 Member Function Documentation

27.138.3.1 `bool gdcm::IconImageFilter::Extract ()`

Extract all Icon found in [File](#).

Examples:

[ExtractIconFromFile.cxx](#).

27.138.3.2 `void gdcm::IconImageFilter::ExtractIconImages ()` [protected]

27.138.3.3 `void gdcm::IconImageFilter::ExtractVeprolIconImages ()` [protected]

27.138.3.4 `File& gdcm::IconImageFilter::GetFile ()` [inline]

27.138.3.5 `const File& gdcm::IconImageFilter::GetFile () const` [inline]

27.138.3.6 `IconImage& gdcm::IconImageFilter::GetIconImage (unsigned int i) const`

Examples:

[ExtractIconFromFile.cxx](#).

27.138.3.7 `unsigned int gdcm::IconImageFilter::GetNumberOfIconImages () const`

Retrieve extract IconImage (need to call Extract first)

Examples:

[ExtractIconFromFile.cxx](#).

27.138.3.8 `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](#)

27.139 gdcm::IconImageGenerator Class Reference

[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:

```
#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.*

27.139.1 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](#).

27.139.2 Constructor & Destructor Documentation

27.139.2.1 `gdcm::IconImageGenerator::IconImageGenerator ()`

27.139.2.2 `gdcm::IconImageGenerator::~~IconImageGenerator ()`

27.139.3 Member Function Documentation

27.139.3.1 `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](#).

27.139.3.2 `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

27.139.3.3 `bool gdcm::IconImageGenerator::Generate ()`

Generate Icon.

Examples:

[ExtractIconFromFile.cxx](#).

27.139.3.4 `const IconImage& gdcm::IconImageGenerator::GetIconImage () const` `[inline]`

Retrieve generated Icon.

Examples:

[ExtractIconFromFile.cxx](#).

27.139.3.5 `Pixmap& gdcm::IconImageGenerator::GetPixmap ()` `[inline]`

27.139.3.6 `const Pixmap& gdcm::IconImageGenerator::GetPixmap () const` `[inline]`

27.139.3.7 `void gdcm::IconImageGenerator::SetOutputDimensions (const unsigned int dims[2])`

Set Target dimension of output Icon.

Examples:

[ExtractIconFromFile.cxx](#).

27.139.3.8 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)`

27.139.3.9 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.

27.139.3.10 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](#)

27.140 gdcm::ignore_char Struct Reference

```
#include <gdcmElement.h>
```

Public Member Functions

- [ignore_char](#) (char c)

Public Attributes

- char [m_char](#)

27.140.1 Constructor & Destructor Documentation

27.140.1.1 `gdcm::ignore_char::ignore_char (char c)` `[inline]`

27.140.2 Member Data Documentation

27.140.2.1 char `gdcm::ignore_char::m_char`

Referenced by `gdcm::operator>>()`.

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

- [gdcmElement.h](#)

- [~Image](#) ()
- 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
print
- void [SetDirectionCosines](#) (const float *dircos)
- void [SetDirectionCosines](#) (const double *dircos)
- void [SetDirectionCosines](#) (unsigned int idx, double dircos)
- void [SetIntercept](#) (double intercept)
intercept
- void [SetOrigin](#) (const float *ori)
- void [SetOrigin](#) (const double *ori)
- void [SetOrigin](#) (unsigned int idx, double ori)
- void [SetSlope](#) (double slope)
slope
- void [SetSpacing](#) (const double *spacing)
- void [SetSpacing](#) (unsigned int idx, double spacing)

Additional Inherited Members

27.141.1 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](#), [csa2img.cxx](#), [ExtractIconFromFile.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenFakeImage.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [iU22tomultisc.cxx](#), [PatchFile.cxx](#), [ReadMultiTimesException.cxx](#), and [threadgdcm.cxx](#).

27.141.2 Constructor & Destructor Documentation

27.141.2.1 `gdcm::Image::Image ()` `[inline]`

27.141.2.2 `gdcm::Image::~~Image ()` `[inline]`

27.141.3 Member Function Documentation

27.141.3.1 `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.

27.141.3.2 `double gdcm::Image::GetDirectionCosines (unsigned int idx)` `const`

27.141.3.3 `double gdcm::Image::GetIntercept ()` `const` `[inline]`

27.141.3.4 `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](#).

27.141.3.5 `double gdcm::Image::GetOrigin (unsigned int idx)` `const`

27.141.3.6 `double gdcm::Image::GetSlope ()` `const` `[inline]`

27.141.3.7 `const double* gdcm::Image::GetSpacing ()` `const`

Return a 3-tuples specifying the spacing NOTE: 3rd value can be an arbitrary 1 value when the spacing was not specified (ex. 2D image). WARNING: when the spacing is not specifier, a default value of 1 will be returned

27.141.3.8 `double gdcm::Image::GetSpacing (unsigned int idx)` `const`

27.141.3.9 `void gdcm::Image::Print (std::ostream & os)` `const` `[virtual]`

print

Reimplemented from [gdcm::Bitmap](#).

Examples:

[CompressImage.cxx](#), and [PatchFile.cxx](#).

27.141.3.10 void `gdcm::Image::SetDirectionCosines` (const float * *dircos*)

27.141.3.11 void `gdcm::Image::SetDirectionCosines` (const double * *dircos*)

27.141.3.12 void `gdcm::Image::SetDirectionCosines` (unsigned int *idx*, double *dircos*)

27.141.3.13 void `gdcm::Image::SetIntercept` (double *intercept*) `[inline]`

intercept

27.141.3.14 void `gdcm::Image::SetOrigin` (const float * *ori*)

27.141.3.15 void `gdcm::Image::SetOrigin` (const double * *ori*)

27.141.3.16 void `gdcm::Image::SetOrigin` (unsigned int *idx*, double *ori*)

27.141.3.17 void `gdcm::Image::SetSlope` (double *slope*) `[inline]`

slope

27.141.3.18 void `gdcm::Image::SetSpacing` (const double * *spacing*)

Examples:

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

27.141.3.19 void `gdcm::Image::SetSpacing` (unsigned int *idx*, double *spacing*)

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

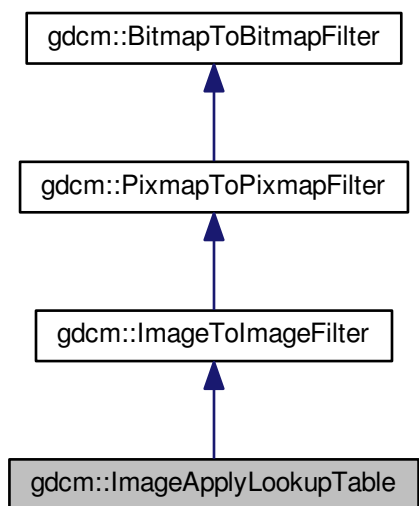
- [gdcmImage.h](#)

27.142 gdcm::ImageApplyLookupTable Class Reference

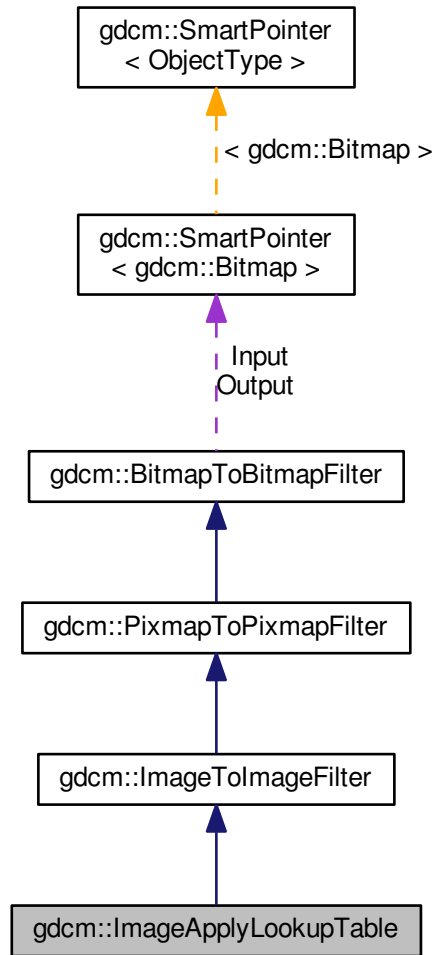
[ImageApplyLookupTable](#) class It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a [PhotometricInterpretation=RGB](#) image.

```
#include <gdcmImageApplyLookupTable.h>
```

Inheritance diagram for gdcm::ImageApplyLookupTable:



Collaboration diagram for `gdcM::ImageApplyLookupTable`:



Public Member Functions

- [ImageApplyLookupTable](#) ()
- [~ImageApplyLookupTable](#) ()
- bool [Apply](#) ()

Apply.

Additional Inherited Members

27.142.1 Detailed Description

[ImageApplyLookupTable](#) class It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a [PhotometricInterpretation](#)=RGB image.

27.142.2 Constructor & Destructor Documentation

27.142.2.1 `gdcm::ImageApplyLookupTable::ImageApplyLookupTable ()` `[inline]`

27.142.2.2 `gdcm::ImageApplyLookupTable::~~ImageApplyLookupTable ()` `[inline]`

27.142.3 Member Function Documentation

27.142.3.1 `bool gdcm::ImageApplyLookupTable::Apply ()`

Apply.

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

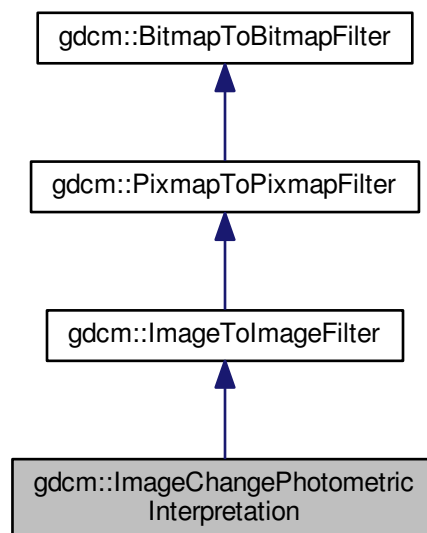
- [gdcmImageApplyLookupTable.h](#)

27.143 gdcm::ImageChangePhotometricInterpretation Class Reference

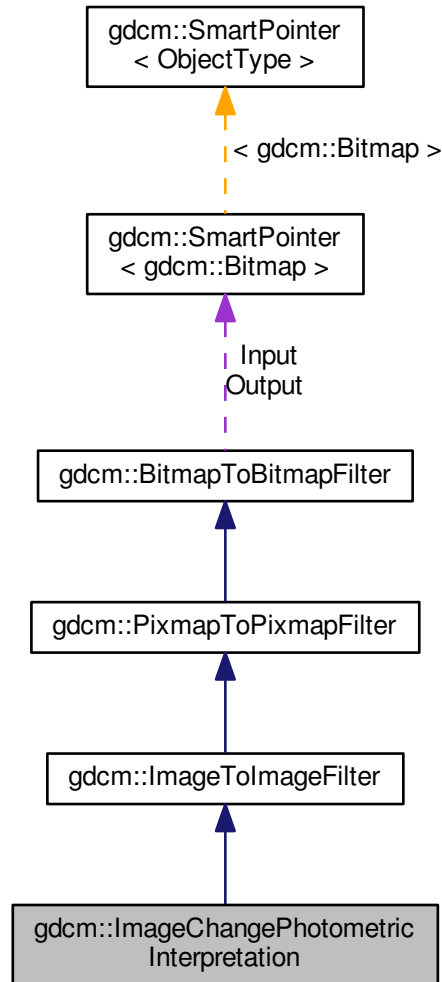
[ImageChangePhotometricInterpretation](#) class Class to change the Photometric Interpretation of an input DICOM.

```
#include <gdcmImageChangePhotometricInterpretation.h>
```

Inheritance diagram for `gdcm::ImageChangePhotometricInterpretation`:



Collaboration diagram for `gdcM::ImageChangePhotometricInterpretation`:



Public Member Functions

- [ImageChangePhotometricInterpretation \(\)](#)
- [~ImageChangePhotometricInterpretation \(\)](#)
- [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])
colorspace conversion (based on CCIR Recommendation 601-2)
- template<typename T >
static void [YBR2RGB](#) (T rgb[3], const T ybr[3])

Protected Member Functions

- bool [ChangeMonochrome](#) ()

Additional Inherited Members

27.143.1 Detailed Description

[ImageChangePhotometricInterpretation](#) class Class to change the Photometric Interpretation of an input DICOM.

27.143.2 Constructor & Destructor Documentation

27.143.2.1 `gdcm::ImageChangePhotometricInterpretation::ImageChangePhotometricInterpretation ()` `[inline]`

27.143.2.2 `gdcm::ImageChangePhotometricInterpretation::~~ImageChangePhotometricInterpretation ()` `[inline]`

27.143.3 Member Function Documentation

27.143.3.1 `bool gdcm::ImageChangePhotometricInterpretation::Change ()`

Change.

27.143.3.2 `bool gdcm::ImageChangePhotometricInterpretation::ChangeMonochrome ()` `[protected]`

27.143.3.3 `const PhotometricInterpretation& gdcm::ImageChangePhotometricInterpretation::GetPhotometricInterpretation ()`
`const` `[inline]`

27.143.3.4 `template<typename T > void gdcm::ImageChangePhotometricInterpretation::RGB2YBR (T ybr[3], const T rgb[3])`
`[static]`

colorspace conversion (based on CCIR Recommendation 601-2)

27.143.3.5 `void gdcm::ImageChangePhotometricInterpretation::SetPhotometricInterpretation (PhotometricInterpretation const & pi)` `[inline]`

Set/Get requested [PhotometricInterpretation](#).

27.143.3.6 `template<typename T > void gdcmm::ImageChangePhotometricInterpretation::YBR2RGB (T rgb[3], const T ybr[3])`
`[static]`

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

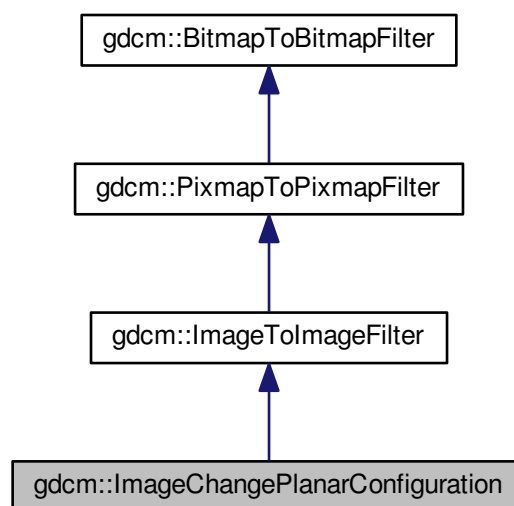
- [gdcmmImageChangePhotometricInterpretation.h](#)

27.144 gdcmm::ImageChangePlanarConfiguration Class Reference

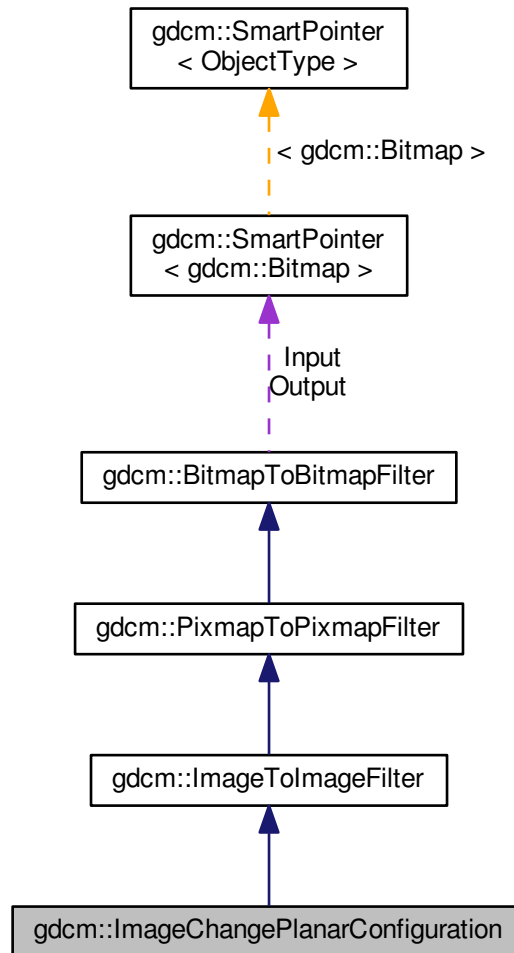
[ImageChangePlanarConfiguration](#) class Class to change the Planar configuration of an input DICOM By default it will change into the more usual representation: PlanarConfiguration = 0.

```
#include <gdcmmImageChangePlanarConfiguration.h>
```

Inheritance diagram for gdcmm::ImageChangePlanarConfiguration:



Collaboration diagram for gdcm::ImageChangePlanarConfiguration:



Public Member Functions

- [ImageChangePlanarConfiguration](#) ()
- [~ImageChangePlanarConfiguration](#) ()
- [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

27.144.1 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.

27.144.2 Constructor & Destructor Documentation

27.144.2.1 `gdcm::ImageChangePlanarConfiguration::ImageChangePlanarConfiguration ()` `[inline]`

27.144.2.2 `gdcm::ImageChangePlanarConfiguration::~~ImageChangePlanarConfiguration ()` `[inline]`

27.144.3 Member Function Documentation

27.144.3.1 `bool gdcm::ImageChangePlanarConfiguration::Change ()`

Change.

27.144.3.2 `unsigned int gdcm::ImageChangePlanarConfiguration::GetPlanarConfiguration () const` `[inline]`

27.144.3.3 `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

27.144.3.4 `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

27.144.3.5 `void gdcm::ImageChangePlanarConfiguration::SetPlanarConfiguration (unsigned int pc)` `[inline]`

Set/Get requested PlanarConfiguration.

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

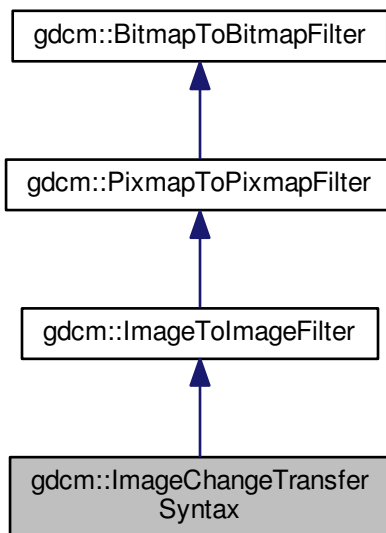
- [gdcmImageChangePlanarConfiguration.h](#)

27.145 gdcm::ImageChangeTransferSyntax Class Reference

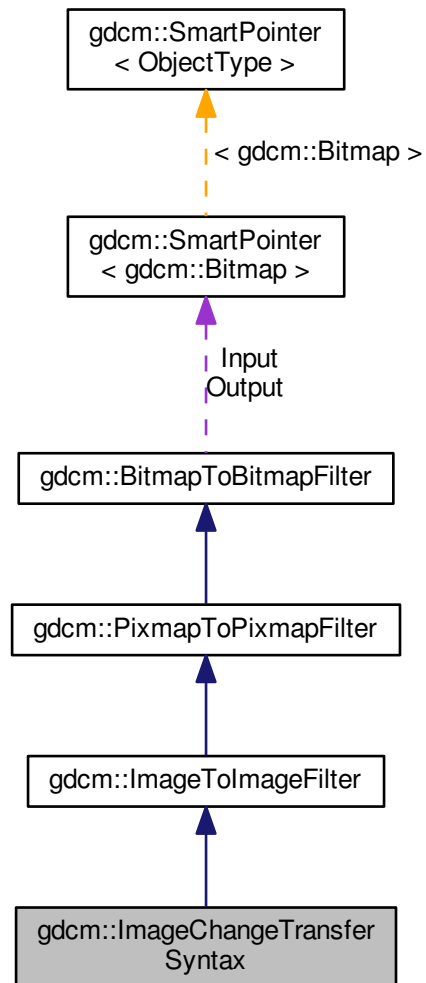
[ImageChangeTransferSyntax](#) class Class to change the transfer syntax of an input DICOM.

```
#include <gdcmImageChangeTransferSyntax.h>
```

Inheritance diagram for gdcm::ImageChangeTransferSyntax:



Collaboration diagram for `gdcm::ImageChangeTransferSyntax`:



Public Member Functions

- `ImageChangeTransferSyntax ()`
- `~ImageChangeTransferSyntax ()`
- `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

27.145.1 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](#).

27.145.2 Constructor & Destructor Documentation

27.145.2.1 `gdcm::ImageChangeTransferSyntax::ImageChangeTransferSyntax ()` [\[inline\]](#)

27.145.2.2 `gdcm::ImageChangeTransferSyntax::~ImageChangeTransferSyntax ()` [\[inline\]](#)

27.145.3 Member Function Documentation

27.145.3.1 `bool gdcm::ImageChangeTransferSyntax::Change ()`

Change.

Examples:

[CompressImage.cxx](#).

27.145.3.2 `const TransferSyntax& gdcm::ImageChangeTransferSyntax::GetTransferSyntax () const` [\[inline\]](#)

Get Transfer Syntax.

27.145.3.3 `void gdcmm::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

27.145.3.4 `void gdcmm::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

27.145.3.5 `void gdcmm::ImageChangeTransferSyntax::SetTransferSyntax (const TransferSyntax & ts) [inline]`

Set target Transfer Syntax.

Examples:

[CompressImage.cxx](#).

27.145.3.6 `void gdcmm::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)`

27.145.3.7 `bool gdcmm::ImageChangeTransferSyntax::TryJPEG2000Codec (const DataElement & pixelde, Bitmap const & input, Bitmap & output) [protected]`

27.145.3.8 `bool gdcmm::ImageChangeTransferSyntax::TryJPEGCodec (const DataElement & pixelde, Bitmap const & input, Bitmap & output) [protected]`

27.145.3.9 `bool gdcmm::ImageChangeTransferSyntax::TryJPEGLSCCodec (const DataElement & pixelde, Bitmap const & input, Bitmap & output) [protected]`

27.145.3.10 `bool gdcmm::ImageChangeTransferSyntax::TryRAWCodec (const DataElement & pixelde, Bitmap const & input, Bitmap & output) [protected]`

27.145.3.11 `bool gdcmm::ImageChangeTransferSyntax::TryRLECodec (const DataElement & pixelde, Bitmap const & input, Bitmap & output) [protected]`

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

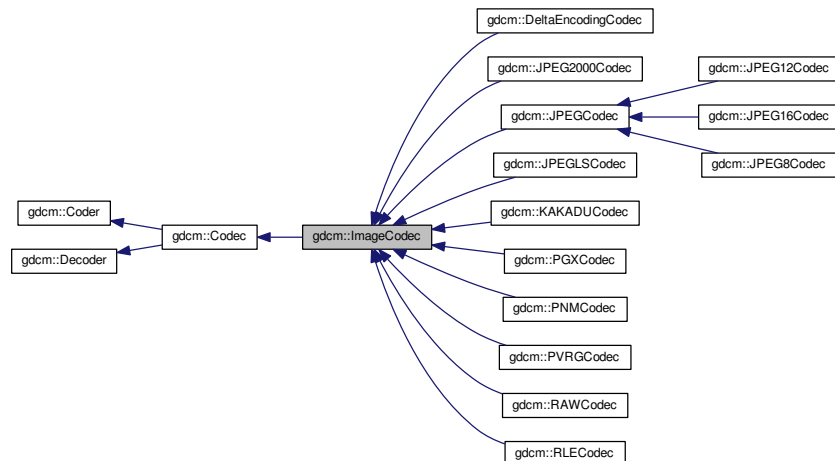
- [gdcmmImageChangeTransferSyntax.h](#)

27.146 gdcm::ImageCodec Class Reference

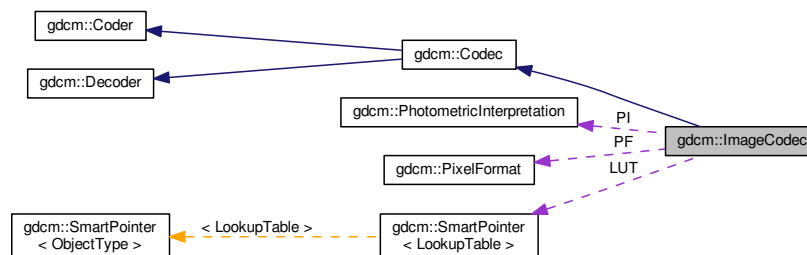
[ImageCodec](#).

```
#include <gdcmImageCodec.h>
```

Inheritance diagram for gdcm::ImageCodec:



Collaboration diagram for gdcm::ImageCodec:



Public Member Functions

- [ImageCodec](#) ()
- [~ImageCodec](#) ()
- bool [CanCode](#) ([TransferSyntax](#) const &) const
Return whether this coder support this transfer syntax (can code it)
- bool [CanDecode](#) ([TransferSyntax](#) const &) const
Return whether this decoder support this transfer syntax (can decode it)
- virtual [ImageCodec](#) * [Clone](#) () const =0
- bool [Decode](#) ([DataElement](#) const &is_, [DataElement](#) &os)

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 unsigned int d[3])
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- 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)
- 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)
- 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](#)

27.146.1 Detailed Description

[ImageCodec](#).

Note

Main codec, this is a central place for all implementation

27.146.2 Member Typedef Documentation

27.146.2.1 `typedef SmartPointer<LookupTable> gdcm::ImageCodec::LUTPtr` [protected]

27.146.3 Constructor & Destructor Documentation

27.146.3.1 `gdcm::ImageCodec::ImageCodec ()`

27.146.3.2 `gdcm::ImageCodec::~~ImageCodec ()`

27.146.4 Member Function Documentation

27.146.4.1 `virtual bool gdcm::ImageCodec::AppendFrameEncode (std::ostream & out, const char * data, size_t datalen)`
[protected], [virtual]

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

27.146.4.2 `virtual bool gdcm::ImageCodec::AppendRowEncode (std::ostream & out, const char * data, size_t datalen)`
[protected], [virtual]

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

27.146.4.3 `bool gdcm::ImageCodec::CanCode (TransferSyntax const &) const` `[inline],[virtual]`

Return whether this coder support this transfer syntax (can code it)

Implements [gdcm::Coder](#).

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

27.146.4.4 `bool gdcm::ImageCodec::CanDecode (TransferSyntax const &) const` `[inline],[virtual]`

Return whether this decoder support this transfer syntax (can decode it)

Implements [gdcm::Decoder](#).

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

27.146.4.5 `virtual ImageCodec* gdcm::ImageCodec::Clone () const` `[pure virtual]`

Implemented in [gdcm::JPEGCodec](#), [gdcm::RLECodec](#), [gdcm::JPEGGLSCodec](#), [gdcm::PVRGCodec](#), [gdcm::JPEG2000Codec](#), [gdcm::PNMCodec](#), [gdcm::RAWCodec](#), [gdcm::KAKADUCodec](#), and [gdcm::PGXCodec](#).

27.146.4.6 `bool gdcm::ImageCodec::Decode (DataElement const & , DataElement &)` `[virtual]`

Decode.

Reimplemented from [gdcm::Decoder](#).

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

27.146.4.7 `bool gdcm::ImageCodec::DecodeByStreams (std::istream & is_ , std::ostream & os)` `[protected],[virtual]`

Reimplemented from [gdcm::Decoder](#).

Reimplemented in [gdcm::JPEGCodec](#), [gdcm::JPEG2000Codec](#), [gdcm::RLECodec](#), [gdcm::RAWCodec](#), [gdcm::JPEGGLSCodec](#), [gdcm::JPEG16Codec](#), and [gdcm::JPEG8Codec](#).

27.146.4.8 `bool gdcm::ImageCodec::DoByteSwap (std::istream & is_ , std::ostream & os)` `[protected]`

27.146.4.9 `bool gdcm::ImageCodec::DoInvertMonochrome (std::istream & is_ , std::ostream & os)` `[protected]`

27.146.4.10 `bool gdcm::ImageCodec::DoOverlayCleanup (std::istream & is_ , std::ostream & os)` `[protected]`

27.146.4.11 `bool gdcm::ImageCodec::DoPaddedCompositePixelCode (std::istream & is_ , std::ostream & os)` `[protected]`

27.146.4.12 `bool gdcm::ImageCodec::DoPlanarConfiguration (std::istream & is_ , std::ostream & os)` `[protected]`

27.146.4.13 `bool gdcm::ImageCodec::DoSimpleCopy (std::istream & is_ , std::ostream & os)` `[protected]`

27.146.4.14 `bool gdcm::ImageCodec::DoYBR (std::istream & is_ , std::ostream & os)` `[protected]`

27.146.4.15 `const unsigned int* gdcm::ImageCodec::GetDimensions () const` [inline]

27.146.4.16 `virtual bool gdcm::ImageCodec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` [virtual]

Reimplemented in [gdcm::JPEGCodec](#), [gdcm::RLECodec](#), [gdcm::JPEGLSCodec](#), [gdcm::JPEG2000Codec](#), [gdcm::PNGCodec](#), [gdcm::JPEG12Codec](#), [gdcm::JPEG16Codec](#), [gdcm::JPEG8Codec](#), [gdcm::RAWCodec](#), and [gdcm::PGXCodec](#).

27.146.4.17 `bool gdcm::ImageCodec::GetLossyFlag () const`

27.146.4.18 `const LookupTable& gdcm::ImageCodec::GetLUT () const` [inline]

27.146.4.19 `bool gdcm::ImageCodec::GetNeedByteSwap () const` [inline]

27.146.4.20 `unsigned int gdcm::ImageCodec::GetNumberOfDimensions () const`

27.146.4.21 `const PhotometricInterpretation& gdcm::ImageCodec::GetPhotometricInterpretation () const`

27.146.4.22 `PixelFormat& gdcm::ImageCodec::GetPixelFormat ()` [inline]

Examples:

[GetJPEGSamplePrecision.cxx](#).

27.146.4.23 `const PixelFormat& gdcm::ImageCodec::GetPixelFormat () const` [inline]

27.146.4.24 `unsigned int gdcm::ImageCodec::GetPlanarConfiguration () const` [inline]

27.146.4.25 `virtual bool gdcm::ImageCodec::IsFrameEncoder ()` [protected],[virtual]

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

27.146.4.26 `bool gdcm::ImageCodec::IsLossy () const`

27.146.4.27 `virtual bool gdcm::ImageCodec::IsRowEncoder ()` [protected],[virtual]

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

27.146.4.28 `virtual bool gdcm::ImageCodec::IsValid (PhotometricInterpretation const & pi)` [protected],[virtual]

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

27.146.4.29 `void gdcm::ImageCodec::SetDimensions (const unsigned int d[3])`

Examples:

[ExtractIconFromFile.cxx](#).

27.146.4.30 void `gdcm::ImageCodec::SetDimensions` (const `std::vector< unsigned int >` & *d*)

27.146.4.31 void `gdcm::ImageCodec::SetLossyFlag` (bool *l*)

27.146.4.32 void `gdcm::ImageCodec::SetLUT` (`LookupTable` const & *lut*) [inline]

Examples:

[ExtractIconFromFile.cxx](#).

27.146.4.33 void `gdcm::ImageCodec::SetNeedByteSwap` (bool *b*) [inline]

27.146.4.34 void `gdcm::ImageCodec::SetNeedOverlayCleanup` (bool *b*) [inline]

27.146.4.35 void `gdcm::ImageCodec::SetNumberOfDimensions` (unsigned int *dim*)

27.146.4.36 void `gdcm::ImageCodec::SetPhotometricInterpretation` (`PhotometricInterpretation` const & *pi*)

Examples:

[ExtractIconFromFile.cxx](#).

27.146.4.37 virtual void `gdcm::ImageCodec::SetPixelFormat` (`PixelFormat` const & *pf*) [inline],[virtual]

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

Examples:

[ExtractIconFromFile.cxx](#).

27.146.4.38 void `gdcm::ImageCodec::SetPlanarConfiguration` (unsigned int *pc*) [inline]

27.146.4.39 virtual bool `gdcm::ImageCodec::StartEncode` (`std::ostream` & *os*) [protected],[virtual]

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

27.146.4.40 virtual bool `gdcm::ImageCodec::StopEncode` (`std::ostream` & *os*) [protected],[virtual]

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

27.146.5 Friends And Related Function Documentation

27.146.5.1 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

27.146.5.2 friend class ImageChangePhotometricInterpretation [friend]

27.146.6 Member Data Documentation

27.146.6.1 unsigned int gdcm::ImageCodec::Dimensions[3] [protected]

27.146.6.2 bool gdcm::ImageCodec::LossyFlag [protected]

27.146.6.3 LUTPtr gdcm::ImageCodec::LUT [protected]

27.146.6.4 bool gdcm::ImageCodec::NeedByteSwap [protected]

27.146.6.5 bool gdcm::ImageCodec::NeedOverlayCleanup [protected]

27.146.6.6 unsigned int gdcm::ImageCodec::NumberOfDimensions [protected]

27.146.6.7 PixelFormat gdcm::ImageCodec::PF [protected]

27.146.6.8 PhotometricInterpretation gdcm::ImageCodec::PI [protected]

27.146.6.9 unsigned int gdcm::ImageCodec::PlanarConfiguration [protected]

27.146.6.10 bool gdcm::ImageCodec::RequestPaddedCompositePixelCode [protected]

27.146.6.11 bool gdcm::ImageCodec::RequestPlanarConfiguration [protected]

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

- [gdcmImageCodec.h](#)

27.147 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)

27.147.1 Detailed Description

[Image](#) Converter.

Note

This is the class used to convert from one [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

27.147.2 Constructor & Destructor Documentation

27.147.2.1 `gdcm::ImageConverter::ImageConverter ()`

27.147.2.2 `gdcm::ImageConverter::~~ImageConverter ()`

27.147.3 Member Function Documentation

27.147.3.1 `void gdcm::ImageConverter::Convert ()`

27.147.3.2 `const Image& gdcm::ImageConverter::GetOutput () const`

27.147.3.3 `void gdcm::ImageConverter::SetInput (Image const & input)`

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

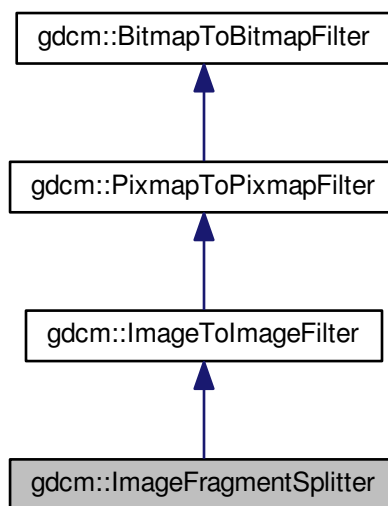
- [gdcmImageConverter.h](#)

27.148 gdcm::ImageFragmentSplitter Class Reference

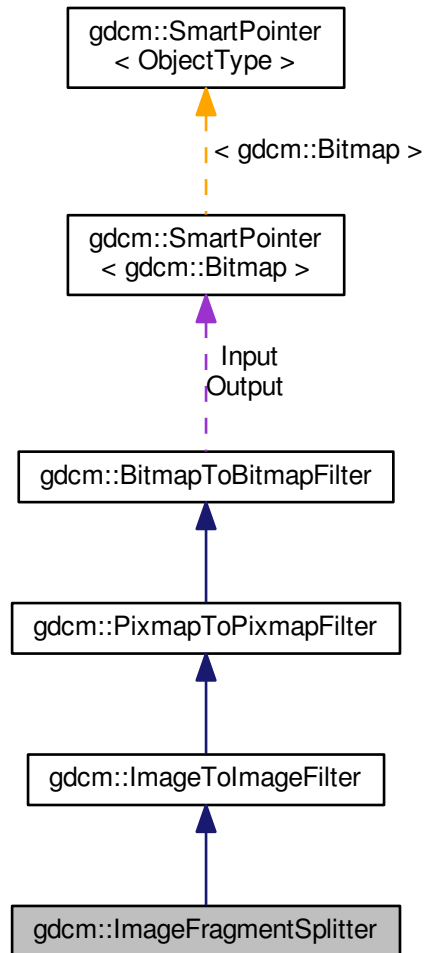
[ImageFragmentSplitter](#) class For single frame image, DICOM standard allow splitting the frame into multiple fragments.

```
#include <gdcmImageFragmentSplitter.h>
```

Inheritance diagram for gdcm::ImageFragmentSplitter:



Collaboration diagram for `gdcM::ImageFragmentSplitter`:



Public Member Functions

- [ImageFragmentSplitter](#) ()
- [~ImageFragmentSplitter](#) ()
- 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

27.148.1 Detailed Description

[ImageFragmentSplitter](#) class For single frame image, DICOM standard allow splitting the frame into multiple fragments.

27.148.2 Constructor & Destructor Documentation

27.148.2.1 `gdcm::ImageFragmentSplitter::ImageFragmentSplitter ()` [\[inline\]](#)

27.148.2.2 `gdcm::ImageFragmentSplitter::~~ImageFragmentSplitter ()` [\[inline\]](#)

27.148.3 Member Function Documentation

27.148.3.1 `unsigned int gdcm::ImageFragmentSplitter::GetFragmentSizeMax () const` [\[inline\]](#)

27.148.3.2 `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 !

27.148.3.3 `void gdcm::ImageFragmentSplitter::SetFragmentSizeMax (unsigned int fragsize)`

FragmentSizeMax needs to be an even number.

27.148.3.4 `bool gdcm::ImageFragmentSplitter::Split ()`

Split.

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

- [gdcmImageFragmentSplitter.h](#)

27.149 gdcm::ImageHelper Class Reference

[ImageHelper](#) (internal class, not intended for user level)

```
#include <gdcmImageHelper.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 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 [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)

27.149.1 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.

27.149.2 Member Function Documentation

- 27.149.2.1 static [MediaStorage](#) [gdcmm::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](#).

27.149.2.2 `static bool gdcm::ImageHelper::ComputeSpacingFromImagePositionPatient (const std::vector< double > & imageposition, std::vector< double > & spacing) [static]`

DO NOT USE.

27.149.2.3 `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](#).

27.149.2.4 `static bool gdcm::ImageHelper::GetDirectionCosinesFromDataSet (DataSet const & ds, std::vector< double > & dircos) [static]`

27.149.2.5 `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

27.149.2.6 `static bool gdcm::ImageHelper::GetForcePixelSpacing () [static]`

27.149.2.7 `static bool gdcm::ImageHelper::GetForceRescaleInterceptSlope () [static]`

27.149.2.8 `static SmartPointer<LookupTable> gdcm::ImageHelper::GetLUT (File const & f) [static]`

returns the lookup table of an image file

27.149.2.9 `static std::vector<double> gdcm::ImageHelper::GetOriginValue (File const & f) [static]`

Set/Get Origin (IPP) from/to a file.

27.149.2.10 `static PhotometricInterpretation gdcm::ImageHelper::GetPhotometricInterpretationValue (File const & f) [static]`

27.149.2.11 `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)

27.149.2.12 `static unsigned int gdcm::ImageHelper::GetPlanarConfigurationValue (const File & f) [static]`

27.149.2.13 `static const ByteValue* gdcm::ImageHelper::GetPointerFromElement (Tag const & tag, File const & f) [static]`

27.149.2.14 `static bool gdcm::ImageHelper::GetRealWorldValueMappingContent (File const & f, RealWorldValueMappingContent & rwvmc) [static]`

27.149.2.15 `static std::vector<double> gdcmm::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

27.149.2.16 `static Tag gdcmm::ImageHelper::GetSpacingTagFromMediaStorage (MediaStorage const & ms) [static],
[protected]`

27.149.2.17 `static std::vector<double> gdcmm::ImageHelper::GetSpacingValue (File const & f) [static]`

Set/Get [Spacing](#) from/to a [File](#).

27.149.2.18 `static Tag gdcmm::ImageHelper::GetZSpacingTagFromMediaStorage (MediaStorage const & ms) [static],
[protected]`

27.149.2.19 `static void gdcmm::ImageHelper::SetDimensionsValue (File & f, const Pixmap & img) [static]`

27.149.2.20 `static void gdcmm::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.

27.149.2.21 `static void gdcmm::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](#)

27.149.2.22 `static void gdcmm::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.

27.149.2.23 `static void gdcmm::ImageHelper::SetOriginValue (DataSet & ds, const Image & img) [static]`

27.149.2.24 `static void gdcmm::ImageHelper::SetRescaleInterceptSlopeValue (File & f, const Image & img) [static]`

27.149.2.25 `static void gdcmm::ImageHelper::SetSpacingValue (DataSet & ds, const std::vector< double > & spacing)
[static]`

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

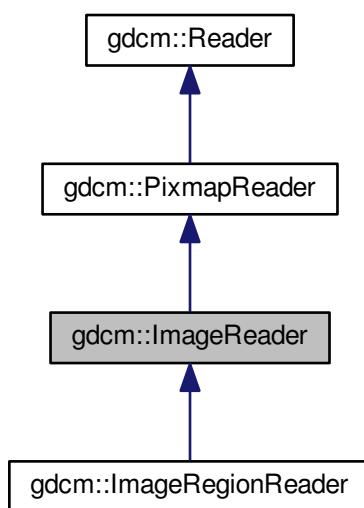
- [gdcmmImageHelper.h](#)

27.150 gdcm::ImageReader Class Reference

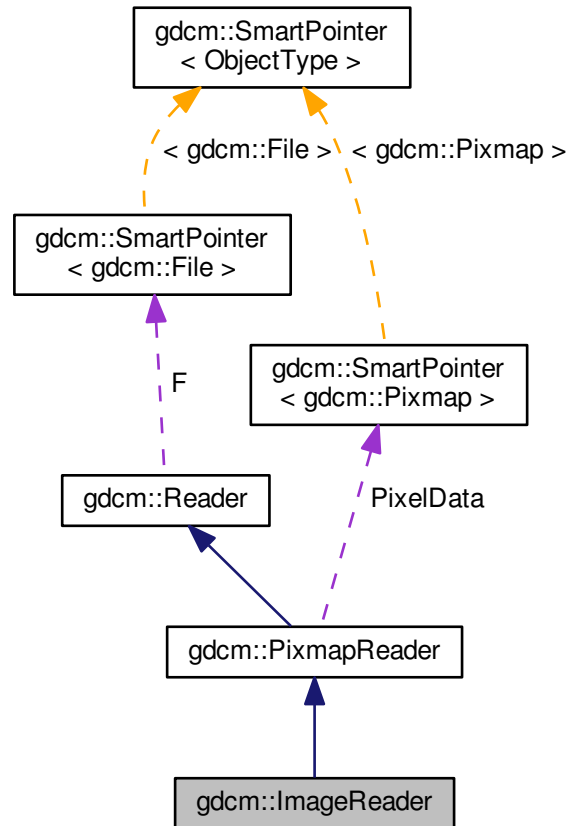
[ImageReader](#).

```
#include <gdcmImageReader.h>
```

Inheritance diagram for gdcm::ImageReader:



Collaboration diagram for `gdcm::ImageReader`:



Public Member Functions

- `ImageReader ()`
- `virtual ~ImageReader ()`
- `const Image & GetImage () const`
Return the read image.
- `Image & GetImage ()`
- `virtual bool Read ()`

Protected Member Functions

- `bool ReadACRNEMAIImage ()`
- `bool ReadImage (MediaStorage const &ms)`

Additional Inherited Members

27.150.1 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](#), [ReadMultiTimesException.cxx](#), and [threadgdcm.cxx](#).

27.150.2 Constructor & Destructor Documentation

27.150.2.1 `gdcm::ImageReader::ImageReader ()`

27.150.2.2 `virtual gdcm::ImageReader::~~ImageReader ()` [virtual]

27.150.3 Member Function Documentation

27.150.3.1 `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](#), [ReadMultiTimesException.cxx](#), and [threadgdcm.cxx](#).

27.150.3.2 `Image& gdcm::ImageReader::GetImage ()`

27.150.3.3 `virtual bool gdcm::ImageReader::Read ()` [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::PixmapReader](#).

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

27.150.3.4 `bool gdcm::ImageReader::ReadACRNEMAIImage () [protected],[virtual]`

Reimplemented from [gdcm::PixmapReader](#).

27.150.3.5 `bool gdcm::ImageReader::ReadImage (MediaStorage const & ms) [protected],[virtual]`

Reimplemented from [gdcm::PixmapReader](#).

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

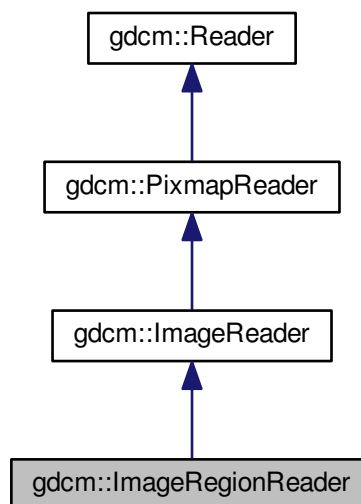
- [gdcmImageReader.h](#)

27.151 gdcm::ImageRegionReader Class Reference

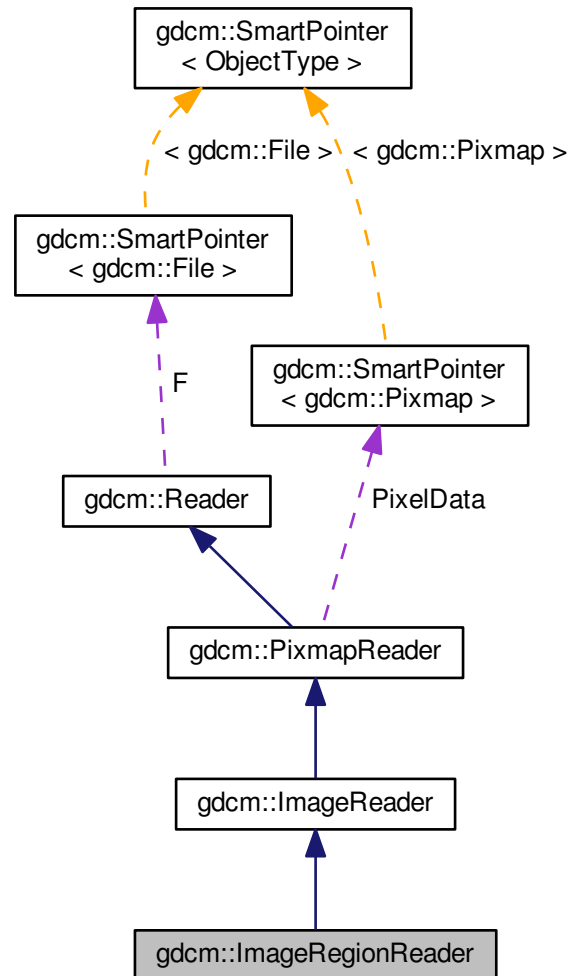
[ImageRegionReader](#).

```
#include <gdcmImageRegionReader.h>
```

Inheritance diagram for `gdcm::ImageRegionReader`:



Collaboration diagram for gdcm::ImageRegionReader:



Public Member Functions

- [ImageRegionReader](#) ()
- [~ImageRegionReader](#) ()
- [size_t ComputeBufferLength](#) () const
- [Region](#) const & [GetRegion](#) () const
- [bool ReadInformation](#) ()
- [bool ReadIntoBuffer](#) (char *inreadbuffer, size_t buflen)
- [void SetRegion](#) ([Region](#) const ®ion)

Set/Get [Region](#) to be read.

Protected Member Functions

- bool [Read](#) ()

To prevent user from calling super class [Read\(\)](#) function.

Additional Inherited Members

27.151.1 Detailed Description

[ImageRegionReader](#).

See also

[ImageReader](#)

27.151.2 Constructor & Destructor Documentation

27.151.2.1 `gdcm::ImageRegionReader::ImageRegionReader ()`

27.151.2.2 `gdcm::ImageRegionReader::~~ImageRegionReader ()`

27.151.3 Member Function Documentation

27.151.3.1 `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

27.151.3.2 `Region const& gdcm::ImageRegionReader::GetRegion () const`

27.151.3.3 `bool gdcm::ImageRegionReader::Read () [protected],[virtual]`

To prevent user from calling super class [Read\(\)](#) function.

Reimplemented from [gdcm::ImageReader](#).

27.151.3.4 `bool gdcm::ImageRegionReader::ReadInformation ()`

Read meta information (not Pixel Data) from the DICOM file.

Returns

false upon error

27.151.3.5 `bool gdcm::ImageRegionReader::ReadIntoBuffer (char * inreadbuffer, size_t buflen)`

Read into buffer:

Returns

false upon error

27.151.3.6 `void gdcm::ImageRegionReader::SetRegion (Region const & region)`

Set/Get [Region](#) to be read.

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

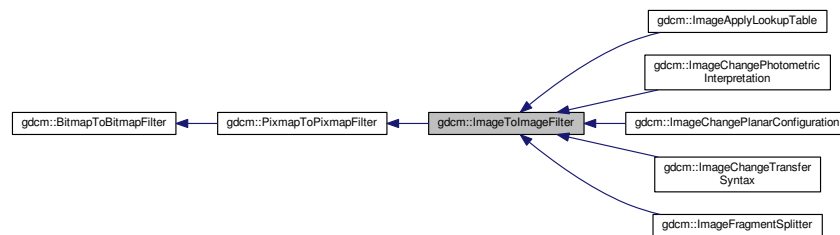
- [gdcmImageRegionReader.h](#)

27.152 gdcm::ImageToImageFilter Class Reference

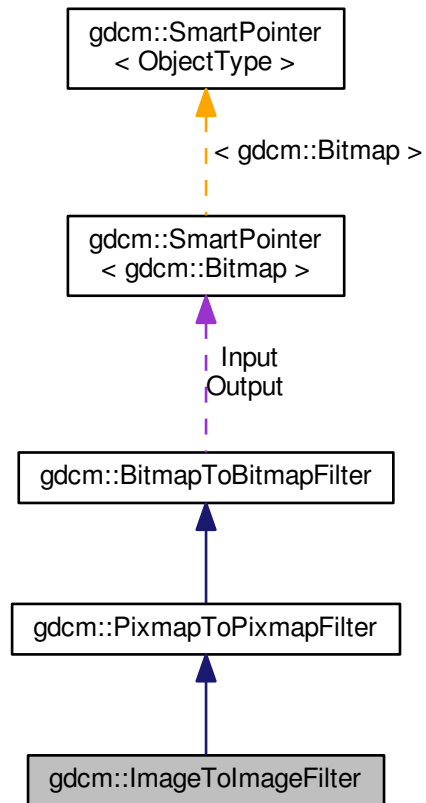
[ImageToImageFilter](#) class Super class for all filter taking an image and producing an output image.

```
#include <gdcmImageToImageFilter.h>
```

Inheritance diagram for `gdcm::ImageToImageFilter`:



Collaboration diagram for `gdcm::ImageToImageFilter`:



Public Member Functions

- [ImageToImageFilter \(\)](#)
- [~ImageToImageFilter \(\)](#)
- [Image & GetInput \(\)](#)
- [const Image & GetOutput \(\) const](#)
Get Output image.

Additional Inherited Members

27.152.1 Detailed Description

[ImageToImageFilter](#) class Super class for all filter taking an image and producing an output image.

27.152.2 Constructor & Destructor Documentation

27.152.2.1 `gdcm::ImageToImageFilter::ImageToImageFilter ()`

27.152.2.2 `gdcm::ImageToImageFilter::~~ImageToImageFilter ()` `[inline]`

27.152.3 Member Function Documentation

27.152.3.1 `Image& gdcm::ImageToImageFilter::GetInput ()`

27.152.3.2 `const Image& gdcm::ImageToImageFilter::GetOutput () const`

Get Output image.

Examples:

[CompressImage.cxx](#).

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

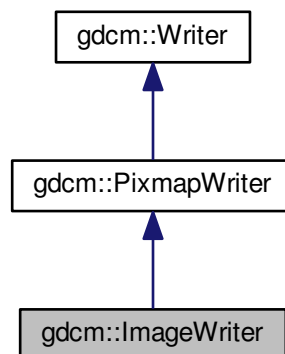
- [gdcmImageToImageFilter.h](#)

27.153 gdcm::ImageWriter Class Reference

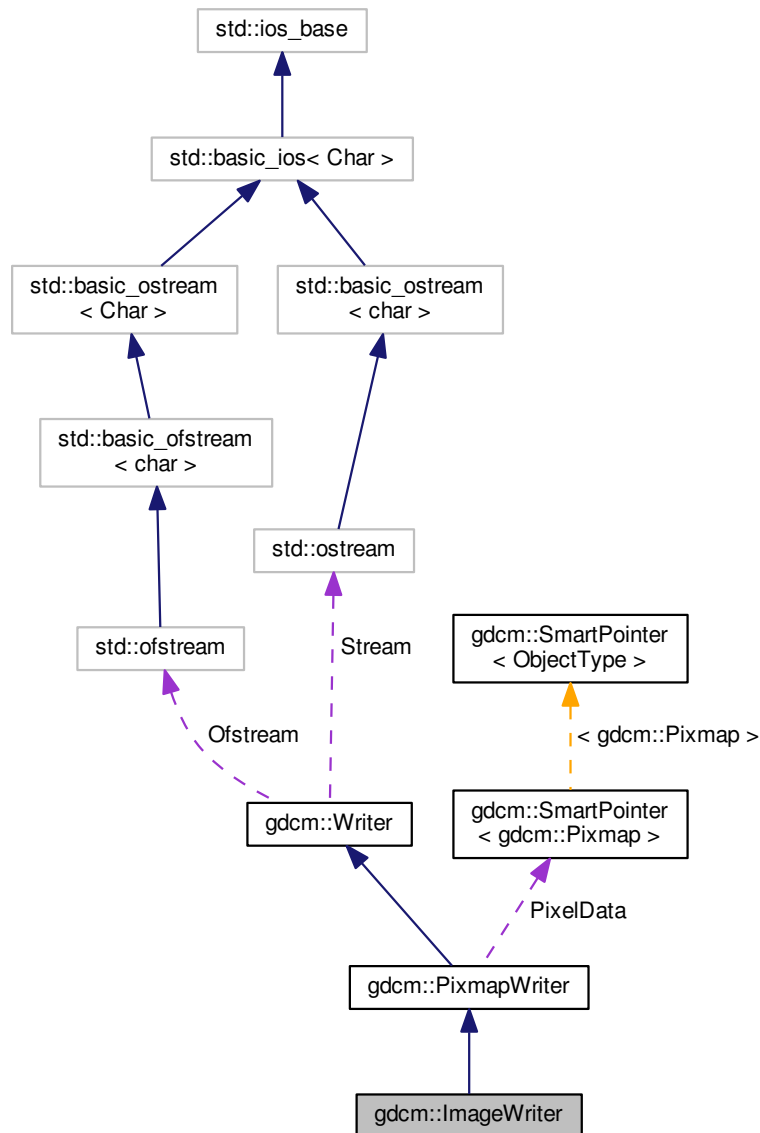
[ImageWriter](#).

```
#include <gdcmImageWriter.h>
```

Inheritance diagram for `gdcm::ImageWriter`:



Collaboration diagram for `gdcm::ImageWriter`:



Public Member Functions

- `ImageWriter ()`
- `~ImageWriter ()`
- `const Image & GetImage () const`
- `Image & GetImage ()`
- `bool Write ()`

Write.

Additional Inherited Members

27.153.1 Detailed Description

[ImageWriter](#).

Examples:

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

27.153.2 Constructor & Destructor Documentation

27.153.2.1 `gdcm::ImageWriter::ImageWriter ()`

27.153.2.2 `gdcm::ImageWriter::~~ImageWriter ()`

27.153.3 Member Function Documentation

27.153.3.1 `const Image& gdcm::ImageWriter::GetImage () const` `[inline],[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](#).

27.153.3.2 `Image& gdcm::ImageWriter::GetImage ()` `[inline],[virtual]`

Reimplemented from [gdcm::PixmapWriter](#).

27.153.3.3 `bool gdcm::ImageWriter::Write ()` `[virtual]`

Write.

Reimplemented from [gdcm::Writer](#).

Examples:

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

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

- [gdcmImageWriter.h](#)

27.154 gdcm::network::ImplementationClassUIDSub Class Reference

[ImplementationClassUIDSub](#) PS 3.7 [Table D.3-1](#) IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

```
#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

27.154.1 Detailed Description

[ImplementationClassUIDSub](#) PS 3.7 [Table D.3-1](#) IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

27.154.2 Constructor & Destructor Documentation

27.154.2.1 `gdcm::network::ImplementationClassUIDSub::ImplementationClassUIDSub ()`

27.154.3 Member Function Documentation

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

27.154.3.2 `std::istream& gdcm::network::ImplementationClassUIDSub::Read (std::istream & is)`

27.154.3.3 `size_t gdcm::network::ImplementationClassUIDSub::Size () const`

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

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

- [gdcmImplementationClassUIDSub.h](#)

27.155 gdcm::network::ImplementationUIDSub Class Reference

[ImplementationUIDSub](#) [Table D.3-2](#) IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)

```
#include <gdcmImplementationUIDSub.h>
```

Public Member Functions

- [ImplementationUIDSub](#) ()
- const std::ostream & [Write](#) (std::ostream &os) const

27.155.1 Detailed Description

[ImplementationUIDSub Table](#) D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)

27.155.2 Constructor & Destructor Documentation

27.155.2.1 `gdcm::network::ImplementationUIDSub::ImplementationUIDSub ()`

27.155.3 Member Function Documentation

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

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

- [gdcmImplementationUIDSub.h](#)

27.156 gdcm::network::ImplementationVersionNameSub Class Reference

[ImplementationVersionNameSub Table](#) D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

```
#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

27.156.1 Detailed Description

[ImplementationVersionNameSub Table](#) D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

27.156.2 Constructor & Destructor Documentation

27.156.2.1 `gdcm::network::ImplementationVersionNameSub::ImplementationVersionNameSub ()`

27.156.3 Member Function Documentation

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

27.156.3.2 `std::istream& gdcm::network::ImplementationVersionNameSub::Read (std::istream & is)`

27.156.3.3 `size_t gdcm::network::ImplementationVersionNameSub::Size () const`

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

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

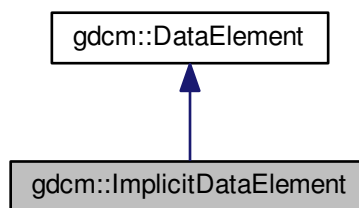
- [gdcmImplementationVersionNameSub.h](#)

27.157 gdcm::ImplicitDataElement Class Reference

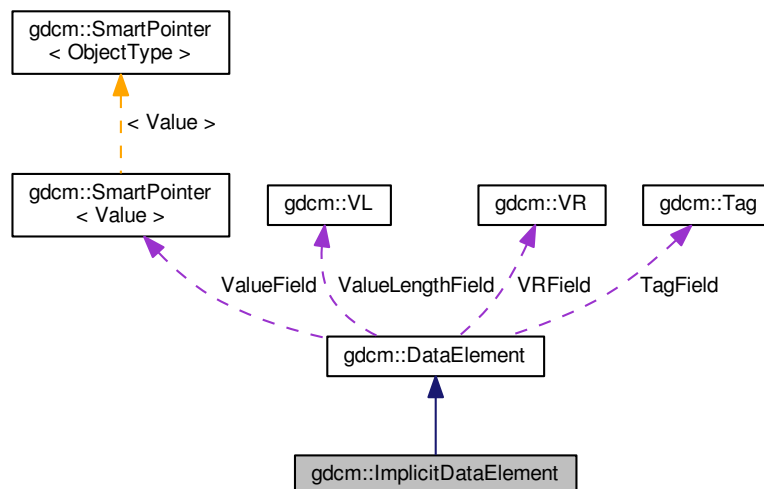
Class to represent an *Implicit VR Data Element*.

```
#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

27.157.1 Detailed Description

Class to represent an *Implicit VR Data Element*.

Note

bla

Examples:

[ReadExplicitLengthSQIVR.cxx](#).

27.157.2 Member Function Documentation

27.157.2.1 [VL gdcm::ImplicitDataElement::GetLength \(\) const](#)

27.157.2.2 [template<typename TSwap > std::istream& gdcm::ImplicitDataElement::Read \(std::istream & is \)](#)

27.157.2.3 [template<typename TSwap > std::istream& gdcm::ImplicitDataElement::ReadPreValue \(std::istream & is \)](#)

27.157.2.4 [template<typename TSwap > std::istream& gdcm::ImplicitDataElement::ReadValue \(std::istream & is, bool readvalues = true \)](#)

27.157.2.5 [template<typename TSwap > std::istream& gdcm::ImplicitDataElement::ReadValueWithLength \(std::istream & is, VL & length, bool readvalues = true \)](#)

27.157.2.6 [template<typename TSwap > std::istream& gdcm::ImplicitDataElement::ReadWithLength \(std::istream & is, VL & length, bool readvalues = true \)](#)

27.157.2.7 [template<typename TSwap > const std::ostream& gdcm::ImplicitDataElement::Write \(std::ostream & os \) const](#)

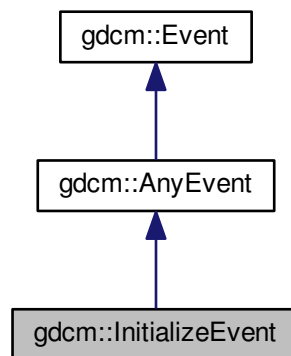
The documentation for this class was generated from the following file:

- [gdcmImplicitDataElement.h](#)

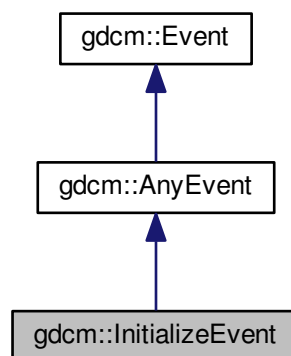
27.158 gdcm::InitializeEvent Class Reference

```
#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](#)

27.159 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](#) ()
- 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)

27.159.1 Detailed Description

Class for representing a [IOD](#).

Note

bla

See also

[Dict](#)

Examples:

[TraverseModules.cxx](#).

27.159.2 Member Typedef Documentation

27.159.2.1 typedef std::vector<[IODEntry](#)> [gdcm::IOD::MapIODEntry](#)

27.159.2.2 typedef MapIODEntry::size_type [gdcm::IOD::SizeType](#)

27.159.3 Constructor & Destructor Documentation

27.159.3.1 [gdcm::IOD::IOD](#) () `[inline]`

27.159.4 Member Function Documentation

27.159.4.1 void `gdcmm::IOD::AddIODEntry (const IODEntry & iode)` `[inline]`

27.159.4.2 void `gdcmm::IOD::Clear ()` `[inline]`

27.159.4.3 const IODEntry& `gdcmm::IOD::GetIODEntry (SizeType idx) const` `[inline]`

Examples:

[TraverseModules.cxx](#).

27.159.4.4 SizeType `gdcmm::IOD::GetNumberOfIODs () const` `[inline]`

Examples:

[TraverseModules.cxx](#).

27.159.4.5 Type `gdcmm::IOD::GetTypeFromTag (const Defs & defs, const Tag & tag) const`

27.159.5 Friends And Related Function Documentation

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

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

- [gdcmmIOD.h](#)

27.160 gdcmm::IODEntry Class Reference

Class for representing a [IODEntry](#).

```
#include <gdcmmIODEntry.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)`

27.160.1 Detailed Description

Class for representing a [IODEntry](#).

Note

A.1.3 [IOD Module Table](#) and Functional Group [Macro Table](#) This Section of each [IOD](#) defines in a tabular form the [Modules](#) comprising the [IOD](#). The following information must be specified for each [Module](#) in the table:

- The name of the [Module](#) or Functional Group
- A reference to the Section in Annex C which defines the [Module](#) or Functional Group
- The usage of the [Module](#) or Functional Group; whether it is:
 - Mandatory (see A.1.3.1) , abbreviated M
 - Conditional (see A.1.3.2) , abbreviated C
 - User Option (see A.1.3.3) , abbreviated U
- The [Modules](#) referenced are defined in Annex C. A.1.3.1 MANDATORY MODULES For each [IOD](#), Mandatory [Modules](#) shall be supported per the definitions, semantics and requirements defined in Annex C. PS 3.3 - 2008 Page 96
- Standard - A.1.3.2 CONDITIONAL MODULES Conditional [Modules](#) are Mandatory [Modules](#) if specific conditions are met. If the specified conditions are not met, this [Module](#) shall not be supported; that is, no information defined in that [Module](#) shall be sent. A.1.3.3 USER OPTION MODULES User Option [Modules](#) may or may not be supported. If an optional [Module](#) is supported, the [Attribute](#) Types specified in the [Modules](#) in Annex C shall be supported.

See also

[DictEntry](#)

Examples:

[TraverseModules.cxx](#).

27.160.2 Constructor & Destructor Documentation

27.160.2.1 `gdcm::IODEntry::IODEntry (const char * name = " ", const char * ref = " ", const char * usag = " ") [inline]`

27.160.3 Member Function Documentation

27.160.3.1 `const char* gdcm::IODEntry::GetIE () const [inline]`

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

27.160.3.3 `const char* gdcm::IODEntry::GetRef () const [inline]`

Examples:

[TraverseModules.cxx](#).

27.160.3.4 `const char* gdcm::IODEntry::GetUsage () const` `[inline]`

27.160.3.5 `Usage::UsageType gdcm::IODEntry::GetUsageType () const`

27.160.3.6 `void gdcm::IODEntry::SetIE (const char * ie)` `[inline]`

27.160.3.7 `void gdcm::IODEntry::SetName (const char * name)` `[inline]`

27.160.3.8 `void gdcm::IODEntry::SetRef (const char * ref)` `[inline]`

27.160.3.9 `void gdcm::IODEntry::SetUsage (const char * usag)` `[inline]`

27.160.4 Friends And Related Function Documentation

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

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

- [gdcmIODEntry.h](#)

27.161 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](#) ()
- 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*)

27.161.1 Detailed Description

Class for representing a [IODs](#).

Note

bla

See also

[IOD](#)

Examples:

[TraverseModules.cxx](#).

27.161.2 Member Typedef Documentation

27.161.2.1 `typedef std::map<IODName, IOD> gdcm::IODs::IODMapType`

27.161.2.2 `typedef IODMapType::const_iterator gdcm::IODs::IODMapTypeConstIterator`

27.161.2.3 `typedef std::string gdcm::IODs::IODName`

27.161.3 Constructor & Destructor Documentation

27.161.3.1 `gdcm::IODs::IODs ()` `[inline]`

27.161.4 Member Function Documentation

27.161.4.1 `void gdcm::IODs::AddIOD (const char * name, const IOD & module)` `[inline]`

27.161.4.2 `IODMapTypeConstIterator gdcm::IODs::Begin () const` `[inline]`

Examples:

[TraverseModules.cxx](#).

27.161.4.3 `void gdcm::IODs::Clear ()` `[inline]`

27.161.4.4 `IODMapTypeConstIterator gdcm::IODs::End () const` `[inline]`

Examples:

[TraverseModules.cxx](#).

27.161.4.5 `const IOD& gdcm::IODs::GetIOD (const char * name) const` `[inline]`

27.161.5 Friends And Related Function Documentation

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

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

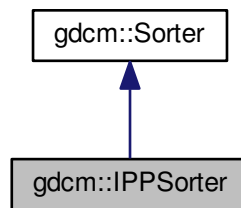
- [gdcmIODs.h](#)

27.162 gdcm::IPPSorter Class Reference

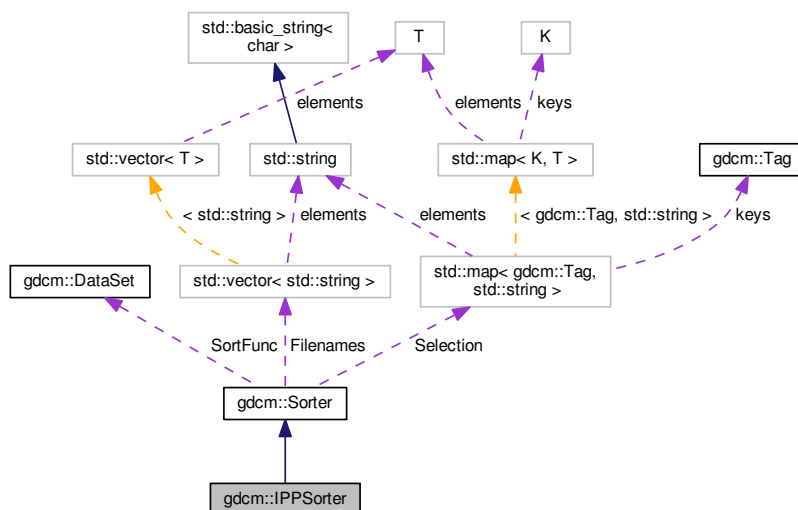
IPPSorter Implement a simple **Image** Position (**Patient**) sorter, along the **Image Orientation** (**Patient**) direction. This algorithm does NOT support duplicate and will FAIL in case of duplicate IPP.

```
#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)
- virtual bool [Sort](#) (std::vector< std::string > const &filenames)

Protected Attributes

- bool [ComputeZSpacing](#)
- double [DirCosTolerance](#)
- bool [DropDuplicatePositions](#)
- double [ZSpacing](#)
- double [ZTolerance](#)

Additional Inherited Members

27.162.1 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

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](#), [PET](#)↔[ImageStorage](#))

Examples:

[Compute3DSpacing.cxx](#), [gdcmorthoplanes.cxx](#), [reslicesphere.cxx](#), and [VolumeSorter.cxx](#).

27.162.2 Constructor & Destructor Documentation

27.162.2.1 `gdcmm::IPPSorter::IPPSorter ()`

27.162.3 Member Function Documentation

27.162.3.1 `double gdcmm::IPPSorter::GetDirectionCosinesTolerance () const` `[inline]`

27.162.3.2 `double gdcmm::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](#), [gdcmmorthoplanes.cxx](#), and [reslicesphere.cxx](#).

27.162.3.3 `double gdcmm::IPPSorter::GetZSpacingTolerance () const` `[inline]`

27.162.3.4 `void gdcmm::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](#), [gdcmmorthoplanes.cxx](#), [reslicesphere.cxx](#), and [VolumeSorter.cxx](#).

27.162.3.5 `void gdcmm::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).

27.162.3.6 `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.

27.162.3.7 void gdcm::IPPSorter::SetZSpacingTolerance (double *tol*) [inline]

1. Another reason for failure is that that Z-Spacing is only slightly changing (eg 1e-3) along the serie, a human can determine that this is ok and change the tolerance from its default value: 1e-6

Examples:

[Compute3DSpacing.cxx](#), [gdcmorthoplanes.cxx](#), and [reslicesphere.cxx](#).

27.162.3.8 virtual bool gdcm::IPPSorter::Sort (std::vector< std::string > const & *filenames*) [virtual]

Main entry point to the sorter. It will execute the filter, option should be set before running this function (SetZSpacing←Tolerance, ...) Return value indicate if sorting could be achieved. Warning this does *NOT* imply that spacing is consistent, it only means the file are sorted according to IPP You should check if ZSpacing is 0 or not to deduce if file are actually a 3D volume

Reimplemented from [gdcm::Sorter](#).

Examples:

[Compute3DSpacing.cxx](#), [gdcmorthoplanes.cxx](#), [reslicesphere.cxx](#), and [VolumeSorter.cxx](#).

27.162.4 Member Data Documentation

27.162.4.1 bool gdcm::IPPSorter::ComputeZSpacing [protected]

27.162.4.2 double gdcm::IPPSorter::DirCosTolerance [protected]

27.162.4.3 bool gdcm::IPPSorter::DropDuplicatePositions [protected]

27.162.4.4 double gdcm::IPPSorter::ZSpacing [protected]

27.162.4.5 double gdcm::IPPSorter::ZTolerance [protected]

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

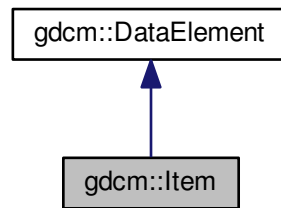
- [gdcmIPPSorter.h](#)

27.163 gdcm::Item Class Reference

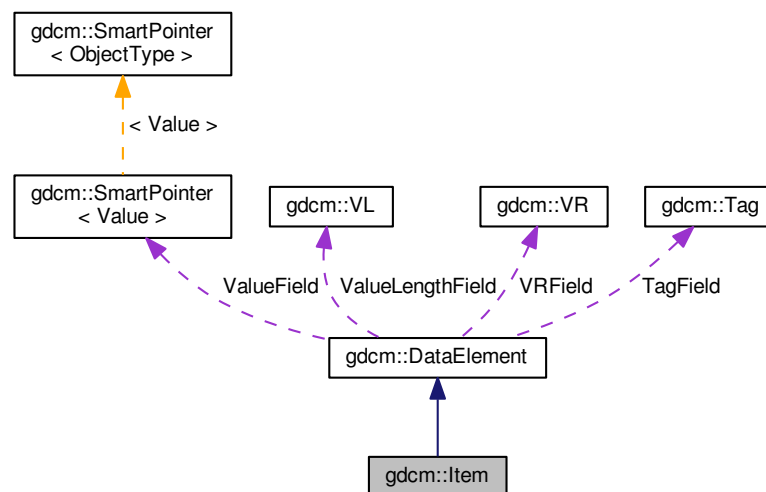
Class to represent an [Item](#) A component of the value of a Data [Element](#) that is of [Value](#) Representation Sequence of Items. An [Item](#) contains a Data Set . See PS 3.5 7.5.1 [Item](#) Encoding Rules Each [Item](#) of a Data [Element](#) of VR SQ shall be encoded as a DICOM Standart Data [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.

```
#include <gdcmItem.h>
```

Inheritance diagram for `gdc::Item`:



Collaboration diagram for `gdc::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
- const `DataSet` & `GetNestedDataSet` () const
- `DataSet` & `GetNestedDataSet` ()

- 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

27.163.1 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](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream↵_Image_Writer.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GenAllIVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.↵cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), and [LargeVRDSExplicit.cxx](#).

27.163.2 Constructor & Destructor Documentation

27.163.2.1 `gdcm::Item::Item () [inline]`

27.163.2.2 `gdcm::Item::Item (Item const & val) [inline]`

27.163.3 Member Function Documentation

27.163.3.1 `void gdcm::Item::Clear () [inline]`

Referenced by `gdcm::SequenceOfItems::Read()`.

27.163.3.2 `bool gdcm::Item::FindDataElement (const Tag & t) const [inline]`

Examples:

[ReadAndDumpDICOMDIR.cxx](#).

27.163.3.3 `const DataElement& gdcM::Item::GetDataElement (const Tag & t) const` `[inline]`

Examples:

[ReadAndDumpDICOMDIR.cxx](#).

27.163.3.4 `template<typename TDE > VL gdcM::Item::GetLength () const`

27.163.3.5 `const DataSet& gdcM::Item::GetNestedDataSet () const` `[inline]`

Examples:

[ChangeSequenceUltrasound.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream↵_Image_Writer.cxx](#), [gdcMrtionplan.cxx](#), [gdcMrtplan.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), and [LargeVRDSExplicit.cxx](#).

Referenced by `gdcM::SequenceOfItems::Read()`.

27.163.3.6 `DataSet& gdcM::Item::GetNestedDataSet ()` `[inline]`

27.163.3.7 `void gdcM::Item::InsertDataElement (const DataElement & de)` `[inline]`

27.163.3.8 `template<typename TDE , typename TSwap > std::istream& gdcM::Item::Read (std::istream & is)` `[inline]`

References `gdcM::DataSet::Clear()`, `gdcMDebugMacro`, `gdcMErrorMacro`, `gdcMWarningMacro`, and `gdcM::DataSet::↵IsEmpty()`.

Referenced by `gdcM::SequenceOfItems::Read()`.

27.163.3.9 `void gdcM::Item::SetNestedDataSet (const DataSet & nested)` `[inline]`

27.163.3.10 `template<typename TDE , typename TSwap > const std::ostream& gdcM::Item::Write (std::ostream & os) const` `[inline]`

References `gdcMWarningMacro`, `gdcM::VL::GetLength()`, `gdcM::VL::Write()`, and `gdcM::Tag::Write()`.

27.163.4 Friends And Related Function Documentation

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

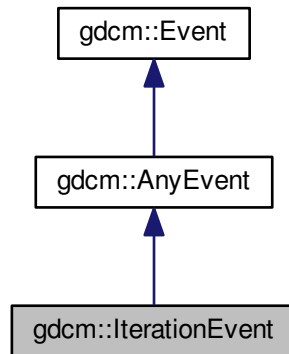
The documentation for this class was generated from the following file:

- [gdcMItem.h](#)

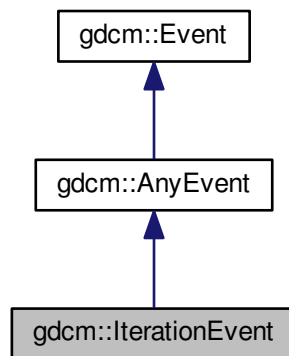
27.164 gdcM::IterationEvent Class Reference

```
#include <gdcMEvent.h>
```


Inheritance diagram for gdcmm::IterationEvent:



Collaboration diagram for gdcmm::IterationEvent:



Additional Inherited Members

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

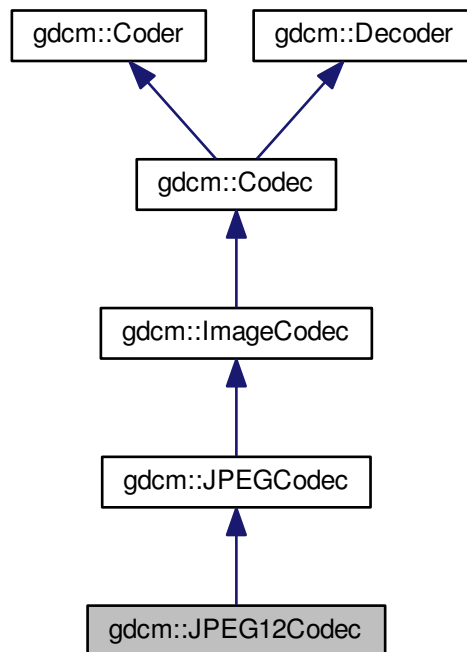
- [gdcmmEvent.h](#)

27.165 gdcm::JPEG12Codec Class Reference

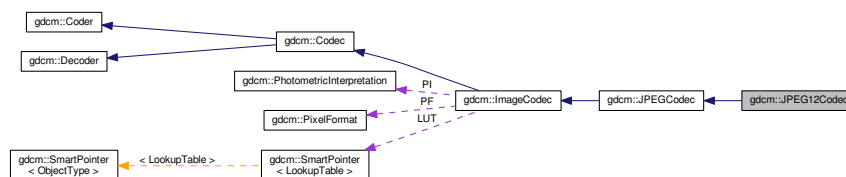
Class to do JPEG 12bits (lossy & lossless)

```
#include <gdcmJPEG12Codec.h>
```

Inheritance diagram for gdcm::JPEG12Codec:



Collaboration diagram for gdcm::JPEG12Codec:



Public Member Functions

- [JPEG12Codec](#) ()
- [~JPEG12Codec](#) ()
- [bool DecodeByStreams](#) (std::istream &is, std::ostream &os)

- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts)
- bool [InternalCode](#) (const char *input, unsigned long len, std::ostream &os)

Protected Member Functions

- virtual bool [EncodeBuffer](#) (std::ostream &os, const char *data, size_t datalen)
- bool [IsStateSuspension](#) () const

Additional Inherited Members

27.165.1 Detailed Description

Class to do JPEG 12bits (lossy & lossless)

Note

internal class

27.165.2 Constructor & Destructor Documentation

27.165.2.1 [gdcm::JPEG12Codec::JPEG12Codec](#) ()

27.165.2.2 [gdcm::JPEG12Codec::~~JPEG12Codec](#) ()

27.165.3 Member Function Documentation

27.165.3.1 [bool gdcm::JPEG12Codec::DecodeByStreams](#) (std::istream & *is*, std::ostream & *os*) [virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.165.3.2 [virtual bool gdcm::JPEG12Codec::EncodeBuffer](#) (std::ostream & *os*, const char * *data*, size_t *datalen*)
[protected], [virtual]

Reimplemented from [gdcm::JPEGCodec](#).

27.165.3.3 [bool gdcm::JPEG12Codec::GetHeaderInfo](#) (std::istream & *is*, [TransferSyntax](#) & *ts*) [virtual]

Reimplemented from [gdcm::JPEGCodec](#).

27.165.3.4 [bool gdcm::JPEG12Codec::InternalCode](#) (const char * *input*, unsigned long *len*, std::ostream & *os*) [virtual]

Reimplemented from [gdcm::Coder](#).

27.165.3.5 [bool gdcm::JPEG12Codec::IsStateSuspension](#) () const [protected], [virtual]

Reimplemented from [gdcm::JPEGCodec](#).

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

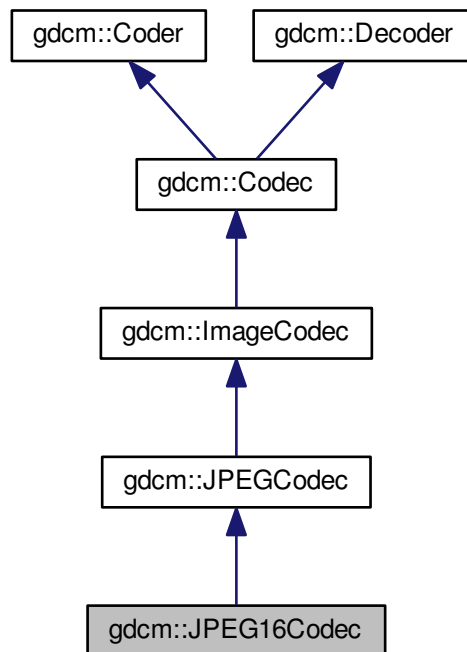
- [gdcmJPEG12Codec.h](#)

27.166 gdcm::JPEG16Codec Class Reference

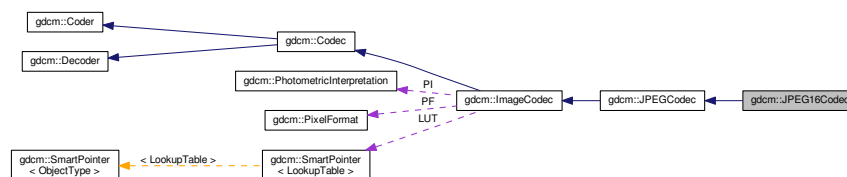
Class to do JPEG 16bits (lossless)

```
#include <gdcmJPEG16Codec.h>
```

Inheritance diagram for gdcm::JPEG16Codec:



Collaboration diagram for gdcm::JPEG16Codec:



Public Member Functions

- [JPEG16Codec](#) ()
- [~JPEG16Codec](#) ()
- [bool DecodeByStreams](#) (std::istream &is, std::ostream &os)

- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts)
- bool [InternalCode](#) (const char *input, unsigned long len, std::ostream &os)

Protected Member Functions

- virtual bool [EncodeBuffer](#) (std::ostream &os, const char *data, size_t datalen)
- bool [IsStateSuspension](#) () const

Additional Inherited Members

27.166.1 Detailed Description

Class to do JPEG 16bits (lossless)

Note

internal class

27.166.2 Constructor & Destructor Documentation

27.166.2.1 [gdcm::JPEG16Codec::JPEG16Codec \(\)](#)

27.166.2.2 [gdcm::JPEG16Codec::~~JPEG16Codec \(\)](#)

27.166.3 Member Function Documentation

27.166.3.1 [bool gdcm::JPEG16Codec::DecodeByStreams \(std::istream & is, std::ostream & os \)](#) [virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.166.3.2 [virtual bool gdcm::JPEG16Codec::EncodeBuffer \(std::ostream & os, const char * data, size_t datalen \)](#)
[protected], [virtual]

Reimplemented from [gdcm::JPEGCodec](#).

27.166.3.3 [bool gdcm::JPEG16Codec::GetHeaderInfo \(std::istream & is, TransferSyntax & ts \)](#) [virtual]

Reimplemented from [gdcm::JPEGCodec](#).

27.166.3.4 [bool gdcm::JPEG16Codec::InternalCode \(const char * input, unsigned long len, std::ostream & os \)](#) [virtual]

Reimplemented from [gdcm::Coder](#).

27.166.3.5 [bool gdcm::JPEG16Codec::IsStateSuspension \(\) const](#) [protected], [virtual]

Reimplemented from [gdcm::JPEGCodec](#).

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

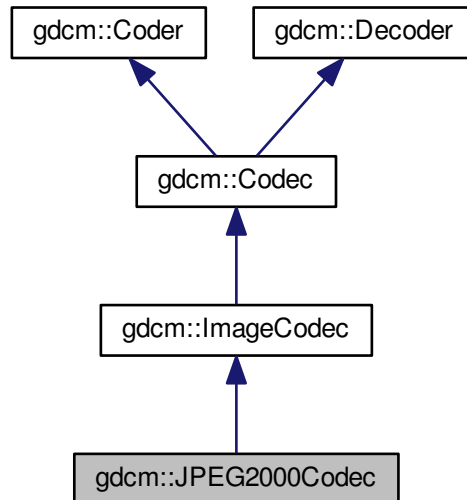
- [gdcmJPEG16Codec.h](#)

27.167 gdcm::JPEG2000Codec Class Reference

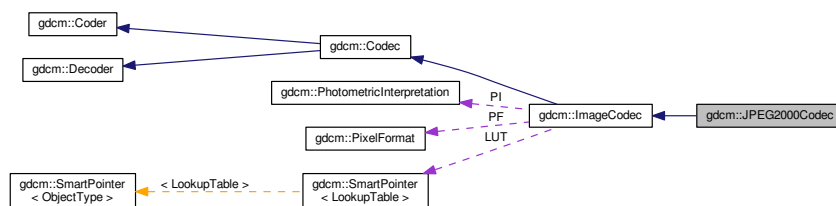
Class to do JPEG 2000.

```
#include <gdcmJPEG2000Codec.h>
```

Inheritance diagram for gdcm::JPEG2000Codec:



Collaboration diagram for gdcm::JPEG2000Codec:



Public Member Functions

- [JPEG2000Codec](#) ()
- [~JPEG2000Codec](#) ()
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const
Return whether this coder support this transfer syntax (can code it)
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const

Return whether this decoder support this transfer syntax (can decode it)

- virtual [ImageCodec](#) * [Clone](#) () const
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out)

Code.

- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os)

Decode.

- virtual bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts)
- double [GetQuality](#) (unsigned int idx=0) const
- double [GetRate](#) (unsigned int idx=0) const
- void [SetNumberOfResolutions](#) (unsigned int nres)
- 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)
- bool [AppendRowEncode](#) (std::ostream &out, const char *data, size_t datalen)
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os)
- 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](#) ()
- bool [IsRowEncoder](#) ()
- bool [StartEncode](#) (std::ostream &)
- bool [StopEncode](#) (std::ostream &)

Friends

- class [Bitmap](#)
- class [ImageRegionReader](#)

Additional Inherited Members

27.167.1 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](#)

27.167.2 Constructor & Destructor Documentation

27.167.2.1 [gdcm::JPEG2000Codec::JPEG2000Codec \(\)](#)

27.167.2.2 [gdcm::JPEG2000Codec::~~JPEG2000Codec \(\)](#)

27.167.3 Member Function Documentation

27.167.3.1 `bool gdcm::JPEG2000Codec::AppendFrameEncode (std::ostream & out, const char * data, size_t datalen)`
[protected], [virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.167.3.2 `bool gdcm::JPEG2000Codec::AppendRowEncode (std::ostream & out, const char * data, size_t datalen)`
[protected], [virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.167.3.3 `bool gdcm::JPEG2000Codec::CanCode (TransferSyntax const &) const` [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

27.167.3.4 `bool gdcm::JPEG2000Codec::CanDecode (TransferSyntax const &) const` [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

27.167.3.5 `virtual ImageCodec* gdcm::JPEG2000Codec::Clone () const` [virtual]

Implements [gdcm::ImageCodec](#).

27.167.3.6 `bool gdcm::JPEG2000Codec::Code (DataElement const & in_, DataElement & out_)` [virtual]

Code.

Reimplemented from [gdcm::Coder](#).

27.167.3.7 `bool gdcm::JPEG2000Codec::Decode (DataElement const &, DataElement &)` [virtual]

Decode.

Reimplemented from [gdcm::ImageCodec](#).

27.167.3.8 `bool gdcm::JPEG2000Codec::DecodeByStreams (std::istream & is, std::ostream & os)` [protected],
[virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.167.3.9 `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]

27.167.3.10 `virtual bool gdcm::JPEG2000Codec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` [virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.167.3.11 `double gdcm::JPEG2000Codec::GetQuality (unsigned int idx = 0) const`

27.167.3.12 `double gdcm::JPEG2000Codec::GetRate (unsigned int idx = 0) const`

27.167.3.13 `bool gdcm::JPEG2000Codec::IsFrameEncoder ()` [protected],[virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.167.3.14 `bool gdcm::JPEG2000Codec::IsRowEncoder ()` [protected],[virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.167.3.15 `void gdcm::JPEG2000Codec::SetNumberOfResolutions (unsigned int nres)`

27.167.3.16 `void gdcm::JPEG2000Codec::SetQuality (unsigned int idx, double q)`

27.167.3.17 `void gdcm::JPEG2000Codec::SetRate (unsigned int idx, double rate)`

27.167.3.18 `void gdcm::JPEG2000Codec::SetReversible (bool res)`

27.167.3.19 `void gdcm::JPEG2000Codec::SetTileSize (unsigned int tx, unsigned int ty)`

27.167.3.20 `bool gdcm::JPEG2000Codec::StartEncode (std::ostream &)` [protected],[virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.167.3.21 `bool gdcm::JPEG2000Codec::StopEncode (std::ostream &)` [protected],[virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.167.4 Friends And Related Function Documentation

27.167.4.1 `friend class Bitmap` [friend]

27.167.4.2 `friend class ImageRegionReader` [friend]

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

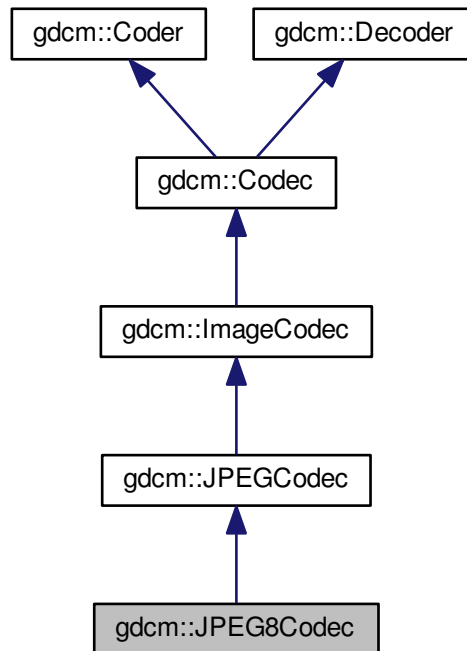
- [gdcmJPEG2000Codec.h](#)

27.168 gdcm::JPEG8Codec Class Reference

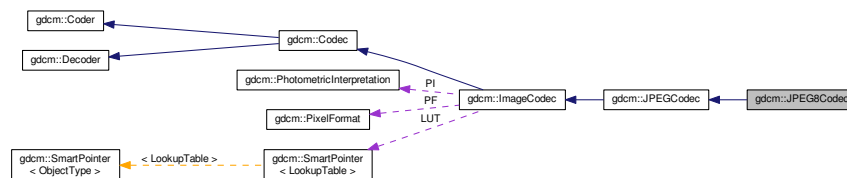
Class to do JPEG 8bits (lossy & lossless)

```
#include <gdcmJPEG8Codec.h>
```

Inheritance diagram for `gdcm::JPEG8Codec`:



Collaboration diagram for `gdcm::JPEG8Codec`:



Public Member Functions

- [JPEG8Codec](#) ()
- [~JPEG8Codec](#) ()
- [DecodeByStreams](#) (std::istream &is, std::ostream &os)
- [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts)
- [InternalCode](#) (const char *input, unsigned long len, std::ostream &os)

Protected Member Functions

- virtual bool [EncodeBuffer](#) (std::ostream &os, const char *data, size_t datalen)
- bool [IsStateSuspension](#) () const

Additional Inherited Members

27.168.1 Detailed Description

Class to do JPEG 8bits (lossy & lossless)

Note

internal class

27.168.2 Constructor & Destructor Documentation

27.168.2.1 `gdcm::JPEG8Codec::JPEG8Codec ()`

27.168.2.2 `gdcm::JPEG8Codec::~~JPEG8Codec ()`

27.168.3 Member Function Documentation

27.168.3.1 `bool gdcm::JPEG8Codec::DecodeByStreams (std::istream & is, std::ostream & os)` [virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.168.3.2 `virtual bool gdcm::JPEG8Codec::EncodeBuffer (std::ostream & os, const char * data, size_t datalen)`
[protected], [virtual]

Reimplemented from [gdcm::JPEGCodec](#).

27.168.3.3 `bool gdcm::JPEG8Codec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` [virtual]

Reimplemented from [gdcm::JPEGCodec](#).

27.168.3.4 `bool gdcm::JPEG8Codec::InternalCode (const char * input, unsigned long len, std::ostream & os)` [virtual]

Reimplemented from [gdcm::Coder](#).

27.168.3.5 `bool gdcm::JPEG8Codec::IsStateSuspension () const` [protected], [virtual]

Reimplemented from [gdcm::JPEGCodec](#).

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

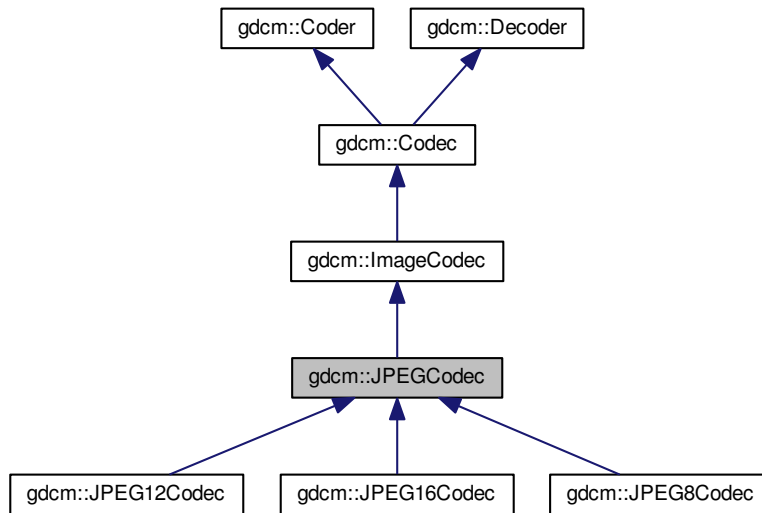
- [gdcmJPEG8Codec.h](#)

27.169 gdcm::JPEGCodec Class Reference

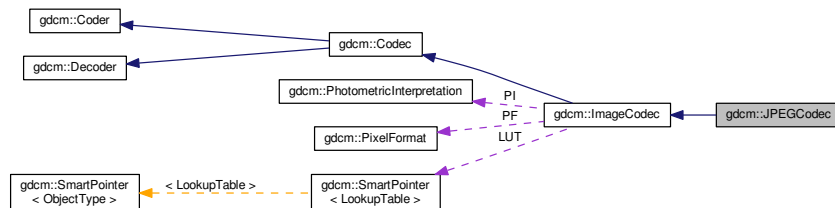
JPEG codec Class to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispach in between the different codec implementation: [JPEG8Codec](#), [JPEG12Codec](#) & [JPEG16Codec](#) It also support inconsistency in between DICO↔M header and JPEG compressed stream [ImageCodec](#) implementation for the JPEG case.

```
#include <gdcmJPEGCodec.h>
```

Inheritance diagram for gdcm::JPEGCodec:



Collaboration diagram for gdcm::JPEGCodec:



Public Member Functions

- [JPEGCodec](#) ()
- [~JPEGCodec](#) ()
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const

Return whether this coder support this transfer syntax (can code it)

- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const
Return whether this decoder support this transfer syntax (can decode it)
- virtual [ImageCodec](#) * [Clone](#) () const
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out)
Compress into JPEG.
- void [ComputeOffsetTable](#) (bool b)
Compute the offset table:
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os)
Decode.
- virtual bool [EncodeBuffer](#) (std::ostream &out, const char *inbuffer, size_t inlen)
- virtual bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts)
- bool [GetLossless](#) () const
- double [GetQuality](#) () const
- void [SetLossless](#) (bool l)
- void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetQuality](#) (double q)

Protected Member Functions

- bool [AppendFrameEncode](#) (std::ostream &out, const char *data, size_t datalen)
- bool [AppendRowEncode](#) (std::ostream &out, const char *data, size_t datalen)
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os)
- 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](#) ()
- bool [IsRowEncoder](#) ()
- virtual bool [IsStateSuspension](#) () const
- bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)
- void [SetBitSample](#) (int bit)
- bool [StartEncode](#) (std::ostream &)
- bool [StopEncode](#) (std::ostream &)

Protected Attributes

- int [BitSample](#)
- int [Quality](#)

Friends

- class [ImageRegionReader](#)

Additional Inherited Members

27.169.1 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 DICO←M 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/75fdfccc65a62
- http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/2d525ef6a2f09
- http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/6b93af410f8c9

Examples:

[GetJPEGSamplePrecision.cxx](#).

27.169.2 Constructor & Destructor Documentation

27.169.2.1 `gdcm::JPEGCodec::JPEGCodec ()`

27.169.2.2 `gdcm::JPEGCodec::~~JPEGCodec ()`

27.169.3 Member Function Documentation

27.169.3.1 `bool gdcm::JPEGCodec::AppendFrameEncode (std::ostream & out, const char * data, size_t datalen)`
[protected], [virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.169.3.2 `bool gdcm::JPEGCodec::AppendRowEncode (std::ostream & out, const char * data, size_t datalen)`
[protected], [virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.169.3.3 `bool gdcm::JPEGCodec::CanCode (TransferSyntax const &) const` [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

27.169.3.4 `bool gdcm::JPEGCodec::CanDecode (TransferSyntax const &) const` [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

27.169.3.5 `virtual ImageCodec* gdcm::JPEGCodec::Clone () const` [virtual]

Implements [gdcm::ImageCodec](#).

27.169.3.6 `bool gdcm::JPEGCodec::Code (DataElement const & in, DataElement & out)` [virtual]

Compress into JPEG.

Reimplemented from [gdcm::Coder](#).

27.169.3.7 `void gdcm::JPEGCodec::ComputeOffsetTable (bool b)`

Compute the offset table:

27.169.3.8 `bool gdcm::JPEGCodec::Decode (DataElement const & , DataElement &)` [virtual]

Decode.

Reimplemented from [gdcm::ImageCodec](#).

27.169.3.9 `bool gdcm::JPEGCodec::DecodeByStreams (std::istream & is, std::ostream & os)` [protected],[virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.169.3.10 `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]

27.169.3.11 `virtual bool gdcm::JPEGCodec::EncodeBuffer (std::ostream & out, const char * inbuffer, size_t inlen)`
[virtual]

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

27.169.3.12 `virtual bool gdcm::JPEGCodec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` [virtual]

Reimplemented from [gdcm::ImageCodec](#).

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

Examples:

[GetJPEGSamplePrecision.cxx](#).

27.169.3.13 `bool gdcm::JPEGCodec::GetLossless ()` const

27.169.3.14 `double gdcm::JPEGCodec::GetQuality ()` const

27.169.3.15 `bool gdcm::JPEGCodec::IsFrameEncoder ()` [protected],[virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.169.3.16 `bool gdcm::JPEGCodec::IsRowEncoder ()` [protected],[virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.169.3.17 `virtual bool gdcM::JPEGCodec::IsStateSuspension () const` [protected],[virtual]

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

27.169.3.18 `bool gdcM::JPEGCodec::IsValid (PhotometricInterpretation const & pi)` [protected],[virtual]

Reimplemented from [gdcM::ImageCodec](#).

27.169.3.19 `void gdcM::JPEGCodec::SetBitSample (int bit)` [protected]

27.169.3.20 `void gdcM::JPEGCodec::SetLossless (bool l)`

27.169.3.21 `void gdcM::JPEGCodec::SetPixelFormat (PixelFormat const & pf)` [virtual]

Reimplemented from [gdcM::ImageCodec](#).

Examples:

[GetJPEGSamplePrecision.cxx](#).

27.169.3.22 `void gdcM::JPEGCodec::SetQuality (double q)`

27.169.3.23 `bool gdcM::JPEGCodec::StartEncode (std::ostream &)` [protected],[virtual]

Reimplemented from [gdcM::ImageCodec](#).

27.169.3.24 `bool gdcM::JPEGCodec::StopEncode (std::ostream &)` [protected],[virtual]

Reimplemented from [gdcM::ImageCodec](#).

27.169.4 Friends And Related Function Documentation

27.169.4.1 `friend class ImageRegionReader` [friend]

27.169.5 Member Data Documentation

27.169.5.1 `int gdcM::JPEGCodec::BitSample` [protected]

27.169.5.2 `int gdcM::JPEGCodec::Quality` [protected]

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

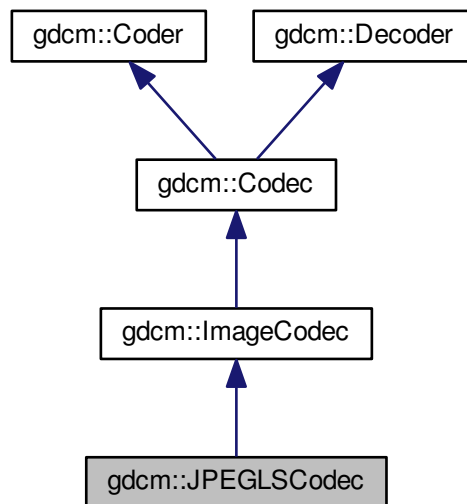
- [gdcMJPEGCodec.h](#)

27.170 gdcM::JPEGLSCodec Class Reference

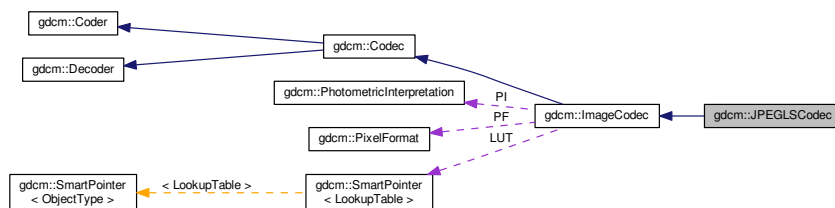
JPEG-LS.


```
#include <gdcmJPEGLSCodec.h>
```

Inheritance diagram for gdcm::JPEGLSCodec:



Collaboration diagram for gdcm::JPEGLSCodec:



Public Member Functions

- `JPEGLSCodec` ()
- `~JPEGLSCodec` ()
- `bool CanCode (TransferSyntax const &ts) const`
Return whether this coder support this transfer syntax (can code it)
- `bool CanDecode (TransferSyntax const &ts) const`
Return whether this decoder support this transfer syntax (can decode it)
- `virtual ImageCodec * Clone () const`
- `bool Code (DataElement const &in, DataElement &out)`
Code.

- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os)
Decode.
 - 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)
 - unsigned long [GetBufferLength](#) () const
 - bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts)
 - 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)
- bool [AppendRowEncode](#) (std::ostream &out, const char *data, size_t datalen)
- 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](#) ()
- bool [IsRowEncoder](#) ()
- bool [StartEncode](#) (std::ostream &)
- bool [StopEncode](#) (std::ostream &)

Friends

- class [ImageRegionReader](#)

Additional Inherited Members

27.170.1 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 <http://charls.codeplex.com>

27.170.2 Constructor & Destructor Documentation

27.170.2.1 `gdcm::JPEGLSCodec::JPEGLSCodec ()`

27.170.2.2 `gdcm::JPEGLSCodec::~~JPEGLSCodec ()`

27.170.3 Member Function Documentation

27.170.3.1 `bool gdcm::JPEGLSCodec::AppendFrameEncode (std::ostream & out, const char * data, size_t datalen)`
[protected], [virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.170.3.2 `bool gdcm::JPEGLSCodec::AppendRowEncode (std::ostream & out, const char * data, size_t datalen)`
[protected], [virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.170.3.3 `bool gdcm::JPEGLSCodec::CanCode (TransferSyntax const &) const` [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

27.170.3.4 `bool gdcm::JPEGLSCodec::CanDecode (TransferSyntax const &) const` [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

27.170.3.5 `virtual ImageCodec* gdcm::JPEGLSCodec::Clone () const` [virtual]

Implements [gdcm::ImageCodec](#).

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

Code.

Reimplemented from [gdcm::Coder](#).

27.170.3.7 `bool gdcm::JPEGLSCodec::Decode (DataElement const & , DataElement &)` [virtual]

Decode.

Reimplemented from [gdcm::ImageCodec](#).

27.170.3.8 `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)`

27.170.3.9 `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]

27.170.3.10 `unsigned long gdcm::JPEGLSCodec::GetBufferLength () const` [inline]

27.170.3.11 `bool gdcm::JPEGLSCodec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` [virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.170.3.12 `bool gdcm::JPEGLSCodec::GetLossless () const`

27.170.3.13 `bool gdcm::JPEGLSCodec::IsFrameEncoder ()` [protected], [virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.170.3.14 `bool gdcm::JPEGLSCodec::IsRowEncoder ()` [protected],[virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.170.3.15 `void gdcm::JPEGLSCodec::SetBufferLength (unsigned long /)` [inline]

27.170.3.16 `void gdcm::JPEGLSCodec::SetLossless (bool /)`

27.170.3.17 `void gdcm::JPEGLSCodec::SetLossyError (int error)`

[0-3] generally

27.170.3.18 `bool gdcm::JPEGLSCodec::StartEncode (std::ostream &)` [protected],[virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.170.3.19 `bool gdcm::JPEGLSCodec::StopEncode (std::ostream &)` [protected],[virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.170.4 Friends And Related Function Documentation

27.170.4.1 `friend class ImageRegionReader` [friend]

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

- [gdcmJPEGLSCodec.h](#)

27.171 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)

27.171.1 Detailed Description

Examples:

[QIDO-RS.cxx](#).

27.171.2 Constructor & Destructor Documentation

27.171.2.1 gdcm::JSON::JSON ()

27.171.2.2 gdcm::JSON::~~JSON ()

27.171.3 Member Function Documentation

27.171.3.1 bool gdcm::JSON::Code (DataSet const & *in*, std::ostream & *os*)

Examples:

[QIDO-RS.cxx](#).

27.171.3.2 bool gdcm::JSON::Decode (std::istream & *is*, DataSet & *out*)

Examples:

[QIDO-RS.cxx](#).

27.171.3.3 bool gdcm::JSON::GetPrettyPrint () const

27.171.3.4 void gdcm::JSON::PrettyPrintOff ()

27.171.3.5 void gdcm::JSON::PrettyPrintOn ()

Examples:

[QIDO-RS.cxx](#).

27.171.3.6 void gdcm::JSON::SetPrettyPrint (bool *onoff*)

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

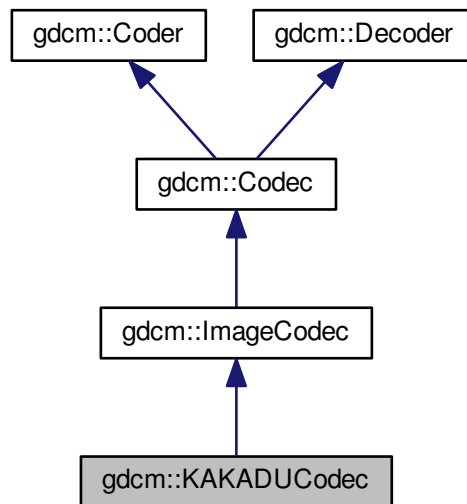
- [gdcmJSON.h](#)

27.172 gdcm::KAKADUCodec Class Reference

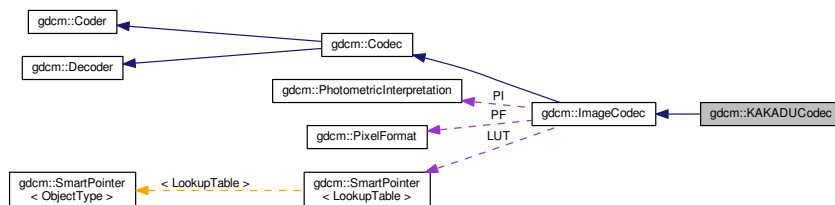
[KAKADUCodec](#).

```
#include <gdcmKAKADUCodec.h>
```

Inheritance diagram for `gdcm::KAKADUCodec`:



Collaboration diagram for `gdcm::KAKADUCodec`:



Public Member Functions

- [KAKADUCodec](#) ()
- [~KAKADUCodec](#) ()
- [CanCode](#) ([TransferSyntax](#) const &ts) const
Return whether this coder support this transfer syntax (can code it)
- [CanDecode](#) ([TransferSyntax](#) const &ts) const
Return whether this decoder support this transfer syntax (can decode it)
- virtual [ImageCodec](#) * [Clone](#) () const
- [Code](#) ([DataElement](#) const &in, [DataElement](#) &out)
Code.
- [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os)
Decode.

Additional Inherited Members

27.172.1 Detailed Description

[KAKADUCodec](#).

27.172.2 Constructor & Destructor Documentation

27.172.2.1 `gdcm::KAKADUCodec::KAKADUCodec ()`

27.172.2.2 `gdcm::KAKADUCodec::~~KAKADUCodec ()`

27.172.3 Member Function Documentation

27.172.3.1 `bool gdcm::KAKADUCodec::CanCode (TransferSyntax const &) const` `[virtual]`

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

27.172.3.2 `bool gdcm::KAKADUCodec::CanDecode (TransferSyntax const &) const` `[virtual]`

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

27.172.3.3 `virtual ImageCodec* gdcm::KAKADUCodec::Clone () const` `[virtual]`

Implements [gdcm::ImageCodec](#).

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

Code.

Reimplemented from [gdcm::Coder](#).

27.172.3.5 `bool gdcm::KAKADUCodec::Decode (DataElement const &, DataElement &)` `[virtual]`

Decode.

Reimplemented from [gdcm::ImageCodec](#).

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

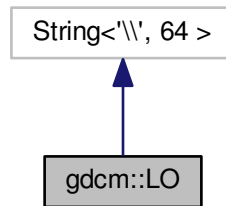
- [gdcmKAKADUCodec.h](#)

27.173 gdcm::LO Class Reference

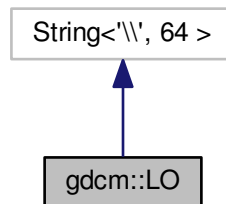
[LO](#).

```
#include <gdcmLO.h>
```

Inheritance diagram for `gdc::LO`:



Collaboration diagram for `gdc::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 [value_type](#) *s)
- [LO](#) (const [value_type](#) *s, [size_type](#) n)
- [LO](#) (const [Superclass](#) &s, [size_type](#) pos=0, [size_type](#) n=npow)
- bool [IsValid](#) () const

27.173.1 Detailed Description

[LO](#).

Note

TODO

27.173.2 Member Typedef Documentation

27.173.2.1 typedef [Superclass::const_iterator](#) [gdcm::LO::const_iterator](#)

27.173.2.2 typedef [Superclass::const_reference](#) [gdcm::LO::const_reference](#)

27.173.2.3 typedef [Superclass::const_reverse_iterator](#) [gdcm::LO::const_reverse_iterator](#)

27.173.2.4 typedef [Superclass::difference_type](#) [gdcm::LO::difference_type](#)

27.173.2.5 typedef [Superclass::iterator](#) [gdcm::LO::iterator](#)

27.173.2.6 typedef [Superclass::pointer](#) [gdcm::LO::pointer](#)

27.173.2.7 typedef [Superclass::reference](#) [gdcm::LO::reference](#)

27.173.2.8 typedef [Superclass::reverse_iterator](#) [gdcm::LO::reverse_iterator](#)

27.173.2.9 typedef [Superclass::size_type](#) [gdcm::LO::size_type](#)

27.173.2.10 typedef [String<'\',64>](#) [gdcm::LO::Superclass](#)

27.173.2.11 typedef [Superclass::value_type](#) [gdcm::LO::value_type](#)

27.173.3 Constructor & Destructor Documentation

27.173.3.1 [gdcm::LO::LO](#) () [\[inline\]](#)

27.173.3.2 [gdcm::LO::LO](#) (const [value_type](#) * s) [\[inline\]](#)

27.173.3.3 [gdcm::LO::LO](#) (const [value_type](#) * s, [size_type](#) n) [\[inline\]](#)

27.173.3.4 [gdcm::LO::LO](#) (const [Superclass](#) & s, [size_type](#) pos = 0, [size_type](#) n = npow) [\[inline\]](#)

27.173.4 Member Function Documentation

27.173.4.1 bool [gdcm::LO::IsValid](#) () const [\[inline\]](#)

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

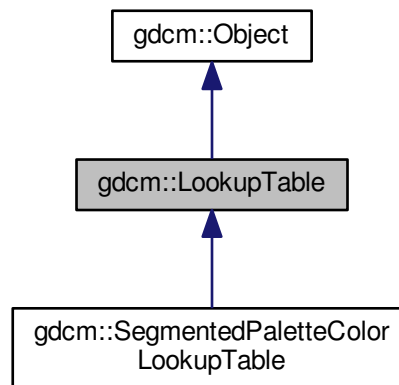
- [gdcmLO.h](#)

27.174 gdcm::LookupTable Class Reference

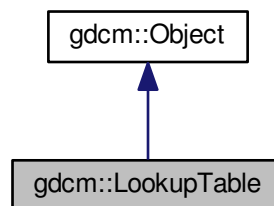
[LookupTable](#) class.

```
#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](#) ()
- void [Allocate](#) (unsigned short bitsample=8)
Allocate the LUT.
- void [Clear](#) ()
Clear the LUT.
- void [Decode](#) (std::istream &is, std::ostream &os) const
Decode the LUT.
- bool [Decode](#) (char *outputbuffer, size_t outlen, const char *inputbuffer, size_t inlen) const
- 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:
- void [Print](#) (std::ostream &) const
- 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

27.174.1 Detailed Description

[LookupTable](#) class.

27.174.2 Member Enumeration Documentation

27.174.2.1 enum gdcm::LookupTable::LookupTableType

Enumerator

RED

GREEN

BLUE

GRAY

UNKNOWN

27.174.3 Constructor & Destructor Documentation

27.174.3.1 gdcm::LookupTable::LookupTable ()

27.174.3.2 gdcm::LookupTable::~~LookupTable ()

27.174.3.3 gdcm::LookupTable::LookupTable ([LookupTable](#) const & *lut*) `[inline]`

27.174.4 Member Function Documentation

27.174.4.1 void gdcm::LookupTable::Allocate (unsigned short *bitsample* = 8)

Allocate the LUT.

27.174.4.2 void gdcm::LookupTable::Clear ()

Clear the LUT.

27.174.4.3 void gdcm::LookupTable::Decode (std::istream & *is*, std::ostream & *os*) const

Decode the LUT.

27.174.4.4 `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

27.174.4.5 `unsigned short gdcmm::LookupTable::GetBitSample () const [inline]`

return the bit sample

27.174.4.6 `bool gdcmm::LookupTable::GetBufferAsRGBA (unsigned char * rgba) const`

return the LUT as RGBA buffer

27.174.4.7 `void gdcmm::LookupTable::GetLUT (LookupTableType type, unsigned char * array, unsigned int & length) const`

Examples:

[ExtractImageRegionWithLUT.cs](#).

27.174.4.8 `void gdcmm::LookupTable::GetLUTDescriptor (LookupTableType type, unsigned short & length, unsigned short & subscript, unsigned short & bitsize) const`

27.174.4.9 `unsigned int gdcmm::LookupTable::GetLUTLength (LookupTableType type) const`

27.174.4.10 `const unsigned char* gdcmm::LookupTable::GetPointer () const`

return a raw pointer to the LUT

27.174.4.11 `void gdcmm::LookupTable::InitializeBlueLUT (unsigned short length, unsigned short subscript, unsigned short bitsize)`

27.174.4.12 `bool gdcmm::LookupTable::Initialized () const`

return whether the LUT has been initialized

27.174.4.13 `void gdcmm::LookupTable::InitializeGreenLUT (unsigned short length, unsigned short subscript, unsigned short bitsize)`

27.174.4.14 `void gdcmm::LookupTable::InitializeLUT (LookupTableType type, unsigned short length, unsigned short subscript, unsigned short bitsize)`

Generic interface:

27.174.4.15 `void gdcmm::LookupTable::InitializeRedLUT (unsigned short length, unsigned short subscript, unsigned short bitsize)`

RED / GREEN / BLUE specific:

27.174.4.16 void gdcM::LookupTable::Print (std::ostream &) const [inline],[virtual]

Reimplemented from [gdcM::Object](#).

Reimplemented in [gdcM::SegmentedPaletteColorLookupTable](#).

27.174.4.17 void gdcM::LookupTable::SetBlueLUT (const unsigned char * *blue*, unsigned int *length*)

27.174.4.18 void gdcM::LookupTable::SetGreenLUT (const unsigned char * *green*, unsigned int *length*)

27.174.4.19 virtual void gdcM::LookupTable::SetLUT (LookupTableType *type*, const unsigned char * *array*, unsigned int *length*) [virtual]

Reimplemented in [gdcM::SegmentedPaletteColorLookupTable](#).

27.174.4.20 void gdcM::LookupTable::SetRedLUT (const unsigned char * *red*, unsigned int *length*)

27.174.4.21 bool gdcM::LookupTable::WriteBufferAsRGBA (const unsigned char * *rgba*)

Write the LUT as RGBA.

27.174.5 Member Data Documentation

27.174.5.1 unsigned short gdcM::LookupTable::BitSample [protected]

27.174.5.2 bool gdcM::LookupTable::IncompleteLUT [protected]

27.174.5.3 LookupTableInternal* gdcM::LookupTable::Internal [protected]

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

- [gdcMLookupTable.h](#)

27.175 gdcM::Scanner::Itstr Struct Reference

```
#include <gdcMScanner.h>
```

Public Member Functions

- bool [operator\(\)](#) (const char **s1*, const char **s2*) const

27.175.1 Member Function Documentation

27.175.1.1 bool gdcM::Scanner::Itstr::operator() (const char * *s1*, const char * *s2*) const [inline]

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

- [gdcMScanner.h](#)

27.176 gdcmm::StrictScanner::ltstr Struct Reference

```
#include <gdcmmStrictScanner.h>
```

Public Member Functions

- bool [operator\(\)](#) (const char *s1, const char *s2) const

27.176.1 Member Function Documentation

27.176.1.1 bool gdcmm::StrictScanner::ltstr::operator() (const char * s1, const char * s2) const [inline]

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

- [gdcmmStrictScanner.h](#)

27.177 gdcmm::Macro Class Reference

Class for representing a [Macro](#).

```
#include <gdcmmMacro.h>
```

Public Types

- typedef std::vector< std::string > [ArrayIncludeMacrosType](#)
- typedef std::map< [Tag](#), [MacroEntry](#) > [MapModuleEntry](#)

Public Member Functions

- [Macro](#) ()
- 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)

27.177.1 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](#)

27.177.2 Member Typedef Documentation

27.177.2.1 `typedef std::vector<std::string> gdcmmacro::Macro::ArrayIncludeMacrosType`

27.177.2.2 `typedef std::map<Tag, MacroEntry> gdcmmacro::Macro::MapModuleEntry`

27.177.3 Constructor & Destructor Documentation

27.177.3.1 `gdcmmacro::Macro::Macro () [inline]`

27.177.4 Member Function Documentation

27.177.4.1 `void gdcmmacro::Macro::AddMacroEntry (const Tag & tag, const MacroEntry & module) [inline]`

Will add a [ModuleEntry](#) directly at root-level. See [Macro](#) for nested-included level.

27.177.4.2 `void gdcmmacro::Macro::Clear () [inline]`

27.177.4.3 `bool gdcmmacro::Macro::FindMacroEntry (const Tag & tag) const`

Find or Get a [ModuleEntry](#). [ModuleEntry](#) are either search are root-level or within nested-macro included in module.

27.177.4.4 `const MacroEntry& gdcmmacro::Macro::GetMacroEntry (const Tag & tag) const`

27.177.4.5 `const char* gdcmmacro::Macro::GetName () const [inline]`

27.177.4.6 `void gdcmmacro::Macro::SetName (const char * name) [inline]`

27.177.4.7 `bool gdcmmacro::Macro::Verify (const DataSet & ds, Usage const & usage) const`

27.177.5 Friends And Related Function Documentation

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

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

- [gdcmmacro.h](#)

27.178 gdcmmacros::Macros Class Reference

Class for representing a [Modules](#).

```
#include <gdcmmacros.h>
```

Public Types

- typedef std::map< std::string, [Macro](#) > [ModuleMapType](#)

Public Member Functions

- [Macros](#) ()
- 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)

27.178.1 Detailed Description

Class for representing a [Modules](#).

Note

bla

See also

[Module](#)

Examples:

[TraverseModules.cxx](#).

27.178.2 Member Typedef Documentation

27.178.2.1 typedef std::map<std::string, [Macro](#)> [gdcmmacros::Macros::ModuleMapType](#)

27.178.3 Constructor & Destructor Documentation

27.178.3.1 [gdcmmacros::Macros](#) () `[inline]`

27.178.4 Member Function Documentation

27.178.4.1 void [gdcmmacros::Macros::AddMacro](#) (const char * *ref*, const [Macro](#) & *module*) `[inline]`

27.178.4.2 `void gdcM::Macros::Clear () [inline]`

27.178.4.3 `const Macro& gdcM::Macros::GetMacro (const char * name) const [inline]`

27.178.4.4 `bool gdcM::Macros::IsEmpty () const [inline]`

27.178.5 Friends And Related Function Documentation

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

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

- [gdcMMacros.h](#)

27.179 gdcM::network::MaximumLengthSub Class Reference

[MaximumLengthSub](#) Annex D [Table D.1-1](#) MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

```
#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`

27.179.1 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)

27.179.2 Constructor & Destructor Documentation

27.179.2.1 `gdcM::network::MaximumLengthSub::MaximumLengthSub ()`

27.179.3 Member Function Documentation

27.179.3.1 `uint32_t gdcM::network::MaximumLengthSub::GetMaximumLength () const [inline]`

27.179.3.2 `void gdcM::network::MaximumLengthSub::Print (std::ostream & os) const`

27.179.3.3 `std::istream& gdcM::network::MaximumLengthSub::Read (std::istream & is)`

27.179.3.4 void gdcm::network::MaximumLengthSub::SetMaximumLength (uint32_t *maximumlength*)

27.179.3.5 size_t gdcm::network::MaximumLengthSub::Size () const

27.179.3.6 const std::ostream& gdcm::network::MaximumLengthSub::Write (std::ostream & *os*) const

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

- [gdcmMaximumLengthSub.h](#)

27.180 gdcm::MD5 Class Reference

Class for [MD5](#).

```
#include <gdcmMD5.h>
```

Public Member Functions

- [MD5](#) ()
- [~MD5](#) ()

Static Public Member Functions

- static bool [Compute](#) (const char *buffer, unsigned long buf_len, char digest_str[33])
- static bool [ComputeFile](#) (const char *filename, char digest_str[33])

27.180.1 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

27.180.2 Constructor & Destructor Documentation

27.180.2.1 gdcm::MD5::MD5 ()

27.180.2.2 gdcm::MD5::~~MD5 ()

27.180.3 Member Function Documentation

27.180.3.1 static bool gdcm::MD5::Compute (const char * *buffer*, unsigned long *buf_len*, char *digest_str*[33]) [static]

27.180.3.2 `static bool gdcM::MD5::ComputeFile (const char * filename, char digest_str[33]) [static]`

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

- [gdcMMD5.h](#)

27.181 `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`,
 - `XRayRadiofluoroscopicImageStorage`,
 - `XRayAngiographicBiPlaneImageStorageRetired`,
 - `NuclearMedicineImageStorage`,
 - `RawDataStorage`,
 - `SpacialRegistrationStorage`,
 - `SpacialFiducialsStorage`,
 - `PETImageStorage`,
 - `RTImageStorage`,
 - `RTDoseStorage`,
 - `RTStructureSetStorage`,
 - `RTPlanStorage`,
 - `CSANonImageStorage`,
 - `Philips3D`,
 - `EnhancedSR`,
 - `BasicTextSR`,
 - `HardcopyGrayscaleImageStorage`,
 - `ComprehensiveSR`,
 - `DetachedStudyManagementSOPClass`,
 - `EncapsulatedCDImageStorage`,
 - `EncapsulatedCDASStorage`,
 - `StudyComponentManagementSOPClass`,
 - `DetachedVisitManagementSOPClass`,
 - `DetachedPatientManagementSOPClass`

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)

27.181.1 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.cxx](#), [gdcmmrtionplan.cxx](#), [gdcmmrtplan.cxx](#), [GenAllVR.cxx](#), [GenerateStandardSOPClasses.cxx](#), [GenFakeIdentifyFile.cxx](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [StreamImageReaderTest.cxx](#), and [TestReader.cxx](#).

27.181.2 Member Enumeration Documentation

27.181.2.1 enum gdcmm::MediaStorage::MSType

Enumerator

MediaStorageDirectoryStorage
ComputedRadiographylImageStorage
DigitalXRayImageStorageForPresentation
DigitalXRayImageStorageForProcessing
DigitalMammographylImageStorageForPresentation
DigitalMammographylImageStorageForProcessing
DigitalIntraoralXrayImageStorageForPresentation
DigitalIntraoralXRayImageStorageForProcessing
CTImageStorage
EnhancedCTImageStorage
UltrasoundImageStorageRetired
UltrasoundImageStorage
UltrasoundMultiFrameImageStorageRetired
UltrasoundMultiFrameImageStorage
MRIImageStorage
EnhancedMRIImageStorage
MRSpectroscopyStorage
NuclearMedicineImageStorageRetired
SecondaryCaptureImageStorage
MultiframeSingleBitSecondaryCaptureImageStorage
MultiframeGrayscaleByteSecondaryCaptureImageStorage

MultiframeGrayscaleWordSecondaryCaptureImageStorage

MultiframeTrueColorSecondaryCaptureImageStorage

StandaloneOverlayStorage

StandaloneCurveStorage

LeadECGWaveformStorage

GeneralECGWaveformStorage

AmbulatoryECGWaveformStorage

HemodynamicWaveformStorage

CardiacElectrophysiologyWaveformStorage

BasicVoiceAudioWaveformStorage

StandaloneModalityLUTStorage

StandaloneVOILUTStorage

GrayscaleSoftcopyPresentationStateStorageSOPClass

XRayAngiographicImageStorage

XRayRadiofluoroscopicImageStorage

XRayAngiographicBiPlaneImageStorageRetired

NuclearMedicineImageStorage

RawDataStorage

SpacialRegistrationStorage

SpacialFiducialsStorage

PETImageStorage

RTImageStorage

RTDoseStorage

RTStructureSetStorage

RTPlanStorage

CSANonImageStorage

Philips3D

EnhancedSR

BasicTextSR

HardcopyGrayscaleImageStorage

ComprehensiveSR

DetachedStudyManagementSOPClass

EncapsulatedPDFStorage

EncapsulatedCDASStorage

StudyComponentManagementSOPClass

DetachedVisitManagementSOPClass

DetachedPatientManagementSOPClass

VideoEndoscopicImageStorage

GeneralElectricMagneticResonanceImageStorage

GEPrivate3DModelStorage

ToshibaPrivateDataStorage

MammographyCADSR

KeyObjectSelectionDocument
HangingProtocolStorage
ModalityPerformedProcedureStepSOPClass
PhilipsPrivateMRSyntheticImageStorage
VLPhotographicImageStorage
SegmentationStorage
RTIonPlanStorage
XRay3DAngiographicImageStorage
EnhancedXAImageStorage
RTIonBeamsTreatmentRecordStorage
SurfaceSegmentationStorage
VLWholeSlideMicroscopyImageStorage
RTTreatmentSummaryRecordStorage
EnhancedUSVolumeStorage
XRayRadiationDoseSR
VLEndoscopicImageStorage
BreastTomosynthesisImageStorage
FujiPrivateCRLImageStorage
OphthalmicPhotography8BitImageStorage
OphthalmicTomographyImageStorage
VLMicroscopicImageStorage
EnhancedPETImageStorage
VideoPhotographicImageStorage
MS_END

Examples:

[GenerateStandardSOPClasses.cxx](#).

27.181.2.2 enum gdcmm::MediaStorage::ObjectType

Enumerator

NoObject
Video
Waveform
Audio
PDF
URI
Segmentation
ObjectEnd

27.181.3 Constructor & Destructor Documentation

27.181.3.1 `gdcm::MediaStorage::MediaStorage (MStype type = MS_END) [inline]`

27.181.4 Member Function Documentation

27.181.4.1 `const char* gdcm::MediaStorage::GetModality () const`

27.181.4.2 `unsigned int gdcm::MediaStorage::GetModalityDimension () const`

27.181.4.3 `static const char* gdcm::MediaStorage::GetMSString (MStype ts) [static]`

Return the Media [String](#) associated. Will return NULL for MS_END.

Examples:

[GenerateStandardSOPClasses.cxx](#).

27.181.4.4 `static MStype gdcm::MediaStorage::GetMStype (const char * str) [static]`

Examples:

[MetaImageMD5Activiz.cs](#), and [TestReader.cxx](#).

27.181.4.5 `static unsigned int gdcm::MediaStorage::GetNumberOfModality () [static]`

27.181.4.6 `static unsigned int gdcm::MediaStorage::GetNumberOfMSString () [static]`

27.181.4.7 `static unsigned int gdcm::MediaStorage::GetNumberOfMStype () [static]`

27.181.4.8 `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](#), [iU22tomultisc.cxx](#), and [StreamImageReaderTest.cxx](#).

27.181.4.9 `void gdcm::MediaStorage::GuessFromModality (const char * modality, unsigned int dimension = 2)`

27.181.4.10 `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](#).

27.181.4.11 `bool gdcm::MediaStorage::IsUndefined () const [inline]`

Examples:

[TestReader.cxx](#).

27.181.4.12 `gdcm::MediaStorage::operator MType () const [inline]`

27.181.4.13 `bool gdcm::MediaStorage::SetFromDataSet (DataSet const & ds)`

Advanced user only (functions should be protected level...) Those function are lower level than SetFromFile

27.181.4.14 `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:

[gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), and [TestReader.cxx](#).

27.181.4.15 `bool gdcm::MediaStorage::SetFromHeader (FileMetaInformation const & fmi)`

27.181.4.16 `bool gdcm::MediaStorage::SetFromModality (DataSet const & ds)`

27.181.4.17 `void gdcm::MediaStorage::SetFromSourceImageSequence (DataSet const & ds) [protected]`

27.181.5 Friends And Related Function Documentation

27.181.5.1 `std::ostream& operator<< (std::ostream & os, const MediaStorage & ms) [friend]`

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

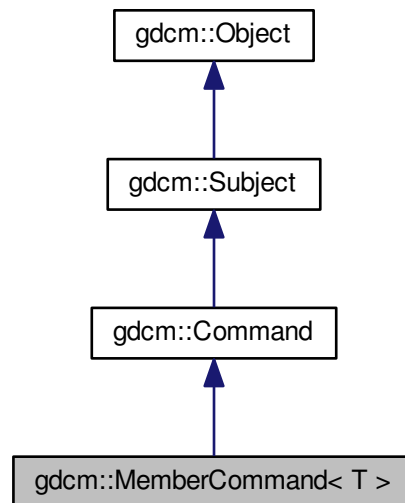
- [gdcmMediaStorage.h](#)

27.182 gdcm::MemberCommand< T > Class Template Reference

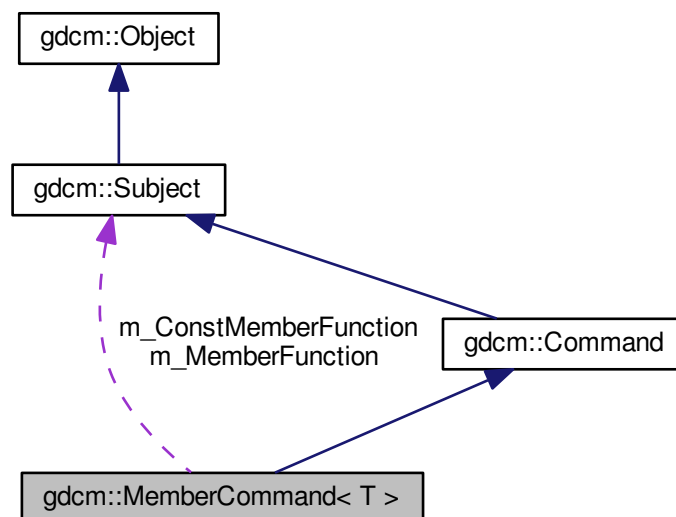
[Command](#) subclass that calls a pointer to a member function.

```
#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

- virtual void [Execute](#) ([Subject](#) *caller, const [Event](#) &event)
- virtual void [Execute](#) (const [Subject](#) *caller, const [Event](#) &event)
- void [SetCallbackFunction](#) (T *object, [TMemberFunctionPointer](#) memberFunction)
- void [SetCallbackFunction](#) (T *object, [TConstMemberFunctionPointer](#) memberFunction)

Static Public Member Functions

- static [SmartPointer](#)< [MemberCommand](#) > [New](#) ()

Protected Member Functions

- [MemberCommand](#) ()
- virtual [~MemberCommand](#) ()

Protected Attributes

- [TConstMemberFunctionPointer](#) m_ConstMemberFunction
- [TMemberFunctionPointer](#) m_MemberFunction
- T * m_This

27.182.1 Detailed Description

template<class T>class gdcmmembercommand< 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](#).

27.182.2 Member Typedef Documentation

27.182.2.1 template<class T > typedef [MemberCommand](#) gdcmmembercommand< T >::Self

Standard class typedefs.

27.182.2.2 template<class T > typedef void(T::* gdcmmembercommand< T >::TConstMemberFunctionPointer) (const [Subject](#) *, const [Event](#) &)

27.182.2.3 template<class T > typedef void(T::* gdcmmembercommand< T >::TMemberFunctionPointer) ([Subject](#) *, const [Event](#) &)

pointer to a member function that takes a [Subject](#)* and the event

27.182.3 Constructor & Destructor Documentation

27.182.3.1 `template<class T> gdcM::MemberCommand< T >::MemberCommand () [inline],
[protected]`

Referenced by `gdcM::MemberCommand< T >::New()`.

27.182.3.2 `template<class T> virtual gdcM::MemberCommand< T >::~~MemberCommand () [inline],
[protected], [virtual]`

27.182.4 Member Function Documentation

27.182.4.1 `template<class T> virtual void gdcM::MemberCommand< T >::Execute (Subject * caller, const Event &
event) [inline], [virtual]`

Invoke the member function.

Implements [gdcM::Command](#).

References `gdcM::MemberCommand< T >::m_MemberFunction`.

27.182.4.2 `template<class T> virtual void gdcM::MemberCommand< T >::Execute (const Subject * caller, const Event
& event) [inline], [virtual]`

Invoke the member function with a const object.

Implements [gdcM::Command](#).

References `gdcM::MemberCommand< T >::m_ConstMemberFunction`.

27.182.4.3 `template<class T> static SmartPointer<MemberCommand> gdcM::MemberCommand< T >::New ()
[inline], [static]`

Method for creation through the object factory.

References `gdcM::MemberCommand< T >::MemberCommand()`.

27.182.4.4 `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`.

27.182.4.5 `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`.

27.182.5 Member Data Documentation

27.182.5.1 `template<class T > TConstMemberFunctionPointer gdcm::MemberCommand< T
>::m_ConstMemberFunction` [protected]

Referenced by `gdcm::MemberCommand< T >::Execute()`, and `gdcm::MemberCommand< T >::SetCallbackFunction()`.

27.182.5.2 `template<class T > TMemberFunctionPointer gdcm::MemberCommand< T >::m_MemberFunction`
[protected]

Referenced by `gdcm::MemberCommand< T >::Execute()`, and `gdcm::MemberCommand< T >::SetCallbackFunction()`.

27.182.5.3 `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:

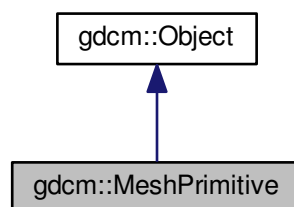
- [gdcmCommand.h](#)

27.183 **gdcm::MeshPrimitive Class Reference**

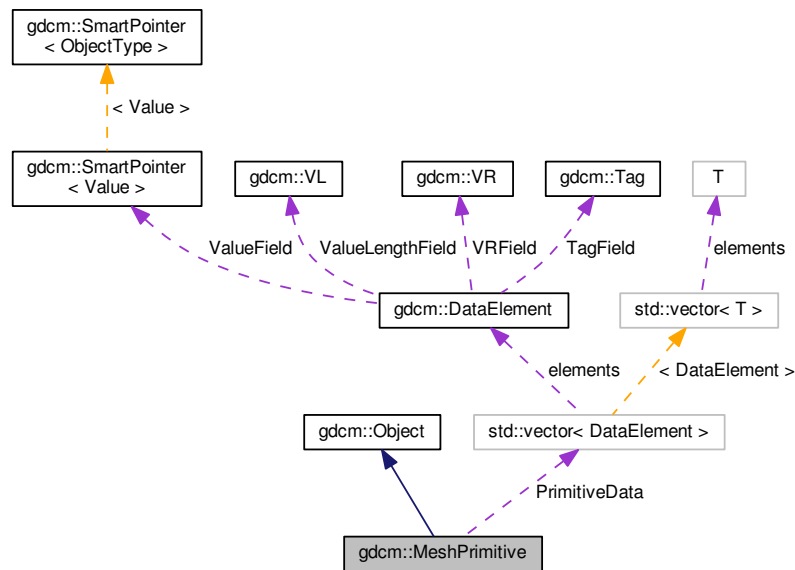
This class defines surface mesh primitives. It is designed from surface mesh primitives macro.

```
#include <gdcmMeshPrimitive.h>
```

Inheritance diagram for `gdcm::MeshPrimitive`:



Collaboration diagram for `gdc::MeshPrimitive`:



Public Types

- enum `MPTType` {
`VERTEX` = 0,
`EDGE`,
`TRIANGLE`,
`TRIANGLE_STRIP`,
`TRIANGLE_FAN`,
`LINE`,
`FACET`,
`MPTType_END` }
- This enumeration defines primitive types.*
- typedef `std::vector< DataElement >` `PrimitivesData`

Public Member Functions

- `MeshPrimitive` ()
- virtual `~MeshPrimitive` ()
- void `AddPrimitiveData` (`DataElement` const &de)
- unsigned int `GetNumberOfPrimitivesData` () const
- 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](#) ()
- [MPTType](#) [GetPrimitiveType](#) () const
- void [SetPrimitiveData](#) ([DataElement](#) const &de)
- void [SetPrimitiveData](#) (const unsigned int idx, [DataElement](#) const &de)
- void [SetPrimitivesData](#) ([PrimitivesData](#) const &DEs)
- void [SetPrimitiveType](#) (const [MPTType](#) type)

Static Public Member Functions

- static [MPTType](#) [GetMPTType](#) (const char *type)
- static const char * [GetMPTTypeString](#) (const [MPTType](#) type)

Protected Attributes

- [PrimitivesData](#) [PrimitiveData](#)
- [MPTType](#) [PrimitiveType](#)

Additional Inherited Members

27.183.1 Detailed Description

This class defines surface mesh primitives. It is designed from surface mesh primitives macro.

See also

PS 3.3 C.27.4

27.183.2 Member Typedef Documentation

27.183.2.1 `typedef std::vector< DataElement > gdcmmeshprimitive::PrimitivesData`

27.183.3 Member Enumeration Documentation

27.183.3.1 `enum gdcmmeshprimitive::MPTType`

This enumeration defines primitive types.

See also

PS 3.3 C.27.4.1

Enumerator

VERTEX
EDGE
TRIANGLE
TRIANGLE_STRIP
TRIANGLE_FAN
LINE
FACET
MPTType_END

27.183.4 Constructor & Destructor Documentation

- 27.183.4.1 `gdcM::MeshPrimitive::MeshPrimitive ()`
- 27.183.4.2 `virtual gdcM::MeshPrimitive::~~MeshPrimitive ()` [virtual]

27.183.5 Member Function Documentation

- 27.183.5.1 `void gdcM::MeshPrimitive::AddPrimitiveData (DataElement const & de)`
- 27.183.5.2 `static MPTYPE gdcM::MeshPrimitive::GetMPTYPE (const char * type)` [static]
- 27.183.5.3 `static const char* gdcM::MeshPrimitive::GetMPTYPEString (const MPTYPE type)` [static]
- 27.183.5.4 `unsigned int gdcM::MeshPrimitive::GetNumberOfPrimitivesData () const`
- 27.183.5.5 `const DataElement& gdcM::MeshPrimitive::GetPrimitiveData () const`
- 27.183.5.6 `DataElement& gdcM::MeshPrimitive::GetPrimitiveData ()`
- 27.183.5.7 `const DataElement& gdcM::MeshPrimitive::GetPrimitiveData (const unsigned int idx) const`
- 27.183.5.8 `DataElement& gdcM::MeshPrimitive::GetPrimitiveData (const unsigned int idx)`
- 27.183.5.9 `const PrimitivesData& gdcM::MeshPrimitive::GetPrimitivesData () const`
- 27.183.5.10 `PrimitivesData& gdcM::MeshPrimitive::GetPrimitivesData ()`
- 27.183.5.11 `MPTYPE gdcM::MeshPrimitive::GetPrimitiveType () const`
- 27.183.5.12 `void gdcM::MeshPrimitive::SetPrimitiveData (DataElement const & de)`
- 27.183.5.13 `void gdcM::MeshPrimitive::SetPrimitiveData (const unsigned int idx, DataElement const & de)`
- 27.183.5.14 `void gdcM::MeshPrimitive::SetPrimitivesData (PrimitivesData const & DEs)`
- 27.183.5.15 `void gdcM::MeshPrimitive::SetPrimitiveType (const MPTYPE type)`

27.183.6 Member Data Documentation

- 27.183.6.1 `PrimitivesData gdcM::MeshPrimitive::PrimitiveData` [protected]
- 27.183.6.2 `MPTYPE gdcM::MeshPrimitive::PrimitiveType` [protected]

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

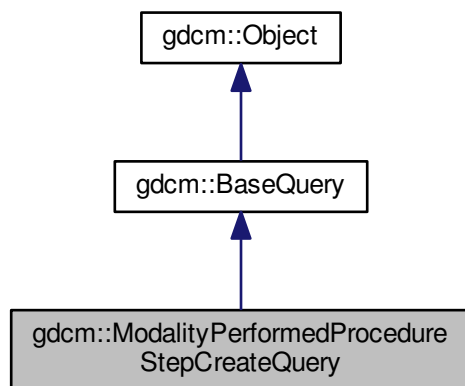
- [gdcMMeshPrimitive.h](#)

27.184 gdcm::ModalityPerformedProcedureStepCreateQuery Class Reference

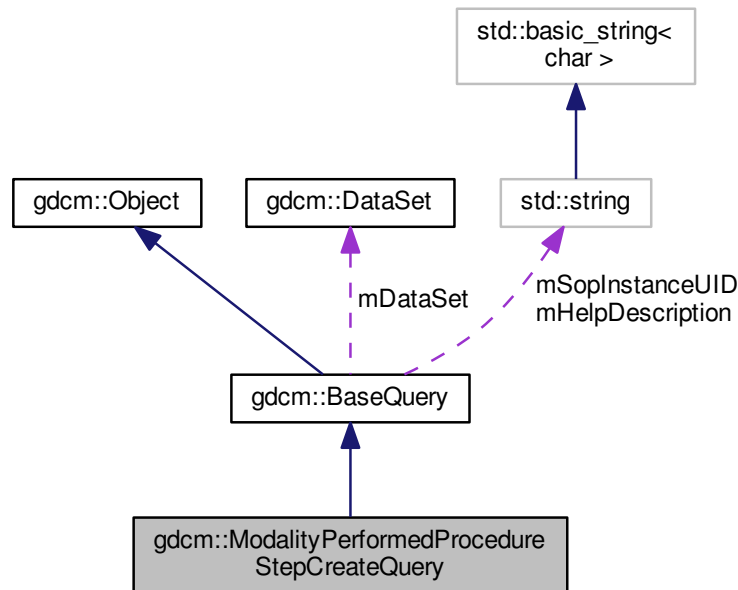
[ModalityPerformedProcedureStepCreateQuery](#) contains: the class which will produce a dataset for n-create for Modality Performed Procedure Step sop class.

```
#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
- [gdcM::DataSet GetRequiredDataSet](#) () const
- bool [ValidateQuery](#) (bool inStrict=true) const

Friends

- class [QueryFactory](#)

Additional Inherited Members

27.184.1 Detailed Description

[ModalityPerformedProcedureStepCreateQuery](#) contains: the class which will produce a dataset for n-create for Modality Performed Procedure Step sop class.

27.184.2 Constructor & Destructor Documentation

27.184.2.1 `gdcm::ModalityPerformedProcedureStepCreateQuery::ModalityPerformedProcedureStepCreateQuery (const std::string & iSopInstanceUID)`

27.184.3 Member Function Documentation

27.184.3.1 **UIDs::TSName** `gdcm::ModalityPerformedProcedureStepCreateQuery::GetAbstractSyntaxUID () const`
[virtual]

Implements [gdcm::BaseQuery](#).

27.184.3.2 **gdcm::DataSet** `gdcm::ModalityPerformedProcedureStepCreateQuery::GetRequiredDataSet () const`

27.184.3.3 **bool** `gdcm::ModalityPerformedProcedureStepCreateQuery::ValidateQuery (bool iStrict = true) const`
[virtual]

Implements [gdcm::BaseQuery](#).

27.184.4 Friends And Related Function Documentation

27.184.4.1 **friend class QueryFactory** [friend]

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

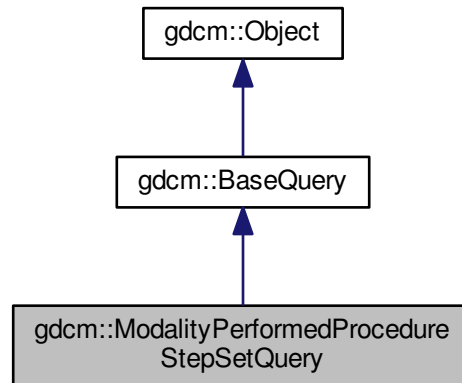
- [gdcmModalityPerformedProcedureStepCreateQuery.h](#)

27.185 gdcm::ModalityPerformedProcedureStepSetQuery Class Reference

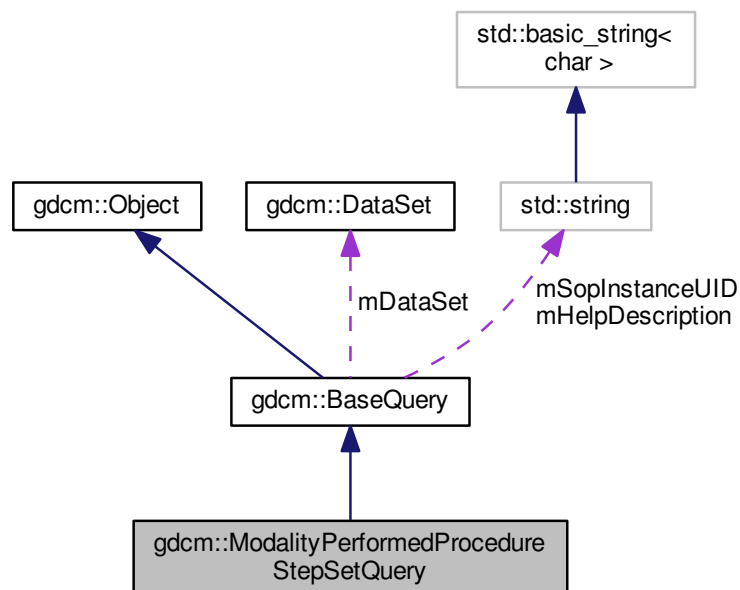
[ModalityPerformedProcedureStepSetQuery](#) contains: the class which will produce a dataset for n-set for Modality Performed Procedure Step sop class.

```
#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
- [gdcm::DataSet GetRequiredDataSet](#) () const
- bool [ValidateQuery](#) (bool inStrict=true) const

Friends

- class [QueryFactory](#)

Additional Inherited Members

27.185.1 Detailed Description

[ModalityPerformedProcedureStepSetQuery](#) contains: the class which will produce a dataset for n-set for Modality Performed Procedure Step sop class.

27.185.2 Constructor & Destructor Documentation

27.185.2.1 `gdcm::ModalityPerformedProcedureStepSetQuery::ModalityPerformedProcedureStepSetQuery (const std::string &iSopInstanceUID)`

27.185.3 Member Function Documentation

27.185.3.1 `UIDs::TSName gdcm::ModalityPerformedProcedureStepSetQuery::GetAbstractSyntaxUID () const` `[virtual]`

Implements [gdcm::BaseQuery](#).

27.185.3.2 `gdcm::DataSet gdcm::ModalityPerformedProcedureStepSetQuery::GetRequiredDataSet () const`

27.185.3.3 `bool gdcm::ModalityPerformedProcedureStepSetQuery::ValidateQuery (bool inStrict = true) const` `[virtual]`

Implements [gdcm::BaseQuery](#).

27.185.4 Friends And Related Function Documentation

27.185.4.1 `friend class QueryFactory` `[friend]`

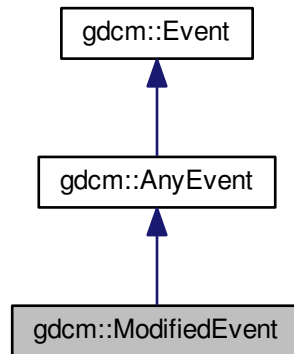
The documentation for this class was generated from the following file:

- [gdcmModalityPerformedProcedureStepSetQuery.h](#)

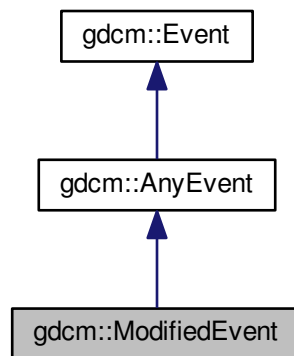
27.186 gdcm::ModifiedEvent Class Reference

```
#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](#)

27.187 gdcmmodule 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](#) ()
- 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 ¯os, const [Tag](#) &tag) const
- const [ModuleEntry](#) & [GetModuleEntryInMacros](#) ([Macros](#) const ¯os, 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)

27.187.1 Detailed Description

Class for representing a [Module](#).

Note

[Module](#): A set of Attributes within an Information Entity or Normalized [IOD](#) which are logically related to each other.

See also

[Macro](#)

Examples:

[TraverseModules.cxx](#).

27.187.2 Member Typedef Documentation

27.187.2.1 typedef std::vector<std::string> gdcmmodule::ArrayIncludeMacrosType

27.187.2.2 typedef std::map<Tag, ModuleEntry> gdcmmodule::MapModuleEntry

27.187.3 Constructor & Destructor Documentation

27.187.3.1 `gdcmmodule::Module () [inline]`

27.187.4 Member Function Documentation

27.187.4.1 `void gdcmmodule::AddMacro (const char * include) [inline]`

27.187.4.2 `void gdcmmodule::AddModuleEntry (const Tag & tag, const ModuleEntry & module) [inline]`

Will add a [ModuleEntry](#) directly at root-level. See [Macro](#) for nested-included level.

27.187.4.3 `void gdcmmodule::Clear () [inline]`

27.187.4.4 `bool gdcmmodule::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.

Examples:

[TraverseModules.cxx](#).

27.187.4.5 `const ModuleEntry& gdcmmodule::GetModuleEntryInMacros (Macros const & macros, const Tag & tag) const`

Examples:

[TraverseModules.cxx](#).

27.187.4.6 `const char* gdcmmodule::GetName () const [inline]`

27.187.4.7 `void gdcmmodule::SetName (const char * name) [inline]`

27.187.4.8 `bool gdcmmodule::Verify (const DataSet & ds, Usage const & usage) const`

27.187.5 Friends And Related Function Documentation

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

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

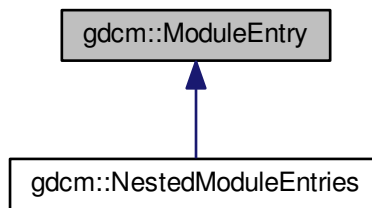
- [gdcmmodule.h](#)

27.188 gdcmmodule::ModuleEntry Class Reference

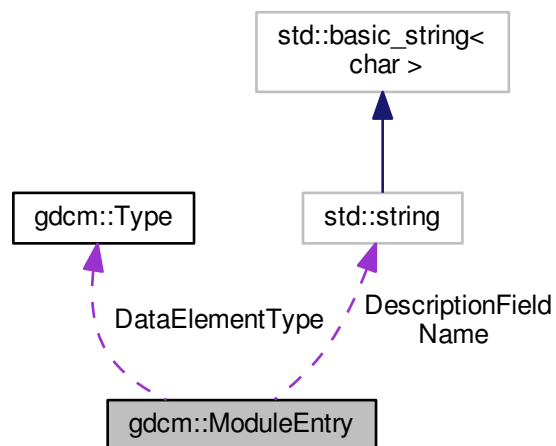
Class for representing a [ModuleEntry](#).

```
#include <gdcmmoduleEntry.h>
```

Inheritance diagram for gdcm::ModuleEntry:



Collaboration diagram for gdcm::ModuleEntry:



Public Types

- typedef std::string [Description](#)

Public Member Functions

- [ModuleEntry](#) (const char *name="", const char *type="3", const char *description="")
- virtual [~ModuleEntry](#) ()
- 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)

27.188.1 Detailed Description

Class for representing a [ModuleEntry](#).

Note

bla

See also

[DictEntry](#)

Examples:

[TraverseModules.cxx](#).

27.188.2 Member Typedef Documentation

27.188.2.1 `typedef std::string gdcmm::ModuleEntry::Description`

27.188.3 Constructor & Destructor Documentation

27.188.3.1 `gdcmm::ModuleEntry::ModuleEntry (const char * name = " ", const char * type = "3", const char * description = " ")`
`[inline]`

References `gdcmm::Type::GetTypeType()`.

27.188.3.2 `virtual gdcmm::ModuleEntry::~~ModuleEntry ()` `[inline]`, `[virtual]`

27.188.4 Member Function Documentation

27.188.4.1 `const Description& gdcmm::ModuleEntry::GetDescription () const` `[inline]`

27.188.4.2 `const char* gdcmm::ModuleEntry::GetName () const` `[inline]`

27.188.4.3 `const Type& gdcm::ModuleEntry::GetType () const` `[inline]`

Examples:

[TraverseModules.cxx](#).

27.188.4.4 `void gdcm::ModuleEntry::SetDescription (const char * d)` `[inline]`

27.188.4.5 `void gdcm::ModuleEntry::SetName (const char * name)` `[inline]`

27.188.4.6 `void gdcm::ModuleEntry::SetType (const Type & type)` `[inline]`

27.188.5 Friends And Related Function Documentation

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

27.188.6 Member Data Documentation

27.188.6.1 `Type gdcm::ModuleEntry::DataElementType` `[protected]`

Referenced by `gdcm::operator<<()`.

27.188.6.2 `Description gdcm::ModuleEntry::DescriptionField` `[protected]`

Referenced by `gdcm::operator<<()`.

27.188.6.3 `std::string gdcm::ModuleEntry::Name` `[protected]`

Referenced by `gdcm::operator<<()`.

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

- [gdcmModuleEntry.h](#)

27.189 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](#) ()
- 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)

27.189.1 Detailed Description

Class for representing a [Modules](#).

Note

bla

See also

[Module](#)

Examples:

[TraverseModules.cxx](#).

27.189.2 Member Typedef Documentation

27.189.2.1 `typedef std::map<std::string, Module> gdcm::Modules::ModuleMapType`

27.189.3 Constructor & Destructor Documentation

27.189.3.1 `gdcm::Modules::Modules ()` `[inline]`

27.189.4 Member Function Documentation

27.189.4.1 `void gdcm::Modules::AddModule (const char * ref, const Module & module)` `[inline]`

27.189.4.2 `void gdcm::Modules::Clear ()` `[inline]`

27.189.4.3 `const Module& gdcm::Modules::GetModule (const char * name) const` `[inline]`

Examples:

[TraverseModules.cxx](#).

27.189.4.4 `bool gdcm::Modules::IsEmpty () const` `[inline]`

27.189.5 Friends And Related Function Documentation

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

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

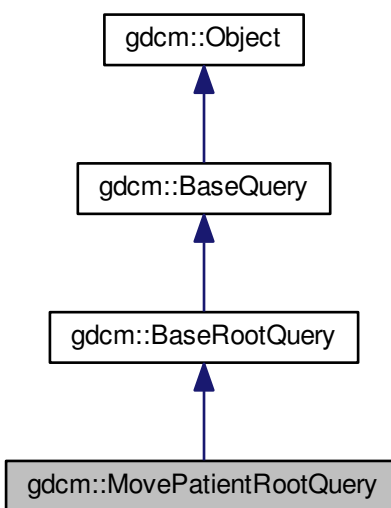
- [gdcmModules.h](#)

27.190 gdcm::MovePatientRootQuery Class Reference

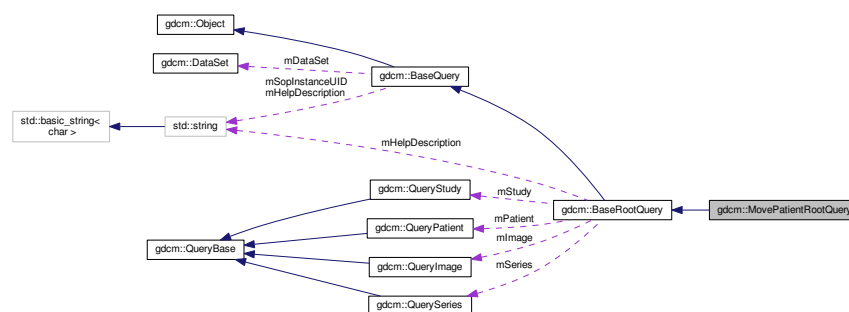
[MovePatientRootQuery](#) contains: the class which will produce a dataset for c-move with patient root.

```
#include <gdcmMovePatientRootQuery.h>
```

Inheritance diagram for gdcm::MovePatientRootQuery:



Collaboration diagram for gdcm::MovePatientRootQuery:



Public Member Functions

- [MovePatientRootQuery](#) ()
- [UIDs::TSName GetAbstractSyntaxUID](#) () const
- `std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)`
- `void InitializeDataSet (const EQueryLevel &inQueryLevel)`
- `bool ValidateQuery (bool inStrict=true) const`

Friends

- class [QueryFactory](#)

Additional Inherited Members

27.190.1 Detailed Description

[MovePatientRootQuery](#) contains: the class which will produce a dataset for c-move with patient root.

27.190.2 Constructor & Destructor Documentation

27.190.2.1 `gdcm::MovePatientRootQuery::MovePatientRootQuery ()`

27.190.3 Member Function Documentation

27.190.3.1 `UIDs::TSName gdcm::MovePatientRootQuery::GetAbstractSyntaxUID () const` [virtual]

Implements [gdcm::BaseQuery](#).

27.190.3.2 `std::vector<Tag> gdcm::MovePatientRootQuery::GetTagListByLevel (const EQueryLevel & inQueryLevel)`
[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](#).

27.190.3.3 `void gdcm::MovePatientRootQuery::InitializeDataSet (const EQueryLevel & inQueryLevel)` [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](#).

27.190.3.4 `bool gdcm::MovePatientRootQuery::ValidateQuery (bool inStrict =true) const` [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](#).

27.190.4 Friends And Related Function Documentation

27.190.4.1 friend class QueryFactory [friend]

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

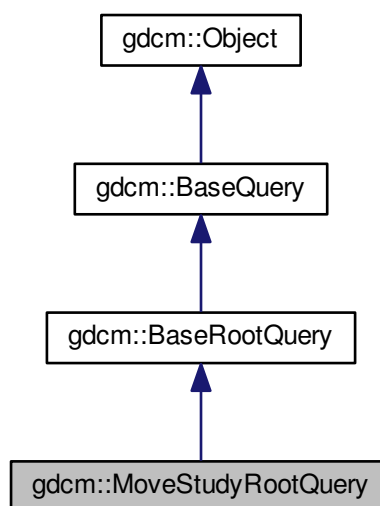
- [gdcmMovePatientRootQuery.h](#)

27.191 gdcm::MoveStudyRootQuery Class Reference

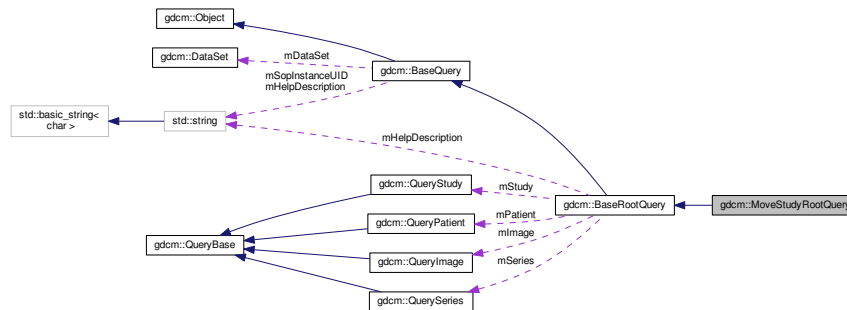
[MoveStudyRootQuery](#) contains: the class which will produce a dataset for C-MOVE with study root.

```
#include <gdcmMoveStudyRootQuery.h>
```

Inheritance diagram for gdcm::MoveStudyRootQuery:



Collaboration diagram for `gdcm::MoveStudyRootQuery`:



Public Member Functions

- [MoveStudyRootQuery](#) ()
- [UIDs::TSName GetAbstractSyntaxUID](#) () const
- `std::vector< Tag > GetTagListByLevel` (const [EQueryLevel](#) &inQueryLevel)
- void [InitializeDataSet](#) (const [EQueryLevel](#) &inQueryLevel)
- bool [ValidateQuery](#) (bool inStrict=true) const

Friends

- class [QueryFactory](#)

Additional Inherited Members

27.191.1 Detailed Description

[MoveStudyRootQuery](#) contains: the class which will produce a dataset for C-MOVE with study root.

27.191.2 Constructor & Destructor Documentation

27.191.2.1 `gdcm::MoveStudyRootQuery::MoveStudyRootQuery ()`

27.191.3 Member Function Documentation

27.191.3.1 `UIDs::TSName gdcm::MoveStudyRootQuery::GetAbstractSyntaxUID () const` `[virtual]`

Implements [gdcm::BaseQuery](#).

27.191.3.2 `std::vector<Tag> gdcm::MoveStudyRootQuery::GetTagListByLevel (const EQueryLevel & inQueryLevel)` `[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](#).

27.191.3.3 void gdcm::MoveStudyRootQuery::InitializeDataSet (const EQueryLevel & *inQueryLevel*) [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](#).

27.191.3.4 bool gdcm::MoveStudyRootQuery::ValidateQuery (bool *inStrict* = true) const [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](#).

27.191.4 Friends And Related Function Documentation

27.191.4.1 friend class QueryFactory [friend]

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

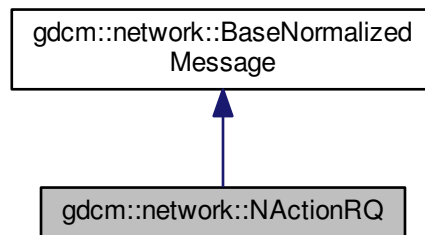
- [gdcmMoveStudyRootQuery.h](#)

27.192 gdcm::network::NActionRQ Class Reference

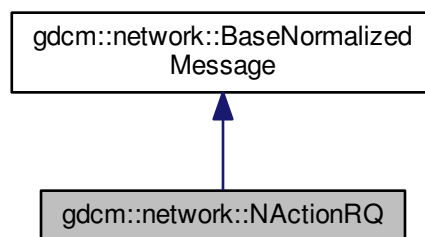
[NActionRQ](#) this file defines the messages for the NAction action.

```
#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)

27.192.1 Detailed Description

[NActionRQ](#) this file defines the messages for the NAction action.

27.192.2 Member Function Documentation

27.192.2.1 `std::vector<PresentationDataValue> gdcm::network::NActionRQ::ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery)` [virtual]

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

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

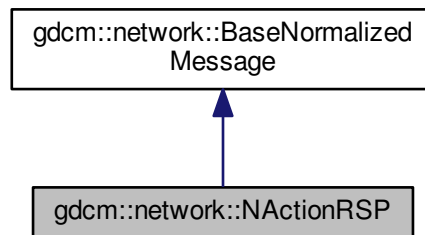
- [gdcmNActionMessages.h](#)

27.193 gdcm::network::NActionRSP Class Reference

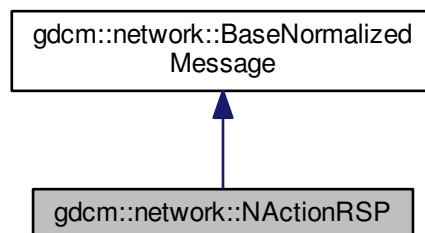
[NActionRSP](#) this file defines the messages for the NAction action.

```
#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)

27.193.1 Detailed Description

[NActionRSP](#) this file defines the messages for the NAction action.

27.193.2 Member Function Documentation

27.193.2.1 `std::vector<PresentationDataValue> gdcmm::network::NActionRSP::ConstructPDVByDataSet (const DataSet * inDataSet)`

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

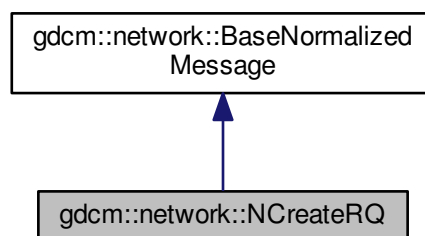
- [gdcmmNActionMessages.h](#)

27.194 gdcmm::network::NCreateRQ Class Reference

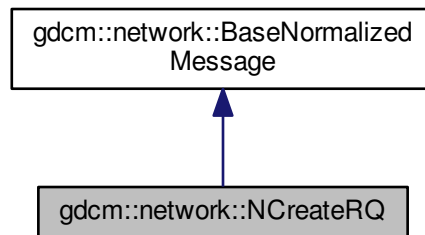
[NCreateRQ](#) this file defines the messages for the ncreate action.

```
#include <gdcmmNCreateMessages.h>
```

Inheritance diagram for gdcmm::network::NCreateRQ:



Collaboration diagram for gdcm::network::NCreateRQ:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (`const ULConnection &inConnection`, `const BaseQuery *inQuery`)

27.194.1 Detailed Description

[NCreateRQ](#) this file defines the messages for the ncreate action.

27.194.2 Member Function Documentation

27.194.2.1 `std::vector<PresentationDataValue> gdcm::network::NCreateRQ::ConstructPDV (const ULConnection & inConnection, const BaseQuery * inQuery)` [virtual]

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

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

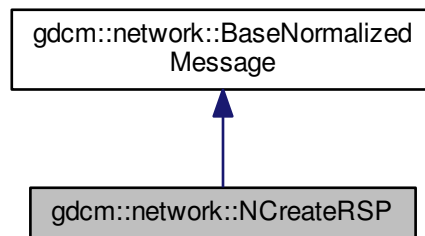
- [gdcmNCreateMessages.h](#)

27.195 gdcm::network::NCreateRSP Class Reference

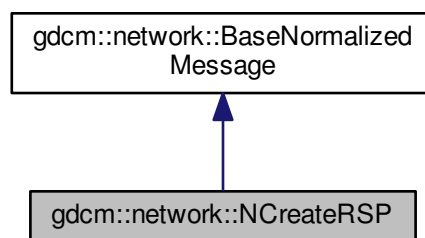
[NCreateRSP](#) this file defines the messages for the ncreate action.

```
#include <gdcmNCreateMessages.h>
```

Inheritance diagram for `gdcm::network::NCreateRSP`:



Collaboration diagram for `gdcm::network::NCreateRSP`:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)`

27.195.1 Detailed Description

[NCreateRSP](#) this file defines the messages for the ncreate action.

27.195.2 Member Function Documentation

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

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

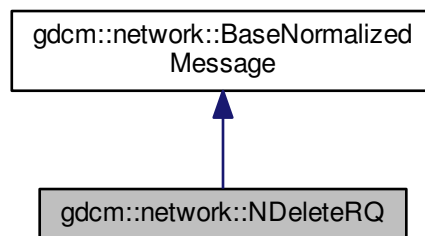
- [gdcmNCreateMessages.h](#)

27.196 gdcm::network::NDeleteRQ Class Reference

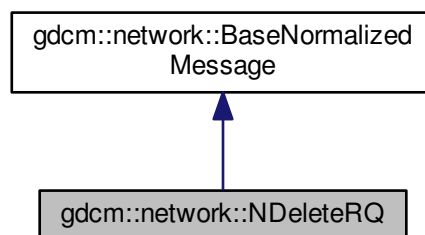
[NDeleteRQ](#) this file defines the messages for the ndelete action.

```
#include <gdcmNDeleteMessages.h>
```

Inheritance diagram for gdcm::network::NDeleteRQ:



Collaboration diagram for gdcm::network::NDeleteRQ:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseQuery](#) *inQuery)

27.196.1 Detailed Description

[NDeleteRQ](#) this file defines the messages for the ndelete action.

27.196.2 Member Function Documentation

27.196.2.1 `std::vector<PresentationDataValue> gdcM::network::NDeleteRQ::ConstructPDV (const ULConnection & inConnection, const BaseQuery * inQuery) [virtual]`

Implements [gdcM::network::BaseNormalizedMessage](#).

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

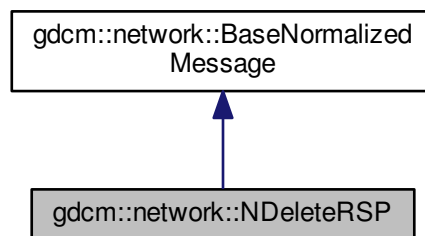
- [gdcMNDeleteMessages.h](#)

27.197 gdcM::network::NDeleteRSP Class Reference

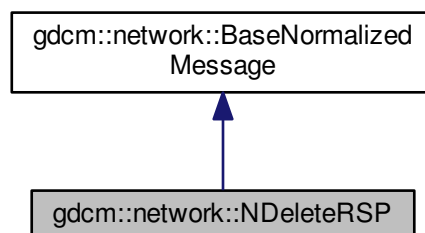
[NDeleteRSP](#) this file defines the messages for the ndelete action.

```
#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)

27.197.1 Detailed Description

[NDeleteRSP](#) this file defines the messages for the ndelete action.

27.197.2 Member Function Documentation

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

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

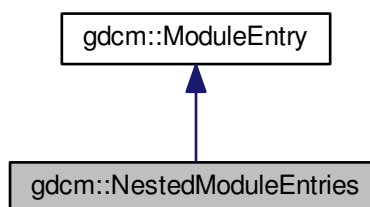
- [gdcmNDeleteMessages.h](#)

27.198 gdcm::NestedModuleEntries Class Reference

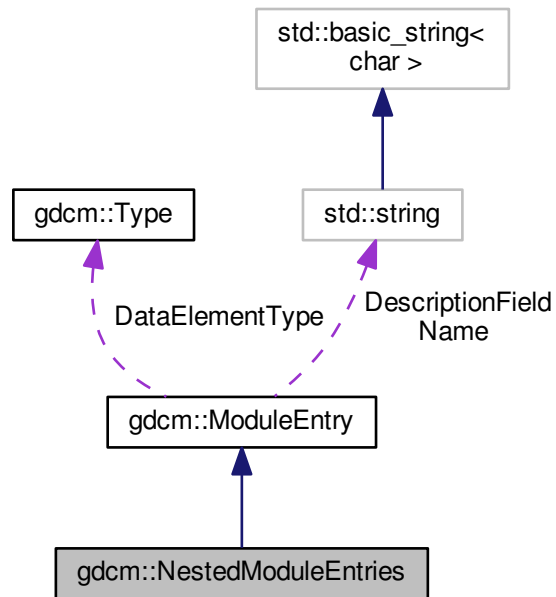
Class for representing a [NestedModuleEntries](#).

```
#include <gdcmNestedModuleEntries.h>
```

Inheritance diagram for `gdcm::NestedModuleEntries`:



Collaboration diagram for `gdcmm::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)
- const [ModuleEntry](#) & [GetModuleEntry](#) ([SizeType](#) idx) const
- [ModuleEntry](#) & [GetModuleEntry](#) ([SizeType](#) idx)
- [SizeType](#) [GetNumberOfModuleEntries](#) ()

Friends

- `std::ostream & operator<< (std::ostream &_os, const NestedModuleEntries &_val)`

Additional Inherited Members

27.198.1 Detailed Description

Class for representing a [NestedModuleEntries](#).

Note

bla

See also

[ModuleEntry](#)

27.198.2 Member Typedef Documentation

27.198.2.1 `typedef std::vector<ModuleEntry>::size_type gdcm::NestedModuleEntries::SizeType`

27.198.3 Constructor & Destructor Documentation

27.198.3.1 `gdcm::NestedModuleEntries::NestedModuleEntries (const char * name = " ", const char * type = "3", const char * description = " ")` `[inline]`

27.198.4 Member Function Documentation

27.198.4.1 `void gdcm::NestedModuleEntries::AddModuleEntry (const ModuleEntry & me)` `[inline]`

27.198.4.2 `const ModuleEntry& gdcm::NestedModuleEntries::GetModuleEntry (SizeType idx) const` `[inline]`

27.198.4.3 `ModuleEntry& gdcm::NestedModuleEntries::GetModuleEntry (SizeType idx)` `[inline]`

27.198.4.4 `SizeType gdcm::NestedModuleEntries::GetNumberOfModuleEntries ()` `[inline]`

27.198.5 Friends And Related Function Documentation

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

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

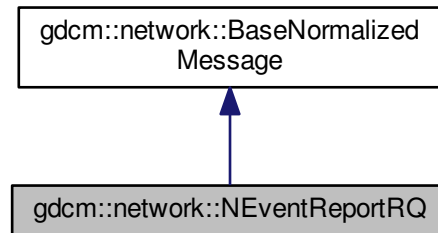
- [gdcmNestedModuleEntries.h](#)

27.199 gdcm::network::NEventReportRQ Class Reference

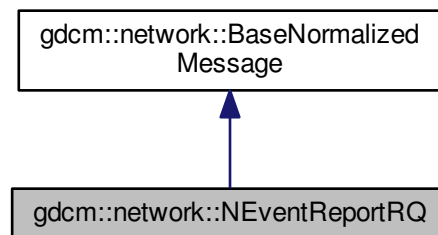
[NEventReportRQ](#) this file defines the messages for the neventreport action.

```
#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`)

27.199.1 Detailed Description

[NEventReportRQ](#) this file defines the messages for the neventreport action.

27.199.2 Member Function Documentation

27.199.2.1 `std::vector<PresentationDataValue> gdcm::network::NEventReportRQ::ConstructPDV (const ULConnection & inConnection, const BaseQuery * inQuery)` `[virtual]`

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

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

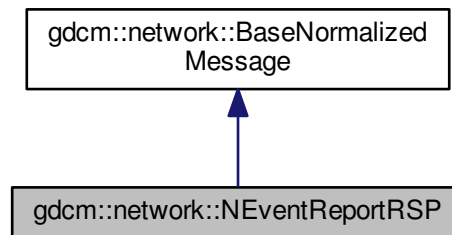
- [gdcmNEventReportMessages.h](#)

27.200 gdcm::network::NEventReportRSP Class Reference

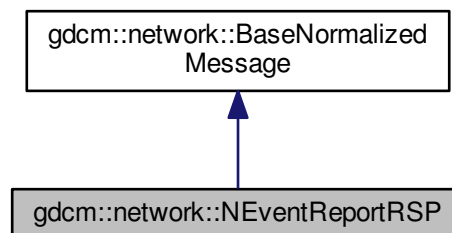
[NEventReportRSP](#) this file defines the messages for the neventreport action.

```
#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)

27.200.1 Detailed Description

[NEventReportRSP](#) this file defines the messages for the neventreport action.

27.200.2 Member Function Documentation

27.200.2.1 `std::vector<PresentationDataValue> gdcmm::network::NEventReportRSP::ConstructPDVByDataSet (const DataSet * inDataSet)`

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

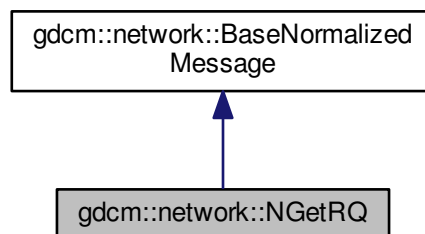
- [gdcmmNEventReportMessages.h](#)

27.201 gdcmm::network::NGetRQ Class Reference

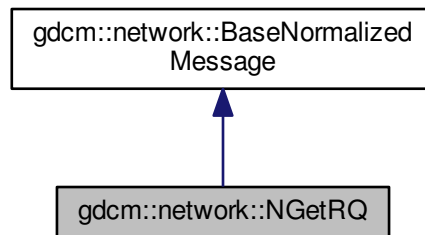
[NGetRQ](#) this file defines the messages for the nget action.

```
#include <gdcmmNGetMessages.h>
```

Inheritance diagram for gdcmm::network::NGetRQ:



Collaboration diagram for gdcm::network::NGetRQ:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (`const ULConnection &inConnection`, `const BaseQuery *inQuery`)

27.201.1 Detailed Description

[NGetRQ](#) this file defines the messages for the nget action.

27.201.2 Member Function Documentation

27.201.2.1 `std::vector<PresentationDataValue> gdcm::network::NGetRQ::ConstructPDV (const ULConnection & inConnection, const BaseQuery * inQuery)` [virtual]

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

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

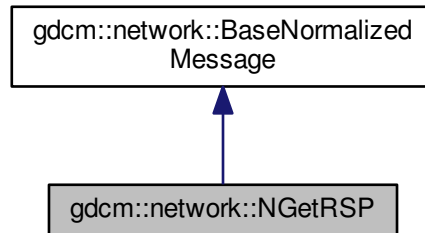
- [gdcmNGetMessages.h](#)

27.202 gdcm::network::NGetRSP Class Reference

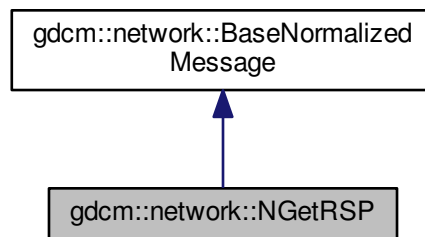
[NGetRSP](#) this file defines the messages for the nget action.

```
#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)`

27.202.1 Detailed Description

[NGetRSP](#) this file defines the messages for the nget action.

27.202.2 Member Function Documentation

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

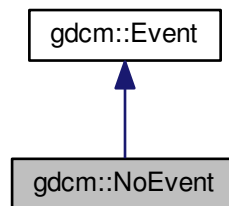
The documentation for this class was generated from the following file:

- [gdcmNGetMessages.h](#)

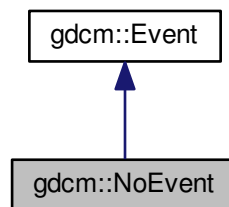
27.203 gdcm::NoEvent Class Reference

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

Inheritance diagram for gdcm::NoEvent:



Collaboration diagram for gdcm::NoEvent:



Additional Inherited Members

27.203.1 Detailed Description

Define some common GDCM events

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

- [gdcmEvent.h](#)

27.204 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)

27.204.1 Member Function Documentation

- 27.204.1.1 static std::vector<[PresentationDataValue](#)> [gdcm::network::NormalizedMessageFactory::ConstructNAction](#) (const [ULConnection](#) & *inConnection*, const [BaseQuery](#) * *inQuery*) [static]
- 27.204.1.2 static std::vector<[PresentationDataValue](#)> [gdcm::network::NormalizedMessageFactory::ConstructNCreate](#) (const [ULConnection](#) & *inConnection*, const [BaseQuery](#) * *inQuery*) [static]
- 27.204.1.3 static std::vector<[PresentationDataValue](#)> [gdcm::network::NormalizedMessageFactory::ConstructNDelete](#) (const [ULConnection](#) & *inConnection*, const [BaseQuery](#) * *inQuery*) [static]
- 27.204.1.4 static std::vector<[PresentationDataValue](#)> [gdcm::network::NormalizedMessageFactory::ConstructNEventReport](#) (const [ULConnection](#) & *inConnection*, const [BaseQuery](#) * *inQuery*) [static]
- 27.204.1.5 static std::vector<[PresentationDataValue](#)> [gdcm::network::NormalizedMessageFactory::ConstructNGet](#) (const [ULConnection](#) & *inConnection*, const [BaseQuery](#) * *inQuery*) [static]
- 27.204.1.6 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](#)

27.205 gdcm::NormalizedNetworkFunctions Class Reference

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:

```
#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)

27.205.1 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

27.205.2 Member Function Documentation

- 27.205.2.1 static [BaseQuery](#)* [gdcmm::NormalizedNetworkFunctions::ConstructQuery](#) (const std::string & *sopInstanceUID*, const [DataSet](#) & *queryds*, [ENQueryType](#) *queryType* = [eCreateMMPS](#)) [static]
- 27.205.2.2 static bool [gdcmm::NormalizedNetworkFunctions::NAction](#) (const char * *remote*, uint16_t *portno*, const [BaseQuery](#) * *query*, std::vector< [DataSet](#) > & *retDataSets*, const char * *aetitle*, const char * *call*) [static]
- 27.205.2.3 static bool [gdcmm::NormalizedNetworkFunctions::NCreate](#) (const char * *remote*, uint16_t *portno*, [BaseQuery](#) * *query*, std::vector< [DataSet](#) > & *retDataSets*, const char * *aetitle*, const char * *call*) [static]
- 27.205.2.4 static bool [gdcmm::NormalizedNetworkFunctions::NDelete](#) (const char * *remote*, uint16_t *portno*, const [BaseQuery](#) * *query*, std::vector< [DataSet](#) > & *retDataSets*, const char * *aetitle*, const char * *call*) [static]

27.205.2.5 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]

27.205.2.6 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]

27.205.2.7 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:

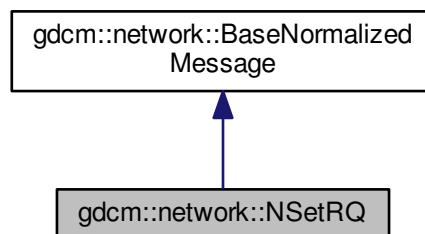
- [gdcMNormalizedNetworkFunctions.h](#)

27.206 gdcM::network::NSetRQ Class Reference

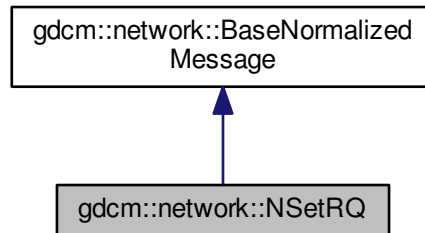
[NSetRQ](#) this file defines the messages for the nset action.

```
#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)`

27.206.1 Detailed Description

[NSetRQ](#) this file defines the messages for the nset action.

27.206.2 Member Function Documentation

27.206.2.1 `std::vector<PresentationDataValue> gdcm::network::NSetRQ::ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery)` [virtual]

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

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

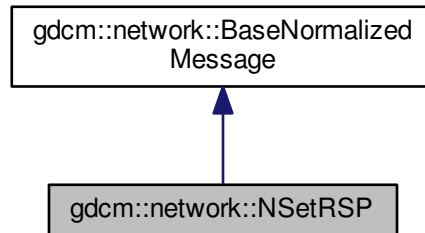
- [gdcmNSetMessages.h](#)

27.207 gdcm::network::NSetRSP Class Reference

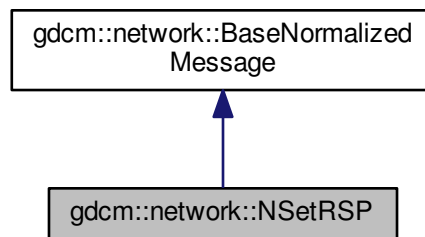
[NSetRSP](#) this file defines the messages for the nset action.

```
#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)`

27.207.1 Detailed Description

[NSetRSP](#) this file defines the messages for the nset action.

27.207.2 Member Function Documentation

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

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

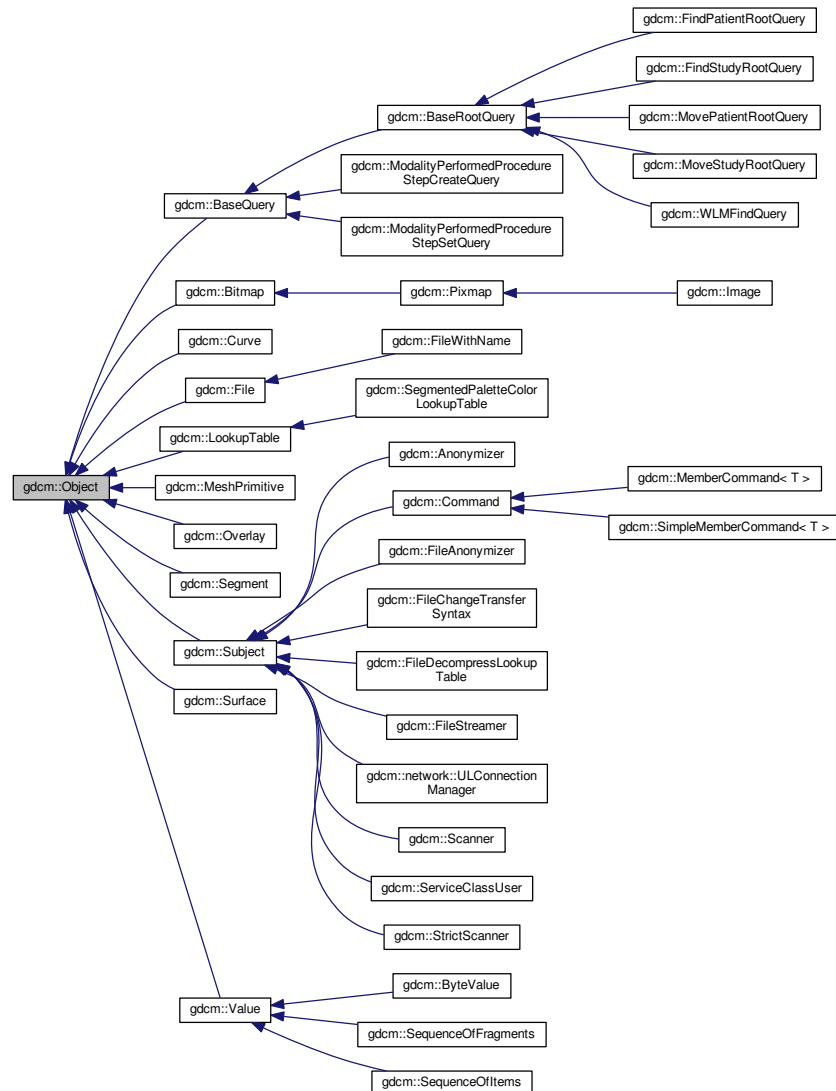
- [gdcmNSetMessages.h](#)

27.208 gdcm::Object Class Reference

Object.

```
#include <gdcmObject.h>
```

Inheritance diagram for gdcm::Object:



Public Member Functions

- [Object](#) ()
 - [Object](#) (const [Object](#) &)
 - virtual [~Object](#) ()
- Special requirement for copy/cstor, assignment operator.*

- 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](#)

27.208.1 Detailed Description

[Object](#).

Note

main superclass for object that want to use [SmartPointer](#) invasive ref counting system

See also

[SmartPointer](#)

27.208.2 Constructor & Destructor Documentation

27.208.2.1 [gdcmm::Object::Object \(\)](#) [\[inline\]](#)

27.208.2.2 [virtual gdcmm::Object::~~Object \(\)](#) [\[inline\]](#), [\[virtual\]](#)

27.208.2.3 [gdcmm::Object::Object \(const \[Object\]\(#\) & \)](#) [\[inline\]](#)

Special requirement for copy/cstor, assignment operator.

27.208.3 Member Function Documentation

27.208.3.1 [void gdcmm::Object::operator= \(const \[Object\]\(#\) & \)](#) [\[inline\]](#)

27.208.3.2 [virtual void gdcmm::Object::Print \(std::ostream & \) const](#) [\[inline\]](#), [\[virtual\]](#)

Reimplemented in [gdcmm::SequenceOfFragments](#), [gdcmm::SequenceOfItems](#), [gdcmm::ByteValue](#), [gdcmm::Scanner](#), [gdcmm::StrictScanner](#), [gdcmm::Image](#), [gdcmm::BaseQuery](#), [gdcmm::Curve](#), [gdcmm::Overlay](#), [gdcmm::Bitmap](#), [gdcmm::LookupTable](#), [gdcmm::Pixmap](#), and [gdcmm::SegmentedPaletteColorLookupTable](#).

Examples:

[ReadAndDumpDICOMDIR.cxx](#).

Referenced by [gdcmm::operator<<\(\)](#).

27.208.3.3 void gdcm::Object::Register () [inline],[protected]

27.208.3.4 void gdcm::Object::UnRegister () [inline],[protected]

27.208.4 Friends And Related Function Documentation

27.208.4.1 std::ostream& operator<< (std::ostream & os, const Object & obj) [friend]

27.208.4.2 template<class ObjectType > friend class SmartPointer [friend]

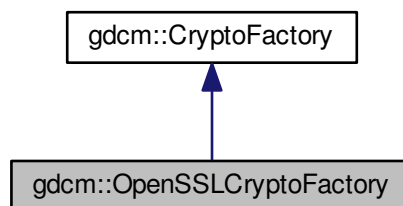
The documentation for this class was generated from the following file:

- [gdcmObject.h](#)

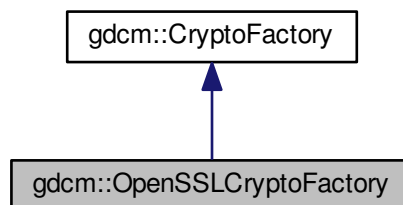
27.209 gdcm::OpenSSLCryptoFactory Class Reference

```
#include <gdcmOpenSSLCryptoFactory.h>
```

Inheritance diagram for gdcm::OpenSSLCryptoFactory:



Collaboration diagram for gdcm::OpenSSLCryptoFactory:



Public Member Functions

- [OpenSSLCryptoFactory](#) ([CryptoLib](#) id)
- [CryptographicMessageSyntax](#) * [CreateCMSProvider](#) ()

Protected Member Functions

- void [InitOpenSSL](#) ()

Additional Inherited Members

27.209.1 Constructor & Destructor Documentation

27.209.1.1 `gdcm::OpenSSLCryptoFactory::OpenSSLCryptoFactory (CryptoLib id) [inline]`

References [gdcmDebugMacro](#).

27.209.2 Member Function Documentation

27.209.2.1 `CryptographicMessageSyntax* gdcm::OpenSSLCryptoFactory::CreateCMSProvider () [inline], [virtual]`

Implements [gdcm::CryptoFactory](#).

27.209.2.2 `void gdcm::OpenSSLCryptoFactory::InitOpenSSL () [protected]`

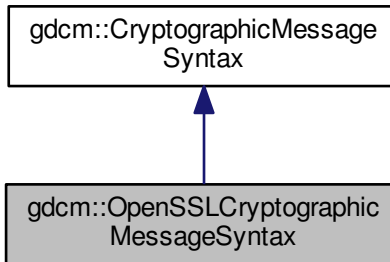
The documentation for this class was generated from the following file:

- [gdcmOpenSSLCryptoFactory.h](#)

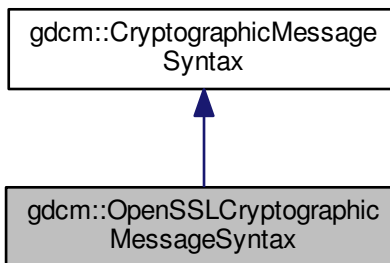
27.210 gdcm::OpenSSLCryptographicMessageSyntax Class Reference

```
#include <gdcmOpenSSLCryptographicMessageSyntax.h>
```

Inheritance diagram for gdcmm::OpenSSLCryptographicMessageSyntax:



Collaboration diagram for gdcmm::OpenSSLCryptographicMessageSyntax:



Public Member Functions

- [OpenSSLCryptographicMessageSyntax](#) ()
- [~OpenSSLCryptographicMessageSyntax](#) ()
- bool [Decrypt](#) (char *output, size_t &outlen, const char *array, size_t len) const
decrypt content from a PKCS#7 envelopedData structure
- bool [Encrypt](#) (char *output, size_t &outlen, const char *array, size_t len) const
create a CMS envelopedData structure
- [CipherTypes](#) [GetCipherType](#) () const
- bool [ParseCertificateFile](#) (const char *filename)
- bool [ParseKeyFile](#) (const char *filename)
- void [SetCipherType](#) ([CipherTypes](#) type)
- bool [SetPassword](#) (const char *pass, size_t passLen)

Additional Inherited Members

27.210.1 Constructor & Destructor Documentation

27.210.1.1 `gdcmm::OpenSSLCryptographicMessageSyntax::OpenSSLCryptographicMessageSyntax ()`

27.210.1.2 `gdcmm::OpenSSLCryptographicMessageSyntax::~~OpenSSLCryptographicMessageSyntax ()`

27.210.2 Member Function Documentation

27.210.2.1 `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](#).

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

create a CMS envelopedData structure

Implements [gdcmm::CryptographicMessageSyntax](#).

27.210.2.3 `CipherTypes gdcmm::OpenSSLCryptographicMessageSyntax::GetCipherType () const` `[virtual]`

Implements [gdcmm::CryptographicMessageSyntax](#).

27.210.2.4 `bool gdcmm::OpenSSLCryptographicMessageSyntax::ParseCertificateFile (const char * filename)` `[virtual]`

Implements [gdcmm::CryptographicMessageSyntax](#).

27.210.2.5 `bool gdcmm::OpenSSLCryptographicMessageSyntax::ParseKeyFile (const char * filename)` `[virtual]`

Implements [gdcmm::CryptographicMessageSyntax](#).

27.210.2.6 `void gdcmm::OpenSSLCryptographicMessageSyntax::SetCipherType (CipherTypes type)` `[virtual]`

Set Cipher [Type](#). Default is: AES256_CIPHER

Implements [gdcmm::CryptographicMessageSyntax](#).

27.210.2.7 `bool gdcmm::OpenSSLCryptographicMessageSyntax::SetPassword (const char * pass, size_t passLen)` `[virtual]`

Implements [gdcmm::CryptographicMessageSyntax](#).

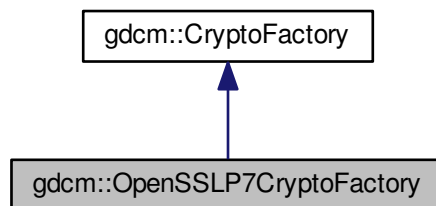
The documentation for this class was generated from the following file:

- [gdcmmOpenSSLCryptographicMessageSyntax.h](#)

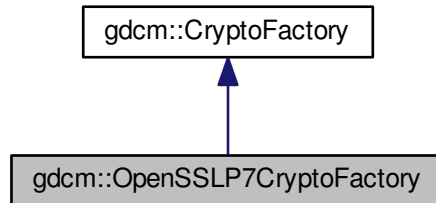
27.211 gdcM::OpenSSL7CryptoFactory Class Reference

```
#include <gdcMOpenSSL7CryptoFactory.h>
```

Inheritance diagram for gdcM::OpenSSL7CryptoFactory:



Collaboration diagram for gdcM::OpenSSL7CryptoFactory:



Public Member Functions

- [OpenSSL7CryptoFactory](#) ([CryptoLib](#) id)
- [CryptographicMessageSyntax](#) * [CreateCMSProvider](#) ()

Additional Inherited Members

27.211.1 Constructor & Destructor Documentation

27.211.1.1 `gdcM::OpenSSL7CryptoFactory::OpenSSL7CryptoFactory (CryptoLib id)` [`inline`]

References `gdcMDebugMacro`.

27.211.2 Member Function Documentation

27.211.2.1 **CryptographicMessageSyntax*** `gdcM::OpenSSLP7CryptoFactory::CreateCMSProvider ()` `[inline]`,
`[virtual]`

Implements [gdcM::CryptoFactory](#).

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

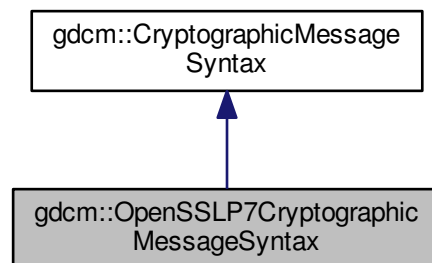
- [gdcMOpenSSLP7CryptoFactory.h](#)

27.212 gdcM::OpenSSLP7CryptographicMessageSyntax Class Reference

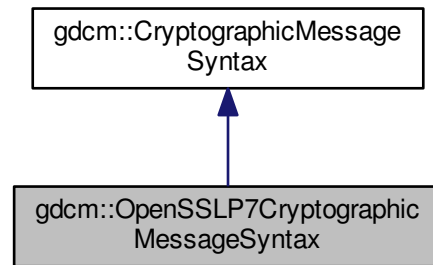
Class for [CryptographicMessageSyntax](#) encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities.

```
#include <gdcMOpenSSLP7CryptographicMessageSyntax.h>
```

Inheritance diagram for `gdcM::OpenSSLP7CryptographicMessageSyntax`:



Collaboration diagram for gdcM::OpenSSLP7CryptographicMessageSyntax:



Public Member Functions

- [OpenSSLP7CryptographicMessageSyntax](#) ()
- [~OpenSSLP7CryptographicMessageSyntax](#) ()
- 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

27.212.1 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

27.212.2 Constructor & Destructor Documentation

27.212.2.1 `gdcM::OpenSSLP7CryptographicMessageSyntax::OpenSSLP7CryptographicMessageSyntax ()`

27.212.2.2 `gdcM::OpenSSLP7CryptographicMessageSyntax::~~OpenSSLP7CryptographicMessageSyntax ()`

27.212.3 Member Function Documentation

27.212.3.1 `bool gdcM::OpenSSL7CryptographicMessageSyntax::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](#).

27.212.3.2 `bool gdcM::OpenSSL7CryptographicMessageSyntax::Encrypt (char * output, size_t & outlen, const char * array, size_t len) const` [virtual]

create a PKCS#7 envelopedData structure

Implements [gdcM::CryptographicMessageSyntax](#).

27.212.3.3 `CipherTypes gdcM::OpenSSL7CryptographicMessageSyntax::GetCipherType () const` [virtual]

Implements [gdcM::CryptographicMessageSyntax](#).

27.212.3.4 `bool gdcM::OpenSSL7CryptographicMessageSyntax::ParseCertificateFile (const char * filename)` [virtual]

Implements [gdcM::CryptographicMessageSyntax](#).

27.212.3.5 `bool gdcM::OpenSSL7CryptographicMessageSyntax::ParseKeyFile (const char * filename)` [virtual]

Implements [gdcM::CryptographicMessageSyntax](#).

27.212.3.6 `void gdcM::OpenSSL7CryptographicMessageSyntax::SetCipherType (CipherTypes type)` [virtual]

Set Cipher [Type](#). Default is: AES256_CIPHER

Implements [gdcM::CryptographicMessageSyntax](#).

27.212.3.7 `bool gdcM::OpenSSL7CryptographicMessageSyntax::SetPassword (const char * , size_t)` [inline], [virtual]

Implements [gdcM::CryptographicMessageSyntax](#).

References `gdcMWarningMacro`.

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

- [gdcMOpenSSL7CryptographicMessageSyntax.h](#)

27.213 gdcM::Orientation Class Reference

class to handle [Orientation](#)

```
#include <gdcMOrientation.h>
```

Public Types

- enum [OrientationType](#) {
 [UNKNOWN](#),
 [AXIAL](#),
 [CORONAL](#),
 [SAGITTAL](#),
 [OBLIQUE](#) }

Public Member Functions

- [Orientation](#) ()
- [~Orientation](#) ()
- 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)

27.213.1 Detailed Description

class to handle [Orientation](#)

27.213.2 Member Enumeration Documentation

27.213.2.1 enum gdcm::Orientation::OrientationType

Enumerator

UNKNOWN
AXIAL
CORONAL
SAGITTAL
OBLIQUE

27.213.3 Constructor & Destructor Documentation

27.213.3.1 `gdcmm::Orientation::Orientation ()`

27.213.3.2 `gdcmm::Orientation::~~Orientation ()`

27.213.4 Member Function Documentation

27.213.4.1 `static const char* gdcmm::Orientation::GetLabel (OrientationType type) [static]`

Return the label of an [Orientation](#).

Examples:

[FixOrientation.cxx](#).

27.213.4.2 `static char gdcmm::Orientation::GetMajorAxisFromPatientRelativeDirectionCosine (double x, double y, double z) [static], [protected]`

27.213.4.3 `static double gdcmm::Orientation::GetObliquityThresholdCosineValue () [static]`

27.213.4.4 `static OrientationType gdcmm::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](#).

27.213.4.5 `void gdcmm::Orientation::Print (std::ostream &) const`

Print.

Referenced by `gdcmm::operator<<()`.

27.213.4.6 `static void gdcmm::Orientation::SetObliquityThresholdCosineValue (double val) [static]`

ObliquityThresholdCosineValue stuff.

27.213.5 Friends And Related Function Documentation

27.213.5.1 `std::ostream& operator<< (std::ostream &_os, const Orientation &o) [friend]`

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

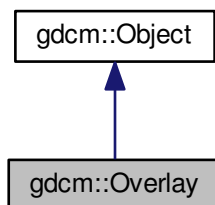
- [gdcmmOrientation.h](#)

27.214 gdcm::Overlay Class Reference

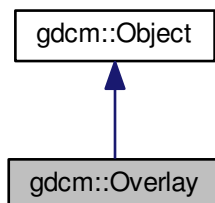
[Overlay](#) class.

```
#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](#) ()
- 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
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

27.214.1 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:

27.214.2 Member Enumeration Documentation

27.214.2.1 enum gdcm::Overlay::OverlayType

Enumerator

Invalid

Graphics

ROI

27.214.3 Constructor & Destructor Documentation

27.214.3.1 [gdcm::Overlay::Overlay](#) ()

27.214.3.2 [gdcm::Overlay::~~Overlay](#) ()

27.214.3.3 [gdcm::Overlay::Overlay](#) ([Overlay](#) const & ov)

27.214.4 Member Function Documentation

27.214.4.1 void `gdcm::Overlay::Decompress (std::ostream & os)` const

Decode the internal `OverlayData` (packed bits) into unpacked representation.

27.214.4.2 unsigned short `gdcm::Overlay::GetBitPosition ()` const

return bit position

27.214.4.3 unsigned short `gdcm::Overlay::GetBitsAllocated ()` const

return bits allocated

27.214.4.4 unsigned short `gdcm::Overlay::GetColumns ()` const

get columns

27.214.4.5 const char* `gdcm::Overlay::GetDescription ()` const

get description

27.214.4.6 unsigned short `gdcm::Overlay::GetGroup ()` const

Get Group number.

27.214.4.7 const signed short* `gdcm::Overlay::GetOrigin ()` const

get origin

27.214.4.8 const `ByteValue&` `gdcm::Overlay::GetOverlayData ()` const

Return the [Overlay](#) Data as [ByteValue](#): Not thread safe

27.214.4.9 static const char* `gdcm::Overlay::GetOverlayTypeAsString (OverlayType ot)` [static]

27.214.4.10 static `OverlayType` `gdcm::Overlay::GetOverlayTypeFromString (const char *)` [static]

27.214.4.11 unsigned short `gdcm::Overlay::GetRows ()` const

get rows

27.214.4.12 const char* `gdcm::Overlay::GetType ()` const

get type

27.214.4.13 **OverlayType** gdcmm::Overlay::GetTypeAsEnum () const

27.214.4.14 **bool** gdcmm::Overlay::GetUnpackBuffer (*char * buffer*, *size_t len*) const

Retrieve the unpack buffer for [Overlay](#). This is an error if the size is below [GetUnpackBufferLength\(\)](#)

27.214.4.15 **size_t** gdcmm::Overlay::GetUnpackBufferLength () const

Retrieve the size of the buffer needed to hold the [Overlay](#) as specified by Col & Row parameters

27.214.4.16 **bool** gdcmm::Overlay::GrabOverlayFromPixelData (*DataSet* const & *ds*)

27.214.4.17 **bool** gdcmm::Overlay::IsEmpty () const

Return whether or not the [Overlay](#) is empty:

27.214.4.18 **bool** gdcmm::Overlay::IsInPixelData () const

return if the [Overlay](#) is stored in the pixel data or not

27.214.4.19 **void** gdcmm::Overlay::IsInPixelData (*bool b*)

Set whether or no the OverlayData is in the Pixel Data:

27.214.4.20 **bool** gdcmm::Overlay::IsZero () const

return true if all bits are set to 0

27.214.4.21 **Overlay&** gdcmm::Overlay::operator= (*Overlay* const & *ov*)

27.214.4.22 **void** gdcmm::Overlay::Print (*std::ostream &*) const [virtual]

Print.

Reimplemented from [gdcmm::Object](#).

27.214.4.23 **void** gdcmm::Overlay::SetBitPosition (*unsigned short bitposition*)

set bit position

27.214.4.24 **void** gdcmm::Overlay::SetBitsAllocated (*unsigned short bitsallocated*)

set bits allocated

27.214.4.25 **void** gdcmm::Overlay::SetColumns (*unsigned short columns*)

set columns

27.214.4.26 void `gdcm::Overlay::SetDescription (const char * description)`

set description

27.214.4.27 void `gdcm::Overlay::SetFrameOrigin (unsigned short frameorigin)`

set frame origin

27.214.4.28 void `gdcm::Overlay::SetGroup (unsigned short group)`

Set Group number.

27.214.4.29 void `gdcm::Overlay::SetNumberOfFrames (unsigned int numberofframes)`

set number of frames

27.214.4.30 void `gdcm::Overlay::SetOrigin (const signed short origin[2])`

set origin

27.214.4.31 void `gdcm::Overlay::SetOverlay (const char * array, size_t length)`

set overlay from byte array + length

27.214.4.32 void `gdcm::Overlay::SetRows (unsigned short rows)`

set rows

27.214.4.33 void `gdcm::Overlay::SetType (const char * type)`

set type

27.214.4.34 void `gdcm::Overlay::Update (const DataElement & de)`

Update overlay from data element de:

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

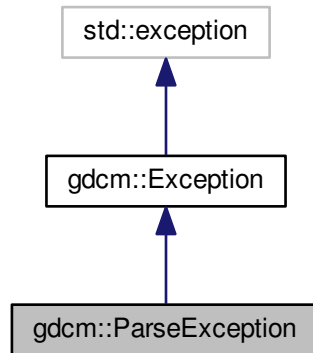
- [gdcmOverlay.h](#)

27.215 gdcm::ParseException Class Reference

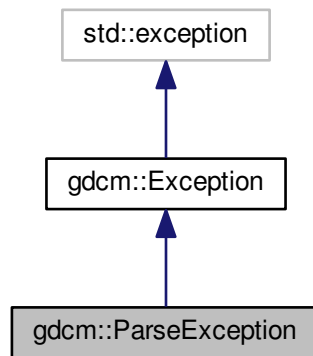
[ParseException](#) Standard exception handling object.

```
#include <gdcmParseException.h>
```

Inheritance diagram for gdcm::ParseException:



Collaboration diagram for gdcm::ParseException:



Public Member Functions

- [ParseException](#) ()
- virtual [~ParseException](#) () throw ()
- const [DataElement](#) & [GetLastElement](#) () const
- [ParseException](#) & [operator=](#) (const [ParseException](#) &orig)
- void [SetLastElement](#) ([DataElement](#) &de)

27.215.1 Detailed Description

[ParseException](#) Standard exception handling object.

27.215.2 Constructor & Destructor Documentation

27.215.2.1 `gdcmm::ParseException::ParseException ()` `[inline]`

27.215.2.2 `virtual gdcmm::ParseException::~~ParseException () throw` `[inline],[virtual]`

27.215.3 Member Function Documentation

27.215.3.1 `const DataElement& gdcmm::ParseException::GetLastElement () const` `[inline]`

27.215.3.2 `ParseException& gdcmm::ParseException::operator= (const ParseException & orig)` `[inline]`

Assignment operator.

27.215.3.3 `void gdcmm::ParseException::SetLastElement (DataElement & de)` `[inline]`

Equivalence operator.

Referenced by `gdcmm::Fragment::ReadBacktrack()`, and `gdcmm::Fragment::ReadValue()`.

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

- [gdcmmParseException.h](#)

27.216 gdcmm::Parser Class Reference

[Parser](#) ala XML_Parser from expat (SAX)

```
#include <gdcmmParser.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](#) ()

27.216.1 Detailed Description

[Parser](#) ala XML_Parser from expat (SAX)

Detailed description here

Note

Simple API for DICOM

27.216.2 Member Typedef Documentation

27.216.2.1 typedef void(* gdcm::Parser::EndElementHandler) (void *userData, const Tag &name)

27.216.2.2 typedef void(* gdcm::Parser::StartElementHandler) (void *userData, const Tag &tag, const char *atts[])

27.216.3 Member Enumeration Documentation

27.216.3.1 enum gdcm::Parser::ErrorType

Enumerator

NoError

NoMemoryError

SyntaxError

NoElementsError

TagMismatchError

DuplicateAttributeError

JunkAfterDocElementError

UndefinedEntityError

UnexpectedStateError

27.216.4 Constructor & Destructor Documentation

27.216.4.1 `gdcm::Parser::Parser () [inline]`

27.216.4.2 `gdcm::Parser::~~Parser () [inline]`

27.216.5 Member Function Documentation

27.216.5.1 `char* gdcm::Parser::GetBuffer (int len) [protected]`

27.216.5.2 `unsigned long gdcm::Parser::GetCurrentByteIndex () const`

27.216.5.3 `ErrorType gdcm::Parser::GetErrorCode () const`

27.216.5.4 `static const char* gdcm::Parser::GetErrorString (ErrorType const & err) [static]`

27.216.5.5 `void* gdcm::Parser::GetUserData () const`

27.216.5.6 `bool gdcm::Parser::Parse (const char * s, int len, bool isFinal)`

27.216.5.7 `bool gdcm::Parser::ParseBuffer (int len, bool isFinal) [protected]`

27.216.5.8 `ErrorType gdcm::Parser::Process () [protected]`

27.216.5.9 `void gdcm::Parser::SetElementHandler (StartElementHandler start, EndElementHandler end)`

27.216.5.10 `void gdcm::Parser::SetUserData (void * userData)`

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

- [gdcmParser.h](#)

27.217 gdcm::Patient Class Reference

See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

```
#include <gdcmPatient.h>
```

Public Member Functions

- [Patient \(\)](#)

27.217.1 Detailed Description

See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

27.217.2 Constructor & Destructor Documentation

27.217.2.1 gdcm::Patient::Patient() [inline]

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

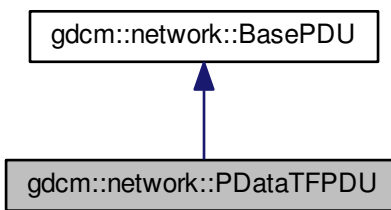
- [gdcmPatient.h](#)

27.218 gdcm::network::PDataTFPDU Class Reference

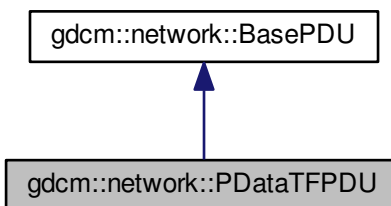
[PDataTFPDU Table](#) 9-22 P-DATA-TF PDU FIELDS.

```
#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
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- size_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

Protected Member Functions

- std::istream & [ReadInto](#) (std::istream &is, std::ostream &os)

27.218.1 Detailed Description

[PDataTFPDU](#) Table 9-22 P-DATA-TF PDU FIELDS.

27.218.2 Member Typedef Documentation

27.218.2.1 typedef std::vector<[PresentationDataValue](#)>::size_type [gdcm::network::PDataTFPDU::SizeType](#)

27.218.3 Constructor & Destructor Documentation

27.218.3.1 [gdcm::network::PDataTFPDU::PDataTFPDU](#) ()

27.218.4 Member Function Documentation

27.218.4.1 void [gdcm::network::PDataTFPDU::AddPresentationDataValue](#) ([PresentationDataValue](#) const & *pdv*)
[inline]

27.218.4.2 [SizeType](#) [gdcm::network::PDataTFPDU::GetNumberOfPresentationDataValues](#) () const [inline]

27.218.4.3 [PresentationDataValue](#) const& [gdcm::network::PDataTFPDU::GetPresentationDataValue](#) ([SizeType](#) *i*) const
[inline]

27.218.4.4 bool [gdcm::network::PDataTFPDU::IsLastFragment](#) () const [virtual]

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

27.218.4.5 void [gdcm::network::PDataTFPDU::Print](#) (std::ostream & *os*) const [virtual]

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

27.218.4.6 std::istream& [gdcm::network::PDataTFPDU::Read](#) (std::istream & *is*) [virtual]

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

27.218.4.7 `std::istream& gdcm::network::PDataTFPDU::ReadInto (std::istream & is, std::ostream & os)` [protected]

27.218.4.8 `size_t gdcm::network::PDataTFPDU::Size () const` [virtual]

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

27.218.4.9 `const std::ostream& gdcm::network::PDataTFPDU::Write (std::ostream & os) const` [virtual]

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

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

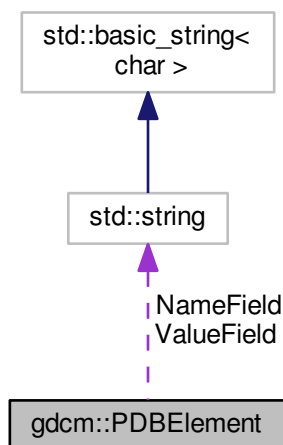
- [gdcmPDataTFPDU.h](#)

27.219 gdcm::PDBelement Class Reference

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

```
#include <gdcmPDBelement.h>
```

Collaboration diagram for gdcm::PDBelement:



Public Member Functions

- [PDBelement](#) ()
- `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)

27.219.1 Detailed Description

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

See also

[PDBHeader](#)

27.219.2 Constructor & Destructor Documentation

27.219.2.1 `gdcm::PDBElement::PDBElement ()` [[inline](#)]

27.219.3 Member Function Documentation

27.219.3.1 `const char* gdcm::PDBElement::GetName ()` const [[inline](#)]

Set/Get Name.

27.219.3.2 `const char* gdcm::PDBElement::GetValue ()` const [[inline](#)]

Set/Get [Value](#).

27.219.3.3 `bool gdcm::PDBElement::operator== (const PDBElement & de)` const [[inline](#)]

References [NameField](#), and [ValueField](#).

27.219.3.4 `void gdcm::PDBElement::SetName (const char * name)` [[inline](#)]

27.219.3.5 `void gdcm::PDBElement::SetValue (const char * value)` [[inline](#)]

27.219.4 Friends And Related Function Documentation

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

27.219.5 Member Data Documentation

27.219.5.1 std::string gdcm::PDBElement::NameField [protected]

Referenced by `gdcm::operator<<()`, and `operator==()`.

27.219.5.2 std::string gdcm::PDBElement::ValueField [protected]

Referenced by `gdcm::operator<<()`, and `operator==()`.

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

- [gdcmPDBElement.h](#)

27.220 gdcm::PDBHeader Class Reference

Class for [PDBHeader](#).

```
#include <gdcmPDBHeader.h>
```

Public Member Functions

- [PDBHeader](#) ()
- [~PDBHeader](#) ()
- 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)

27.220.1 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](#)

27.220.2 Constructor & Destructor Documentation

27.220.2.1 `gdcm::PDBHeader::PDBHeader () [inline]`

27.220.2.2 `gdcm::PDBHeader::~~PDBHeader () [inline]`

27.220.3 Member Function Documentation

27.220.3.1 `bool gdcm::PDBHeader::FindPDBElementByName (const char * name)`

Return true if the PDB element matching name is found or not.

27.220.3.2 `const PDBElement& gdcm::PDBHeader::GetPDBEEnd () const [protected]`

27.220.3.3 `const PDBElement& gdcm::PDBHeader::GetPDBElementByName (const char * name)`

Lookup in the PDB header if a PDB element match the name 'name':

Warning

Case Sensitive

27.220.3.4 `static const PrivateTag& gdcm::PDBHeader::GetPDBInfoTag () [static]`

Return the Private [Tag](#) where the PDB header is stored within a DICOM [DataSet](#).

27.220.3.5 `bool gdcm::PDBHeader::LoadFromDataElement (DataElement const & de)`

Load the PDB Header from a [DataElement](#) of a [DataSet](#).

27.220.3.6 void gdcm::PDBHeader::Print (std::ostream & os) const

Print.

Referenced by gdcm::operator<<().

27.220.4 Friends And Related Function Documentation

27.220.4.1 std::ostream& operator<< (std::ostream & _os, const PDBHeader & d) [friend]

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

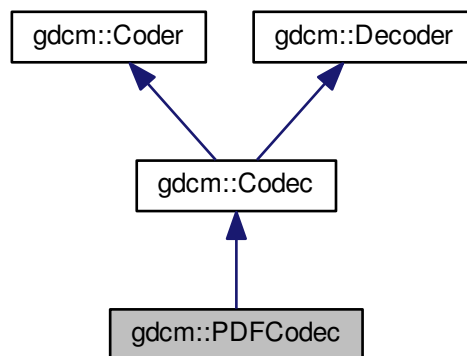
- [gdcmPDBHeader.h](#)

27.221 gdcm::PDFCodec Class Reference

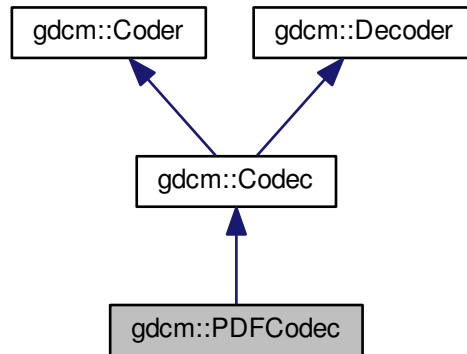
[PDFCodec](#) class.

```
#include <gdcmPDFCodec.h>
```

Inheritance diagram for gdcm::PDFCodec:



Collaboration diagram for `gdcm::PDFCodec`:



Public Member Functions

- [PDFCodec](#) ()
- [~PDFCodec](#) ()
- bool [CanCode](#) ([TransferSyntax](#) const &) const
Return whether this coder support this transfer syntax (can code it)
- bool [CanDecode](#) ([TransferSyntax](#) const &) const
Return whether this decoder support this transfer syntax (can decode it)
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os)
Decode.

Additional Inherited Members

27.221.1 Detailed Description

[PDFCodec](#) class.

27.221.2 Constructor & Destructor Documentation

27.221.2.1 `gdcm::PDFCodec::PDFCodec ()`

27.221.2.2 `gdcm::PDFCodec::~~PDFCodec ()`

27.221.3 Member Function Documentation

27.221.3.1 `bool gdcm::PDFCodec::CanCode (TransferSyntax const &) const` `[inline]`, `[virtual]`

Return whether this coder support this transfer syntax (can code it)

Implements [gdcm::Coder](#).

27.221.3.2 `bool gdcm::PDFCodec::CanDecode (TransferSyntax const &) const` `[inline],[virtual]`

Return whether this decoder support this transfer syntax (can decode it)

Implements [gdcm::Decoder](#).

27.221.3.3 `bool gdcm::PDFCodec::Decode (DataElement const & , DataElement &)` `[virtual]`

Decode.

Reimplemented from [gdcm::Decoder](#).

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

- [gdcmPDFCodec.h](#)

27.222 gdcm::network::PDUFactory Class Reference

[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.

```
#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](#) * > [CreateCStoreRSPDU](#) (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)

27.222.1 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.

27.222.2 Member Function Documentation

- 27.222.2.1 static **BasePDU*** `gdcmm::network::PDUFactory::ConstructAbortPDU ()` [static]
- 27.222.2.2 static **BasePDU*** `gdcmm::network::PDUFactory::ConstructPDU (uint8_t itemtype)` [static]
- 27.222.2.3 static **BasePDU*** `gdcmm::network::PDUFactory::ConstructReleasePDU ()` [static]
- 27.222.2.4 static **std::vector<BasePDU*>** `gdcmm::network::PDUFactory::CreateCEchoPDU (const ULConnection & inConnection)` [static]
- 27.222.2.5 static **std::vector<BasePDU*>** `gdcmm::network::PDUFactory::CreateCFindPDU (const ULConnection & inConnection, const BaseRootQuery * inRootQuery)` [static]
- 27.222.2.6 static **std::vector<BasePDU*>** `gdcmm::network::PDUFactory::CreateCMovePDU (const ULConnection & inConnection, const BaseRootQuery * inRootQuery)` [static]
- 27.222.2.7 static **std::vector<BasePDU*>** `gdcmm::network::PDUFactory::CreateCStoreRQPDU (const ULConnection & inConnection, const File & file, bool writeDataSet = true)` [static]
- 27.222.2.8 static **std::vector<BasePDU*>** `gdcmm::network::PDUFactory::CreateCStoreRSPPDU (const DataSet * inDataSet, const BasePDU * inPC)` [static]
- 27.222.2.9 static **std::vector<BasePDU*>** `gdcmm::network::PDUFactory::CreateNActionPDU (const ULConnection & inConnection, const BaseQuery * inQuery)` [static]
- 27.222.2.10 static **std::vector<BasePDU*>** `gdcmm::network::PDUFactory::CreateNCreatePDU (const ULConnection & inConnection, const BaseQuery * inQuery)` [static]
- 27.222.2.11 static **std::vector<BasePDU*>** `gdcmm::network::PDUFactory::CreateNDeletePDU (const ULConnection & inConnection, const BaseQuery * inQuery)` [static]
- 27.222.2.12 static **std::vector<BasePDU*>** `gdcmm::network::PDUFactory::CreateNEventReportPDU (const ULConnection & inConnection, const BaseQuery * inQuery)` [static]
- 27.222.2.13 static **std::vector<BasePDU*>** `gdcmm::network::PDUFactory::CreateNGetPDU (const ULConnection & inConnection, const BaseQuery * inQuery)` [static]
- 27.222.2.14 static **std::vector<BasePDU*>** `gdcmm::network::PDUFactory::CreateNSetPDU (const ULConnection & inConnection, const BaseQuery * inQuery)` [static]
- 27.222.2.15 static **EEventID** `gdcmm::network::PDUFactory::DetermineEventByPDU (const BasePDU * inPDU)` [static]
- 27.222.2.16 static **std::vector<PresentationDataValue>** `gdcmm::network::PDUFactory::GetPDVs (const std::vector<BasePDU * > & inDataPDUs)` [static]

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

- [gdcmPDUFactory.h](#)

27.223 gdcm::PersonName Class Reference

[PersonName](#) class.

```
#include <gdcmPersonName.h>
```

Public Member Functions

- unsigned int [GetMaxLength](#) () const
- unsigned int [GetNumberOfComponents](#) () const
- void [Print](#) (std::ostream &os) const
- void [SetBlob](#) (const std::vector< char > &v)
- void [SetComponents](#) (const char *comp1="", const char *comp2="", const char *comp3="", const char *comp4="", const char *comp5="")
- void [SetComponents](#) (const char *components[])

Public Attributes

- char [Component](#) [[MaxNumberOfComponents](#)][[MaxLength](#)+1]

Static Public Attributes

- static const unsigned int [MaxLength](#) = 64
- static const unsigned int [MaxNumberOfComponents](#) = 5
- static const char [Padding](#) = ' '
- static const char [Separator](#) = '^'

27.223.1 Detailed Description

[PersonName](#) class.

27.223.2 Member Function Documentation

27.223.2.1 unsigned int gdcm::PersonName::GetMaxLength () const [inline]

27.223.2.2 unsigned int gdcm::PersonName::GetNumberOfComponents () const [inline]

27.223.2.3 void gdcm::PersonName::Print (std::ostream & os) const [inline]

27.223.2.4 void gdcm::PersonName::SetBlob (const std::vector< char > & v) [inline]

27.223.2.5 void gdcm::PersonName::SetComponents (const char * comp1 = " ", const char * comp2 = " ", const char * comp3 = " ", const char * comp4 = " ", const char * comp5 = " ") [inline]

27.223.2.6 void gdcm::PersonName::SetComponents (const char * components[]) [inline]

27.223.3 Member Data Documentation

27.223.3.1 `char gdcM::PersonName::Component[MaxNumberOfComponents][MaxLength+1]`

27.223.3.2 `const unsigned int gdcM::PersonName::MaxLength = 64` `[static]`

27.223.3.3 `const unsigned int gdcM::PersonName::MaxNumberOfComponents = 5` `[static]`

27.223.3.4 `const char gdcM::PersonName::Padding = ''` `[static]`

27.223.3.5 `const char gdcM::PersonName::Separator = '^'` `[static]`

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

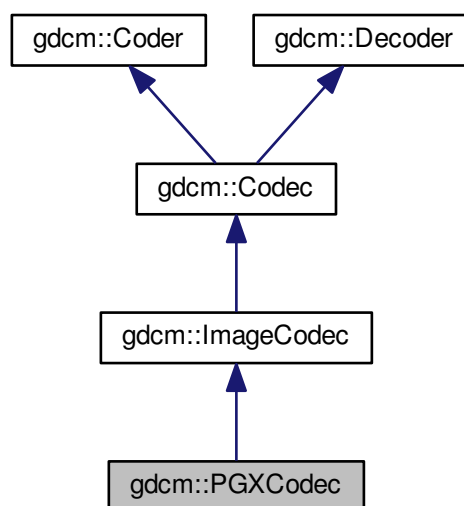
- [gdcMPersonName.h](#)

27.224 gdcM::PGXCodec Class Reference

Class to do PGX See PGX as used in JPEG 2000 implementation and reference images.

```
#include <gdcM_PGXCodec.h>
```

Inheritance diagram for gdcM::PGXCodec:



```

graph TD
    PGXCodec[gdcm::PGXCodec] --> ImageCodec[gdcm::ImageCodec]
    ImageCodec --> Codec[gdcm::Codec]
    ImageCodec --> PI[gdcm::PhotometricInterpretation]
    ImageCodec --> PF[gdcm::PixelFormat]
    ImageCodec --> SP[gdcm::SmartPointer]
    Codec --> Coder[gdcm::Coder]
    Codec --> Decoder[gdcm::Decoder]
    PI --> PF
    PF --> SP_LUT[gdcm::SmartPointer < LookupTable >]
    SP --> SP_LUT
    SP_LUT -.-> SP_OT[gdcm::SmartPointer < ObjectType >]
  
```

- PGXCodec ()
- ~PGXCodec ()
- bool CanCode (TransferSyntax const &ts) const
Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const
Return whether this decoder support this transfer syntax (can decode it)
- virtual ImageCodec * Clone () const
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- bool Read (const char *filename, DataElement &out) const
- bool Write (const char *filename, const DataElement &out) const

27.224.1 Detailed Description

27.224.2 Constructor & Destructor Documentation

27.224.2.2 gdcmm::PGXCodec::~PGXCodec ()

```
27.224.3.1 bool gdcmm::PGXCodec::CanCode ( TransferSyntax const & ) const [virtual]
```

Reimplemented from `gdcm::ImageCodec`.

27.224.3.2 `bool gdcm::PGXCodec::CanDecode (TransferSyntax const &) const` [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

27.224.3.3 `virtual ImageCodec* gdcm::PGXCodec::Clone () const` [virtual]

Implements [gdcm::ImageCodec](#).

27.224.3.4 `bool gdcm::PGXCodec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` [virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.224.3.5 `bool gdcm::PGXCodec::Read (const char * filename, DataElement & out) const`

27.224.3.6 `bool gdcm::PGXCodec::Write (const char * filename, const DataElement & out) const`

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

- [gdcmPGXCodec.h](#)

27.225 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](#) ([PType](#) 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)

27.225.1 Detailed Description

Class to represent an [PhotometricInterpretation](#).

Examples:

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

27.225.2 Member Enumeration Documentation

27.225.2.1 enum gdcm::PhotometricInterpretation::PIType

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

27.225.3 Constructor & Destructor Documentation

27.225.3.1 `gdcm::PhotometricInterpretation::PhotometricInterpretation (PType pi = UNKNOWN)` `[inline]`

27.225.4 Member Function Documentation

27.225.4.1 `static const char* gdcm::PhotometricInterpretation::GetPIString (PType pi)` `[static]`

Referenced by `gdcm::operator<<()`.

27.225.4.2 `static PType gdcm::PhotometricInterpretation::GetPIType (const char * pi)` `[static]`

27.225.4.3 `unsigned short gdcm::PhotometricInterpretation::GetSamplesPerPixel () const`

return the value for Sample Per Pixel associated with a particular Photometric Interpretation

27.225.4.4 `const char* gdcm::PhotometricInterpretation::GetString () const`

27.225.4.5 `PType gdcm::PhotometricInterpretation::GetType () const` `[inline]`

27.225.4.6 `bool gdcm::PhotometricInterpretation::IsLossless () const`

27.225.4.7 `bool gdcm::PhotometricInterpretation::IsLossy () const`

27.225.4.8 `static bool gdcm::PhotometricInterpretation::IsRetired (PType pi)` `[static]`

27.225.4.9 `bool gdcm::PhotometricInterpretation::IsSameColorSpace (PhotometricInterpretation const & pi) const`

27.225.4.10 `gdcm::PhotometricInterpretation::operator PType () const` `[inline]`

27.225.5 Friends And Related Function Documentation

27.225.5.1 `std::ostream& operator<< (std::ostream & os, const PhotometricInterpretation & pi)` `[friend]`

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

- [gdcmPhotometricInterpretation.h](#)

27.226 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](#) (unsigned short samplesperpixel=1, unsigned short bitsallocated=8, unsigned short bitsstored=8, unsigned short highbit=7, unsigned short pixelrepresentation=0)
- [PixelFormat](#) ([ScalarType](#) st)
- 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!=](#) ([ScalarType](#) st) const
- bool [operator!=](#) (const [PixelFormat](#) &pf) const
- bool [operator==](#) ([ScalarType](#) st) const
- bool [operator==](#) (const [PixelFormat](#) &pf) 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)

27.226.1 Detailed Description

[PixelFormat](#).

Note

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](#), [csa2img.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetJPEGSample←Precision.cxx](#), [iU22tomultisc.cxx](#), and [threadgdcmm.cxx](#).

27.226.2 Member Enumeration Documentation

27.226.2.1 enum gdcm::PixelFormat::ScalarType

Enumerator

UINT8

INT8

UINT12
INT12
UINT16
INT16
UINT32
INT32
UINT64
INT64
FLOAT16
FLOAT32
FLOAT64
SINGLEBIT
UNKNOWN

27.226.3 Constructor & Destructor Documentation

27.226.3.1 `gdcm::PixelFormat::PixelFormat (unsigned short samplesperpixel = 1, unsigned short bitsallocated = 8, unsigned short bitsstored = 8, unsigned short highbit = 7, unsigned short pixelrepresentation = 0)` `[inline]`, `[explicit]`

27.226.3.2 `gdcm::PixelFormat::PixelFormat (ScalarType st)`

27.226.4 Member Function Documentation

27.226.4.1 `unsigned short gdcm::PixelFormat::GetBitsAllocated () const` `[inline]`

BitsAllocated see [Tag](#) (0028,0100) US Bits Allocated.

Examples:

[GetJPEGSamplePrecision.cxx](#).

27.226.4.2 `unsigned short gdcm::PixelFormat::GetBitsStored () const` `[inline]`

BitsStored see [Tag](#) (0028,0101) US Bits Stored.

Examples:

[GetJPEGSamplePrecision.cxx](#).

27.226.4.3 `unsigned short gdcm::PixelFormat::GetHighBit () const` `[inline]`

HighBit see [Tag](#) (0028,0102) US High Bit.

27.226.4.4 `int64_t gdcm::PixelFormat::GetMax () const`

return the max possible of the pixel

27.226.4.5 `int64_t gdcm::PixelFormat::GetMin () const`

return the min possible of the pixel

27.226.4.6 `unsigned short gdcm::PixelFormat::GetPixelRepresentation () const` `[inline]`

PixelRepresentation: 0 or 1, see [Tag](#) (0028,0103) US Pixel Representation.

27.226.4.7 `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](#).

27.226.4.8 `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](#).

27.226.4.9 `ScalarType gdcm::PixelFormat::GetScalarType () const`

ScalarType does not take into account the sample per pixel.

27.226.4.10 `const char* gdcm::PixelFormat::GetScalarTypeAsString () const`

27.226.4.11 `bool gdcm::PixelFormat::IsCompatible (const TransferSyntax & ts) const`

27.226.4.12 `bool gdcm::PixelFormat::IsValid () const`

return IsValid

27.226.4.13 `gdcm::PixelFormat::operator ScalarType () const` `[inline]`

27.226.4.14 `bool gdcm::PixelFormat::operator!= (ScalarType st) const` `[inline]`

27.226.4.15 `bool gdcm::PixelFormat::operator!= (const PixelFormat & pf) const` `[inline]`

27.226.4.16 `bool gdcm::PixelFormat::operator==(ScalarType st) const` `[inline]`

27.226.4.17 `bool gdcm::PixelFormat::operator==(const PixelFormat & pf) const` `[inline]`

27.226.4.18 `void gdcm::PixelFormat::Print (std::ostream & os) const`

Print.

Referenced by `gdcm::operator<<()`.

27.226.4.19 `void gdcm::PixelFormat::SetBitsAllocated (unsigned short ba)` `[inline]`

27.226.4.20 `void gdcm::PixelFormat::SetBitsStored (unsigned short bs)` `[inline]`

27.226.4.21 `void gdcm::PixelFormat::SetHighBit (unsigned short hb)` `[inline]`

27.226.4.22 `void gdcm::PixelFormat::SetPixelRepresentation (unsigned short pr)` `[inline]`

27.226.4.23 `void gdcm::PixelFormat::SetSamplesPerPixel (unsigned short spp)` `[inline]`

Examples:

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

References `gdcmAssertMacro`.

27.226.4.24 `void gdcm::PixelFormat::SetScalarType (ScalarType st)`

Set [PixelFormat](#) based only on the `ScalarType`

Warning

: You need to call `SetScalarType` *before* `SetSamplesPerPixel`

27.226.4.25 `bool gdcm::PixelFormat::Validate ()` `[protected]`

When image with 24/24/23 was read, need to validate.

Referenced by `gdcm::Bitmap::SetPixelFormat()`.

27.226.5 Friends And Related Function Documentation

27.226.5.1 `friend class Bitmap` `[friend]`

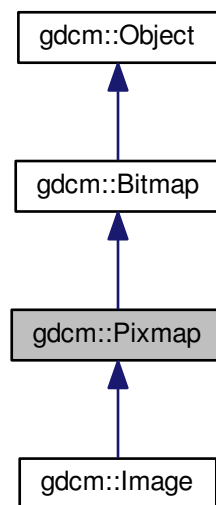
27.226.5.2 `std::ostream& operator<< (std::ostream & _os, const PixelFormat & pf)` `[friend]`

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

- [gdcmPixelFormat.h](#)

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)

Inheritance diagram for gdcm::Pixmap:

[illegible]

Public Member Functions

- [Pixmap](#) ()
- [~Pixmap](#) ()
- bool [AreOverlaysInPixelData](#) () const
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
- const [IconImage](#) & [GetIconImage](#) () const
Set/Get Icon Image.
- [IconImage](#) & [GetIconImage](#) ()
- 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
- void [RemoveOverlay](#) (size_t i)
- void [SetIconImage](#) ([IconImage](#) const &ii)
- void [SetNumberOfCurves](#) (size_t n)
- void [SetNumberOfOverlays](#) (size_t n)

Protected Attributes

- std::vector< [Curve](#) > [Curves](#)
- [SmartPointer](#)< [IconImage](#) > [Icon](#)
- std::vector< [Overlay](#) > [Overlays](#)

Additional Inherited Members

27.227.1 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](#)

27.227.2 Constructor & Destructor Documentation

27.227.2.1 [gdcm::Pixmap::Pixmap](#) ()

27.227.2.2 [gdcm::Pixmap::~~Pixmap](#) ()

27.227.3 Member Function Documentation

27.227.3.1 `bool gdcm::Pixmap::AreOverlaysInPixelData () const` [virtual]

returns if Overlays are stored in the unused bit of the pixel data:

Reimplemented from [gdcm::Bitmap](#).

27.227.3.2 `Curve& gdcm::Pixmap::GetCurve (size_t i = 0)` [inline]

[Curve](#): group 50xx.

27.227.3.3 `const Curve& gdcm::Pixmap::GetCurve (size_t i = 0) const` [inline]

27.227.3.4 `const IconImage& gdcm::Pixmap::GetIconImage () const` [inline]

Set/Get Icon [Image](#).

27.227.3.5 `IconImage& gdcm::Pixmap::GetIconImage ()` [inline]

27.227.3.6 `size_t gdcm::Pixmap::GetNumberOfCurves () const` [inline]

27.227.3.7 `size_t gdcm::Pixmap::GetNumberOfOverlays () const` [inline]

27.227.3.8 `Overlay& gdcm::Pixmap::GetOverlay (size_t i = 0)` [inline]

[Overlay](#): group 60xx.

27.227.3.9 `const Overlay& gdcm::Pixmap::GetOverlay (size_t i = 0) const` [inline]

27.227.3.10 `void gdcm::Pixmap::Print (std::ostream &) const` [virtual]

Reimplemented from [gdcm::Bitmap](#).

27.227.3.11 `void gdcm::Pixmap::RemoveOverlay (size_t i)` [inline]

27.227.3.12 `void gdcm::Pixmap::SetIconImage (IconImage const & ii)` [inline]

27.227.3.13 `void gdcm::Pixmap::SetNumberOfCurves (size_t n)` [inline]

27.227.3.14 `void gdcm::Pixmap::SetNumberOfOverlays (size_t n)` [inline]

27.227.4 Member Data Documentation

27.227.4.1 `std::vector<Curve> gdcm::Pixmap::Curves` [protected]

27.227.4.2 `SmartPointer<IconImage> gdcm::Pixmap::Icon` [protected]

27.227.4.3 `std::vector<Overlay> gdcm::Pixmap::Overlays` [protected]

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

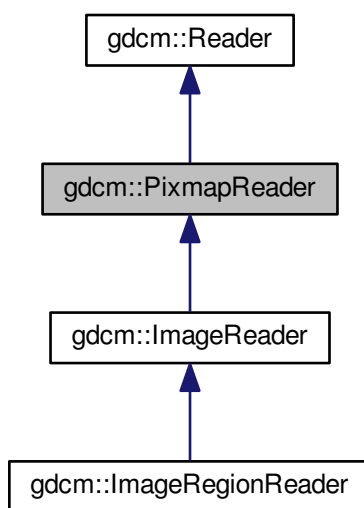
- [gdcmPixmap.h](#)

27.228 gdcm::PixmapReader Class Reference

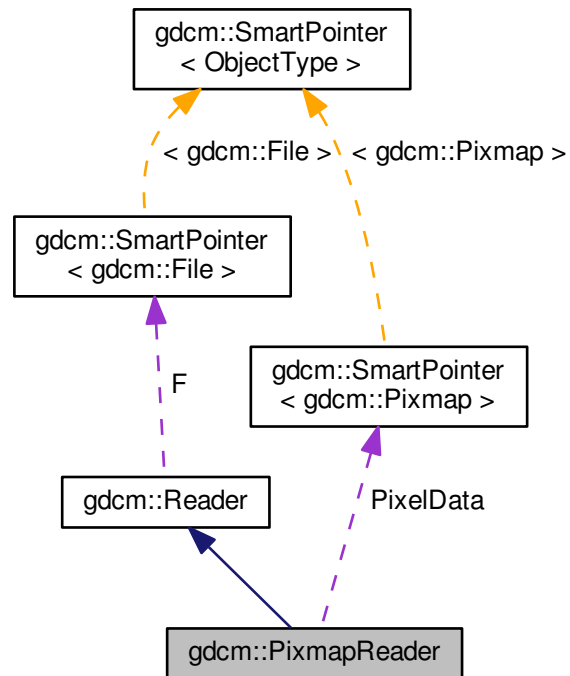
[PixmapReader](#).

```
#include <gdcmPixmapReader.h>
```

Inheritance diagram for gdcm::PixmapReader:



Collaboration diagram for `gdcm::PixmapReader`:



Public Member Functions

- [PixmapReader](#) ()
- virtual [~PixmapReader](#) ()
- const [Pixmap](#) & [GetPixmap](#) () const
Return the read image (need to call [Read\(\)](#) first)
- [Pixmap](#) & [GetPixmap](#) ()
- virtual bool [Read](#) ()

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](#)

27.228.1 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](#)

27.228.2 Constructor & Destructor Documentation

27.228.2.1 `gdcm::PixmapReader::PixmapReader ()`

27.228.2.2 `virtual gdcm::PixmapReader::~~PixmapReader () [virtual]`

27.228.3 Member Function Documentation

27.228.3.1 `const Pixmap& gdcm::PixmapReader::GetPixmap () const`

Return the read image (need to call [Read\(\)](#) first)

27.228.3.2 `Pixmap& gdcm::PixmapReader::GetPixmap ()`

27.228.3.3 `virtual bool gdcm::PixmapReader::Read () [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](#).

Reimplemented in [gdcm::ImageRegionReader](#), and [gdcm::ImageReader](#).

27.228.3.4 `virtual bool gdcm::PixmapReader::ReadACRNEMAIImage () [protected], [virtual]`

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

27.228.3.5 `virtual bool gdcm::PixmapReader::ReadImage (MediaStorage const & ms) [protected], [virtual]`

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

27.228.3.6 `bool gdcM::PixmapReader::ReadImageInternal (MediaStorage const & ms, bool handlepixeldata = true)`
`[protected]`

27.228.4 Member Data Documentation

27.228.4.1 `SmartPointer<Pixmap> gdcM::PixmapReader::PixelData` `[protected]`

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

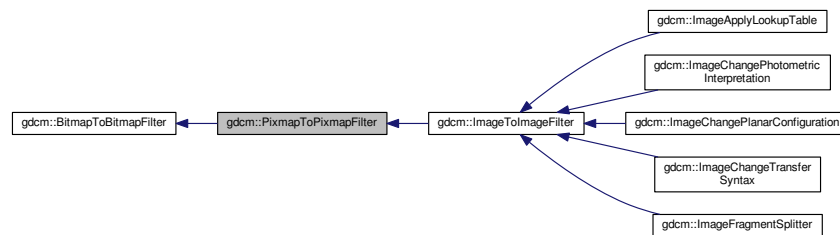
- [gdcMPixmapReader.h](#)

27.229 gdcM::PixmapToPixmapFilter Class Reference

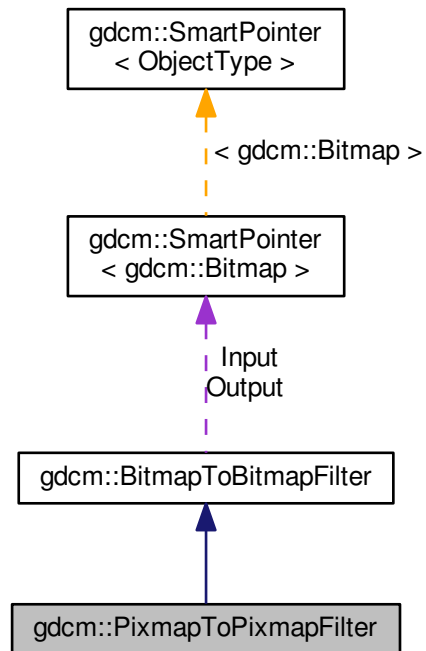
[PixmapToPixmapFilter](#) class Super class for all filter taking an image and producing an output image.

```
#include <gdcMPixmapToPixmapFilter.h>
```

Inheritance diagram for `gdcM::PixmapToPixmapFilter`:



Collaboration diagram for gdcm::PixmapToPixmapFilter:



Public Member Functions

- [PixmapToPixmapFilter \(\)](#)
- [~PixmapToPixmapFilter \(\)](#)
- [Pixmap & GetInput \(\)](#)
- [const Pixmap & GetOutput \(\) const](#)
Get Output image.
- [const Pixmap & GetOutputAsPixmap \(\) const](#)

Additional Inherited Members

27.229.1 Detailed Description

[PixmapToPixmapFilter](#) class Super class for all filter taking an image and producing an output image.

27.229.2 Constructor & Destructor Documentation

27.229.2.1 gdcm::PixmapToPixmapFilter::PixmapToPixmapFilter ()

27.229.2.2 `gdcm::PixmapToPixmapFilter::~~PixmapToPixmapFilter ()` `[inline]`

27.229.3 Member Function Documentation

27.229.3.1 `Pixmap& gdcm::PixmapToPixmapFilter::GetInput ()`

27.229.3.2 `const Pixmap& gdcm::PixmapToPixmapFilter::GetOutput () const`

Get Output image.

27.229.3.3 `const Pixmap& gdcm::PixmapToPixmapFilter::GetOutputAsPixmap () const`

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

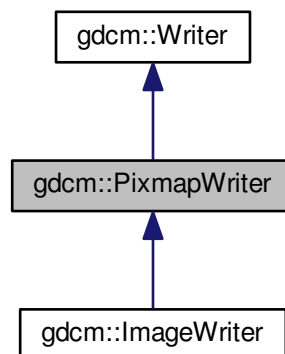
- [gdcmPixmapToPixmapFilter.h](#)

27.230 gdcm::PixmapWriter Class Reference

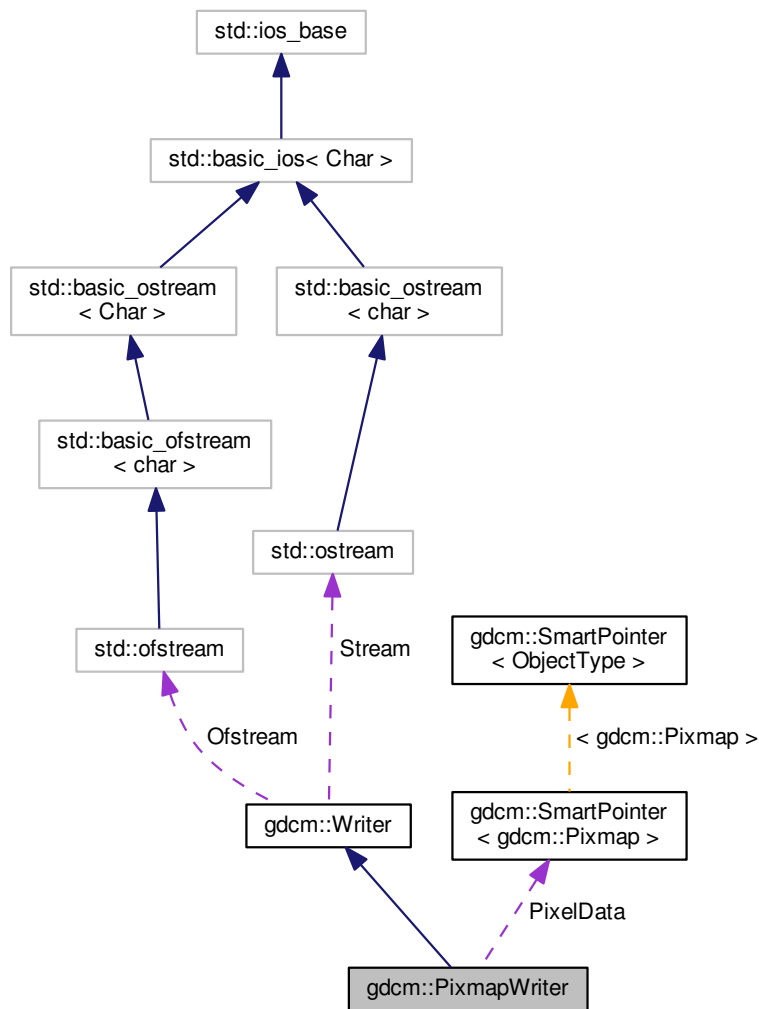
[PixmapWriter](#) This class will takes two inputs:

```
#include <gdcmPixmapWriter.h>
```

Inheritance diagram for `gdcm::PixmapWriter`:



Collaboration diagram for gdcm::PixmapWriter:



Public Member Functions

- `PixmapWriter ()`
- `~PixmapWriter ()`
- `virtual const Pixmap & GetImage () const`
- `virtual Pixmap & GetImage ()`
- `const Pixmap & GetPixmap () const`
- `Pixmap & GetPixmap ()`
- `virtual void SetImage (Pixmap const &img)`
- `void SetPixmap (Pixmap const &img)`
- `bool Write ()`

Write.

Protected Member Functions

- void [DolconImage](#) ([DataSet](#) &ds, [Pixmap](#) const &image)
- bool [PrepareWrite](#) ()
- bool [PrepareWrite](#) ([MediaStorage](#) const &refms)

Protected Attributes

- [SmartPointer](#)< [Pixmap](#) > [PixelData](#)

27.230.1 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)

27.230.2 Constructor & Destructor Documentation

27.230.2.1 `gdcm::PixmapWriter::PixmapWriter ()`

27.230.2.2 `gdcm::PixmapWriter::~~PixmapWriter ()`

27.230.3 Member Function Documentation

27.230.3.1 `void gdcm::PixmapWriter::DolconImage (DataSet & ds, Pixmap const & image)` `[protected]`

27.230.3.2 `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](#).

27.230.3.3 `virtual Pixmap& gdcm::PixmapWriter::GetImage ()` `[inline],[virtual]`

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

27.230.3.4 `const Pixmap& gdcm::PixmapWriter::GetPixmap () const` `[inline]`

27.230.3.5 `Pixmap& gdcm::PixmapWriter::GetPixmap ()` `[inline]`

27.230.3.6 `bool gdcm::PixmapWriter::PrepareWrite ()` `[protected]`

27.230.3.7 `bool gdcm::PixmapWriter::PrepareWrite (MediaStorage const & refms)` `[protected]`

27.230.3.8 virtual void gdcm::PixmapWriter::SetImage (Pixmap const & *img*) [virtual]

Examples:

[CompressImage.cxx](#), [GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), and [MergeTwoFiles.cxx](#).

27.230.3.9 void gdcm::PixmapWriter::SetPixmap (Pixmap const & *img*)

27.230.3.10 bool gdcm::PixmapWriter::Write () [virtual]

Write.

Reimplemented from [gdcm::Writer](#).

27.230.4 Member Data Documentation

27.230.4.1 `SmartPointer<Pixmap>` gdcm::PixmapWriter::PixelData [protected]

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

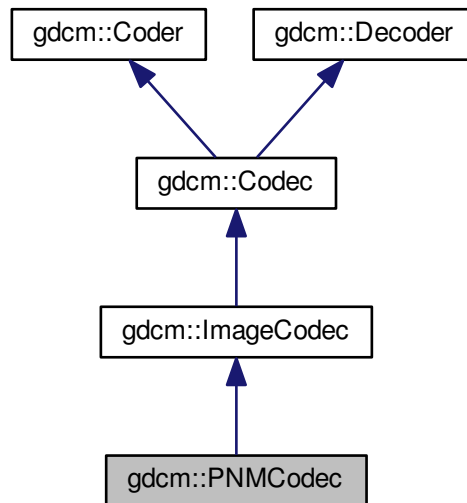
- [gdcmPixmapWriter.h](#)

27.231 gdcm::PNMCodec Class Reference

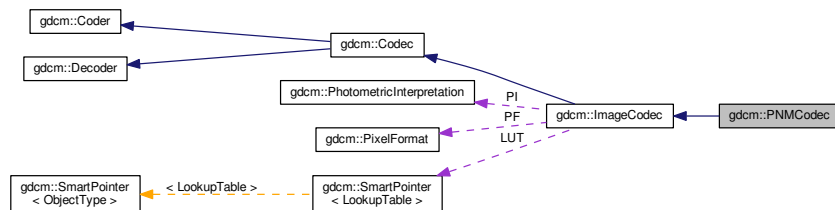
Class to do PNM PNM is the Portable anymap file format. The main web page can be found at: <http://netpbm.sourceforge.net/>.↵

```
#include <gdcmPNMCodec.h>
```

Inheritance diagram for `gdcm::PNMCodec`:



Collaboration diagram for `gdcm::PNMCodec`:



Public Member Functions

- [PNMCodec](#) ()
- [~PNMCodec](#) ()
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const
Return whether this coder support this transfer syntax (can code it)
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const
Return whether this decoder support this transfer syntax (can decode it)
- virtual [ImageCodec](#) * [Clone](#) () const
- unsigned long [GetBufferLength](#) () const
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts)
- 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

27.231.1 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](#).

27.231.2 Constructor & Destructor Documentation

27.231.2.1 `gdcm::PNMCodec::PNMCodec ()`

27.231.2.2 `gdcm::PNMCodec::~~PNMCodec ()`

27.231.3 Member Function Documentation

27.231.3.1 `bool gdcm::PNMCodec::CanCode (TransferSyntax const &) const` [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

27.231.3.2 `bool gdcm::PNMCodec::CanDecode (TransferSyntax const &) const` [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

27.231.3.3 `virtual ImageCodec* gdcm::PNMCodec::Clone () const` [virtual]

Implements [gdcm::ImageCodec](#).

27.231.3.4 `unsigned long gdcm::PNMCodec::GetBufferLength () const` [inline]

27.231.3.5 `bool gdcm::PNMCodec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` [virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.231.3.6 `bool gdcmm::PNMCodec::Read (const char * filename, DataElement & out) const`

27.231.3.7 `void gdcmm::PNMCodec::SetBufferLength (unsigned long l) [inline]`

27.231.3.8 `bool gdcmm::PNMCodec::Write (const char * filename, const DataElement & out) const`

Examples:

[ExtractIconFromFile.cxx](#).

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

- [gdcmmPNMCodec.h](#)

27.232 gdcmm::Preamble Class Reference

DICOM [Preamble](#) (Part 10)

```
#include <gdcmmPreamble.h>
```

Public Member Functions

- [Preamble](#) ()
- [Preamble](#) ([Preamble](#) const &)
- [~Preamble](#) ()
- void [Clear](#) ()
- void [Create](#) ()
- const char * [GetInternal](#) () const
- [VL GetLength](#) () const
- bool [IsEmpty](#) () const
- [Preamble](#) & [operator=](#) ([Preamble](#) const &)
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [Remove](#) ()
- void [Valid](#) ()
- std::ostream const & [Write](#) (std::ostream &os) const

Protected Member Functions

- bool [IsValid](#) () const

Friends

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

27.232.1 Detailed Description

DICOM [Preamble](#) (Part 10)

27.232.2 Constructor & Destructor Documentation

27.232.2.1 `gdcm::Preamble::Preamble ()`

27.232.2.2 `gdcm::Preamble::~~Preamble ()`

27.232.2.3 `gdcm::Preamble::Preamble (Preamble const &)` `[inline]`

27.232.3 Member Function Documentation

27.232.3.1 `void gdcm::Preamble::Clear ()`

27.232.3.2 `void gdcm::Preamble::Create ()`

27.232.3.3 `const char* gdcm::Preamble::GetInternal () const` `[inline]`

27.232.3.4 `VL gdcm::Preamble::GetLength () const` `[inline]`

27.232.3.5 `bool gdcm::Preamble::IsEmpty () const` `[inline]`

27.232.3.6 `bool gdcm::Preamble::IsValid () const` `[inline],[protected]`

27.232.3.7 `Preamble& gdcm::Preamble::operator= (Preamble const &)` `[inline]`

27.232.3.8 `void gdcm::Preamble::Print (std::ostream & os) const`

27.232.3.9 `std::istream& gdcm::Preamble::Read (std::istream & is)`

27.232.3.10 `void gdcm::Preamble::Remove ()`

27.232.3.11 `void gdcm::Preamble::Valid ()`

27.232.3.12 `std::ostream const& gdcm::Preamble::Write (std::ostream & os) const`

27.232.4 Friends And Related Function Documentation

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

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

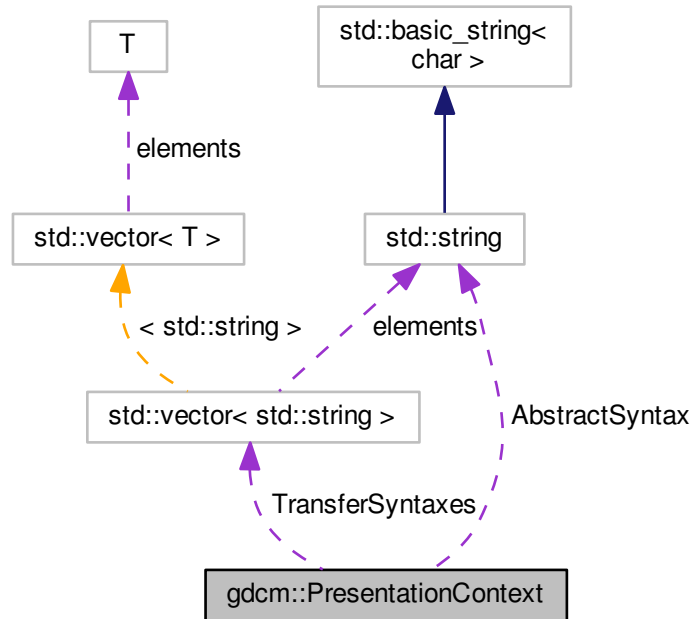
- [gdcmPreamble.h](#)

27.233 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::ImplicitVRLittleEndianDefault](#)↔[TransferSyntaxforDICOM](#))
- 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](#)

27.233.1 Detailed Description

[PresentationContext](#).

See also

[PresentationContextAC](#) [PresentationContextRQ](#)

27.233.2 Member Typedef Documentation

27.233.2.1 `typedef TransferSyntaxArrayType::size_type gdcm::PresentationContext::SizeType`

27.233.2.2 `typedef std::vector<std::string> gdcm::PresentationContext::TransferSyntaxArrayType`

27.233.3 Constructor & Destructor Documentation

27.233.3.1 `gdcm::PresentationContext::PresentationContext ()`

27.233.3.2 `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).

27.233.4 Member Function Documentation

27.233.4.1 `void gdcm::PresentationContext::AddTransferSyntax (const char * tsstr)`

27.233.4.2 `const char* gdcm::PresentationContext::GetAbstractSyntax () const` `[inline]`

27.233.4.3 `SizeType gdcm::PresentationContext::GetNumberOfTransferSyntaxes () const` `[inline]`

27.233.4.4 `uint8_t gdcm::PresentationContext::GetPresentationContextID () const`

27.233.4.5 `const char* gdcm::PresentationContext::GetTransferSyntax (SizeType i) const` `[inline]`

27.233.4.6 `bool gdcm::PresentationContext::operator== (const PresentationContext & pc) const` `[inline]`

References [AbstractSyntax](#), and [TransferSyntaxes](#).

27.233.4.7 `void gdcm::PresentationContext::Print (std::ostream & os) const`

27.233.4.8 `void gdcm::PresentationContext::SetAbstractSyntax (const char * absyn)` `[inline]`

27.233.4.9 void `gdcm::PresentationContext::SetPresentationContextID (uint8_t id)`

27.233.5 Member Data Documentation

27.233.5.1 `std::string gdcm::PresentationContext::AbstractSyntax` [protected]

Referenced by operator==().

27.233.5.2 `uint8_t gdcm::PresentationContext::ID` [protected]

27.233.5.3 `std::vector<std::string> gdcm::PresentationContext::TransferSyntaxes` [protected]

Referenced by operator==().

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

- [gdcmPresentationContext.h](#)

27.234 gdcm::network::PresentationContextAC Class Reference

[PresentationContextAC Table](#) 9-18 PRESENTATION CONTEXT ITEM FIELDS.

```
#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`

27.234.1 Detailed Description

[PresentationContextAC Table](#) 9-18 PRESENTATION CONTEXT ITEM FIELDS.

See also

[PresentationContext](#)

27.234.2 Constructor & Destructor Documentation

27.234.2.1 `gdcm::network::PresentationContextAC::PresentationContextAC ()`

27.234.3 Member Function Documentation

27.234.3.1 `uint8_t gdcm::network::PresentationContextAC::GetPresentationContextID () const` `[inline]`

27.234.3.2 `uint8_t gdcm::network::PresentationContextAC::GetReason () const` `[inline]`

27.234.3.3 `TransferSyntaxSub const& gdcm::network::PresentationContextAC::GetTransferSyntax () const` `[inline]`

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

27.234.3.5 `std::istream& gdcm::network::PresentationContextAC::Read (std::istream & is)`

27.234.3.6 `void gdcm::network::PresentationContextAC::SetPresentationContextID (uint8_t id)`

27.234.3.7 `void gdcm::network::PresentationContextAC::SetReason (uint8_t r)` `[inline]`

27.234.3.8 `void gdcm::network::PresentationContextAC::SetTransferSyntax (TransferSyntaxSub const & ts)`

27.234.3.9 `size_t gdcm::network::PresentationContextAC::Size () const`

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

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

- [gdcmPresentationContextAC.h](#)

27.235 gdcm::PresentationContextGenerator Class Reference

[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.

```
#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

27.235.1 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](#).

27.235.2 Member Typedef Documentation

27.235.2.1 `typedef std::vector<PresentationContext> gdcm::PresentationContextGenerator::PresentationContextArrayType`

27.235.2.2 `typedef PresentationContextArrayType::size_type gdcm::PresentationContextGenerator::SizeType`

27.235.3 Constructor & Destructor Documentation

27.235.3.1 `gdcm::PresentationContextGenerator::PresentationContextGenerator ()`

27.235.4 Member Function Documentation

27.235.4.1 `bool gdcm::PresentationContextGenerator::AddFromFile (const File & file)`

Add a single [PresentationContext](#) from a single [File](#). Call multiple times when dealing with multiple files.

27.235.4.2 `bool gdcm::PresentationContextGenerator::AddPresentationContext (const char * absyn, const char * ts)`
`[protected]`

27.235.4.3 `bool gdcm::PresentationContextGenerator::GenerateFromFilenames (const Directory::FilenamesType & files)`

Generate the [PresentationContext](#) array from a File-Set. [File](#) specified needs to be valid DICOM files. Used for C-ST↔ORE operations

Examples:

[CStoreQtProgress.cxx](#).

27.235.4.4 `bool gdcm::PresentationContextGenerator::GenerateFromUID (UIDs::TSName asname)`

Generate the [PresentationContext](#) array from a UID (eg. VerificationSOPClass)

27.235.4.5 `const char* gdcm::PresentationContextGenerator::GetDefaultTransferSyntax () const` `[protected]`

27.235.4.6 `PresentationContextArrayType const& gdcm::PresentationContextGenerator::GetPresentationContexts ()`
`[inline]`

Examples:

[CStoreQtProgress.cxx](#).

27.235.4.7 `void gdcm::PresentationContextGenerator::SetDefaultTransferSyntax (const TransferSyntax & ts)`

Not implemented for now. GDCM internally uses Implicit Little Endian.

27.235.4.8 `void gdcm::PresentationContextGenerator::SetMergeModeToAbstractSyntax ()`

27.235.4.9 `void gdcm::PresentationContextGenerator::SetMergeModeToTransferSyntax ()`

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

- [gdcmPresentationContextGenerator.h](#)

27.236 gdcm::network::PresentationContextRQ Class Reference

[PresentationContextRQ](#) Table 9-13 PRESENTATION CONTEXT ITEM FIELDS.

```
#include <gdcmPresentationContextRQ.h>
```

Public Types

- `typedef std::vector< TransferSyntaxSub >::size_type SizeType`

Public Member Functions

- [PresentationContextRQ](#) ()
- [PresentationContextRQ](#) ([UIDs::TSName](#) asname, [UIDs::TSName](#) tsname=[UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM](#))
- [PresentationContextRQ](#) (const [PresentationContext](#) &pc)
- void [AddTransferSyntax](#) ([TransferSyntaxSub](#) const &ts)
- [AbstractSyntax](#) const & [GetAbstractSyntax](#) () const
- [AbstractSyntax](#) & [GetAbstractSyntax](#) ()
- [SizeType](#) [GetNumberOfTransferSyntaxes](#) () const
- [uint8_t](#) [GetPresentationContextID](#) () const
- [TransferSyntaxSub](#) const & [GetTransferSyntax](#) ([SizeType](#) i) const
- [TransferSyntaxSub](#) & [GetTransferSyntax](#) ([SizeType](#) i)
- [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

27.236.1 Detailed Description

[PresentationContextRQ](#) Table 9-13 PRESENTATION CONTEXT ITEM FIELDS.

See also

[PresentationContextAC](#)

27.236.2 Member Typedef Documentation

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

27.236.3 Constructor & Destructor Documentation

27.236.3.1 `gdcm::network::PresentationContextRQ::PresentationContextRQ ()`

27.236.3.2 `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).

27.236.3.3 `gdcm::network::PresentationContextRQ::PresentationContextRQ (const PresentationContext & pc)`

27.236.4 Member Function Documentation

27.236.4.1 `void gdcm::network::PresentationContextRQ::AddTransferSyntax (TransferSyntaxSub const & ts)`

- 27.236.4.2 **AbstractSyntax** const& gdcm::network::PresentationContextRQ::GetAbstractSyntax () const [inline]
- 27.236.4.3 **AbstractSyntax&** gdcm::network::PresentationContextRQ::GetAbstractSyntax () [inline]
- 27.236.4.4 **SizeType** gdcm::network::PresentationContextRQ::GetNumberOfTransferSyntaxes () const [inline]
- 27.236.4.5 **uint8_t** gdcm::network::PresentationContextRQ::GetPresentationContextID () const
- 27.236.4.6 **TransferSyntaxSub** const& gdcm::network::PresentationContextRQ::GetTransferSyntax (**SizeType** *i*) const [inline]
- 27.236.4.7 **TransferSyntaxSub&** gdcm::network::PresentationContextRQ::GetTransferSyntax (**SizeType** *i*) [inline]
- 27.236.4.8 **std::vector<TransferSyntaxSub>** const& gdcm::network::PresentationContextRQ::GetTransferSyntaxes () const [inline]
- 27.236.4.9 **bool** gdcm::network::PresentationContextRQ::operator== (const **PresentationContextRQ** & *pc*) const [inline]
- 27.236.4.10 **void** gdcm::network::PresentationContextRQ::Print (std::ostream & *os*) const
- 27.236.4.11 **std::istream&** gdcm::network::PresentationContextRQ::Read (std::istream & *is*)
- 27.236.4.12 **void** gdcm::network::PresentationContextRQ::SetAbstractSyntax (**AbstractSyntax** const & *absyn*)
- 27.236.4.13 **void** gdcm::network::PresentationContextRQ::SetPresentationContextID (**uint8_t** *id*)
- 27.236.4.14 **size_t** gdcm::network::PresentationContextRQ::Size () const
- 27.236.4.15 **const std::ostream&** gdcm::network::PresentationContextRQ::Write (std::ostream & *os*) const

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

- [gdcmPresentationContextRQ.h](#)

27.237 gdcm::network::PresentationDataValue Class Reference

[PresentationDataValue Table](#) 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS.

```
#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)

27.237.1 Detailed Description

[PresentationDataValue](#) Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS.

27.237.2 Constructor & Destructor Documentation

27.237.2.1 `gdcm::network::PresentationDataValue::PresentationDataValue ()`

27.237.3 Member Function Documentation

27.237.3.1 `static DataSet gdcm::network::PresentationDataValue::ConcatenatePDVBlobs (const std::vector< PresentationDataValue > & inPDVs) [static]`

Warning

[DataSet](#) will be read as Implicit Little Endian TS

27.237.3.2 `static DataSet gdcm::network::PresentationDataValue::ConcatenatePDVBlobsAsExplicit (const std::vector< PresentationDataValue > & inPDVs) [static]`

27.237.3.3 `const std::string& gdcm::network::PresentationDataValue::GetBlob () const`

27.237.3.4 `bool gdcm::network::PresentationDataValue::GetIsCommand () const`

27.237.3.5 `bool gdcm::network::PresentationDataValue::GetIsLastFragment () const`

27.237.3.6 `uint8_t gdcm::network::PresentationDataValue::GetMessageHeader () const [inline]`

27.237.3.7 `uint8_t gdcm::network::PresentationDataValue::GetPresentationContextID () const [inline]`

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

27.237.3.9 `std::istream& gdcm::network::PresentationDataValue::Read (std::istream & is)`

27.237.3.10 `std::istream& gdcm::network::PresentationDataValue::ReadInto (std::istream & is, std::ostream & os)`

27.237.3.11 `void gdcm::network::PresentationDataValue::SetBlob (const std::string & partialblob)`

27.237.3.12 `void gdcm::network::PresentationDataValue::SetCommand (bool inCommand)`

27.237.3.13 `void gdcm::network::PresentationDataValue::SetDataSet (const DataSet & ds)`

Set [DataSet](#). Write [DataSet](#) in implicit.

Warning

size of dataset should be below maxpdusize

27.237.3.14 `void gdcm::network::PresentationDataValue::SetLastFragment (bool inLast)`

27.237.3.15 `void gdcm::network::PresentationDataValue::SetMessageHeader (uint8_t messageheader) [inline]`

27.237.3.16 `void gdcm::network::PresentationDataValue::SetPresentationContextID (uint8_t id) [inline]`

27.237.3.17 `size_t gdcm::network::PresentationDataValue::Size () const`

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

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

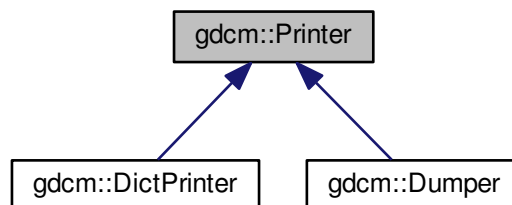
- [gdcmPresentationDataValue.h](#)

27.238 gdcm::Printer Class Reference

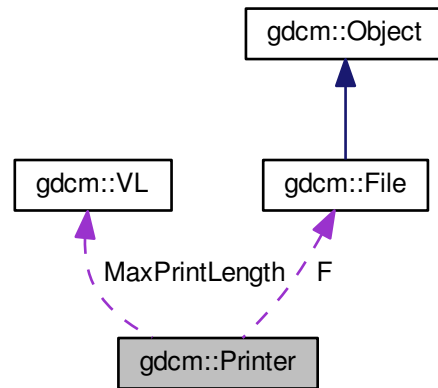
[Printer](#) class.

```
#include <gdcmPrinter.h>
```

Inheritance diagram for `gdcm::Printer`:



Collaboration diagram for `gdcm::Printer`:



Public Types

- enum `PrintStyles` {
`VERBOSE_STYLE` = 0,
`CONDENSED_STYLE`,
`XML` }

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::ostringstream &os, const Dicts &dicts, const DataSet &ds, const DataElement &de, std::ostream &out, std::string const &indent)`

- void [PrintSQ](#) (const [SequenceOfItems](#) *sqi, std::ostream &os, std::string const &indent)

Protected Attributes

- const [File](#) * F
- [VL MaxPrintLength](#)
- [PrintStyles](#) [PrintStyle](#)

27.238.1 Detailed Description

[Printer](#) class.

Examples:

[DumpToshibaDTI.cxx](#).

27.238.2 Member Enumeration Documentation

27.238.2.1 enum gdcm::Printer::PrintStyles

Enumerator

VERBOSE_STYLE

CONDENSED_STYLE

XML

27.238.3 Constructor & Destructor Documentation

27.238.3.1 `gdcm::Printer::Printer ()`

27.238.3.2 `gdcm::Printer::~~Printer ()`

27.238.4 Member Function Documentation

27.238.4.1 `PrintStyles gdcm::Printer::GetPrintStyle () const` `[inline]`

Get [PrintStyle](#) value.

27.238.4.2 `void gdcm::Printer::Print (std::ostream & os)`

Print.

Examples:

[DumpToshibaDTI.cxx](#).

27.238.4.3 **VR** `gdcM::Printer::PrintDataElement (std::ostream & os, const Dicts & dicts, const DataSet & ds, const DataElement & de, std::ostream & out, std::string const & indent)` `[protected]`

27.238.4.4 `void gdcM::Printer::PrintDataSet (const DataSet & ds, std::ostream & os, const std::string & s = " ")`

Print an individual dataset.

27.238.4.5 `void gdcM::Printer::PrintSQ (const SequenceOfItems * sqi, std::ostream & os, std::string const & indent)` `[protected]`

27.238.4.6 `void gdcM::Printer::SetColor (bool c)`

Set color mode or not.

Examples:

[DumpToshibaDTI.cxx](#).

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

Set file.

Examples:

[DumpToshibaDTI.cxx](#).

27.238.4.8 `void gdcM::Printer::SetStyle (PrintStyles ps)` `[inline]`

Set PrintStyle value.

27.238.5 Member Data Documentation

27.238.5.1 `const File* gdcM::Printer::F` `[protected]`

27.238.5.2 `VL gdcM::Printer::MaxPrintLength` `[protected]`

27.238.5.3 `PrintStyles gdcM::Printer::PrintStyle` `[protected]`

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

- [gdcMPrinter.h](#)

27.239 gdcM::PrivateDict Class Reference

Private [Dict](#).

```
#include <gdcMDict.h>
```


Public Member Functions

- [PrivateDict](#) ()
- [~PrivateDict](#) ()
- 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)

27.239.1 Detailed Description

Private [Dict](#).

27.239.2 Constructor & Destructor Documentation

27.239.2.1 `gdcmm::PrivateDict::PrivateDict ()` [\[inline\]](#)

27.239.2.2 `gdcmm::PrivateDict::~~PrivateDict ()` [\[inline\]](#)

27.239.3 Member Function Documentation

27.239.3.1 `void gdcmm::PrivateDict::AddDictEntry (const PrivateTag & tag, const DictEntry & de)` [\[inline\]](#)

References [gdcmm::DictEntry::GetVM\(\)](#), [gdcmm::DictEntry::GetVR\(\)](#), [gdcmm::DictEntry::SetVR\(\)](#), and [gdcmm::VR::UN](#).

27.239.3.2 `bool gdcmm::PrivateDict::FindDictEntry (const PrivateTag & tag) const` [\[inline\]](#)

27.239.3.3 `const DictEntry& gdcmm::PrivateDict::GetDictEntry (const PrivateTag & tag) const` [\[inline\]](#)

27.239.3.4 `bool gdcmm::PrivateDict::IsEmpty () const` [\[inline\]](#)

27.239.3.5 `void gdcmm::PrivateDict::LoadDefault ()` [\[protected\]](#)

27.239.3.6 `void gdcmm::PrivateDict::PrintXML () const` [\[inline\]](#)

References [gdcmm::Tag::GetElement\(\)](#), [gdcmm::Tag::GetGroup\(\)](#), [gdcmm::DictEntry::GetName\(\)](#), [gdcmm::PrivateTag::GetOwner\(\)](#), [gdcmm::DictEntry::GetVM\(\)](#), and [gdcmm::DictEntry::GetVR\(\)](#).

27.239.3.7 `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.

27.239.4 Friends And Related Function Documentation

27.239.4.1 `friend class Dicts [friend]`

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

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

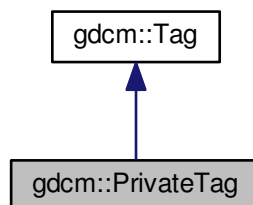
- [gdcmDict.h](#)

27.240 gdcm::PrivateTag Class Reference

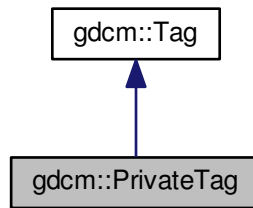
Class to represent a Private DICOM Data [Element](#) ([Attribute](#)) [Tag](#) (Group, [Element](#), Owner)

```
#include <gdcmPrivateTag.h>
```

Inheritance diagram for gdcm::PrivateTag:



Collaboration diagram for gdcm::PrivateTag:



Public Member Functions

- [PrivateTag](#) (uint16_t group=0, uint16_t element=0, const char *owner="")
- [PrivateTag](#) ([Tag](#) const &t, 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)

27.240.1 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](#), [csa2img.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [ELSCINT1WaveToText.cxx](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), [MrProtocol.cxx](#), [pmsct_rgb1.cxx](#), [PublicDict.cxx](#), [ReadGEMSSDO.cxx](#), and [rle2img.cxx](#).

27.240.2 Constructor & Destructor Documentation

27.240.2.1 `gdcm::PrivateTag::PrivateTag (uint16_t group = 0, uint16_t element = 0, const char * owner = " ") [inline]`

27.240.2.2 `gdcm::PrivateTag::PrivateTag (Tag const & t, const char * owner = " ") [inline]`

References `gdcm::Tag::GetElement()`.

27.240.3 Member Function Documentation

27.240.3.1 `DataElement gdcm::PrivateTag::GetAsDataElement () const`

27.240.3.2 `const char* gdcm::PrivateTag::GetOwner () const [inline]`

Examples:

[PublicDict.cxx](#).

Referenced by `gdcm::PrivateDict::PrintXML()`.

27.240.3.3 `bool gdcm::PrivateTag::operator< (const PrivateTag & _val) const`

27.240.3.4 `bool gdcm::PrivateTag::ReadFromCommaSeparatedString (const char * str)`

Read [PrivateTag](#) from a string. [Element](#) number will be truncated to 8bits. Eg: "1234,5678,GDCM" is private tag: (1234,78,"GDCM")

27.240.3.5 `void gdcm::PrivateTag::SetOwner (const char * owner) [inline]`

27.240.4 Friends And Related Function Documentation

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

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

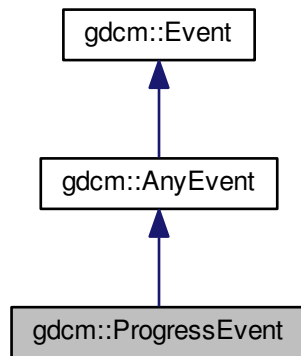
- [gdcmPrivateTag.h](#)

27.241 gdcm::ProgressEvent Class Reference

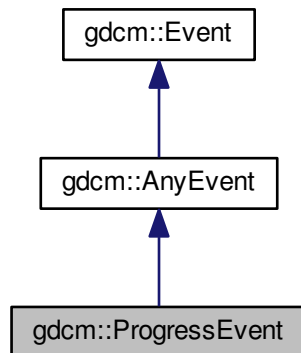
[ProgressEvent](#) Special type of event triggered during.

```
#include <gdcmProgressEvent.h>
```

Inheritance diagram for gdcm::ProgressEvent:



Collaboration diagram for gdcm::ProgressEvent:



Public Types

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

Public Member Functions

- [ProgressEvent](#) (double p=0)
- [ProgressEvent](#) (const [Self](#) &s)

- virtual [~ProgressEvent](#) ()
- virtual bool [CheckEvent](#) (const [::gdcm::Event](#) *e) const
- virtual const char * [GetEventName](#) () const
- double [GetProgress](#) () const
- virtual [::gdcm::Event](#) * [MakeObject](#) () const
- void [SetProgress](#) (double p)

27.241.1 Detailed Description

[ProgressEvent](#) Special type of event triggered during.

See also

[AnyEvent](#)

27.241.2 Member Typedef Documentation

27.241.2.1 typedef [ProgressEvent](#) [gdcm::ProgressEvent::Self](#)

27.241.2.2 typedef [AnyEvent](#) [gdcm::ProgressEvent::Superclass](#)

27.241.3 Constructor & Destructor Documentation

27.241.3.1 [gdcm::ProgressEvent::ProgressEvent](#) (double *p* = 0) [\[inline\]](#)

27.241.3.2 virtual [gdcm::ProgressEvent::~~ProgressEvent](#) () [\[inline\]](#),[\[virtual\]](#)

27.241.3.3 [gdcm::ProgressEvent::ProgressEvent](#) (const [Self](#) & *s*) [\[inline\]](#)

27.241.4 Member Function Documentation

27.241.4.1 virtual bool [gdcm::ProgressEvent::CheckEvent](#) (const [::gdcm::Event](#) * *e*) const [\[inline\]](#),[\[virtual\]](#)

27.241.4.2 virtual const char* [gdcm::ProgressEvent::GetEventName](#) () const [\[inline\]](#),[\[virtual\]](#)

Return the StringName associated with the event.

Implements [gdcm::Event](#).

27.241.4.3 double [gdcm::ProgressEvent::GetProgress](#) () const [\[inline\]](#)

27.241.4.4 virtual [::gdcm::Event](#)* [gdcm::ProgressEvent::MakeObject](#) () const [\[inline\]](#),[\[virtual\]](#)

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

Implements [gdcm::Event](#).

27.241.4.5 void [gdcm::ProgressEvent::SetProgress](#) (double *p*) [\[inline\]](#)

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

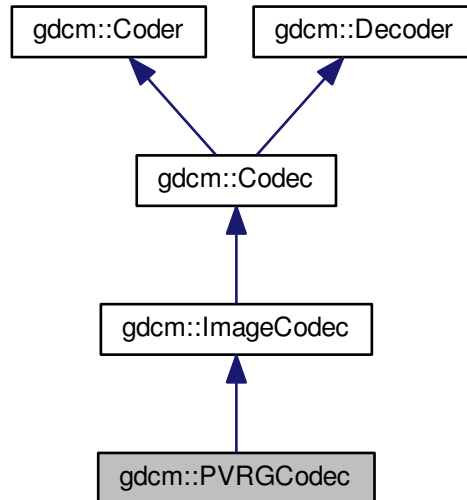
- [gdcmProgressEvent.h](#)

27.242 gdcm::PVRGCodec Class Reference

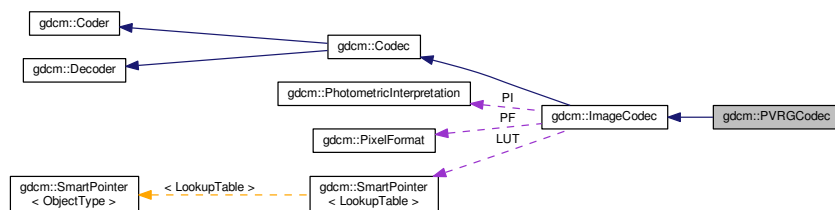
PVRGCodec.

```
#include <gdcmPVRGCodec.h>
```

Inheritance diagram for gdcm::PVRGCodec:



Collaboration diagram for gdcm::PVRGCodec:



Public Member Functions

- [PVRGCodec](#) ()
- [~PVRGCodec](#) ()
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const
Return whether this coder support this transfer syntax (can code it)
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const

Return whether this decoder support this transfer syntax (can decode it)

- virtual [ImageCodec](#) * [Clone](#) () const
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out)
Code.
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os)
Decode.
- void [SetLossyFlag](#) (bool l)

Additional Inherited Members

27.242.1 Detailed Description

[PVRGCodec](#).

Note

pvr 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...

27.242.2 Constructor & Destructor Documentation

27.242.2.1 `gdcm::PVRGCodec::PVRGCodec ()`

27.242.2.2 `gdcm::PVRGCodec::~~PVRGCodec ()`

27.242.3 Member Function Documentation

27.242.3.1 `bool gdcm::PVRGCodec::CanCode (TransferSyntax const &) const` `[virtual]`

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

27.242.3.2 `bool gdcm::PVRGCodec::CanDecode (TransferSyntax const &) const` `[virtual]`

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

27.242.3.3 `virtual ImageCodec* gdcm::PVRGCodec::Clone () const` `[virtual]`

Implements [gdcm::ImageCodec](#).

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

Code.

Reimplemented from [gdcm::Coder](#).

27.242.3.5 `bool gdcm::PVRGCodec::Decode (DataElement const &, DataElement &) [virtual]`

Decode.

Reimplemented from [gdcm::ImageCodec](#).

27.242.3.6 `void gdcm::PVRGCodec::SetLossyFlag (bool l)`

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

- [gdcmPVRGCodec.h](#)

27.243 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 use)

27.243.1 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.

27.243.2 Constructor & Destructor Documentation

27.243.2.1 `gdcm::PythonFilter::PythonFilter ()`

27.243.2.2 `gdcm::PythonFilter::~~PythonFilter ()`

27.243.3 Member Function Documentation

- 27.243.3.1 `File& gdcM::PythonFilter::GetFile () [inline]`
- 27.243.3.2 `const File& gdcM::PythonFilter::GetFile () const [inline]`
- 27.243.3.3 `void gdcM::PythonFilter::SetDicts (const Dicts & dicts)`
- 27.243.3.4 `void gdcM::PythonFilter::SetFile (const File & f) [inline]`
- 27.243.3.5 `PyObject* gdcM::PythonFilter::ToPyObject (const Tag & t) const`
- 27.243.3.6 `void gdcM::PythonFilter::UseDictAlways (bool use) [inline]`

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

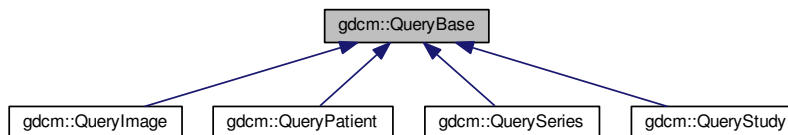
- [gdcMPythonFilter.h](#)

27.244 gdcM::QueryBase Class Reference

[QueryBase](#) contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE.

```
#include <gdcMQueryBase.h>
```

Inheritance diagram for gdcM::QueryBase:



Public Member Functions

- virtual `~QueryBase ()`
- `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`

27.244.1 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.

27.244.2 Constructor & Destructor Documentation

27.244.2.1 `virtual gdcm::QueryBase::~~QueryBase () [inline],[virtual]`

27.244.3 Member Function Documentation

27.244.3.1 `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

27.244.3.2 `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

27.244.3.3 `virtual std::vector<Tag> gdcm::QueryBase::GetHierachicalSearchTags (const ERootType & inRootType) const [pure virtual]`

Return all Unique Key for a particular Query Root type (from the same level and above).

Implemented in [gdcm::QueryImage](#), [gdcm::QueryPatient](#), [gdcm::QuerySeries](#), and [gdcm::QueryStudy](#).

27.244.3.4 `virtual const char* gdcm::QueryBase::GetName () const [pure virtual]`

Implemented in [gdcm::QueryImage](#), [gdcm::QueryPatient](#), [gdcm::QuerySeries](#), and [gdcm::QueryStudy](#).

27.244.3.5 `virtual std::vector<Tag> gdcm::QueryBase::GetOptionalTags (const ERootType & inRootType) const [pure virtual]`

Implemented in [gdcm::QueryImage](#), [gdcm::QueryPatient](#), [gdcm::QuerySeries](#), and [gdcm::QueryStudy](#).

27.244.3.6 `virtual DataElement gdcM::QueryBase::GetQueryLevel () const [pure virtual]`

Implemented in [gdcM::QueryImage](#), [gdcM::QueryPatient](#), [gdcM::QuerySeries](#), and [gdcM::QueryStudy](#).

27.244.3.7 `virtual std::vector<Tag> gdcM::QueryBase::GetRequiredTags (const ERootType & inRootType) const [pure virtual]`

Implemented in [gdcM::QueryImage](#), [gdcM::QueryPatient](#), [gdcM::QuerySeries](#), and [gdcM::QueryStudy](#).

27.244.3.8 `virtual std::vector<Tag> gdcM::QueryBase::GetUniqueTags (const ERootType & inRootType) const [pure virtual]`

Implemented in [gdcM::QueryImage](#), [gdcM::QueryPatient](#), [gdcM::QuerySeries](#), and [gdcM::QueryStudy](#).

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

- [gdcMQueryBase.h](#)

27.245 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)

27.245.1 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

27.245.2 Member Function Documentation

27.245.2.1 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()).

27.245.2.2 static void gdcm::QueryFactory::ListCharSets (std::ostream & os) [static]

List all possible CharSet.

27.245.2.3 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

27.245.2.4 static **BaseQuery*** gdcm::QueryFactory::ProduceQuery (const std::string & sopInstanceUID, ENQueryType inQueryType) [static]

27.245.2.5 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:

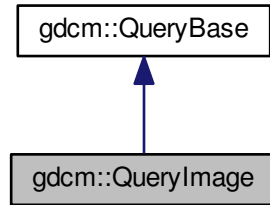
- [gdcmQueryFactory.h](#)

27.246 gdcm::QueryImage Class Reference

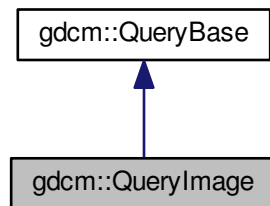
QueryImage contains: class to construct an image-based query for C-FIND and C-MOVE.

```
#include <gdcmQueryImage.h>
```

Inheritance diagram for `gdcm::QueryImage`:



Collaboration diagram for `gdcm::QueryImage`:



Public Member Functions

- `std::vector< Tag > GetHierachicalSearchTags` (const `ERootType` &inRootType) const
Return all Unique Key for a particular Query Root type (from the same level and above).
- `const char * GetName` () const
- `std::vector< Tag > GetOptionalTags` (const `ERootType` &inRootType) const
- `DataElement GetQueryLevel` () const
- `std::vector< Tag > GetRequiredTags` (const `ERootType` &inRootType) const
- `std::vector< Tag > GetUniqueTags` (const `ERootType` &inRootType) const

27.246.1 Detailed Description

`QueryImage` contains: class to construct an image-based query for C-FIND and C-MOVE.

27.246.2 Member Function Documentation

27.246.2.1 `std::vector<Tag> gdcm::QueryImage::GetHierachicalSearchTags (const ERootType & inRootType) const` [virtual]

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements [gdcm::QueryBase](#).

27.246.2.2 `const char* gdcm::QueryImage::GetName () const` [virtual]

Implements [gdcm::QueryBase](#).

27.246.2.3 `std::vector<Tag> gdcm::QueryImage::GetOptionalTags (const ERootType & inRootType) const` [virtual]

Implements [gdcm::QueryBase](#).

27.246.2.4 `DataElement gdcm::QueryImage::GetQueryLevel () const` [virtual]

Implements [gdcm::QueryBase](#).

27.246.2.5 `std::vector<Tag> gdcm::QueryImage::GetRequiredTags (const ERootType & inRootType) const` [virtual]

Implements [gdcm::QueryBase](#).

27.246.2.6 `std::vector<Tag> gdcm::QueryImage::GetUniqueTags (const ERootType & inRootType) const` [virtual]

Implements [gdcm::QueryBase](#).

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

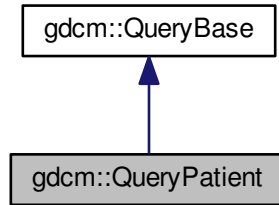
- [gdcmQueryImage.h](#)

27.247 gdcm::QueryPatient Class Reference

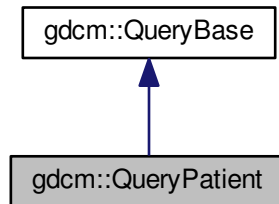
[QueryPatient](#) contains: class to construct a patient-based query for c-find and c-move.

```
#include <gdcmQueryPatient.h>
```

Inheritance diagram for `gdcm::QueryPatient`:



Collaboration diagram for `gdcm::QueryPatient`:



Public Member Functions

- `std::vector< Tag > GetHierachicalSearchTags (const ERootType &inRootType) const`
Return all Unique Key for a particular Query Root type (from the same level and above).
- `const char * GetName () const`
- `std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const`
- `DataElement GetQueryLevel () const`
- `std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const`
- `std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const`

27.247.1 Detailed Description

`QueryPatient` contains: class to construct a patient-based query for c-find and c-move.

27.247.2 Member Function Documentation

27.247.2.1 `std::vector<Tag> gdcm::QueryPatient::GetHierachicalSearchTags (const ERootType & inRootType) const` [virtual]

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements [gdcm::QueryBase](#).

27.247.2.2 `const char* gdcm::QueryPatient::GetName () const` [virtual]

Implements [gdcm::QueryBase](#).

27.247.2.3 `std::vector<Tag> gdcm::QueryPatient::GetOptionalTags (const ERootType & inRootType) const` [virtual]

Implements [gdcm::QueryBase](#).

27.247.2.4 `DataElement gdcm::QueryPatient::GetQueryLevel () const` [virtual]

Implements [gdcm::QueryBase](#).

27.247.2.5 `std::vector<Tag> gdcm::QueryPatient::GetRequiredTags (const ERootType & inRootType) const` [virtual]

Implements [gdcm::QueryBase](#).

27.247.2.6 `std::vector<Tag> gdcm::QueryPatient::GetUniqueTags (const ERootType & inRootType) const` [virtual]

Implements [gdcm::QueryBase](#).

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

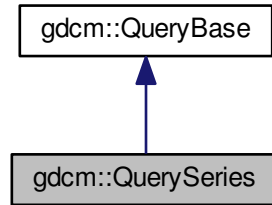
- [gdcmQueryPatient.h](#)

27.248 gdcm::QuerySeries Class Reference

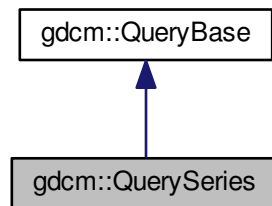
[QuerySeries](#) contains: class to construct a series-based query for c-find and c-move.

```
#include <gdcmQuerySeries.h>
```

Inheritance diagram for `gdcm::QuerySeries`:



Collaboration diagram for `gdcm::QuerySeries`:



Public Member Functions

- `std::vector< Tag > GetHierachicalSearchTags` (const `ERootType` &inRootType) const
Return all Unique Key for a particular Query Root type (from the same level and above).
- `const char * GetName` () const
- `std::vector< Tag > GetOptionalTags` (const `ERootType` &inRootType) const
- `DataElement GetQueryLevel` () const
- `std::vector< Tag > GetRequiredTags` (const `ERootType` &inRootType) const
- `std::vector< Tag > GetUniqueTags` (const `ERootType` &inRootType) const

27.248.1 Detailed Description

`QuerySeries` contains: class to construct a series-based query for c-find and c-move.

27.248.2 Member Function Documentation

27.248.2.1 `std::vector<Tag> gdcm::QuerySeries::GetHierachicalSearchTags (const ERootType & inRootType) const` [virtual]

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements [gdcm::QueryBase](#).

27.248.2.2 `const char* gdcm::QuerySeries::GetName () const` [virtual]

Implements [gdcm::QueryBase](#).

27.248.2.3 `std::vector<Tag> gdcm::QuerySeries::GetOptionalTags (const ERootType & inRootType) const` [virtual]

Implements [gdcm::QueryBase](#).

27.248.2.4 `DataElement gdcm::QuerySeries::GetQueryLevel () const` [virtual]

Implements [gdcm::QueryBase](#).

27.248.2.5 `std::vector<Tag> gdcm::QuerySeries::GetRequiredTags (const ERootType & inRootType) const` [virtual]

Implements [gdcm::QueryBase](#).

27.248.2.6 `std::vector<Tag> gdcm::QuerySeries::GetUniqueTags (const ERootType & inRootType) const` [virtual]

Implements [gdcm::QueryBase](#).

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

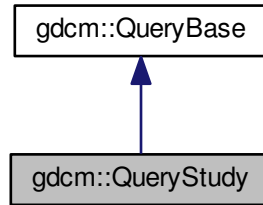
- [gdcmQuerySeries.h](#)

27.249 gdcm::QueryStudy Class Reference

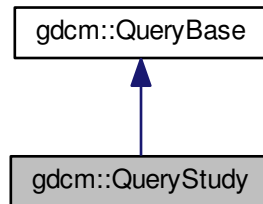
QueryStudy.h contains: class to construct a study-based query for C-FIND and C-MOVE.

```
#include <gdcmQueryStudy.h>
```

Inheritance diagram for `gdcm::QueryStudy`:



Collaboration diagram for `gdcm::QueryStudy`:



Public Member Functions

- `std::vector< Tag > GetHierachicalSearchTags` (const `ERootType` &`inRootType`) const
Return all Unique Key for a particular Query Root type (from the same level and above).
- `const char * GetName` () const
- `std::vector< Tag > GetOptionalTags` (const `ERootType` &`inRootType`) const
- `DataElement GetQueryLevel` () const
- `std::vector< Tag > GetRequiredTags` (const `ERootType` &`inRootType`) const
- `std::vector< Tag > GetUniqueTags` (const `ERootType` &`inRootType`) const

27.249.1 Detailed Description

`QueryStudy.h` contains: class to construct a study-based query for C-FIND and C-MOVE.

27.249.2 Member Function Documentation

27.249.2.1 `std::vector<Tag> gdcm::QueryStudy::GetHierachicalSearchTags (const ERootType & inRootType) const` [virtual]

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements [gdcm::QueryBase](#).

27.249.2.2 `const char* gdcm::QueryStudy::GetName () const` [virtual]

Implements [gdcm::QueryBase](#).

27.249.2.3 `std::vector<Tag> gdcm::QueryStudy::GetOptionalTags (const ERootType & inRootType) const` [virtual]

Implements [gdcm::QueryBase](#).

27.249.2.4 `DataElement gdcm::QueryStudy::GetQueryLevel () const` [virtual]

Implements [gdcm::QueryBase](#).

27.249.2.5 `std::vector<Tag> gdcm::QueryStudy::GetRequiredTags (const ERootType & inRootType) const` [virtual]

Implements [gdcm::QueryBase](#).

27.249.2.6 `std::vector<Tag> gdcm::QueryStudy::GetUniqueTags (const ERootType & inRootType) const` [virtual]

Implements [gdcm::QueryBase](#).

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

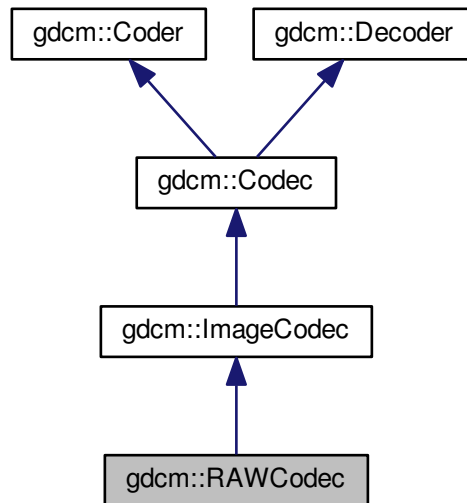
- [gdcmQueryStudy.h](#)

27.250 gdcm::RAWCodec Class Reference

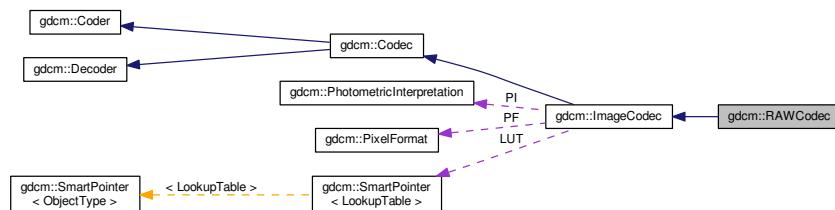
[RAWCodec](#) class.

```
#include <gdcmRAWCodec.h>
```

Inheritance diagram for `gdcm::RAWCodec`:



Collaboration diagram for `gdcm::RAWCodec`:



Public Member Functions

- [RAWCodec](#) ()
- [~RAWCodec](#) ()
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const
Return whether this coder support this transfer syntax (can code it)
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const
Return whether this decoder support this transfer syntax (can decode it)
- virtual [ImageCodec](#) * [Clone](#) () const
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out)
Code.
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os)

Decode.

- bool [DecodeBytes](#) (const char *inBytes, size_t inBufferLength, char *outBytes, size_t inOutBufferLength)
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts)

Protected Member Functions

- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os)

Additional Inherited Members

27.250.1 Detailed Description

[RAWCodec](#) class.

27.250.2 Constructor & Destructor Documentation

27.250.2.1 `gdcm::RAWCodec::RAWCodec ()`

27.250.2.2 `gdcm::RAWCodec::~~RAWCodec ()`

27.250.3 Member Function Documentation

27.250.3.1 `bool gdcm::RAWCodec::CanCode (TransferSyntax const &) const` `[virtual]`

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

27.250.3.2 `bool gdcm::RAWCodec::CanDecode (TransferSyntax const &) const` `[virtual]`

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

27.250.3.3 `virtual ImageCodec* gdcm::RAWCodec::Clone () const` `[virtual]`

Implements [gdcm::ImageCodec](#).

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

Code.

Reimplemented from [gdcm::Coder](#).

27.250.3.5 `bool gdcm::RAWCodec::Decode (DataElement const &, DataElement &)` `[virtual]`

Decode.

Reimplemented from [gdcm::ImageCodec](#).

27.250.3.6 `bool gdcm::RAWCodec::DecodeByStreams (std::istream & is, std::ostream & os)` `[protected]`, `[virtual]`

Reimplemented from [gdcm::ImageCodec](#).

27.250.3.7 `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.

27.250.3.8 `bool gdcm::RAWCodec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` `[virtual]`

Reimplemented from [gdcm::ImageCodec](#).

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

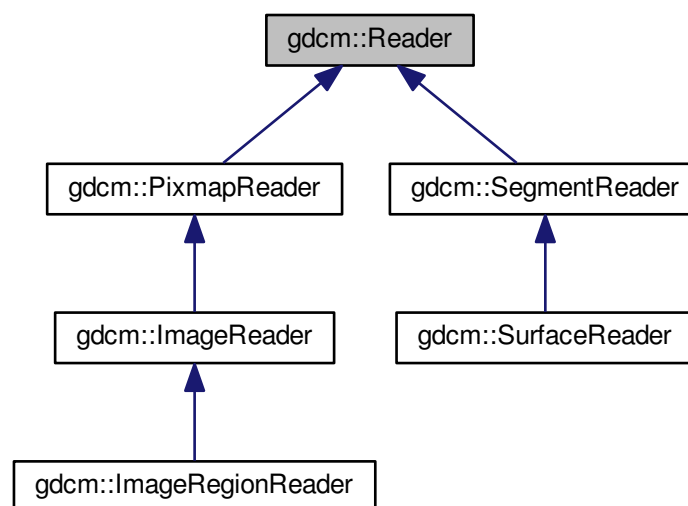
- [gdcmRAWCodec.h](#)

27.251 gdcm::Reader Class Reference

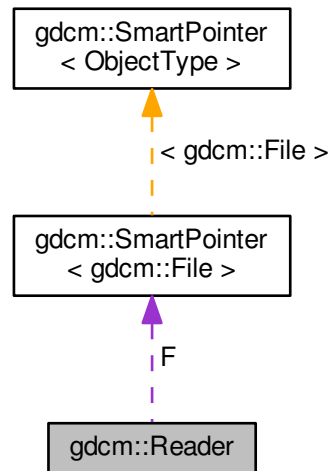
[Reader](#) ala DOM (Document [Object](#) Model)

```
#include <gdcmReader.h>
```

Inheritance diagram for `gdcm::Reader`:



Collaboration diagram for gdcm::Reader:



Public Member Functions

- [Reader](#) ()
- virtual [~Reader](#) ()
- bool [CanRead](#) () const
- const [File](#) & [GetFile](#) () const
Set/Get File.
- [File](#) & [GetFile](#) ()
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](#)

27.251.1 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](#), [csa2img.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [FixBrokenJ2K.cxx](#), [FixOrientation.cxx](#), [gdcmrtonplan.cxx](#), [gdcmrplan.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [ReadUTF8QtDir.cxx](#), [rle2img.cxx](#), and [TestReader.cxx](#).

27.251.2 Constructor & Destructor Documentation

27.251.2.1 `gdcm::Reader::Reader ()`

27.251.2.2 `virtual gdcm::Reader::~~Reader ()` `[virtual]`

27.251.3 Member Function Documentation

27.251.3.1 `bool gdcm::Reader::CanRead () const`

Test whether this is a DICOM file

Warning

need to call either `SetFileName` or `SetStream` first

Examples:

[ReadUTF8QtDir.cxx](#).

27.251.3.2 `const File& gdcm::Reader::GetFile () const` `[inline]`

Set/Get [File](#).

Examples:

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CompressImage.cxx](#), [CreateFakeRTDOSE.cxx](#), [csa2img.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [gdcmrtonplan.cxx](#), [gdcmrtpplan.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [rle2img.cxx](#), and [TestReader.cxx](#).

27.251.3.3 `File& gdcm::Reader::GetFile ()` `[inline]`

Set/Get [File](#).

27.251.3.4 `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++

27.251.3.5 `std::istream* gdcm::Reader::GetStreamPtr () const` `[inline]`, `[protected]`

27.251.3.6 `virtual bool gdcm::Reader::Read ()` `[virtual]`

Main function to read a file.

Reimplemented in [gdcm::ImageRegionReader](#), [gdcm::PixmapReader](#), [gdcm::ImageReader](#), [gdcm::SegmentReader](#), and [gdcm::SurfaceReader](#).

Examples:

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CreateFakeRTDOSE.cxx](#), [csa2img.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [FixBrokenJ2K.cxx](#), [FixOrientation.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [rle2img.cxx](#), and [TestReader.cxx](#).

27.251.3.7 `bool gdcm::Reader::ReadDataSet ()` [protected]

27.251.3.8 `bool gdcm::Reader::ReadMetaInformation ()` [protected]

27.251.3.9 `bool gdcm::Reader::ReadPreamble ()` [protected]

27.251.3.10 `bool gdcm::Reader::ReadSelectedPrivateTags (std::set< PrivateTag > const & ptags, bool readvalues = true)`

Will only read the specified selected private tags.

27.251.3.11 `bool gdcm::Reader::ReadSelectedTags (std::set< Tag > const & tags, bool readvalues = true)`

Will only read the specified selected tags.

27.251.3.12 `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>	

27.251.3.13 `void gdcm::Reader::SetFile (File & file)` [inline]

Set/Get [File](#).

27.251.3.14 `void gdcm::Reader::SetFileName (const char * filename_native)`

Set the filename to open. This will create a `std::ifstream` internally See [SetStream](#) if you are dealing with different `std::istream` object

Examples:

[BasicImageAnonymizer.cs](#), [ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CheckBigEndianBug.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CompressImage.cxx](#), [ConvertToQImage.cxx](#), [CreateFakeRTDOSE.cxx](#), [csa2img.cxx](#),

[DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [ReadMultiTimesException.cxx](#), [ReadUTF8QtDir.cxx](#), [rle2img.cxx](#), [SimplePrintPatientName.cs](#), [TestReader.cxx](#), and [threadgdcm.cxx](#).

27.251.3.15 `void gdcm::Reader::SetStream (std::istream & input_stream)` `[inline]`

Set the open-ed stream directly.

Examples:

[DumpToshibaDTI.cxx](#), and [ReadUTF8QtDir.cxx](#).

27.251.4 Friends And Related Function Documentation

27.251.4.1 `friend class StreamImageReader` `[friend]`

27.251.5 Member Data Documentation

27.251.5.1 `SmartPointer<File> gdcm::Reader::F` `[protected]`

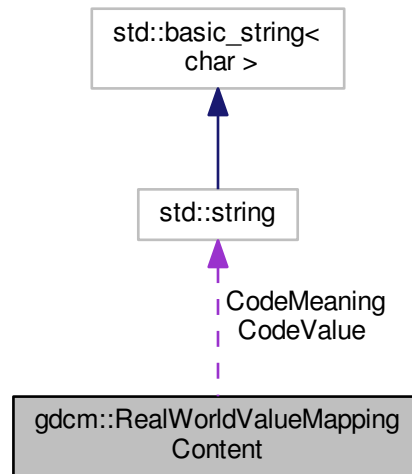
The documentation for this class was generated from the following file:

- [gdcmReader.h](#)

27.252 gdcm::RealWorldValueMappingContent Struct Reference

```
#include <gdcmImageHelper.h>
```

Collaboration diagram for `gdcm::RealWorldValueMappingContent`:



Public Attributes

- `std::string` [CodeMeaning](#)
- `std::string` [CodeValue](#)
- `double` [RealWorldValueIntercept](#)
- `double` [RealWorldValueSlope](#)

27.252.1 Member Data Documentation

27.252.1.1 `std::string` `gdcm::RealWorldValueMappingContent::CodeMeaning`

27.252.1.2 `std::string` `gdcm::RealWorldValueMappingContent::CodeValue`

27.252.1.3 `double` `gdcm::RealWorldValueMappingContent::RealWorldValueIntercept`

27.252.1.4 `double` `gdcm::RealWorldValueMappingContent::RealWorldValueSlope`

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

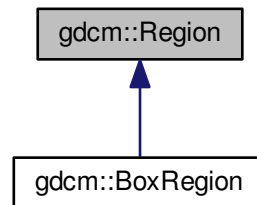
- [gdcmImageHelper.h](#)

27.253 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.

27.253.1 Detailed Description

Class for manipulation region.

27.253.2 Constructor & Destructor Documentation

27.253.2.1 `gdcm::Region::Region ()`

27.253.2.2 `virtual gdcm::Region::~~Region ()` [virtual]

27.253.3 Member Function Documentation

27.253.3.1 `virtual size_t gdcm::Region::Area () const` [pure virtual]

compute the area

Implemented in [gdcm::BoxRegion](#).

27.253.3.2 `virtual Region* gdcm::Region::Clone () const [pure virtual]`

Implemented in [gdcm::BoxRegion](#).

27.253.3.3 `virtual BoxRegion gdcm::Region::ComputeBoundingBox () [pure virtual]`

Return the Axis-Aligned minimum bounding box for all regions.

Implemented in [gdcm::BoxRegion](#).

27.253.3.4 `virtual bool gdcm::Region::Empty () const [pure virtual]`

return whether this domain is empty:

Implemented in [gdcm::BoxRegion](#).

27.253.3.5 `virtual bool gdcm::Region::IsValid () const [pure virtual]`

return whether this is valid domain

Implemented in [gdcm::BoxRegion](#).

27.253.3.6 `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](#)

27.254 gdcm::Rescaler Class Reference

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.

```
#include <gdcmRescaler.h>
```

Public Member Functions

- [Rescaler \(\)](#)

- [~Rescaler](#) ()
- [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)

27.254.1 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 $V \leftrightarrow R:DS$ 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 ouput, 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](#)

27.254.2 Constructor & Destructor Documentation

27.254.2.1 `gdcm::Rescaler::Rescaler ()` `[inline]`

27.254.2.2 `gdcm::Rescaler::~~Rescaler ()` `[inline]`

27.254.3 Member Function Documentation

27.254.3.1 `PixelFormat::ScalarType gdcm::Rescaler::ComputeInterceptSlopePixelType ()`

Compute the Pixel Format of the output data Used for direct transformation

27.254.3.2 `PixelFormat gdcm::Rescaler::ComputePixelTypeFromMinMax ()`

Compute the Pixel Format of the output data Used for inverse transformation

27.254.3.3 `double gdcm::Rescaler::GetIntercept () const` `[inline]`

27.254.3.4 `double gdcm::Rescaler::GetSlope () const` `[inline]`

27.254.3.5 `bool gdcm::Rescaler::InverseRescale (char * out, const char * in, size_t n)`

Inverse transform.

27.254.3.6 `template<typename TIn > void gdcm::Rescaler::InverseRescaleFunctionIntoBestFit (char * out, const TIn * in, size_t n)` `[protected]`

27.254.3.7 `bool gdcm::Rescaler::Rescale (char * out, const char * in, size_t n)`

Direct transform.

27.254.3.8 `template<typename TIn > void gdcm::Rescaler::RescaleFunctionIntoBestFit (char * out, const TIn * in, size_t n)` `[protected]`

27.254.3.9 `void gdcm::Rescaler::SetIntercept (double i)` `[inline]`

Set Intercept: used for both direct&inverse transformation.

27.254.3.10 void gdcm::Rescaler::SetMinMaxForPixelType (double *min*, double *max*) [inline]

Set target interval for output data. A best match will be computed (if possible) Used for inverse transformation

27.254.3.11 void gdcm::Rescaler::SetPixelFormat (PixelFormat const & *pf*) [inline]

Set Pixel Format of input data.

27.254.3.12 void gdcm::Rescaler::SetSlope (double *s*) [inline]

Set Slope: user for both direct&inverse transformation.

27.254.3.13 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](#)

27.254.3.14 void gdcm::Rescaler::SetUseTargetPixelType (bool *b*)

Override default behavior of Rescale.

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

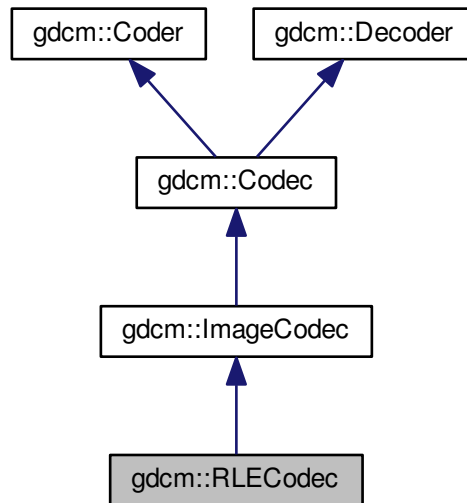
- [gdcmRescaler.h](#)

27.255 gdcm::RLECodec Class Reference

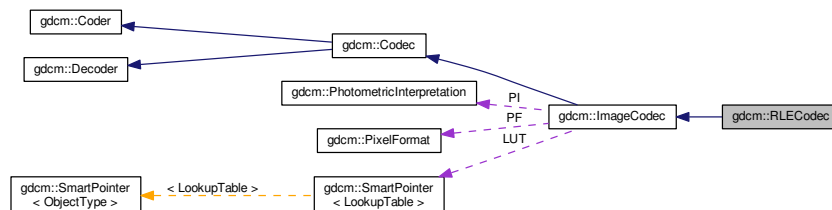
Class to do RLE.

```
#include <gdcmRLECodec.h>
```

Inheritance diagram for `gdcm::RLECodec`:



Collaboration diagram for `gdcm::RLECodec`:



Public Member Functions

- [RLECodec](#) ()
- [~RLECodec](#) ()
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const
Return whether this coder support this transfer syntax (can code it)
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const
Return whether this decoder support this transfer syntax (can decode it)
- virtual [ImageCodec](#) * [Clone](#) () const
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out)
Code.
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os)

Decode.

- unsigned long [GetBufferLength](#) () const
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts)
- void [SetBufferLength](#) (unsigned long l)
- void [SetLength](#) (unsigned long l)

Protected Member Functions

- bool [AppendFrameEncode](#) (std::ostream &out, const char *data, size_t datalen)
- bool [AppendRowEncode](#) (std::ostream &out, const char *data, size_t datalen)
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os)
- 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](#) ()
- bool [IsRowEncoder](#) ()
- bool [StartEncode](#) (std::ostream &)
- bool [StopEncode](#) (std::ostream &)

Friends

- class [ImageRegionReader](#)

Additional Inherited Members

27.255.1 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).

27.255.2 Constructor & Destructor Documentation

27.255.2.1 [gdcm::RLECodec::RLECodec](#) ()

27.255.2.2 [gdcm::RLECodec::~~RLECodec](#) ()

27.255.3 Member Function Documentation

27.255.3.1 [bool gdcm::RLECodec::AppendFrameEncode](#) ([std::ostream & out](#), [const char * data](#), [size_t datalen](#))
[protected], [virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.255.3.2 `bool gdcmm::RLECodec::AppendRowEncode (std::ostream & out, const char * data, size_t datalen)`
[protected], [virtual]

Reimplemented from [gdcmm::ImageCodec](#).

27.255.3.3 `bool gdcmm::RLECodec::CanCode (TransferSyntax const &) const` [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcmm::ImageCodec](#).

27.255.3.4 `bool gdcmm::RLECodec::CanDecode (TransferSyntax const &) const` [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcmm::ImageCodec](#).

27.255.3.5 `virtual ImageCodec* gdcmm::RLECodec::Clone () const` [virtual]

Implements [gdcmm::ImageCodec](#).

27.255.3.6 `bool gdcmm::RLECodec::Code (DataElement const & in_, DataElement & out_)` [virtual]

Code.

Reimplemented from [gdcmm::Coder](#).

27.255.3.7 `bool gdcmm::RLECodec::Decode (DataElement const & , DataElement &)` [virtual]

Decode.

Reimplemented from [gdcmm::ImageCodec](#).

27.255.3.8 `bool gdcmm::RLECodec::DecodeByStreams (std::istream & is, std::ostream & os)` [protected], [virtual]

Reimplemented from [gdcmm::ImageCodec](#).

27.255.3.9 `bool gdcmm::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]

27.255.3.10 `unsigned long gdcmm::RLECodec::GetBufferLength () const` [inline]

27.255.3.11 `bool gdcmm::RLECodec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` [virtual]

Reimplemented from [gdcmm::ImageCodec](#).

27.255.3.12 `bool gdcmm::RLECodec::IsFrameEncoder ()` [protected], [virtual]

Reimplemented from [gdcmm::ImageCodec](#).

27.255.3.13 `bool gdcm::RLECodec::IsRowEncoder () [protected], [virtual]`

Reimplemented from [gdcm::ImageCodec](#).

27.255.3.14 `void gdcm::RLECodec::SetBufferLength (unsigned long l) [inline]`

27.255.3.15 `void gdcm::RLECodec::SetLength (unsigned long l) [inline]`

27.255.3.16 `bool gdcm::RLECodec::StartEncode (std::ostream &) [protected], [virtual]`

Reimplemented from [gdcm::ImageCodec](#).

27.255.3.17 `bool gdcm::RLECodec::StopEncode (std::ostream &) [protected], [virtual]`

Reimplemented from [gdcm::ImageCodec](#).

27.255.4 Friends And Related Function Documentation

27.255.4.1 `friend class ImageRegionReader [friend]`

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

- [gdcmRLECodec.h](#)

27.256 gdcm::network::RoleSelectionSub Class Reference

[RoleSelectionSub](#) PS 3.7 [Table D.3-9 SCP/SCU ROLE SELECTION SUB-ITEM FIELDS \(A-ASSOCIATE-RQ\)](#)

```
#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

27.256.1 Detailed Description

[RoleSelectionSub](#) PS 3.7 [Table D.3-9 SCP/SCU ROLE SELECTION SUB-ITEM FIELDS \(A-ASSOCIATE-RQ\)](#)

27.256.2 Constructor & Destructor Documentation

27.256.2.1 `gdcmm::network::RoleSelectionSub::RoleSelectionSub ()`

27.256.3 Member Function Documentation

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

27.256.3.2 `std::istream& gdcmm::network::RoleSelectionSub::Read (std::istream & is)`

27.256.3.3 `void gdcmm::network::RoleSelectionSub::SetTuple (const char * uid, uint8_t scurole, uint8_t scprole)`

27.256.3.4 `size_t gdcmm::network::RoleSelectionSub::Size () const`

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

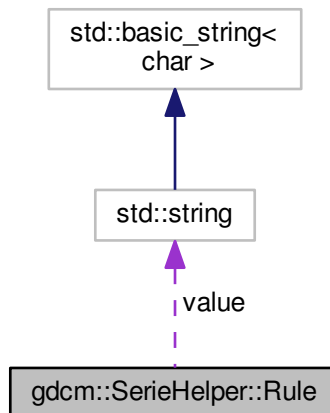
The documentation for this class was generated from the following file:

- [gdcmmRoleSelectionSub.h](#)

27.257 gdcmm::SerieHelper::Rule Struct Reference

```
#include <gdcmmSerieHelper.h>
```

Collaboration diagram for `gdcmm::SerieHelper::Rule`:



Public Attributes

- `uint16_t elem`
- `uint16_t group`

- int [op](#)
- std::string [value](#)

27.257.1 Member Data Documentation

27.257.1.1 uint16_t gdcm::SerieHelper::Rule::elem

27.257.1.2 uint16_t gdcm::SerieHelper::Rule::group

27.257.1.3 int gdcm::SerieHelper::Rule::op

27.257.1.4 std::string gdcm::SerieHelper::Rule::value

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

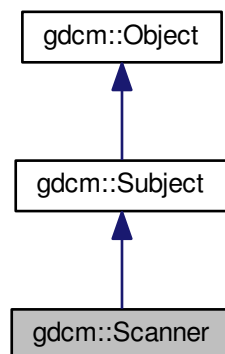
- [gdcmSerieHelper.h](#)

27.258 gdcm::Scanner Class Reference

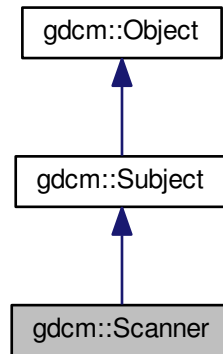
[Scanner](#) This filter is meant for quickly browsing a [FileSet](#) (a set of files on disk). Special consideration are taken so as to read the minimum amount of information in each file in order to retrieve the user specified set of DICOM [Attribute](#).

```
#include <gdcmScanner.h>
```

Inheritance diagram for gdcm::Scanner:



Collaboration diagram for `gdcm::Scanner`:



Classes

- struct [Itstr](#)

Public Types

- typedef `MappingType::const_iterator` [ConstIterator](#)
- typedef `std::map< const char *, TagToValue, Itstr >` [MappingType](#)
- typedef `std::map< Tag, const char * >` [TagToValue](#)
- typedef `TagToValue::value_type` [TagToValueValueType](#)
- typedef `std::set< std::string >` [ValuesType](#)

Public Member Functions

- [Scanner](#) ()
- [~Scanner](#) ()
- void [AddPrivateTag](#) ([PrivateTag](#) const &t)
- void [AddSkipTag](#) ([Tag](#) const &t)
Add a tag that will need to be skipped. Those are root level skip tags.
- void [AddTag](#) ([Tag](#) const &t)
Add a tag that will need to be read. Those are root level tags.
- [ConstIterator](#) [Begin](#) () const
- void [ClearSkipTags](#) ()
- void [ClearTags](#) ()
- [ConstIterator](#) [End](#) () const
- [Directory::FilenameType](#) [GetAllFileNamesFromTagToValue](#) ([Tag](#) const &t, const char *valueref) const
- const char * [GetFilenameFromTagToValue](#) ([Tag](#) const &t, const char *valueref) const
- [Directory::FilenameType](#) const & [GetFileNames](#) () const

- [Directory::FilenamesType GetKeys](#) () const
- [TagToValue](#) const & [GetMapping](#) (const char *filename) const
Get the std::map mapping filenames to value for file 'filename'.
- [TagToValue](#) const & [GetMappingFromTagToValue](#) ([Tag](#) const &t, const char *value) const
See [GetFilenameFromTagToValue\(\)](#). This is simply [GetFilenameFromTagToValue](#) followed.
- [MappingType](#) const & [GetMappings](#) () const
Mappings are the mapping from a particular tag to the map, mapping filename to value:
- [Directory::FilenamesType GetOrderedValues](#) ([Tag](#) const &t) const
- const char * [GetValue](#) (const char *filename, [Tag](#) const &t) const
- [ValuesType](#) const & [GetValues](#) () const
Get all the values found (in lexicographic order)
- [ValuesType GetValues](#) ([Tag](#) const &t) const
Get all the values found (in lexicographic order) associated with [Tag](#) 't'.
- bool [IsKey](#) (const char *filename) const
- void [Print](#) (std::ostream &os) const
Print result.
- bool [Scan](#) ([Directory::FilenamesType](#) 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)

27.258.1 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](#).

27.258.2 Member Typedef Documentation

27.258.2.1 `typedef MappingType::const_iterator gdcm::Scanner::ConstIterator`

27.258.2.2 `typedef std::map<const char *, TagToValue, Itstr> gdcm::Scanner::MappingType`

27.258.2.3 `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)`

27.258.2.4 `typedef TagToValue::value_type gdcm::Scanner::TagToValueValueType`

27.258.2.5 `typedef std::set< std::string > gdcm::Scanner::ValuesType`

27.258.3 Constructor & Destructor Documentation

27.258.3.1 `gdcm::Scanner::Scanner () [inline]`

27.258.3.2 `gdcm::Scanner::~~Scanner ()`

27.258.4 Member Function Documentation

27.258.4.1 `void gdcm::Scanner::AddPrivateTag (PrivateTag const & t)`

27.258.4.2 `void gdcm::Scanner::AddSkipTag (Tag const & t)`

Add a tag that will need to be skipped. Those are root level skip tags.

27.258.4.3 `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](#).

27.258.4.4 **ConstIterator** gdcm::Scanner::Begin () const [inline]

27.258.4.5 void gdcm::Scanner::ClearSkipTags ()

27.258.4.6 void gdcm::Scanner::ClearTags ()

27.258.4.7 **ConstIterator** gdcm::Scanner::End () const [inline]

27.258.4.8 **Directory::FilenameType** gdcm::Scanner::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'

27.258.4.9 const char* gdcm::Scanner::GetFilenameFromTagToValue (Tag const & t, const char * *valueref*) const

Will loop over all files and return the first file where value match the reference value 'valueref'

27.258.4.10 **Directory::FilenameType** const& gdcm::Scanner::GetFilenames () const [inline]

27.258.4.11 **Directory::FilenameType** 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](#).

27.258.4.12 TagToValue const& gdcm::Scanner::GetMapping (const char * *filename*) const

Get the std::map mapping filenames to value for file 'filename'.

Examples:

[DumpToSQLITE3.cxx](#).

27.258.4.13 TagToValue const& gdcm::Scanner::GetMappingFromTagToValue (Tag const & t, const char * *value*) const

See [GetFilenameFromTagToValue\(\)](#). This is simply GetFilenameFromTagToValue followed.

27.258.4.14 **MappingType** const& gdcm::Scanner::GetMappings () const [inline]

Mappings are the mapping from a particular tag to the map, mapping filename to value:

27.258.4.15 **Directory::FilenameType** gdcm::Scanner::GetOrderedValues (Tag const & t) const

Get all the values found (in a vector) associated with Tag 't' This function is identical to GetValues, but is accessible from the wrapped layer (python, C#, java)

27.258.4.16 `const char* gdcmm::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 !

27.258.4.17 `ValueType const& gdcmm::Scanner::GetValues () const [inline]`

Get all the values found (in lexicographic order)

Examples:

[SortImage.cxx](#), and [VolumeSorter.cxx](#).

27.258.4.18 `ValueType gdcmm::Scanner::GetValues (Tag const & t) const`

Get all the values found (in lexicographic order) associated with Tag 't'.

27.258.4.19 `bool gdcmm::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](#).

27.258.4.20 `static SmartPointer<Scanner> gdcmm::Scanner::New () [inline],[static]`

for wrapped language: instantiate a reference counted object

27.258.4.21 `void gdcmm::Scanner::Print (std::ostream & os) const [virtual]`

Print result.

Reimplemented from [gdcmm::Object](#).

Referenced by `gdcmm::operator<<()`.

27.258.4.22 `void gdcmm::Scanner::ProcessPublicTag (StringFilter & sf, const char * filename) [protected]`

27.258.4.23 `bool gdcmm::Scanner::Scan (Directory::FileNamesType const & filenames)`

Start the scan !

Examples:

[DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).

27.258.5 Friends And Related Function Documentation

```
27.258.5.1 std::ostream& operator<< ( std::ostream &_os, const Scanner &s ) [friend]
```

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

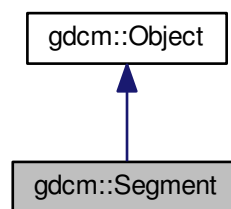
- [gdcmScanner.h](#)

27.259 gdc::Segment Class Reference

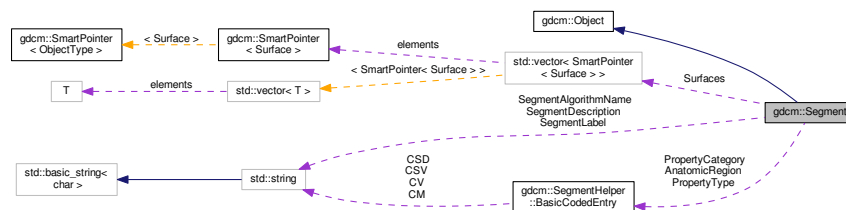
This class defines a segment. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface.

```
#include <gdcmSegment.h>
```

Inheritance diagram for `gdcm::Segment`:



Collaboration diagram for `gdc::Segment`:



Public Types

- enum `ALGOType` {
`MANUAL = 0,`
`AUTOMATIC,`
`ALGOType_END }`
- typedef std::vector< `SmartPointer< Surface >` > `SurfaceVector`

Public Member Functions

- [Segment](#) ()
- virtual [~Segment](#) ()
- void [AddSurface](#) ([SmartPointer](#)< [Surface](#) > surface)
- [SegmentHelper::BasicCodedEntry](#) const & [GetAnatomicRegion](#) () const
- [SegmentHelper::BasicCodedEntry](#) & [GetAnatomicRegion](#) ()
- [SegmentHelper::BasicCodedEntry](#) const & [GetPropertyCategory](#) () const
- [SegmentHelper::BasicCodedEntry](#) & [GetPropertyCategory](#) ()
- [SegmentHelper::BasicCodedEntry](#) const & [GetPropertyType](#) () const
- [SegmentHelper::BasicCodedEntry](#) & [GetPropertyType](#) ()
- 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](#) const & [GetSurfaces](#) () const
- [SurfaceVector](#) & [GetSurfaces](#) ()
- void [SetAnatomicRegion](#) ([SegmentHelper::BasicCodedEntry](#) const &BSE)
- void [SetPropertyCategory](#) ([SegmentHelper::BasicCodedEntry](#) const &BSE)
- void [SetPropertyType](#) ([SegmentHelper::BasicCodedEntry](#) const &BSE)
- 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](#)
- [SegmentHelper::BasicCodedEntry](#) [PropertyCategory](#)
- [SegmentHelper::BasicCodedEntry](#) [PropertyType](#)
- std::string [SegmentAlgorithmName](#)
- [ALGOType](#) [SegmentAlgorithmType](#)
- std::string [SegmentDescription](#)
- std::string [SegmentLabel](#)
- unsigned short [SegmentNumber](#)
- unsigned long [SurfaceCount](#)
- [SurfaceVector](#) [Surfaces](#)

Additional Inherited Members

27.259.1 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

27.259.2 Member Typedef Documentation

27.259.2.1 `typedef std::vector< SmartPointer< Surface > > gdcm::Segment::SurfaceVector`

27.259.3 Member Enumeration Documentation

27.259.3.1 `enum gdcm::Segment::ALGOType`

Enumerator

MANUAL

AUTOMATIC

ALGOType_END

27.259.4 Constructor & Destructor Documentation

27.259.4.1 `gdcm::Segment::Segment ()`

27.259.4.2 `virtual gdcm::Segment::~~Segment () [virtual]`

27.259.5 Member Function Documentation

27.259.5.1 `void gdcm::Segment::AddSurface (SmartPointer< Surface > surface)`

27.259.5.2 `static ALGOType gdcm::Segment::GetALGOType (const char * type) [static]`

27.259.5.3 `static const char* gdcm::Segment::GetALGOTypeString (ALGOType type) [static]`

27.259.5.4 `SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetAnatomicRegion () const`

27.259.5.5 `SegmentHelper::BasicCodedEntry& gdcm::Segment::GetAnatomicRegion ()`

27.259.5.6 `SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetPropertyCategory () const`

27.259.5.7 `SegmentHelper::BasicCodedEntry& gdcm::Segment::GetPropertyCategory ()`

27.259.5.8 `SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetPropertyType () const`

27.259.5.9 `SegmentHelper::BasicCodedEntry& gdcm::Segment::GetPropertyType ()`

- 27.259.5.10 `const char* gdcmm::Segment::GetSegmentAlgorithmName () const`
- 27.259.5.11 `ALGOType gdcmm::Segment::GetSegmentAlgorithmType () const`
- 27.259.5.12 `const char* gdcmm::Segment::GetSegmentDescription () const`
- 27.259.5.13 `const char* gdcmm::Segment::GetSegmentLabel () const`
- 27.259.5.14 `unsigned short gdcmm::Segment::GetSegmentNumber () const`
- 27.259.5.15 `SmartPointer< Surface > gdcmm::Segment::GetSurface (const unsigned int idx = 0) const`
- 27.259.5.16 `unsigned long gdcmm::Segment::GetSurfaceCount ()`
- 27.259.5.17 `SurfaceVector const& gdcmm::Segment::GetSurfaces () const`
- 27.259.5.18 `SurfaceVector& gdcmm::Segment::GetSurfaces ()`
- 27.259.5.19 `void gdcmm::Segment::SetAnatomicRegion (SegmentHelper::BasicCodedEntry const & BSE)`
- 27.259.5.20 `void gdcmm::Segment::SetPropertyCategory (SegmentHelper::BasicCodedEntry const & BSE)`
- 27.259.5.21 `void gdcmm::Segment::SetPropertyType (SegmentHelper::BasicCodedEntry const & BSE)`
- 27.259.5.22 `void gdcmm::Segment::SetSegmentAlgorithmName (const char * name)`
- 27.259.5.23 `void gdcmm::Segment::SetSegmentAlgorithmType (ALGOType type)`
- 27.259.5.24 `void gdcmm::Segment::SetSegmentAlgorithmType (const char * typeStr)`
- 27.259.5.25 `void gdcmm::Segment::SetSegmentDescription (const char * description)`
- 27.259.5.26 `void gdcmm::Segment::SetSegmentLabel (const char * label)`
- 27.259.5.27 `void gdcmm::Segment::SetSegmentNumber (const unsigned short num)`
- 27.259.5.28 `void gdcmm::Segment::SetSurfaceCount (const unsigned long nb)`

27.259.6 Member Data Documentation

- 27.259.6.1 `SegmentHelper::BasicCodedEntry gdcmm::Segment::AnatomicRegion` [protected]
- 27.259.6.2 `SegmentHelper::BasicCodedEntry gdcmm::Segment::PropertyCategory` [protected]
- 27.259.6.3 `SegmentHelper::BasicCodedEntry gdcmm::Segment::PropertyType` [protected]
- 27.259.6.4 `std::string gdcmm::Segment::SegmentAlgorithmName` [protected]
- 27.259.6.5 `ALGOType gdcmm::Segment::SegmentAlgorithmType` [protected]
- 27.259.6.6 `std::string gdcmm::Segment::SegmentDescription` [protected]

27.259.6.7 `std::string` `gdcm::Segment::SegmentLabel` [protected]

27.259.6.8 `unsigned short` `gdcm::Segment::SegmentNumber` [protected]

27.259.6.9 `unsigned long` `gdcm::Segment::SurfaceCount` [protected]

27.259.6.10 `SurfaceVector` `gdcm::Segment::Surfaces` [protected]

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

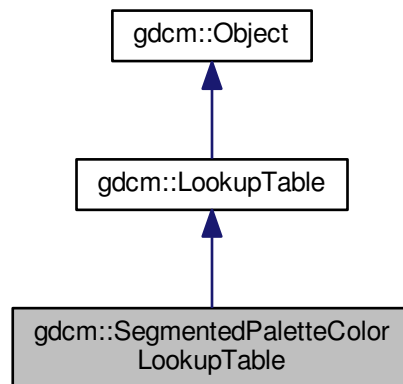
- [gdcmSegment.h](#)

27.260 gdcm::SegmentedPaletteColorLookupTable Class Reference

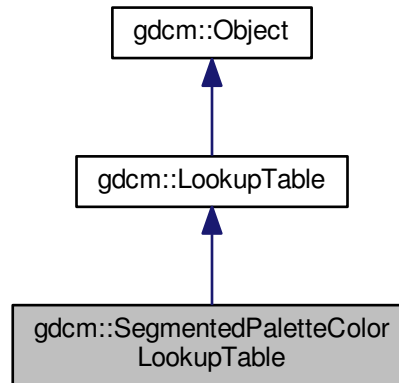
[SegmentedPaletteColorLookupTable](#) class.

```
#include <gdcmSegmentedPaletteColorLookupTable.h>
```

Inheritance diagram for `gdcm::SegmentedPaletteColorLookupTable`:



Collaboration diagram for `gdcm::SegmentedPaletteColorLookupTable`:



Public Member Functions

- [SegmentedPaletteColorLookupTable](#) ()
- [~SegmentedPaletteColorLookupTable](#) ()
- void [Print](#) (std::ostream &) const
- void [SetLUT](#) ([LookupTableType](#) type, const unsigned char *array, unsigned int length)

Initialize a [SegmentedPaletteColorLookupTable](#).

Additional Inherited Members

27.260.1 Detailed Description

[SegmentedPaletteColorLookupTable](#) class.

27.260.2 Constructor & Destructor Documentation

27.260.2.1 `gdcm::SegmentedPaletteColorLookupTable::SegmentedPaletteColorLookupTable ()`

27.260.2.2 `gdcm::SegmentedPaletteColorLookupTable::~~SegmentedPaletteColorLookupTable ()`

27.260.3 Member Function Documentation

27.260.3.1 `void gdcm::SegmentedPaletteColorLookupTable::Print (std::ostream &) const` `[inline]`, `[virtual]`

Reimplemented from [gdcm::LookupTable](#).

27.260.3.2 void gdcm::SegmentedPaletteColorLookupTable::SetLUT (LookupTableType *type*, const unsigned char * *array*, unsigned int *length*) [virtual]

Initialize a [SegmentedPaletteColorLookupTable](#).

Reimplemented from [gdcm::LookupTable](#).

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

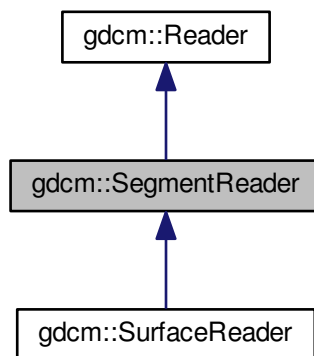
- [gdcmSegmentedPaletteColorLookupTable.h](#)

27.261 gdcm::SegmentReader Class Reference

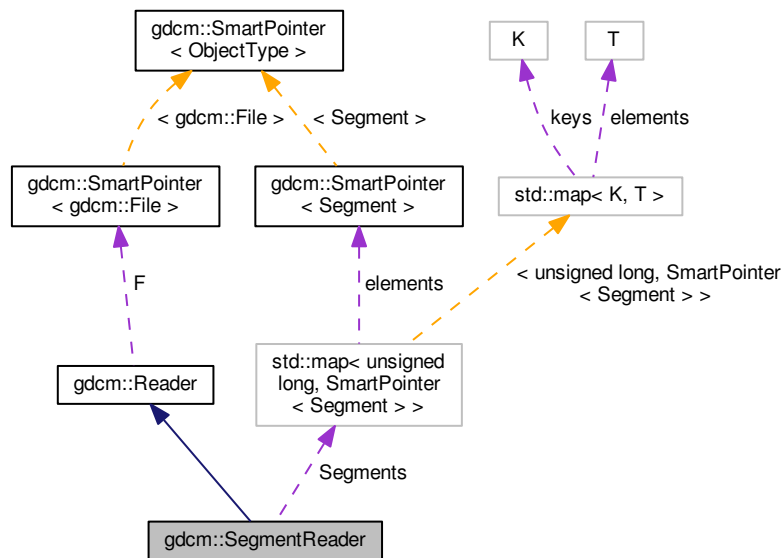
This class defines a segment reader. It reads attributes of group 0x0062.

```
#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](#) ()
- virtual `~SegmentReader` ()
- const [SegmentVector](#) [GetSegments](#) () const
- [SegmentVector](#) [GetSegments](#) ()
- virtual bool [Read](#) ()

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](#)

27.261.1 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

27.261.2 Member Typedef Documentation

27.261.2.1 `typedef std::map< unsigned long, SmartPointer< Segment > > gdcm::SegmentReader::SegmentMap`
[protected]

27.261.2.2 `typedef std::vector< SmartPointer< Segment > > gdcm::SegmentReader::SegmentVector`

27.261.3 Constructor & Destructor Documentation

27.261.3.1 `gdcm::SegmentReader::SegmentReader ()`

27.261.3.2 `virtual gdcm::SegmentReader::~~SegmentReader ()` [virtual]

27.261.4 Member Function Documentation

27.261.4.1 `const SegmentVector gdcm::SegmentReader::GetSegments () const`

27.261.4.2 `SegmentVector gdcm::SegmentReader::GetSegments ()`

27.261.4.3 `virtual bool gdcm::SegmentReader::Read ()` [virtual]

Read.

Reimplemented from [gdcm::Reader](#).

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

27.261.4.4 `bool gdcm::SegmentReader::ReadSegment (const Item & segmentItem, const unsigned int idx)` [protected]

27.261.4.5 `bool gdcm::SegmentReader::ReadSegments ()` [protected]

27.261.5 Member Data Documentation

27.261.5.1 `SegmentMap gdcm::SegmentReader::Segments` [protected]

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

- [gdcmSegmentReader.h](#)

- bool [Write](#) ()
Write.

Protected Member Functions

- bool [PrepareWrite](#) ()

Protected Attributes

- [SegmentVector](#) [Segments](#)

27.262.1 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

27.262.2 Member Typedef Documentation

27.262.2.1 `typedef std::vector< SmartPointer< Segment > > gdcm::SegmentWriter::SegmentVector`

27.262.3 Constructor & Destructor Documentation

27.262.3.1 `gdcm::SegmentWriter::SegmentWriter ()`

27.262.3.2 `virtual gdcm::SegmentWriter::~~SegmentWriter ()` `[virtual]`

27.262.4 Member Function Documentation

27.262.4.1 `void gdcm::SegmentWriter::AddSegment (SmartPointer< Segment > segment)`

27.262.4.2 `unsigned int gdcm::SegmentWriter::GetNumberOfSegments ()` `const`

27.262.4.3 `SmartPointer< Segment > gdcm::SegmentWriter::GetSegment (const unsigned int idx = 0)` `const`

27.262.4.4 `const SegmentVector& gdcm::SegmentWriter::GetSegments ()` `const`

27.262.4.5 `SegmentVector& gdcm::SegmentWriter::GetSegments ()`

27.262.4.6 `bool gdcm::SegmentWriter::PrepareWrite ()` `[protected]`

27.262.4.7 `void gdcm::SegmentWriter::SetNumberOfSegments (const unsigned int size)`

27.262.4.8 `void gdcm::SegmentWriter::SetSegments (SegmentVector & segments)`

27.262.4.9 `bool gdcm::SegmentWriter::Write ()` `[virtual]`

Write.

Reimplemented from [gdcm::Writer](#).

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

27.262.5 Member Data Documentation

27.262.5.1 SegmentVector `gdcm::SegmentWriter::Segments` [protected]

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

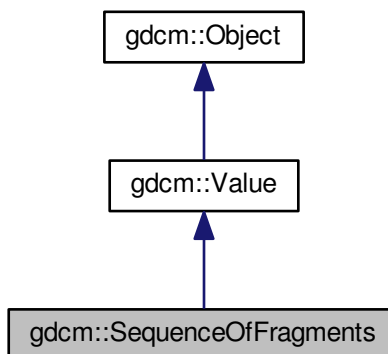
- [gdcmSegmentWriter.h](#)

27.263 `gdcm::SequenceOfFragments` Class Reference

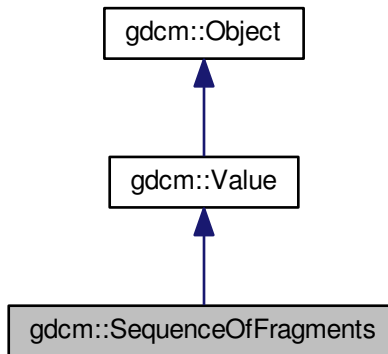
Class to represent a Sequence Of Fragments.

```
#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](#) ()
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
Returns the SQ length, as read from disk.
- [SizeType](#) [GetNumberOfFragments](#) () const
- const [BasicOffsetTable](#) & [GetTable](#) () const
- [BasicOffsetTable](#) & [GetTable](#) ()

- bool `operator==` (const `Value` &val) const
- void `Print` (std::ostream &os) const
- 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)
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

27.263.1 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`.

27.263.2 Member Typedef Documentation

27.263.2.1 `typedef FragmentVector::const_iterator gdcmm::SequenceOfFragments::ConstIterator`

27.263.2.2 `typedef std::vector<Fragment> gdcmm::SequenceOfFragments::FragmentVector`

27.263.2.3 `typedef FragmentVector::iterator gdcmm::SequenceOfFragments::Iterator`

27.263.2.4 `typedef FragmentVector::size_type gdcmm::SequenceOfFragments::SizeType`

27.263.3 Constructor & Destructor Documentation

27.263.3.1 `gdcmm::SequenceOfFragments::SequenceOfFragments () [inline]`

constructor (UndefinedLength by default)

27.263.4 Member Function Documentation

27.263.4.1 void gdcm::SequenceOfFragments::AddFragment (**Fragment** const & *item*)

Appends a [Fragment](#) to the already added ones.

Examples:

[FixBrokenJ2K.cxx](#).

27.263.4.2 Iterator gdcm::SequenceOfFragments::Begin () [inline]

27.263.4.3 ConstIterator gdcm::SequenceOfFragments::Begin () const [inline]

27.263.4.4 void gdcm::SequenceOfFragments::Clear () [virtual]

Clear.

Implements [gdcm::Value](#).

27.263.4.5 unsigned long gdcm::SequenceOfFragments::ComputeByteLength () const

27.263.4.6 VL gdcm::SequenceOfFragments::ComputeLength () const

27.263.4.7 Iterator gdcm::SequenceOfFragments::End () [inline]

27.263.4.8 ConstIterator gdcm::SequenceOfFragments::End () const [inline]

27.263.4.9 bool gdcm::SequenceOfFragments::GetBuffer (char * *buffer*, unsigned long *length*) const

27.263.4.10 bool gdcm::SequenceOfFragments::GetFragBuffer (unsigned int *fragNb*, char * *buffer*, unsigned long & *length*) const

27.263.4.11 const **Fragment**& gdcm::SequenceOfFragments::GetFragment (**SizeType** *num*) const

Examples:

[FixBrokenJ2K.cxx](#), and [FixJAIBugJPEGLS.cxx](#).

27.263.4.12 VL gdcm::SequenceOfFragments::GetLength () const [inline],[virtual]

Returns the SQ length, as read from disk.

Implements [gdcm::Value](#).

27.263.4.13 **SizeType** gdcm::SequenceOfFragments::GetNumberOfFragments () const

Examples:

[FixJAIBugJPEGLS.cxx](#).

27.263.4.14 `const BasicOffsetTable& gdcM::SequenceOfFragments::GetTable () const` `[inline]`

27.263.4.15 `BasicOffsetTable& gdcM::SequenceOfFragments::GetTable ()` `[inline]`

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

27.263.4.17 `bool gdcM::SequenceOfFragments::operator== (const Value & val) const` `[inline]`,`[virtual]`

Implements [gdcM::Value](#).

27.263.4.18 `void gdcM::SequenceOfFragments::Print (std::ostream & os) const` `[inline]`,`[virtual]`

Reimplemented from [gdcM::Object](#).

27.263.4.19 `template<typename TSwap > std::istream& gdcM::SequenceOfFragments::Read (std::istream & is, bool readvalues = true)` `[inline]`

27.263.4.20 `template<typename TSwap > std::istream& gdcM::SequenceOfFragments::ReadPreValue (std::istream & is)`
`[inline]`

References [gdcMDebugMacro](#).

27.263.4.21 `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\(\)](#).

27.263.4.22 `void gdcM::SequenceOfFragments::SetLength (VL length)` `[inline]`,`[virtual]`

Sets the actual SQ length.

Implements [gdcM::Value](#).

27.263.4.23 `template<typename TSwap > std::ostream const& gdcM::SequenceOfFragments::Write (std::ostream & os) const`
`[inline]`

References [gdcM::VL::Write\(\)](#), and [gdcM::Tag::Write\(\)](#).

27.263.4.24 `bool gdcM::SequenceOfFragments::WriteBuffer (std::ostream & os) const`

Examples:

[GetJPEGSamplePrecision.cxx](#).

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

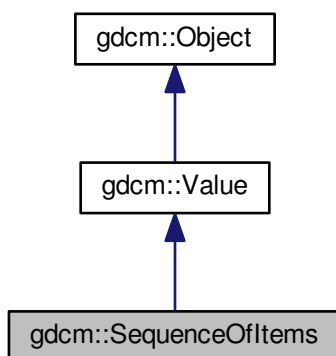
- [gdcMSequenceOfFragments.h](#)

27.264 gdcm::SequenceOfItems Class Reference

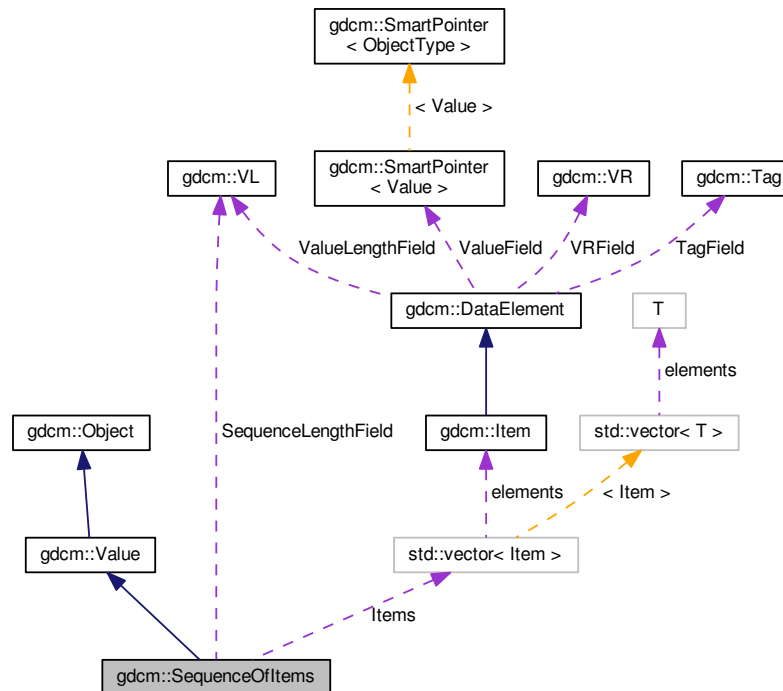
Class to represent a Sequence Of Items (value representation : SQ)

```
#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](#) ()
remove all items within the sequence
- `template<typename TDE >`
[VL ComputeLength](#) () const

- [Iterator End](#) ()
- [ConstIterator End](#) () const
- bool [FindDataElement](#) (const [Tag](#) &t) const
- const [Item](#) & [GetItem](#) ([SizeType](#) position) const
- [Item](#) & [GetItem](#) ([SizeType](#) position)
- [VL GetLength](#) () const
Returns the SQ length, as read from disk.
- [SizeType GetNumberOfItems](#) () const
- bool [IsUndefinedLength](#) () const
return if [Value](#) Length if of undefined length
- [SequenceOfItems](#) & [operator=](#) (const [SequenceOfItems](#) &val)
- bool [operator==](#) (const [Value](#) &val) const
- void [Print](#) (std::ostream &os) const
- template<typename TDE , typename TSwap >
std::istream & [Read](#) (std::istream &is, bool readvalues=true)
- bool [RemoveItemByIndex](#) (const [SizeType](#) index)
- void [SetLength](#) ([VL](#) length)
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

27.264.1 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](#), [GetSequenceUltrasound.cxx](#), and [ReadExplicitLengthSQIVR.cxx](#).

27.264.2 Member Typedef Documentation

27.264.2.1 `typedef ItemVector::const_iterator gdcm::SequenceOfItems::ConstIterator`

27.264.2.2 `typedef std::vector< Item > gdcm::SequenceOfItems::ItemVector`

27.264.2.3 `typedef ItemVector::iterator gdcm::SequenceOfItems::Iterator`

27.264.2.4 `typedef ItemVector::size_type gdcm::SequenceOfItems::SizeType`

27.264.3 Constructor & Destructor Documentation

27.264.3.1 `gdcm::SequenceOfItems::SequenceOfItems () [inline]`

constructor (UndefinedLength by default)

27.264.4 Member Function Documentation

27.264.4.1 `void gdcm::SequenceOfItems::AddItem (Item const & item)`

Appends an [Item](#) to the already added ones.

Examples:

[Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [GenAllVR.cxx](#), [GenLongSeqs.cxx](#), and [GenSeqs.cxx](#).

27.264.4.2 `Item& gdcm::SequenceOfItems::AddNewUndefinedLengthItem ()`

Appends an [Item](#) to the already added ones.

27.264.4.3 `Iterator gdcm::SequenceOfItems::Begin () [inline]`

27.264.4.4 `ConstIterator gdcm::SequenceOfItems::Begin () const [inline]`

27.264.4.5 `void gdcm::SequenceOfItems::Clear () [virtual]`

remove all items within the sequence

Implements [gdcm::Value](#).

27.264.4.6 `template<typename TDE > VL gdcmm::SequenceOfItems::ComputeLength () const`

27.264.4.7 `Iterator gdcmm::SequenceOfItems::End () [inline]`

27.264.4.8 `ConstIterator gdcmm::SequenceOfItems::End () const [inline]`

27.264.4.9 `bool gdcmm::SequenceOfItems::FindDataElement (const Tag & t) const`

27.264.4.10 `const Item& gdcmm::SequenceOfItems::GetItem (SizeType position) const`

Examples:

[ChangeSequenceUltrasound.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [ExtractEncryptedContent.cxx](#), [gdcmmrtionplan.cxx](#), [gdcmmrtplan.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), and [ReadAndDumpDICOMDIR.cxx](#).

27.264.4.11 `Item& gdcmm::SequenceOfItems::GetItem (SizeType position)`

27.264.4.12 `VL gdcmm::SequenceOfItems::GetLength () const [inline],[virtual]`

Returns the SQ length, as read from disk.

Implements [gdcmm::Value](#).

27.264.4.13 `SizeType gdcmm::SequenceOfItems::GetNumberOfItems () const [inline]`

Examples:

[ChangeSequenceUltrasound.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [ExtractEncryptedContent.cxx](#), [gdcmmrtionplan.cxx](#), [gdcmmrtplan.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), and [LargeVRDSExplicit.cxx](#).

27.264.4.14 `bool gdcmm::SequenceOfItems::IsUndefinedLength () const [inline]`

return if [Value](#) Length if of undefined length

27.264.4.15 `static SmartPointer<SequenceOfItems> gdcmm::SequenceOfItems::New () [inline],[static]`

Examples:

[NewSequence.cs](#).

27.264.4.16 `SequenceOfItems& gdcmm::SequenceOfItems::operator= (const SequenceOfItems & val) [inline]`

References Items, and SequenceLengthField.

27.264.4.17 `bool gdcmm::SequenceOfItems::operator==(const Value & val) const [inline],[virtual]`

Implements [gdcmm::Value](#).

References Items, and SequenceLengthField.

27.264.4.18 `void gdcM::SequenceOfItems::Print (std::ostream & os) const` `[inline],[virtual]`

Reimplemented from [gdcM::Object](#).

27.264.4.19 `template<typename TDE , typename TSwap > std::istream& gdcM::SequenceOfItems::Read (std::istream & is, bool readvalues = true)` `[inline]`

Examples:

[ReadExplicitLengthSQIVR.cxx](#).

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\(\)](#).

27.264.4.20 `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

27.264.4.21 `void gdcM::SequenceOfItems::SetLength (VL length)` `[inline],[virtual]`

Sets the actual SQ length.

Implements [gdcM::Value](#).

Examples:

[ReadExplicitLengthSQIVR.cxx](#).

27.264.4.22 `void gdcM::SequenceOfItems::SetLengthToUndefined ()`

Properly set the Sequence of [Item](#) to be undefined length.

Examples:

[Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [GenAllIVR.cxx](#), [GenLongSeqs.cxx](#), and [GenSeqs.cxx](#).

27.264.4.23 `void gdcM::SequenceOfItems::SetNumberOfItems (SizeType n)` `[inline]`

27.264.4.24 `template<typename TDE , typename TSwap > std::ostream const& gdcM::SequenceOfItems::Write (std::ostream & os) const` `[inline]`

References [gdcM::VL::Write\(\)](#), and [gdcM::Tag::Write\(\)](#).

27.264.5 Member Data Documentation

27.264.5.1 ItemVector gdcM::SequenceOfItems::Items

Vector of Sequence Items.

Referenced by [operator=\(\)](#), and [operator==\(\)](#).

27.264.5.2 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:

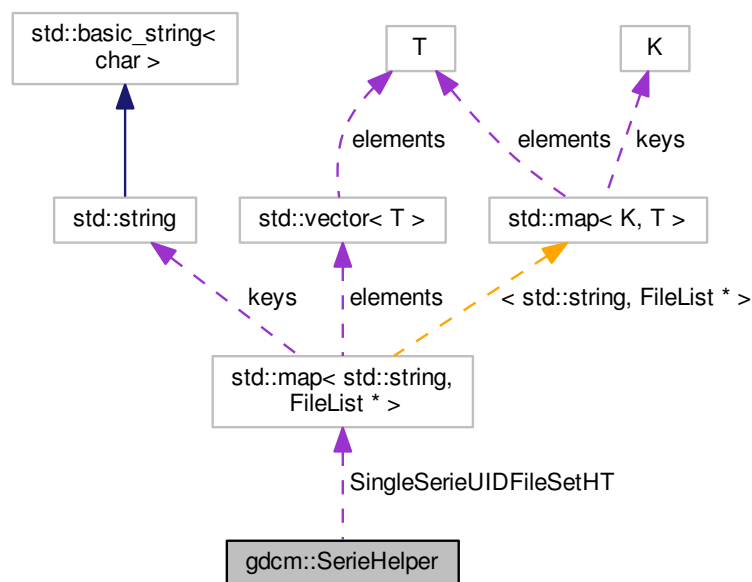
- [gdcmSequenceOfItems.h](#)

27.265 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:



Classes

- struct [Rule](#)

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

- 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 [ImagePositionPatientOrdering](#) ([FileList](#) *fileSet)
- bool [UserOrdering](#) ([FileList](#) *fileSet)

Protected Attributes

- [SingleSerieUIDFileSetmap::iterator](#) [ItFileSetHt](#)
- [SingleSerieUIDFileSetmap](#) [SingleSerieUIDFileSetHT](#)

27.265.1 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](#)

27.265.2 Member Typedef Documentation

27.265.2.1 typedef std::vector<[Rule](#)> [gdcm::SerieHelper::SerieRestrictions](#) [protected]

27.265.2.2 typedef std::map<std::string, [FileList](#) *> [gdcm::SerieHelper::SingleSerieUIDFileSetmap](#) [protected]

27.265.3 Constructor & Destructor Documentation

27.265.3.1 [gdcm::SerieHelper::SerieHelper](#) ()

27.265.3.2 `gdcm::SerieHelper::~~SerieHelper ()`

27.265.4 Member Function Documentation

27.265.4.1 `bool gdcm::SerieHelper::AddFile (FileWithName & header)` `[protected]`

27.265.4.2 `void gdcm::SerieHelper::AddFileName (std::string const & filename)` `[protected]`

27.265.4.3 `void gdcm::SerieHelper::AddRestriction (const std::string & tag)`

27.265.4.4 `void gdcm::SerieHelper::AddRestriction (uint16_t group, uint16_t elem, std::string const & value, int op)`

27.265.4.5 `void gdcm::SerieHelper::AddRestriction (const Tag & tag)` `[protected]`

27.265.4.6 `void gdcm::SerieHelper::Clear ()`

27.265.4.7 `void gdcm::SerieHelper::CreateDefaultUniqueSeriesIdentifier ()`

27.265.4.8 `std::string gdcm::SerieHelper::CreateUniqueSeriesIdentifier (File * inFile)`

27.265.4.9 `bool gdcm::SerieHelper::FileNameOrdering (FileList * fileList)` `[protected]`

27.265.4.10 `FileList* gdcm::SerieHelper::GetFirstSingleSerieUIDFileSet ()`

27.265.4.11 `FileList* gdcm::SerieHelper::GetNextSingleSerieUIDFileSet ()`

27.265.4.12 `bool gdcm::SerieHelper::ImagePositionPatientOrdering (FileList * fileSet)` `[protected]`

27.265.4.13 `void gdcm::SerieHelper::OrderFileList (FileList * fileSet)`

27.265.4.14 `void gdcm::SerieHelper::SetDirectory (std::string const & dir, bool recursive = false)`

27.265.4.15 `void gdcm::SerieHelper::SetLoadMode (int)` `[inline]`

27.265.4.16 `void gdcm::SerieHelper::SetUseSeriesDetails (bool useSeriesDetails)`

27.265.4.17 `bool gdcm::SerieHelper::UserOrdering (FileList * fileSet)` `[protected]`

27.265.5 Member Data Documentation

27.265.5.1 `SingleSerieUIDFileSetmap::iterator gdcm::SerieHelper::ItFileSetHt` `[protected]`

27.265.5.2 `SingleSerieUIDFileSetmap gdcm::SerieHelper::SingleSerieUIDFileSetHT` `[protected]`

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

- [gdcmSerieHelper.h](#)

27.266 gdcm::Series Class Reference

[Series.](#)

```
#include <gdcmSeries.h>
```

Public Member Functions

- [Series](#) ()

27.266.1 Detailed Description

[Series](#).

27.266.2 Constructor & Destructor Documentation

27.266.2.1 `gdcm::Series::Series ()` `[inline]`

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

- [gdcmSeries.h](#)

27.267 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

27.267.1 Detailed Description

PS 3.4 [Table B.3-1](#) SERVICE-CLASS-APPLICATION-INFORMATION (A-ASSOCIATE-RQ)

27.267.2 Constructor & Destructor Documentation

27.267.2.1 `gdcm::network::ServiceClassApplicationInformation::ServiceClassApplicationInformation ()`

27.267.3 Member Function Documentation

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

27.267.3.2 `std::istream& gdcm::network::ServiceClassApplicationInformation::Read (std::istream & is)`

27.267.3.3 void gdcm::network::ServiceClassApplicationInformation::SetTuple (uint8_t *levelofsupport*, uint8_t *levelofdigitalsig*, uint8_t *elementcoercion*)

27.267.3.4 size_t gdcm::network::ServiceClassApplicationInformation::Size () const

27.267.3.5 const std::ostream& gdcm::network::ServiceClassApplicationInformation::Write (std::ostream & *os*) const

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

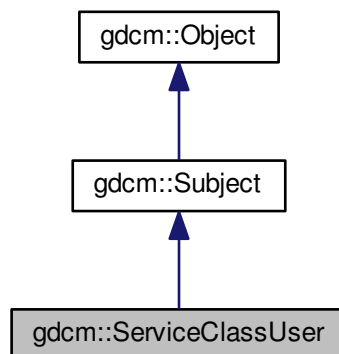
- [gdcmServiceClassApplicationInformation.h](#)

27.268 gdcm::ServiceClassUser Class Reference

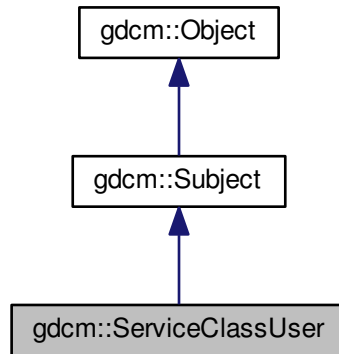
[ServiceClassUser](#).

```
#include <gdcmServiceClassUser.h>
```

Inheritance diagram for gdcm::ServiceClassUser:



Collaboration diagram for `gdcm::ServiceClassUser`:



Public Member Functions

- [ServiceClassUser](#) ()
- [~ServiceClassUser](#) ()
- `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.
- `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](#) ([File](#) `const` &file)
- `bool` [SendStore](#) ([DataSet](#) `const` &ds)
Execute a C-STORE on a DataSet, the transfer syntax used will be Implicit.
- `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: instantiate a reference counted object

Additional Inherited Members

27.268.1 Detailed Description

[ServiceClassUser](#).

Examples:

[CStoreQtProgress.cxx](#).

27.268.2 Constructor & Destructor Documentation

27.268.2.1 gdcm::ServiceClassUser::ServiceClassUser ()

Construct a SCU with default:

- hostname = localhost
- port = 104

27.268.2.2 gdcm::ServiceClassUser::~~ServiceClassUser ()

27.268.3 Member Function Documentation

27.268.3.1 const char* gdcm::ServiceClassUser::GetAETitle () const

27.268.3.2 `const char* gdcm::ServiceClassUser::GetCalledAETitle () const`

27.268.3.3 `double gdcm::ServiceClassUser::GetTimeout () const`

27.268.3.4 `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](#).

27.268.3.5 `bool gdcm::ServiceClassUser::IsPresentationContextAccepted (const PresentationContext & pc) const`

Return if the passed in presentation was accepted during association negotiation.

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

for wrapped language: instantiate a reference counted object

27.268.3.7 `bool gdcm::ServiceClassUser::SendEcho ()`

C-ECHO.

27.268.3.8 `bool gdcm::ServiceClassUser::SendFind (const BaseRootQuery * query, std::vector< DataSet > & retDatasets)`

C-FIND a query, return result are in retDatasets.

27.268.3.9 `bool gdcm::ServiceClassUser::SendMove (const BaseRootQuery * query, const char * outputdir)`

Execute a C-MOVE, based on query, return files are written in outputdir.

27.268.3.10 `bool gdcm::ServiceClassUser::SendMove (const BaseRootQuery * query, std::vector< DataSet > & retDatasets)`

Execute a C-MOVE, based on query, returned dataset are Implicit.

27.268.3.11 `bool gdcm::ServiceClassUser::SendMove (const BaseRootQuery * query, std::vector< File > & retFile)`

Execute a C-MOVE, based on query, returned Files are stored in vector.

27.268.3.12 `bool gdcm::ServiceClassUser::SendStore (const char * filename)`

Execute a C-STORE on file on disk, named filename.

Examples:

[CStoreQtProgress.cxx](#).

27.268.3.13 `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.

27.268.3.14 `bool gdcm::ServiceClassUser::SendStore (DataSet const & ds)`

Execute a C-STORE on a [DataSet](#), the transfer syntax used will be Implicit.

27.268.3.15 `void gdcm::ServiceClassUser::SetAETitle (const char * aetitle)`

set calling ae title

27.268.3.16 `void gdcm::ServiceClassUser::SetCalledAETitle (const char * aetitle)`

set called ae title

Examples:

[CStoreQtProgress.cxx](#).

27.268.3.17 `void gdcm::ServiceClassUser::SetHostname (const char * hostname)`

Set the name of the called hostname (hostname or IP address)

Examples:

[CStoreQtProgress.cxx](#).

27.268.3.18 `void gdcm::ServiceClassUser::SetPort (uint16_t port)`

Set port of remote host (called application)

Examples:

[CStoreQtProgress.cxx](#).

27.268.3.19 `void gdcm::ServiceClassUser::SetPortSCP (uint16_t portscp)`

Set the port for any incoming C-STORE-SCP operation (typically in a return of C-MOVE)

27.268.3.20 `void gdcm::ServiceClassUser::SetPresentationContexts (std::vector< PresentationContext > const & pcs)`

Set the Presentation Context used for the Association.

Examples:

[CStoreQtProgress.cxx](#).

27.268.3.21 void gdcmm::ServiceClassUser::SetTimeout (double t)

set/get Timeout

Examples:

[CStoreQtProgress.cxx](#).

27.268.3.22 bool gdcmm::ServiceClassUser::StartAssociation ()

Start the association. Need to call SetPresentationContexts before.

Examples:

[CStoreQtProgress.cxx](#).

27.268.3.23 bool gdcmm::ServiceClassUser::StopAssociation ()

Stop the running association.

Examples:

[CStoreQtProgress.cxx](#).

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

- [gdcmmServiceClassUser.h](#)

27.269 gdcmm::SHA1 Class Reference

Class for [SHA1](#).

```
#include <gdcmmSHA1.h>
```

Public Member Functions

- [SHA1](#) ()
- [~SHA1](#) ()

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])

27.269.1 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

27.269.2 Constructor & Destructor Documentation

27.269.2.1 `gdcm::SHA1::SHA1 ()`

27.269.2.2 `gdcm::SHA1::~~SHA1 ()`

27.269.3 Member Function Documentation

27.269.3.1 `static bool gdcm::SHA1::Compute (const char * buffer, unsigned long buf_len, char digest_str[20*2+1])`
[static]

27.269.3.2 `static bool gdcm::SHA1::ComputeFile (const char * filename, char digest_str[20*2+1])` [static]

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

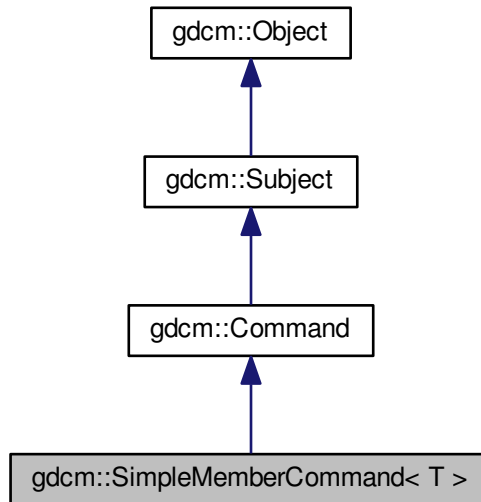
- [gdcmSHA1.h](#)

27.270 `gdcm::SimpleMemberCommand< T >` Class Template Reference

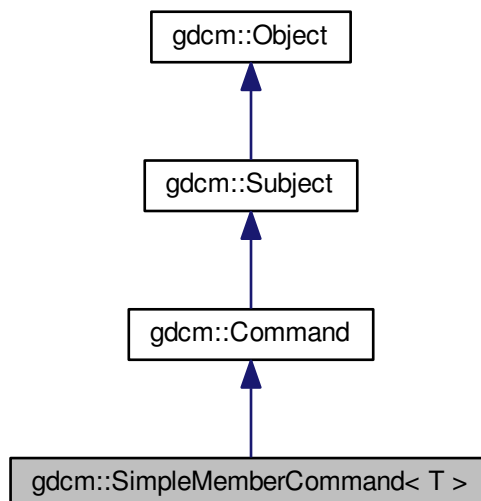
[Command](#) subclass that calls a pointer to a member function.

```
#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

- virtual void [Execute](#) ([Subject](#) *, const [Event](#) &)
- virtual void [Execute](#) (const [Subject](#) *, const [Event](#) &)
- void [SetCallbackFunction](#) (T *object, [TMemberFunctionPointer](#) memberFunction)

Static Public Member Functions

- static [SmartPointer](#)< [SimpleMemberCommand](#) > [New](#) ()

Protected Member Functions

- [SimpleMemberCommand](#) ()
- virtual [~SimpleMemberCommand](#) ()

Protected Attributes

- [TMemberFunctionPointer](#) m_MemberFunction
- T * [m_This](#)

27.270.1 Detailed Description

template<typename T>class gdcmm::SimpleMemberCommand< T >

[Command](#) subclass that calls a pointer to a member function.

[SimpleMemberCommand](#) calls a pointer to a member function with no arguments.

27.270.2 Member Typedef Documentation

27.270.2.1 template<typename T > typedef [SimpleMemberCommand](#) gdcmm::SimpleMemberCommand< T >::Self

Standard class typedefs.

27.270.2.2 template<typename T > typedef void(T::* [gdcmm::SimpleMemberCommand](#)< T >::TMemberFunctionPointer) ()

A method callback.

27.270.3 Constructor & Destructor Documentation

27.270.3.1 `template<typename T> gdcM::SimpleMemberCommand< T>::SimpleMemberCommand ()`
`[inline], [protected]`

Referenced by `gdcM::SimpleMemberCommand< T>::New()`.

27.270.3.2 `template<typename T> virtual gdcM::SimpleMemberCommand< T>::~~SimpleMemberCommand ()`
`[inline], [protected], [virtual]`

27.270.4 Member Function Documentation

27.270.4.1 `template<typename T> virtual void gdcM::SimpleMemberCommand< T>::Execute (Subject *, const Event &)` `[inline], [virtual]`

Invoke the callback function.

Implements [gdcM::Command](#).

References `gdcM::SimpleMemberCommand< T>::m_MemberFunction`.

27.270.4.2 `template<typename T> virtual void gdcM::SimpleMemberCommand< T>::Execute (const Subject * caller, const Event & event)` `[inline], [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`.

27.270.4.3 `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()`.

27.270.4.4 `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`.

27.270.5 Member Data Documentation

27.270.5.1 `template<typename T> TMemberFunctionPointer gdcM::SimpleMemberCommand< T>::m_MemberFunction` `[protected]`

Referenced by `gdcM::SimpleMemberCommand< T>::Execute()`, and `gdcM::SimpleMemberCommand< T>::SetCallbackFunction()`.

27.270.5.2 `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](#)

27.271 gdcm::SimpleSubjectWatcher Class Reference

[SimpleSubjectWatcher](#) This is a typical [Subject](#) Watcher class. It will observe all events.

```
#include <gdcmSimpleSubjectWatcher.h>
```

Public Member Functions

- [SimpleSubjectWatcher](#) ([Subject](#) *s, const char *comment="")
- virtual [~SimpleSubjectWatcher](#) ()

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](#) ()

27.271.1 Detailed Description

[SimpleSubjectWatcher](#) This is a typical [Subject](#) Watcher class. It will observe all events.

Examples:

[SimpleScanner.cxx](#).

27.271.2 Constructor & Destructor Documentation

27.271.2.1 `gdcm::SimpleSubjectWatcher::SimpleSubjectWatcher (Subject * s, const char * comment = " ")`

27.271.2.2 `virtual gdcm::SimpleSubjectWatcher::~~SimpleSubjectWatcher ()` [virtual]

27.271.3 Member Function Documentation

- 27.271.3.1 `virtual void gdcM::SimpleSubjectWatcher::EndFilter () [protected],[virtual]`
- 27.271.3.2 `virtual void gdcM::SimpleSubjectWatcher::ShowAbort () [protected],[virtual]`
- 27.271.3.3 `virtual void gdcM::SimpleSubjectWatcher::ShowAnonymization (Subject * caller, const Event & evt) [protected],[virtual]`
- 27.271.3.4 `virtual void gdcM::SimpleSubjectWatcher::ShowData (Subject * caller, const Event & evt) [protected],[virtual]`
- 27.271.3.5 `virtual void gdcM::SimpleSubjectWatcher::ShowDataSet (Subject * caller, const Event & evt) [protected],[virtual]`
- 27.271.3.6 `virtual void gdcM::SimpleSubjectWatcher::ShowFileName (Subject * caller, const Event & evt) [protected],[virtual]`

Examples:

[SimpleScanner.cxx](#).

- 27.271.3.7 `virtual void gdcM::SimpleSubjectWatcher::ShowIteration () [protected],[virtual]`
- 27.271.3.8 `virtual void gdcM::SimpleSubjectWatcher::ShowProgress (Subject * caller, const Event & evt) [protected],[virtual]`
- 27.271.3.9 `virtual void gdcM::SimpleSubjectWatcher::StartFilter () [protected],[virtual]`
- 27.271.3.10 `void gdcM::SimpleSubjectWatcher::TestAbortOff () [protected]`
- 27.271.3.11 `void gdcM::SimpleSubjectWatcher::TestAbortOn () [protected]`

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

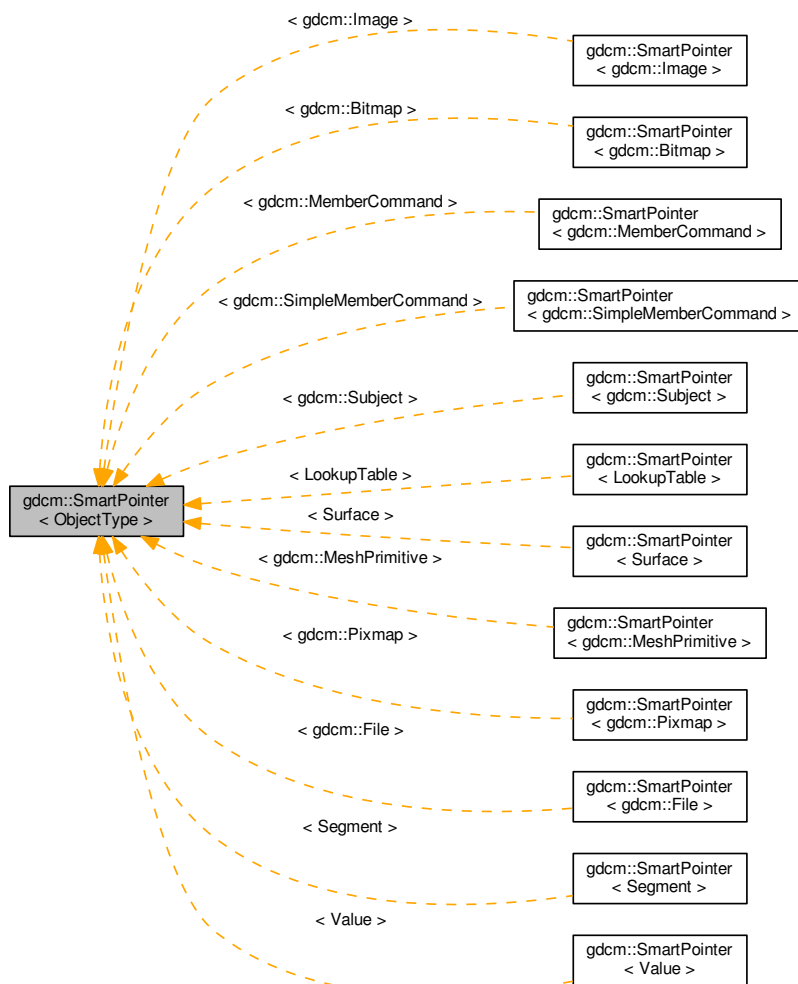
- [gdcMSimpleSubjectWatcher.h](#)

27.272 `gdcM::SmartPointer< ObjectType >` Class Template Reference

Class for Smart Pointer.

```
#include <gdcMObject.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= (SmartPointer const &r)`

Overload operator assignment.

- `SmartPointer & operator= (ObjectType *r)`

Overload operator assignment.

- `SmartPointer & operator= (ObjectType const &r)`

27.272.1 Detailed Description

```
template<class ObjectType>class gdcmm::SmartPointer< ObjectType >
```

Class for Smart Pointer.

Will only work for subclass of `gdcmm::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:

[ChangeSequenceUltrasound.cxx](#), [CStoreQtProgress.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [Fix_BrokenJ2K.cxx](#), [gdcmmrtionplan.cxx](#), [gdcmmrtplan.cxx](#), [GenAIIVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), and [SimpleScanner.cxx](#).

27.272.2 Constructor & Destructor Documentation

27.272.2.1 `template<class ObjectType> gdcmm::SmartPointer< ObjectType >::SmartPointer () [inline]`

27.272.2.2 `template<class ObjectType> gdcmm::SmartPointer< ObjectType >::SmartPointer (const SmartPointer< ObjectType > & p) [inline]`

27.272.2.3 `template<class ObjectType> gdcmm::SmartPointer< ObjectType >::SmartPointer (ObjectType * p) [inline]`

27.272.2.4 `template<class ObjectType> gdcmm::SmartPointer< ObjectType >::SmartPointer (ObjectType const & p) [inline]`

27.272.2.5 `template<class ObjectType> gdcmm::SmartPointer< ObjectType >::~~SmartPointer () [inline]`

27.272.3 Member Function Documentation

27.272.3.1 `template<class ObjectType> ObjectType* gdcm::SmartPointer< ObjectType >::GetPointer () const`
`[inline]`

Explicit function to retrieve the pointer.

27.272.3.2 `template<class ObjectType> gdcm::SmartPointer< ObjectType >::operator ObjectType * () const`
`[inline]`

Return pointer to object.

27.272.3.3 `template<class ObjectType> ObjectType& gdcm::SmartPointer< ObjectType >::operator* () const`
`[inline]`

27.272.3.4 `template<class ObjectType> ObjectType* gdcm::SmartPointer< ObjectType >::operator-> () const`
`[inline]`

Overload operator ->

27.272.3.5 `template<class ObjectType> SmartPointer& gdcm::SmartPointer< ObjectType >::operator= (SmartPointer< ObjectType > const & r)` `[inline]`

Overload operator assignment.

Referenced by `gdcm::SmartPointer< Value >::operator=()`.

27.272.3.6 `template<class ObjectType> SmartPointer& gdcm::SmartPointer< ObjectType >::operator= (ObjectType * r)`
`[inline]`

Overload operator assignment.

27.272.3.7 `template<class ObjectType> SmartPointer& gdcm::SmartPointer< ObjectType >::operator= (ObjectType const & r)` `[inline]`

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

- [gdcmObject.h](#)
- [gdcmSmartPointer.h](#)

27.273 gdcm::network::SOPClassExtendedNegociationSub Class Reference

[SOPClassExtendedNegociationSub](#) PS 3.7 [Table D.3-11](#) SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-ASSOCIATE-AC)

```
#include <gdcmSOPClassExtendedNegociationSub.h>
```

Public Member Functions

- [SOPClassExtendedNegociationSub](#) ()

- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [SetTuple](#) (const char *uid, uint8_t levelofsupport=3, uint8_t levelofdignalsig=0, uint8_t elementcoercion=2)
- size_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

27.273.1 Detailed Description

[SOPClassExtendedNegociationSub](#) PS 3.7 [Table D.3-11](#) SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-ASSOCIATE-AC)

27.273.2 Constructor & Destructor Documentation

27.273.2.1 `gdcm::network::SOPClassExtendedNegociationSub::SOPClassExtendedNegociationSub ()`

27.273.3 Member Function Documentation

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

27.273.3.2 `std::istream& gdcm::network::SOPClassExtendedNegociationSub::Read (std::istream & is)`

27.273.3.3 `void gdcm::network::SOPClassExtendedNegociationSub::SetTuple (const char * uid, uint8_t levelofsupport = 3, uint8_t levelofdignalsig = 0, uint8_t elementcoercion = 2)`

27.273.3.4 `size_t gdcm::network::SOPClassExtendedNegociationSub::Size () const`

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

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

- [gdcmSOPClassExtendedNegociationSub.h](#)

27.274 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](#) ()

27.274.1 Detailed Description

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

Reference PS 3.4 [Table](#) B.5-1 STANDARD SOP CLASSES

27.274.2 Member Typedef Documentation

27.274.2.1 `typedef const char* gdcm::SOPClassUIDToIOD::const(SOPClassUIDToIODType)[2]`

27.274.3 Member Function Documentation

27.274.3.1 `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](#).

27.274.3.2 `static const char* gdcm::SOPClassUIDToIOD::GetIODFromSOPClassUID (const char * sopclassuid) [static]`

27.274.3.3 `static unsigned int gdcm::SOPClassUIDToIOD::GetNumberOfSOPClassToIOD () [static]`

Return the number of SOP Class UID listed internally.

27.274.3.4 `static const char* gdcm::SOPClassUIDToIOD::GetSOPClassUIDFromIOD (const char * iod) [static]`

27.274.3.5 `static SOPClassUIDToIODType& gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIOD (unsigned int i) [static]`

27.274.3.6 `static SOPClassUIDToIODType* gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIODs () [static]`

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

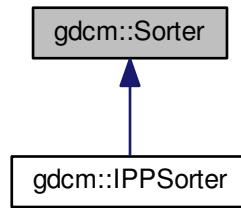
- [gdcmSOPClassUIDToIOD.h](#)

27.275 gdcm::Sorter Class Reference

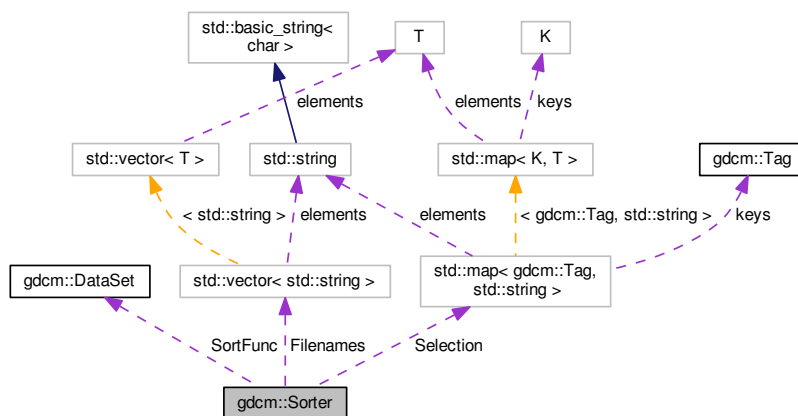
[Sorter](#) General class to do sorting using a custom function You simply need to provide a function of type: [Sorter::Sort←Function](#).

```
#include <gdcmSorter.h>
```

Inheritance diagram for `gdc::Sorter`:



Collaboration diagram for `gdc::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)
- 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](#)

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [Sorter](#) &s)

27.275.1 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](#).

27.275.2 Member Typedef Documentation

27.275.2.1 typedef std::map<[Tag](#),std::string> [gdcm::Sorter::SelectionMap](#) [protected]

27.275.2.2 typedef bool(* [gdcm::Sorter::SortFunction](#)) ([DataSet](#) const &, [DataSet](#) const &)

Set the sort function which compares one dataset to the other.

27.275.3 Constructor & Destructor Documentation

27.275.3.1 `gdcmm::Sorter::Sorter ()`

27.275.3.2 `virtual gdcmm::Sorter::~~Sorter ()` `[virtual]`

27.275.4 Member Function Documentation

27.275.4.1 `bool gdcmm::Sorter::AddSelect (Tag const & tag, const char * value)`

UNSUPPORTED FOR NOW.

27.275.4.2 `const std::vector<std::string>& gdcmm::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](#), [gdcmmorphoplanes.cxx](#), [reslicesphere.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).

27.275.4.3 `void gdcmm::Sorter::Print (std::ostream & os) const`

Print.

Examples:

[gdcmmorphoplanes.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).

Referenced by `gdcmm::operator<<()`.

27.275.4.4 `void gdcmm::Sorter::SetSortFunction (SortFunction f)`

Examples:

[SortImage.cxx](#), and [VolumeSorter.cxx](#).

27.275.4.5 `virtual bool gdcmm::Sorter::Sort (std::vector< std::string > const & filenames)` `[virtual]`

Typically the output of [Directory::GetFileNames\(\)](#)

Reimplemented in [gdcmm::IPPSorter](#).

Examples:

[SortImage.cxx](#).

27.275.4.6 `virtual bool gdcmm::Sorter::StableSort (std::vector< std::string > const & filenames)` `[virtual]`

Examples:

[SortImage.cxx](#), and [VolumeSorter.cxx](#).

27.275.5 Friends And Related Function Documentation

27.275.5.1 `std::ostream& operator<< (std::ostream & _os, const Sorter & s)` [*friend*]

27.275.6 Member Data Documentation

27.275.6.1 `std::vector<std::string> gdcmm::Sorter::FileNames` [*protected*]

27.275.6.2 `std::map<Tag, std::string> gdcmm::Sorter::Selection` [*protected*]

27.275.6.3 **SortFunction** `gdcmm::Sorter::SortFunc` [*protected*]

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

- [gdcmmSorter.h](#)

27.276 gdcmm::Spacing Class Reference

Class for [Spacing](#).

```
#include <gdcmmSpacing.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)

27.276.1 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

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

27.276.2 Member Enumeration Documentation

27.276.2.1 enum gdcm::Spacing::SpacingType

Enumerator

DETECTOR
MAGNIFIED
CALIBRATED
UNKNOWN

27.276.3 Constructor & Destructor Documentation

27.276.3.1 gdcm::Spacing::Spacing ()

27.276.3.2 gdcm::Spacing::~~Spacing ()

27.276.4 Member Function Documentation

27.276.4.1 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](#)

27.277 gdcm::Spectroscopy Class Reference

[Spectroscopy](#) class.

```
#include <gdcmSpectroscopy.h>
```

Public Member Functions

- [Spectroscopy](#) ()

27.277.1 Detailed Description

[Spectroscopy](#) class.

27.277.2 Constructor & Destructor Documentation

27.277.2.1 [gdcm::Spectroscopy::Spectroscopy](#) () [inline]

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

- [gdcmSpectroscopy.h](#)

27.278 gdcm::SplitMosaicFilter Class Reference

[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.

```
#include <gdcmSplitMosaicFilter.h>
```

Public Member Functions

- [SplitMosaicFilter](#) ()
- [~SplitMosaicFilter](#) ()
- bool [ComputeMOSAICDimensions](#) (unsigned int dims[3])
- [File](#) & [GetFile](#) ()
- const [File](#) & [GetFile](#) () const
- const [Image](#) & [GetImage](#) () const
- [Image](#) & [GetImage](#) ()
- void [SetFile](#) (const [File](#) &f)
- void [SetImage](#) (const [Image](#) &image)
- bool [Split](#) ()

Split the SIEMENS MOSAIC image.

27.278.1 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.

27.278.2 Constructor & Destructor Documentation

27.278.2.1 `gdcm::SplitMosaicFilter::SplitMosaicFilter ()`

27.278.2.2 `gdcm::SplitMosaicFilter::~~SplitMosaicFilter ()`

27.278.3 Member Function Documentation

27.278.3.1 `bool gdcm::SplitMosaicFilter::ComputeMOSAICDimensions (unsigned int dims[3])`

Compute the new dimensions according to private information stored in the MOSAIC header.

27.278.3.2 `File& gdcm::SplitMosaicFilter::GetFile ()` `[inline]`

27.278.3.3 `const File& gdcm::SplitMosaicFilter::GetFile () const` `[inline]`

27.278.3.4 `const Image& gdcm::SplitMosaicFilter::GetImage () const` `[inline]`

27.278.3.5 `Image& gdcm::SplitMosaicFilter::GetImage ()` `[inline]`

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

27.278.3.7 `void gdcm::SplitMosaicFilter::SetImage (const Image & image)`

27.278.3.8 `bool gdcm::SplitMosaicFilter::Split ()`

Split the SIEMENS MOSAIC image.

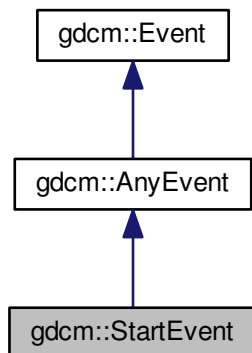
The documentation for this class was generated from the following file:

- [gdcmSplitMosaicFilter.h](#)

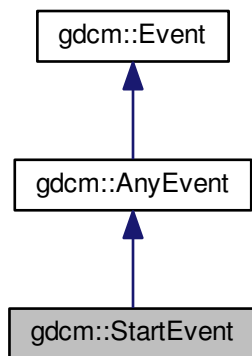
27.279 gdcm::StartEvent Class Reference

```
#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](#)

27.280 `gdcm::static_assert_test< x >` Struct Template Reference

```
#include <gdcmStaticAssert.h>
```

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

- [gdcmStaticAssert.h](#)

27.281 `gdcm::STATIC_ASSERTION_FAILURE< x >` Struct Template Reference

```
#include <gdcmStaticAssert.h>
```

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

- [gdcmStaticAssert.h](#)

27.282 `gdcm::STATIC_ASSERTION_FAILURE< true >` Struct Template Reference

```
#include <gdcmStaticAssert.h>
```

Public Types

- enum { [value](#) = 1 }

27.282.1 Member Enumeration Documentation

27.282.1.1 anonymous enum

Enumerator

value

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

- [gdcmStaticAssert.h](#)

27.283 `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)

27.283.1 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](#).

27.283.2 Constructor & Destructor Documentation

27.283.2.1 `gdcm::StreamImageReader::StreamImageReader ()`

27.283.2.2 `virtual gdcm::StreamImageReader::~~StreamImageReader () [virtual]`

27.283.3 Member Function Documentation

27.283.3.1 `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](#).

27.283.3.2 `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](#).

27.283.3.3 `uint32_t gdcmm::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](#).

27.283.3.4 `std::vector<unsigned int> gdcmm::StreamImageReader::GetDimensionsValueForResolution (unsigned int)`

27.283.3.5 `File const& gdcmm::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](#).

27.283.3.6 `bool gdcmm::StreamImageReader::Read (char * inReadBuffer, const std::size_t & inBufferLength)`

Read the DICOM image. There are three reasons for failure:

1. The extent is not set
2. the conversion from `char*` to `std::ostream` (internally) fails
3. the given buffer isn't large enough to accommodate the desired pixel extent. This method has been implemented to look similar to the `metainageio` in `itk` MUST have an extent defined, or else `Read` will return false. If no particular extent is required, use [ImageReader](#) instead.

Examples:

[StreamImageReaderTest.cxx](#).

27.283.3.7 `virtual bool gdcmm::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](#).

27.283.3.8 `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](#).

27.283.3.9 `void gdcm::StreamImageReader::SetStream (std::istream & inStream)`

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

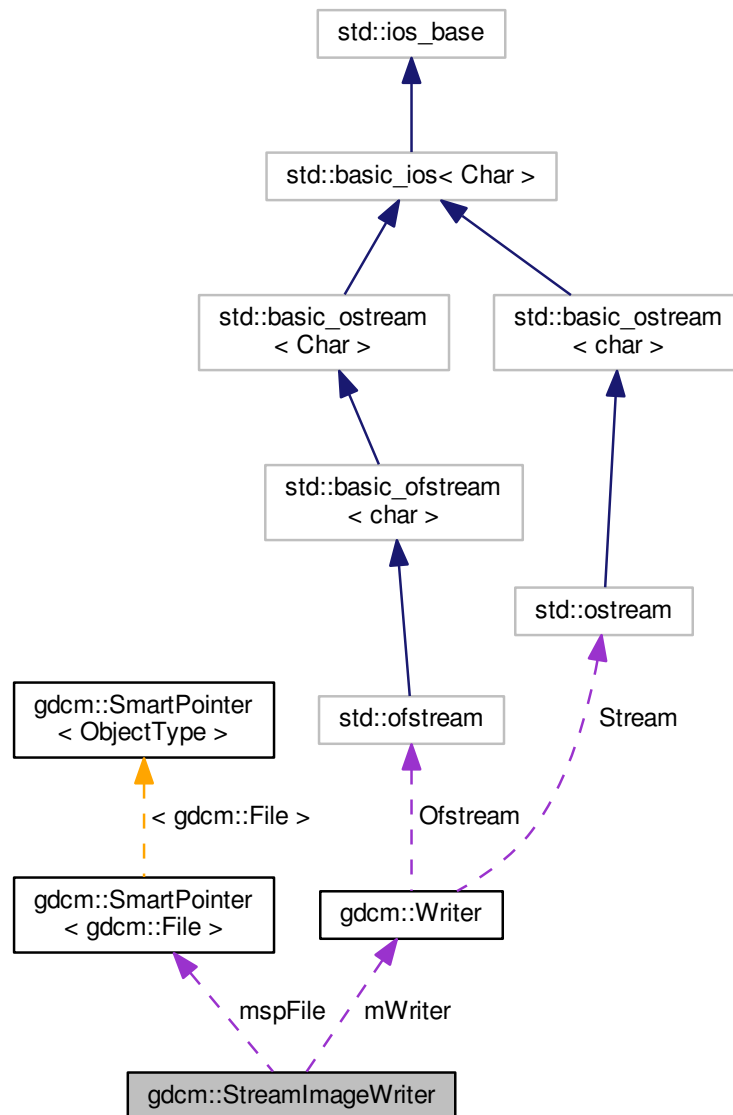
- [gdcmStreamImageReader.h](#)

27.284 `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](#)

27.284.1 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](#).

27.284.2 Constructor & Destructor Documentation

27.284.2.1 `gdcm::StreamImageWriter::StreamImageWriter ()`

27.284.2.2 `virtual gdcm::StreamImageWriter::~~StreamImageWriter ()` [virtual]

27.284.3 Member Function Documentation

27.284.3.1 `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](#).

27.284.3.2 `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](#).

27.284.3.3 `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](#).

27.284.3.4 `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](#).

27.284.3.5 `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.

27.284.3.6 `void gdcm::StreamImageWriter::SetStream (std::ostream & inStream)`

Examples:

[Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

27.284.3.7 `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 `metainageio` 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](#).

27.284.3.8 `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](#).

27.284.3.9 `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

27.284.3.10 `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.

27.284.4 Member Data Documentation

27.284.4.1 `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.

27.284.4.2 `int gdcM::StreamImageWriter::mElementOffsets1` [protected]

27.284.4.3 `SmartPointer<File> gdcM::StreamImageWriter::mspFile` [protected]

27.284.4.4 `Writer gdcM::StreamImageWriter::mWriter` [protected]

27.284.4.5 `uint16_t gdcM::StreamImageWriter::mXMax` [protected]

27.284.4.6 `uint16_t gdcM::StreamImageWriter::mXMin` [protected]

27.284.4.7 `uint16_t gdcM::StreamImageWriter::mYMax` [protected]

27.284.4.8 `uint16_t gdcM::StreamImageWriter::mYMin` [protected]

27.284.4.9 `uint16_t gdcM::StreamImageWriter::mZMax` [protected]

27.284.4.10 `uint16_t gdcM::StreamImageWriter::mZMin` [protected]

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

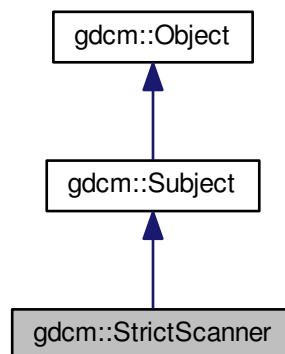
- [gdcMStreamImageWriter.h](#)

27.285 gdcM::StrictScanner Class Reference

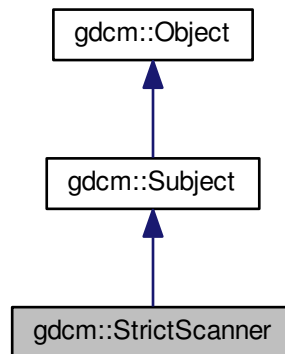
StrictScanner This filter is meant for quickly browsing a [FileSet](#) (a set of files on disk). Special consideration are taken so as to read the minimum amount of information in each file in order to retrieve the user specified set of DICOM [Attribute](#).

```
#include <gdcMStrictScanner.h>
```

Inheritance diagram for gdcM::StrictScanner:



Collaboration diagram for gdcm::StrictScanner:



Classes

- struct [Itstr](#)

Public Types

- typedef MappingType::const_iterator [ConstIterator](#)
- typedef std::map< const char *, [TagToValue](#), [Itstr](#) > [MappingType](#)
- typedef std::map< [Tag](#), const char * > [TagToValue](#)
- typedef TagToValue::value_type [TagToValueValueType](#)
- typedef std::set< std::string > [ValuesType](#)

Public Member Functions

- [StrictScanner](#) ()
- [~StrictScanner](#) ()
- 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::FilenamesType GetKeys](#) () const
- [TagToValue](#) const & [GetMapping](#) (const char *filename) const
Get the std::map mapping filenames to value for file 'filename'.
- [TagToValue](#) const & [GetMappingFromTagToValue](#) ([Tag](#) const &t, const char *value) const
See [GetFilenameFromTagToValue\(\)](#). This is simply [GetFilenameFromTagToValue](#) followed.
- [MappingType](#) const & [GetMappings](#) () const
Mappings are the mapping from a particular tag to the map, mapping filename to value:
- [Directory::FilenamesType GetOrderedValues](#) ([Tag](#) const &t) const
- const char * [GetValue](#) (const char *filename, [Tag](#) const &t) const
- [ValuesType](#) const & [GetValues](#) () const
Get all the values found (in lexicographic order)
- [ValuesType GetValues](#) ([Tag](#) const &t) const
Get all the values found (in lexicographic order) associated with [Tag](#) 't'.
- bool [IsKey](#) (const char *filename) const
- void [Print](#) (std::ostream &os) const
Print result.
- bool [Scan](#) ([Directory::FilenamesType](#) 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)

27.285.1 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](#).

27.285.2 Member Typedef Documentation

27.285.2.1 `typedef MappingType::const_iterator gdcm::StrictScanner::ConstIterator`

27.285.2.2 `typedef std::map<const char *, TagToValue, Itstr> gdcm::StrictScanner::MappingType`

27.285.2.3 `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)`

27.285.2.4 `typedef TagToValue::value_type gdcm::StrictScanner::TagToValueValueType`

27.285.2.5 `typedef std::set< std::string > gdcm::StrictScanner::ValuesType`

27.285.3 Constructor & Destructor Documentation

27.285.3.1 `gdcm::StrictScanner::StrictScanner () [inline]`

27.285.3.2 `gdcm::StrictScanner::~~StrictScanner ()`

27.285.4 Member Function Documentation

27.285.4.1 `void gdcm::StrictScanner::AddPrivateTag (PrivateTag const & t)`

27.285.4.2 `void gdcm::StrictScanner::AddSkipTag (Tag const & t)`

Add a tag that will need to be skipped. Those are root level skip tags.

27.285.4.3 `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](#).

27.285.4.4 **ConstIterator** `gdcmm::StrictScanner::Begin () const` `[inline]`

27.285.4.5 `void gdcmm::StrictScanner::ClearSkipTags ()`

27.285.4.6 `void gdcmm::StrictScanner::ClearTags ()`

27.285.4.7 **ConstIterator** `gdcmm::StrictScanner::End () const` `[inline]`

27.285.4.8 **Directory::FilenameType** `gdcmm::StrictScanner::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'

27.285.4.9 `const char* gdcmm::StrictScanner::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'

27.285.4.10 **Directory::FilenameType** `const& gdcmm::StrictScanner::GetFileNames () const` `[inline]`

27.285.4.11 **Directory::FilenameType** `gdcmm::StrictScanner::GetKeys () const`

Return the list of filename that are key in the internal map, which means those filename were properly parsed

27.285.4.12 **TagToValue** `const& gdcmm::StrictScanner::GetMapping (const char * filename) const`

Get the std::map mapping filenames to value for file 'filename'.

Examples:

[SimpleScanner.cxx](#).

27.285.4.13 **TagToValue** `const& gdcmm::StrictScanner::GetMappingFromTagToValue (Tag const & t, const char * value) const`

See [GetFilenameFromTagToValue\(\)](#). This is simply GetFilenameFromTagToValue followed.

27.285.4.14 **MappingType** `const& gdcmm::StrictScanner::GetMappings () const` `[inline]`

Mappings are the mapping from a particular tag to the map, mapping filename to value:

27.285.4.15 **Directory::FilenameType** `gdcmm::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)

27.285.4.16 `const char* gdcmm::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 !

27.285.4.17 `ValueType const& gdcm::StrictScanner::GetValues () const` `[inline]`

Get all the values found (in lexicographic order)

27.285.4.18 `ValueType gdcm::StrictScanner::GetValues (Tag const & t) const`

Get all the values found (in lexicographic order) associated with [Tag 't'](#).

27.285.4.19 `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](#).

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

for wrapped language: instantiate a reference counted object

27.285.4.21 `void gdcm::StrictScanner::Print (std::ostream & os) const` `[virtual]`

Print result.

Reimplemented from [gdcm::Object](#).

Referenced by `gdcm::operator<<()`.

27.285.4.22 `void gdcm::StrictScanner::ProcessPublicTag (StringFilter & sf, const char * filename)` `[protected]`

27.285.4.23 `bool gdcm::StrictScanner::Scan (Directory::FilenamesType const & filenames)`

Start the scan !

Examples:

[SimpleScanner.cxx](#).

27.285.5 Friends And Related Function Documentation

27.285.5.1 `std::ostream& operator<< (std::ostream & _os, const StrictScanner & s)` `[friend]`

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

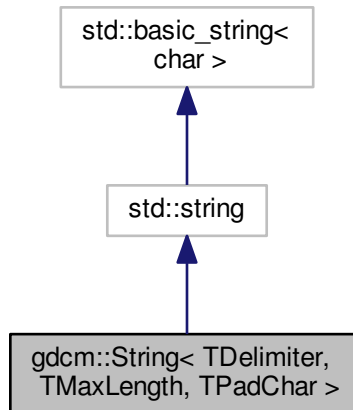
- [gdcmStrictScanner.h](#)

27.286 `gdcm::String< TDelimiter, TMaxLength, TPadChar >` Class Template Reference

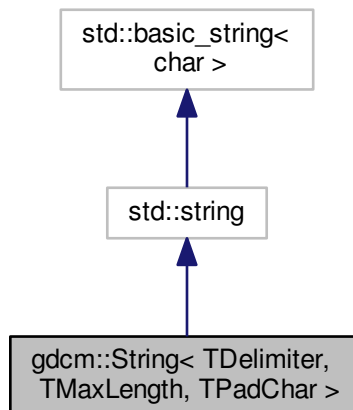
[String](#).

```
#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 [value_type](#) *s)
- [String](#) (const [value_type](#) *s, [size_type](#) n)
- [String](#) (const std::string &s, [size_type](#) pos=0, [size_type](#) n=npos)
- bool [IsValid](#) () const
return if string is valid
- [operator const char *](#) () const
WARNING: Trailing \0 might be lost in this operation:
- std::string [Trim](#) () const
- [gdcm::String](#)< TDelimiter, TMaxLength, TPadChar > [Truncate](#) () const

Static Public Member Functions

- static std::string [Trim](#) (const char *input)

27.286.1 Detailed Description

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>class gdcm::String< TDelimiter, TMaxLength, TPadChar >
```

[String](#).

Note

TDelimiter template parameter is used to separate multiple [String](#) (VM1 >) TMaxLength is only a hint. Noone actually respect the max length TPadChar is the string padding (0 or space)

27.286.2 Member Typedef Documentation

27.286.2.1 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::const_iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::const_iterator`

27.286.2.2 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::const_reference gdcmm::String< TDelimiter, TMaxLength, TPadChar >::const_reference`

27.286.2.3 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::const_reverse_iterator gdcmm::String< TDelimiter, TMaxLength, TPadChar >::const_reverse_iterator`

27.286.2.4 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::difference_type gdcmm::String< TDelimiter, TMaxLength, TPadChar >::difference_type`

27.286.2.5 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::iterator gdcmm::String< TDelimiter, TMaxLength, TPadChar >::iterator`

27.286.2.6 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::pointer gdcmm::String< TDelimiter, TMaxLength, TPadChar >::pointer`

27.286.2.7 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::reference gdcmm::String< TDelimiter, TMaxLength, TPadChar >::reference`

27.286.2.8 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::reverse_iterator gdcmm::String< TDelimiter, TMaxLength, TPadChar >::reverse_iterator`

27.286.2.9 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::size_type gdcmm::String< TDelimiter, TMaxLength, TPadChar >::size_type`

27.286.2.10 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::value_type gdcmm::String< TDelimiter, TMaxLength, TPadChar >::value_type`

27.286.3 Constructor & Destructor Documentation

27.286.3.1 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> gdcmm::String< TDelimiter, TMaxLength, TPadChar >::String () [inline]`

[String](#) constructors.

27.286.3.2 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> gdcmm::String< TDelimiter, TMaxLength, TPadChar >::String (const value_type * s) [inline]`

27.286.3.3 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> gdcmm::String< TDelimiter, TMaxLength, TPadChar >::String (const value_type * s, size_type n) [inline]`

27.286.3.4 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> gdcmm::String< TDelimiter, TMaxLength, TPadChar >::String (const std::string & s, size_type pos = 0, size_type n = npos) [inline]`

27.286.4 Member Function Documentation

27.286.4.1 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> bool gdcmm::String< TDelimiter, TMaxLength, TPadChar >::IsValid () const [inline]`

return if string is valid

Referenced by `gdcmm::String< TDelimiter, TMaxLength, TPadChar >::Truncate()`.

27.286.4.2 `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:

27.286.4.3 `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...

27.286.4.4 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = '> static std::string gdcm::String< TDelimiter, TMaxLength, TPadChar >::Trim (const char * input) [inline], [static]`

27.286.4.5 `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](#)

27.287 gdcm::StringFilter Class Reference

[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.

```
#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, [VL](#) const &vl)
- 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 [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

27.287.1 Detailed Description

[StringFilter](#) [StringFilter](#) is the class that make gdc2.x looks more like gdc1 and transform the binary blob contained in a [DataElement](#) into a string, typically this is a nice feature to have for wrapped language.

Examples:

[ReadAndPrintAttributes.cxx](#).

27.287.2 Constructor & Destructor Documentation

27.287.2.1 `gdc2::StringFilter::StringFilter ()`

27.287.2.2 `gdc2::StringFilter::~~StringFilter ()`

27.287.3 Member Function Documentation

27.287.3.1 `bool gdc2::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

27.287.3.2 `bool gdc2::StringFilter::ExecuteQuery (std::string const & query, DataSet const & ds, std::string & value) const`
[protected]

27.287.3.3 `std::string gdc2::StringFilter::FromString (const Tag & t, const char * value, VL const & vl)`

27.287.3.4 `std::string gdc2::StringFilter::FromString (const Tag & t, const char * value, size_t len)`

Convert to string the char array defined by the pair (value,len)

27.287.3.5 `File& gdc2::StringFilter::GetFile ()` [inline]

27.287.3.6 `const File& gdc2::StringFilter::GetFile () const` [inline]

27.287.3.7 `void gdc2::StringFilter::SetDicts (const Dicts & dicts)`

Allow user to pass in there own dicts.

27.287.3.8 `void gdc2::StringFilter::SetFile (const File & f)` [inline]

Set/Get [File](#).

Examples:

[ReadAndPrintAttributes.cxx](#), and [SimplePrintPatientName.cs](#).

27.287.3.9 `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:

[ReadAndPrintAttributes.cxx](#).

27.287.3.10 `std::string gdcm::StringFilter::ToString (const Tag & t) const`

Directly from a [Tag](#):

27.287.3.11 `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](#).

27.287.3.12 `std::pair<std::string, std::string> gdcm::StringFilter::ToStringPair (const Tag & t) const`

Directly from a [Tag](#):

27.287.3.13 `std::pair<std::string, std::string> gdcm::StringFilter::ToStringPair (const Tag & t, DataSet const & ds) const`
[protected]

27.287.3.14 `void gdcm::StringFilter::UseDictAlways (bool)` [inline]

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

- [gdcmStringFilter.h](#)

27.288 gdcm::Study Class Reference

[Study](#).

```
#include <gdcmStudy.h>
```

Public Member Functions

- [Study](#) ()

27.288.1 Detailed Description

[Study](#).

27.288.2 Constructor & Destructor Documentation

27.288.2.1 `gdcm::Study::Study ()` `[inline]`

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

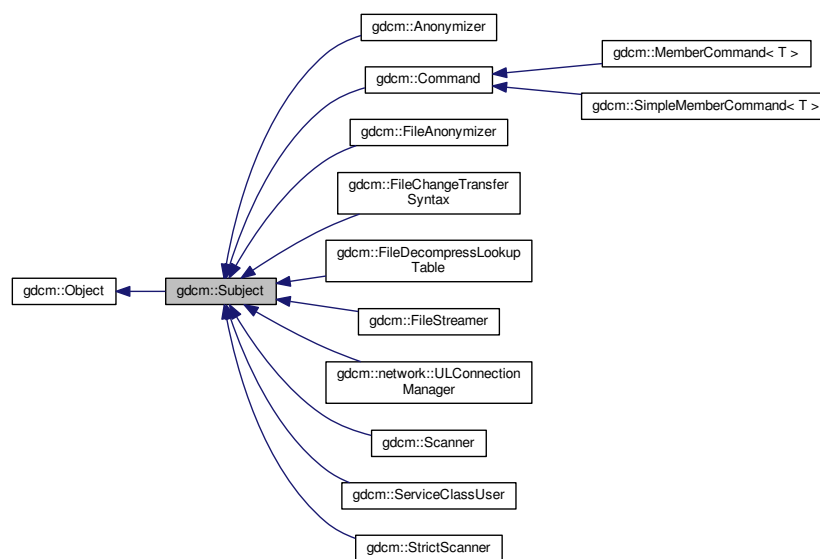
- [gdcmStudy.h](#)

27.289 `gdcm::Subject` Class Reference

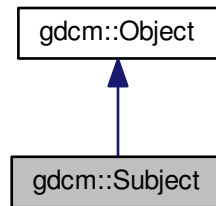
[Subject.](#)

```
#include <gdcmSubject.h>
```

Inheritance diagram for `gdcm::Subject`:



Collaboration diagram for gdcm::Subject:



Public Member Functions

- [Subject](#) ()
- [~Subject](#) ()
- 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

27.289.1 Detailed Description

[Subject](#).

See also

[Command Event](#)

Examples:

[SimpleScanner.cxx](#).

27.289.2 Constructor & Destructor Documentation

27.289.2.1 `gdcm::Subject::Subject ()`

27.289.2.2 `gdcm::Subject::~~Subject ()`

27.289.3 Member Function Documentation

27.289.3.1 unsigned long gdcM::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 [gdcM::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

27.289.3.2 unsigned long gdcM::Subject::AddObserver (const Event & event, Command *) const

27.289.3.3 Command* gdcM::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.

27.289.3.4 bool gdcM::Subject::HasObserver (const Event & event) const

Return true if an observer is registered for this event.

27.289.3.5 void gdcM::Subject::InvokeEvent (const Event &)

Call Execute on all the Commands observing this event id.

27.289.3.6 void gdcM::Subject::InvokeEvent (const Event &) const

Call Execute on all the Commands observing this event id. The actions triggered by this call doesn't modify this object.

27.289.3.7 void gdcM::Subject::RemoveAllObservers ()

Remove all observers .

27.289.3.8 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:

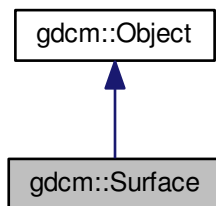
- [gdcMSubject.h](#)

27.290 gdcM::Surface Class Reference

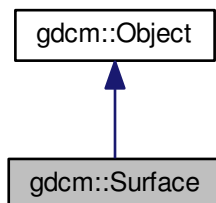
This class defines a SURFACE IE. This members are taken from required surface mesh module attributes.

```
#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](#) ()
- virtual [~Surface](#) ()

- [SegmentHelper::BasicCodedEntry](#) const & [GetAlgorithmFamily](#) () const
- [SegmentHelper::BasicCodedEntry](#) & [GetAlgorithmFamily](#) ()
- 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](#) const & [GetMeshPrimitive](#) () const
- [MeshPrimitive](#) & [GetMeshPrimitive](#) ()
- unsigned long [GetNumberOfSurfacePoints](#) () const
- unsigned long [GetNumberOfVectors](#) () const
- const [DataElement](#) & [GetPointCoordinatesData](#) () const
- [DataElement](#) & [GetPointCoordinatesData](#) ()
- const float * [GetPointPositionAccuracy](#) () const
- const float * [GetPointsBoundingBoxCoordinates](#) () const
- [SegmentHelper::BasicCodedEntry](#) const & [GetProcessingAlgorithm](#) () const
- [SegmentHelper::BasicCodedEntry](#) & [GetProcessingAlgorithm](#) ()
- 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
- const [DataElement](#) & [GetVectorCoordinateData](#) () const
- [DataElement](#) & [GetVectorCoordinateData](#) ()
- 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 unsigned short vl[3])

- void [SetRecommendedDisplayCIELabValue](#) (const unsigned short vl, const unsigned int idx=0)
- void [SetRecommendedDisplayCIELabValue](#) (const std::vector< unsigned short > &vl)
- 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

27.290.1 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

27.290.2 Member Enumeration Documentation

27.290.2.1 enum gdcm::Surface::STATES

Enumerator

NO

YES

UNKNOWN

STATES_END

27.290.2.2 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

27.290.3 Constructor & Destructor Documentation

27.290.3.1 `gdcmm::Surface::Surface ()`

27.290.3.2 `virtual gdcmm::Surface::~~Surface () [virtual]`

27.290.4 Member Function Documentation

27.290.4.1 `SegmentHelper::BasicCodedEntry const& gdcmm::Surface::GetAlgorithmFamily () const`

27.290.4.2 `SegmentHelper::BasicCodedEntry& gdcmm::Surface::GetAlgorithmFamily ()`

27.290.4.3 `const char* gdcmm::Surface::GetAlgorithmName () const`

27.290.4.4 `const char* gdcmm::Surface::GetAlgorithmVersion () const`

27.290.4.5 `const float* gdcmm::Surface::GetAxisOfRotation () const`

Note

Pointer is null if undefined

27.290.4.6 `const float* gdcmm::Surface::GetCenterOfRotation () const`

Note

Pointer is null if undefined

27.290.4.7 `STATES gdcmm::Surface::GetFiniteVolume () const`

27.290.4.8 `STATES gdcmm::Surface::GetManifold () const`

27.290.4.9 `float gdcmm::Surface::GetMaximumPointDistance () const`

27.290.4.10 `float gdcmm::Surface::GetMeanPointDistance () const`

27.290.4.11 `MeshPrimitive const& gdcmm::Surface::GetMeshPrimitive () const`

27.290.4.12 `MeshPrimitive& gdcmm::Surface::GetMeshPrimitive ()`

27.290.4.13 unsigned long gdcm::Surface::GetNumberOfSurfacePoints () const

27.290.4.14 unsigned long gdcm::Surface::GetNumberOfVectors () const

27.290.4.15 const DataElement& gdcm::Surface::GetPointCoordinatesData () const

27.290.4.16 DataElement& gdcm::Surface::GetPointCoordinatesData ()

27.290.4.17 const float* gdcm::Surface::GetPointPositionAccuracy () const

Note

Pointer is null if undefined

27.290.4.18 const float* gdcm::Surface::GetPointsBoundingBoxCoordinates () const

Note

Pointer is null if undefined

27.290.4.19 SegmentHelper::BasicCodedEntry const& gdcm::Surface::GetProcessingAlgorithm () const

27.290.4.20 SegmentHelper::BasicCodedEntry& gdcm::Surface::GetProcessingAlgorithm ()

27.290.4.21 const unsigned short* gdcm::Surface::GetRecommendedDisplayCIELabValue () const

27.290.4.22 unsigned short gdcm::Surface::GetRecommendedDisplayCIELabValue (const unsigned int *idx*) const

27.290.4.23 unsigned short gdcm::Surface::GetRecommendedDisplayGrayscaleValue () const

27.290.4.24 float gdcm::Surface::GetRecommendedPresentationOpacity () const

27.290.4.25 VIEWType gdcm::Surface::GetRecommendedPresentationType () const

27.290.4.26 static STATES gdcm::Surface::GetSTATES (const char * *state*) [static]

27.290.4.27 static const char* gdcm::Surface::GetSTATESString (STATES *state*) [static]

27.290.4.28 const char* gdcm::Surface::GetSurfaceComments () const

27.290.4.29 unsigned long gdcm::Surface::GetSurfaceNumber () const

27.290.4.30 bool gdcm::Surface::GetSurfaceProcessing () const

27.290.4.31 const char* gdcm::Surface::GetSurfaceProcessingDescription () const

27.290.4.32 float gdcm::Surface::GetSurfaceProcessingRatio () const

27.290.4.33 const float* gdcm::Surface::GetVectorAccuracy () const

27.290.4.34 const DataElement& gdcm::Surface::GetVectorCoordinateData () const

- 27.290.4.35 **DataElement& gdcmm::Surface::GetVectorCoordinateData ()**
- 27.290.4.36 **unsigned short gdcmm::Surface::GetVectorDimensionality () const**
- 27.290.4.37 **static VIEWType gdcmm::Surface::GetVIEWType (const char * *type*) [static]**
- 27.290.4.38 **static const char* gdcmm::Surface::GetVIEWTypeString (VIEWType *type*) [static]**
- 27.290.4.39 **void gdcmm::Surface::SetAlgorithmFamily (SegmentHelper::BasicCodedEntry const & *BSE*)**
- 27.290.4.40 **void gdcmm::Surface::SetAlgorithmName (const char * *str*)**
- 27.290.4.41 **void gdcmm::Surface::SetAlgorithmVersion (const char * *str*)**
- 27.290.4.42 **void gdcmm::Surface::SetAxisOfRotation (const float * *axis*)**
- 27.290.4.43 **void gdcmm::Surface::SetCenterOfRotation (const float * *center*)**
- 27.290.4.44 **void gdcmm::Surface::SetFiniteVolume (STATES *state*)**
- 27.290.4.45 **void gdcmm::Surface::SetManifold (STATES *state*)**
- 27.290.4.46 **void gdcmm::Surface::SetMaximumPointDistance (float *maximum*)**
- 27.290.4.47 **void gdcmm::Surface::SetMeanPointDistance (float *average*)**
- 27.290.4.48 **void gdcmm::Surface::SetMeshPrimitive (MeshPrimitive & *mp*)**
- 27.290.4.49 **void gdcmm::Surface::SetNumberOfSurfacePoints (const unsigned long *nb*)**
- 27.290.4.50 **void gdcmm::Surface::SetNumberOfVectors (const unsigned long *nb*)**
- 27.290.4.51 **void gdcmm::Surface::SetPointCoordinatesData (DataElement const & *de*)**
- 27.290.4.52 **void gdcmm::Surface::SetPointPositionAccuracy (const float * *accuracies*)**
- 27.290.4.53 **void gdcmm::Surface::SetPointsBoundingBoxCoordinates (const float * *coordinates*)**
- 27.290.4.54 **void gdcmm::Surface::SetProcessingAlgorithm (SegmentHelper::BasicCodedEntry const & *BSE*)**
- 27.290.4.55 **void gdcmm::Surface::SetRecommendedDisplayCIELabValue (const unsigned short *vl*[3])**
- 27.290.4.56 **void gdcmm::Surface::SetRecommendedDisplayCIELabValue (const unsigned short *vl*, const unsigned int *idx* = 0)**
- 27.290.4.57 **void gdcmm::Surface::SetRecommendedDisplayCIELabValue (const std::vector< unsigned short > & *vl*)**
- 27.290.4.58 **void gdcmm::Surface::SetRecommendedDisplayGrayscaleValue (const unsigned short *vl*)**
- 27.290.4.59 **void gdcmm::Surface::SetRecommendedPresentationOpacity (const float *opacity*)**
- 27.290.4.60 **void gdcmm::Surface::SetRecommendedPresentationType (VIEWType *type*)**

- 27.290.4.61 void gdcm::Surface::SetSurfaceComments (const char * *comment*)
- 27.290.4.62 void gdcm::Surface::SetSurfaceNumber (const unsigned long *nb*)
- 27.290.4.63 void gdcm::Surface::SetSurfaceProcessing (bool *b*)
- 27.290.4.64 void gdcm::Surface::SetSurfaceProcessingDescription (const char * *description*)
- 27.290.4.65 void gdcm::Surface::SetSurfaceProcessingRatio (const float *ratio*)
- 27.290.4.66 void gdcm::Surface::SetVectorAccuracy (const float * *accuracy*)
- 27.290.4.67 void gdcm::Surface::SetVectorCoordinateData (DataElement const & *de*)
- 27.290.4.68 void gdcm::Surface::SetVectorDimensionality (const unsigned short *dim*)

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

- [gdcmSurface.h](#)

27.291 gdcm::SurfaceHelper Class Reference

[SurfaceHelper](#) Helper class for [Surface](#) object.

```
#include <gdcmSurfaceHelper.h>
```

Public Types

- typedef std::vector< unsigned short > [ColorArray](#)

Static Public Member Functions

- template<typename T, typename U >
static std::vector< T > [RecommendedDisplayCIELabToRGB](#) (const [ColorArray](#) &CIELab, const U range↔
Max=255)
Convert a DICOM CIE-Lab (after reading) color into RGB.
- template<typename U >
static std::vector< float > [RecommendedDisplayCIELabToRGB](#) (const [ColorArray](#) &CIELab, const U range↔
Max=255)
Convert a DICOM CIE-Lab (after reading) color into RGB.
- template<typename T, typename U >
static [ColorArray](#) [RGBToRecommendedDisplayCIELab](#) (const std::vector< T > &RGB, const U rangeMax=255)
Convert a RGB color into DICOM CIE-Lab (ready to write).
- template<typename T, typename U >
static unsigned short [RGBToRecommendedDisplayGrayscale](#) (const std::vector< T > &RGB, const U range↔
Max=255)
Convert a RGB color into DICOM grayscale (ready to write).

27.291.1 Detailed Description

[SurfaceHelper](#) Helper class for [Surface](#) object.

27.291.2 Member Typedef Documentation

27.291.2.1 `typedef std::vector< unsigned short > gdcm::SurfaceHelper::ColorArray`

27.291.3 Member Function Documentation

27.291.3.1 `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.

27.291.3.2 `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.
----------	---

27.291.3.3 `template<typename T , typename U > SurfaceHelper::ColorArray gdcm::SurfaceHelper::RGBToRecommendedDisplayCIELab (const std::vector< T > & RGB, const U rangeMax = 255) [static]`

Convert a RGB color into DICOM CIE-Lab (ready to write).

See also

PS 3.3 C.10.7.1.1

Parameters

<i>RGB</i>	RGB array.
<i>rangeMax</i>	Max value of the RGB range.

Template Parameters

<i>T</i>	Type of RGB components.
<i>U</i>	Type of rangeMax value.

27.291.3.4 `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:

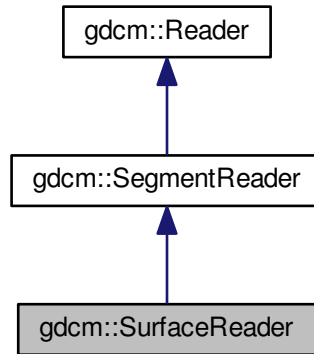
- [gdcmSurfaceHelper.h](#)

27.292 gdcm::SurfaceReader Class Reference

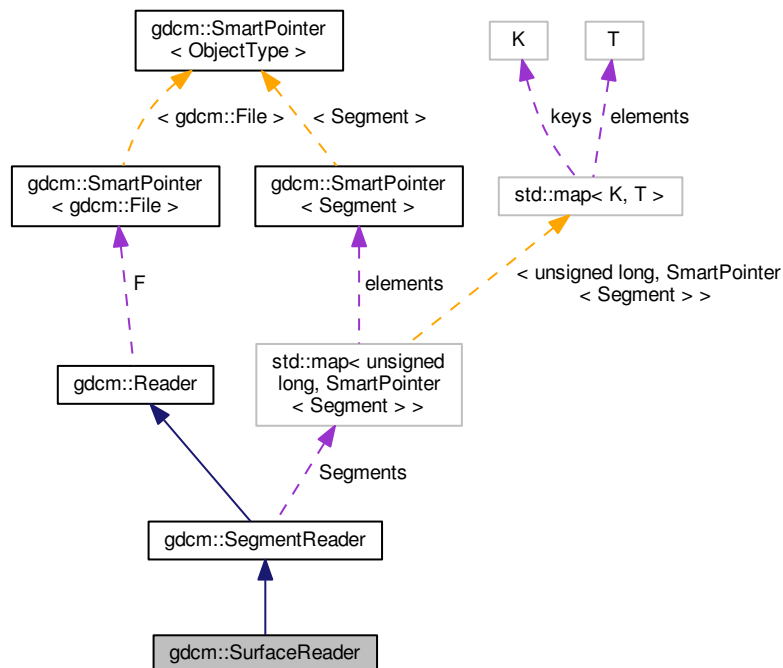
This class defines a SURFACE IE reader. It reads surface mesh module attributes.

```
#include <gdcmSurfaceReader.h>
```

Inheritance diagram for `gdcM::SurfaceReader`:



Collaboration diagram for `gdcM::SurfaceReader`:



Public Member Functions

- [SurfaceReader](#) ()
- virtual [~SurfaceReader](#) ()
- unsigned long [GetNumberOfSurfaces](#) () const
- virtual bool [Read](#) ()

Read.

Protected Member Functions

- bool [ReadPointMacro](#) ([SmartPointer](#)< [Surface](#) > surface, const [DataSet](#) &surfaceDS)
- bool [ReadSurface](#) (const [Item](#) &surfacerItem, const unsigned long idx)
- bool [ReadSurfaces](#) ()

Additional Inherited Members

27.292.1 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

27.292.2 Constructor & Destructor Documentation

27.292.2.1 `gdcm::SurfaceReader::SurfaceReader ()`

27.292.2.2 `virtual gdcm::SurfaceReader::~~SurfaceReader ()` [virtual]

27.292.3 Member Function Documentation

27.292.3.1 `unsigned long gdcm::SurfaceReader::GetNumberOfSurfaces ()` const

27.292.3.2 `virtual bool gdcm::SurfaceReader::Read ()` [virtual]

Read.

Reimplemented from [gdcm::SegmentReader](#).

27.292.3.3 `bool gdcm::SurfaceReader::ReadPointMacro (SmartPointer< Surface > surface, const DataSet & surfaceDS)`
[protected]

27.292.3.4 `bool gdcm::SurfaceReader::ReadSurface (const Item & surfacerItem, const unsigned long idx)` [protected]

27.292.3.5 `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

27.293.1 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

27.293.2 Constructor & Destructor Documentation

27.293.2.1 `gdcm::SurfaceWriter::SurfaceWriter ()`

27.293.2.2 `virtual gdcm::SurfaceWriter::~~SurfaceWriter () [virtual]`

27.293.3 Member Function Documentation

27.293.3.1 `void gdcm::SurfaceWriter::ComputeNumberOfSurfaces () [protected]`

27.293.3.2 `unsigned long gdcm::SurfaceWriter::GetNumberOfSurfaces ()`

27.293.3.3 `bool gdcm::SurfaceWriter::PrepareWrite () [protected]`

27.293.3.4 `bool gdcm::SurfaceWriter::PrepareWritePointMacro (SmartPointer< Surface > surface, DataSet & surfaceDS, const TransferSyntax & ts) [protected]`

27.293.3.5 `void gdcm::SurfaceWriter::SetNumberOfSurfaces (const unsigned long nb)`

27.293.3.6 `bool gdcm::SurfaceWriter::Write () [virtual]`

Write.

Reimplemented from [gdcm::SegmentWriter](#).

27.293.4 Member Data Documentation

27.293.4.1 `unsigned long gdcm::SurfaceWriter::NumberOfSurfaces [protected]`

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

- [gdcmSurfaceWriter.h](#)

27.294 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)

27.294.1 Detailed Description

[SwapCode](#) representation.

Examples:

[TestByteSwap.cxx](#).

27.294.2 Member Enumeration Documentation

27.294.2.1 enum gdcm::SwapCode::SwapCodeType

Enumerator

Unknown

LittleEndian

BigEndian

BadLittleEndian

BadBigEndian

27.294.3 Constructor & Destructor Documentation

27.294.3.1 `gdcm::SwapCode::SwapCode (SwapCodeType sc = Unknown)` `[inline]`

27.294.4 Member Function Documentation

27.294.4.1 `static int gdcm::SwapCode::GetIndex (SwapCode const & sc)` `[static]`, `[protected]`

27.294.4.2 `static const char* gdcm::SwapCode::GetSwapCodeString (SwapCode const & sc)` `[static]`

Referenced by `gdcm::operator<<()`.

27.294.4.3 `gdcm::SwapCode::operator SwapCode::SwapCodeType () const` `[inline]`

27.294.5 Friends And Related Function Documentation

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

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

- [gdcmSwapCode.h](#)

27.295 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)`

27.295.1 Member Function Documentation

27.295.1.1 `template<typename T > static T gdcm::SwapperDoOp::Swap (T val)` `[static]`

27.295.1.2 `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](#)

27.296 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)

27.296.1 Detailed Description

Examples:

[ReadExplicitLengthSQIVR.cxx](#).

27.296.2 Member Function Documentation

27.296.2.1 `template<typename T > static T gdcm::SwapperNoOp::Swap (T val)` `[inline],[static]`

27.296.2.2 `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](#)

27.297 gdcm::System Class Reference

Class to do system operation.

```
#include <gdcmSystem.h>
```

Static Public Member Functions

- 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)

27.297.1 Detailed Description

Class to do system operation.

OS independent functionalities

27.297.2 Member Function Documentation

27.297.2.1 static bool gdcm::System::DeleteDirectory (const char * *source*) [static]

remove a directory named source

27.297.2.2 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

27.297.2.3 static bool gdcm::System::FileExists (const char * *filename*) [static]

Check whether the specified file exist on the sytem.

Examples:

[EncapsulateFileInRawData.cxx](#), [gdcmorthoplanes.cxx](#), and [MagnifyFile.cxx](#).

27.297.2.4 `static bool gdcm::System::FileIsDirectory (const char * name) [static]`

Check whether the file specified is a directory:

Examples:

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

27.297.2.5 `static bool gdcm::System::FileIsSymlink (const char * name) [static]`

Check whether name is a symlink.

27.297.2.6 `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](#).

27.297.2.7 `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

27.297.2.8 `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]

27.297.2.9 `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

27.297.2.10 `static const char* gdcm::System::GetCurrentModuleFileName () [static]`

Return the directory the current module is located: NOT THREAD SAFE

27.297.2.11 `static const char* gdcm::System::GetCurrentProcessFileName () [static]`

Return the directory the current process (executable) is located: NOT THREAD SAFE

27.297.2.12 `static const char* gdcmm::System::GetCurrentResourcesDirectory () [static]`

On some system (Apple) return the path to the current bundled 'Resources' directory NOT THREAD SAFE

27.297.2.13 `static const char* gdcmm::System::GetCurrentWorkingDirectory () [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

27.297.2.14 `static bool gdcmm::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

27.297.2.15 `static const char* gdcmm::System::GetLastError () [static]`

Return the last error.

27.297.2.16 `static const char* gdcmm::System::GetLocaleCharSet () [static]`

return locale charmap

27.297.2.17 `static bool gdcmm::System::GetPermissions (const char * file, unsigned short & mode) [static],
[protected]`

NOT THREAD SAFE.

27.297.2.18 `static const char* gdcmm::System::GetTimezoneOffsetFromUTC () [static]`

Return the value for Timezone Offset From UTC as string.

Warning

not thread safe

27.297.2.19 `static bool gdcmm::System::MakeDirectory (const char * path) [static]`

Create a directory name path.

27.297.2.20 `static bool gdcmm::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)

27.297.2.21 `static bool gdcmm::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](#)

27.297.2.22 `static bool gdcm::System::RemoveFile (const char * source) [static]`

remove a file named source

27.297.2.23 `static bool gdcm::System::SetPermissions (const char * file, unsigned short mode) [static],
[protected]`

27.297.2.24 `static int gdcm::System::StrCaseCmp (const char * s1, const char * s2) [static]`

consistent func for C99 spec of strcasecmp/strncasecmp

27.297.2.25 `static int gdcm::System::StrNCaseCmp (const char * s1, const char * s2, size_t n) [static]`

Precondition

`n != 0`

27.297.2.26 `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

27.297.2.27 `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](#)

27.298 gdcm::Table Class Reference

[Table](#).

```
#include <gdcmTable.h>
```

Public Types

- typedef std::map< [Tag](#), [TableEntry](#) > [MapTableEntry](#)

Public Member Functions

- [Table](#) ()
- [~Table](#) ()
- const [TableEntry](#) & [GetTableEntry](#) (const [Tag](#) &tag) const
- void [InsertEntry](#) ([Tag](#) const &tag, [TableEntry](#) const &te)

Friends

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

27.298.1 Detailed Description

[Table](#).

27.298.2 Member Typedef Documentation

27.298.2.1 `typedef std::map<Tag, TableEntry> gdcm::Table::MapTableEntry`

27.298.3 Constructor & Destructor Documentation

27.298.3.1 `gdcm::Table::Table ()` `[inline]`

27.298.3.2 `gdcm::Table::~~Table ()` `[inline]`

27.298.4 Member Function Documentation

27.298.4.1 `const TableEntry& gdcm::Table::GetTableEntry (const Tag & tag) const` `[inline]`

27.298.4.2 `void gdcm::Table::InsertEntry (Tag const & tag, TableEntry const & te)` `[inline]`

27.298.5 Friends And Related Function Documentation

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

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

- [gdcmTable.h](#)

27.299 gdcm::TableEntry Class Reference

[TableEntry](#).

```
#include <gdcmTableEntry.h>
```

Public Member Functions

- [TableEntry](#) (const char *attribute=0, [Type](#) const &type=[Type](#)(), const char *des=0)
- [~TableEntry](#) ()

27.299.1 Detailed Description

[TableEntry](#).

27.299.2 Constructor & Destructor Documentation

27.299.2.1 `gdcm::TableEntry::TableEntry (const char * attribute = 0, Type const & type = Type (), const char * des = 0)`
`[inline]`

27.299.2.2 `gdcm::TableEntry::~~TableEntry ()` `[inline]`

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

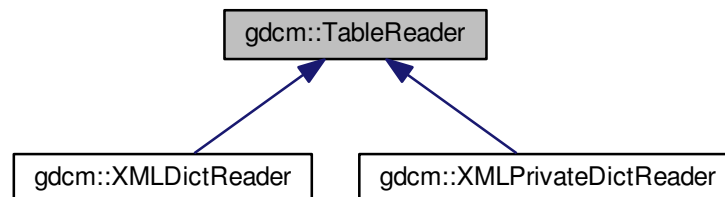
- [gdcmTableEntry.h](#)

27.300 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` ()
- 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)

27.300.1 Detailed Description

Class for representing a [TableReader](#).

Note

This class is an empty shell meant to be derived

27.300.2 Constructor & Destructor Documentation

27.300.2.1 `gdcmm::TableReader::TableReader (Defs & defs)` `[inline]`

27.300.2.2 `virtual gdcmm::TableReader::~~TableReader ()` `[inline]`, `[virtual]`

27.300.3 Member Function Documentation

27.300.3.1 `virtual void gdcmm::TableReader::CharacterDataHandler (const char * data, int length)` `[virtual]`

Reimplemented in [gdcmm::XMLDictReader](#), and [gdcmm::XMLPrivateDictReader](#).

27.300.3.2 `virtual void gdcmm::TableReader::EndElement (const char * name)` `[virtual]`

Reimplemented in [gdcmm::XMLDictReader](#), and [gdcmm::XMLPrivateDictReader](#).

27.300.3.3 `const Defs& gdcmm::TableReader::GetDefs () const` `[inline]`

27.300.3.4 `const char* gdcmm::TableReader::GetFilename ()` `[inline]`

27.300.3.5 `void gdcmm::TableReader::HandleIOD (const char ** atts)`

27.300.3.6 `void gdcmm::TableReader::HandleIODEntry (const char ** atts)`

27.300.3.7 `void gdcmm::TableReader::HandleMacro (const char ** atts)`

27.300.3.8 `void gdcmm::TableReader::HandleMacroEntry (const char ** atts)`

27.300.3.9 `void gdcmm::TableReader::HandleMacroEntryDescription (const char ** atts)`

27.300.3.10 `void gdcmm::TableReader::HandleModule (const char ** atts)`

27.300.3.11 `void gdcmm::TableReader::HandleModuleEntry (const char ** atts)`

27.300.3.12 void gdcM::TableReader::HandleModuleEntryDescription (const char ** *atts*)

27.300.3.13 void gdcM::TableReader::HandleModuleInclude (const char ** *atts*)

27.300.3.14 int gdcM::TableReader::Read ()

27.300.3.15 void gdcM::TableReader::SetFilename (const char * *filename*) [inline]

27.300.3.16 virtual void gdcM::TableReader::StartElement (const char * *name*, const char ** *atts*) [virtual]

Reimplemented in [gdcM::XMLDictReader](#), and [gdcM::XMLPrivateDictReader](#).

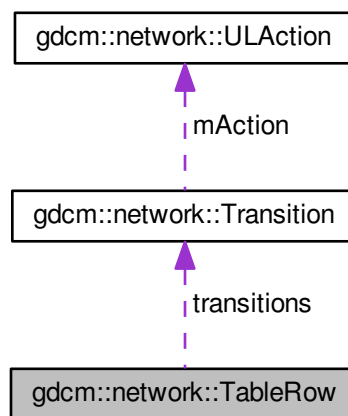
The documentation for this class was generated from the following file:

- [gdcMTableReader.h](#)

27.301 gdcM::network::TableRow Class Reference

```
#include <gdcMULTransitionTable.h>
```

Collaboration diagram for gdcM::network::TableRow:



Public Member Functions

- [TableRow](#) ()
- [~TableRow](#) ()

Public Attributes

- [Transition](#) * [transitions](#) [cMaxStateID]

27.301.1 Constructor & Destructor Documentation

27.301.1.1 `gdcm::network::TableRow::TableRow ()` `[inline]`

References `gdcm::network::cMaxStateID`, and transitions.

27.301.1.2 `gdcm::network::TableRow::~~TableRow ()` `[inline]`

References `gdcm::network::cMaxStateID`, and transitions.

27.301.2 Member Data Documentation

27.301.2.1 `Transition* gdcm::network::TableRow::transitions[cMaxStateID]`

Referenced by `TableRow()`, and `~TableRow()`.

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

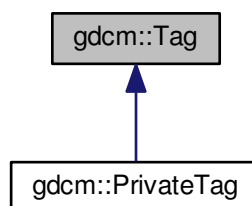
- [gdcmULTransitionTable.h](#)

27.302 gdcm::Tag Class Reference

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)

```
#include <gdcmTag.h>
```

Inheritance diagram for `gdcm::Tag`:



Public Member Functions

- [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.*
- [Tag](#) (const [Tag](#) &_val)

- `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`
- `const uint16_t & operator[] (const unsigned int &_id) const`
Returns the Group or *Element* of the given *Tag*, depending on id (0/1)
- `uint16_t & operator[] (const unsigned int &_id)`
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)`

27.302.1 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](#), [DumpToSQLITE3.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [MergeTwoFiles.cxx](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [PublicDict.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [rle2img.cxx](#), [SimpleScanner.cxx](#), [SortImage.cxx](#), [StreamImageReaderTest.cxx](#), [TraverseModules.cxx](#), and [VolumeSorter.cxx](#).

27.302.2 Constructor & Destructor Documentation

27.302.2.1 `gdcm::Tag::Tag (uint16_t group, uint16_t element) [inline]`

Constructor with 2*`uint16_t`.

27.302.2.2 `gdcm::Tag::Tag (uint32_t tag = 0) [inline]`

Constructor with 1*`uint32_t` Prefer the ctor that takes two `uint16_t`.

27.302.2.3 `gdcm::Tag::Tag (const Tag &_val) [inline]`

References tag.

27.302.3 Member Function Documentation

27.302.3.1 `uint16_t gdcm::Tag::GetElement () const [inline]`

Returns the '[Element](#) number' of the given [Tag](#).

Examples:

[DuplicatePCDE.cxx](#), and [PublicDict.cxx](#).

Referenced by `gdcM::DataSet::ComputeGroupLength()`, `IsGroupXX()`, `gdcM::PrivateDict::PrintXML()`, `gdcM::PrivateTag::PrivateTag()`, `gdcM::SequenceOfFragments::ReadValue()`, and `SetPrivateCreator()`.

27.302.3.2 `uint32_t gdcM::Tag::GetElementTag () const [inline]`

Returns the full tag value of the given [Tag](#).

27.302.3.3 `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::CommandDataSet::Insert()`, `gdcM::FileMetaInformation::Insert()`, `gdcM::DataSet::Insert()`, `IsGroupXX()`, `gdcM::PrivateDict::PrintXML()`, `gdcM::SequenceOfFragments::ReadValue()`, `gdcM::Attribute< Group, Element, TVR, TVM >::SetFromDataElement()`, `gdcM::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement()`, `gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement()`, and `SetPrivateCreator()`.

27.302.3.4 `uint32_t gdcM::Tag::GetLength () const [inline]`

return the length of tag (read: size on disk)

27.302.3.5 `Tag gdcM::Tag::GetPrivateCreator () const [inline]`

Return the Private Creator Data [Element](#) tag of a private data element.

References `SetElement()`.

27.302.3.6 `bool gdcM::Tag::IsGroupLength () const [inline]`

return whether the tag correspond to a group length tag:

27.302.3.7 `bool gdcM::Tag::IsGroupXX (const Tag & t) const [inline]`

e.g 6002,3000 belong to groupXX: 6000,3000

References `GetElement()`, `GetGroup()`, and `IsPrivate()`.

27.302.3.8 `bool gdcM::Tag::IsIllegal () const [inline]`

return if the tag is considered to be an illegal tag

27.302.3.9 `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()`.

27.302.3.10 `bool gdcmm::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](#).

27.302.3.11 `bool gdcmm::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...

27.302.3.12 `bool gdcmm::Tag::operator!= (const Tag &_val) const [inline]`

References tag.

27.302.3.13 `bool gdcmm::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.

27.302.3.14 `bool gdcmm::Tag::operator<= (const Tag &t2) const [inline]`

27.302.3.15 `Tag& gdcmm::Tag::operator= (const Tag &_val) [inline]`

References tag.

27.302.3.16 `bool gdcmm::Tag::operator== (const Tag &_val) const [inline]`

References tag.

27.302.3.17 `const uint16_t& gdcmm::Tag::operator[] (const unsigned int &_id) const [inline]`

Returns the Group or [Element](#) of the given [Tag](#), depending on id (0/1)

27.302.3.18 `uint16_t& gdcmm::Tag::operator[] (const unsigned int &_id) [inline]`

Returns the Group or [Element](#) of the given [Tag](#), depending on id (0/1)

27.302.3.19 `std::string gdcmm::Tag::PrintAsContinuousString () const`

Print tag value with no separating comma: eg. tag = "12345678" It comes in useful when reading tag values from XML file(in NativeDICOMModel)

27.302.3.20 `std::string gdcmm::Tag::PrintAsContinuousUpperCaseString () const`

Same as PrintAsContinuousString, but hexadecimal [a-f] are printed using upper case.

27.302.3.21 `std::string gdcmm::Tag::PrintAsPipeSeparatedString () const`

Print as a pipe separated string (GDCM 1.x compat only). Do not use in newer code

See also

[ReadFromPipeSeparatedString](#)

27.302.3.22 `template<typename TSwap> std::istream& gdcmm::Tag::Read (std::istream & is) [inline]`

Read a tag from binary representation.

27.302.3.23 `bool gdcmm::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

27.302.3.24 `bool gdcmm::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)

27.302.3.25 `bool gdcmm::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](#)

27.302.3.26 `void gdcmm::Tag::SetElement (uint16_t element) [inline]`

Sets the '[Element](#) number' of the given [Tag](#).

Examples:

[DuplicatePCDE.cxx](#), and [PublicDict.cxx](#).

Referenced by [GetPrivateCreator\(\)](#), and [gdcmm::operator>>\(\)](#).

27.302.3.27 `void gdcm::Tag::SetElementTag (uint16_t group, uint16_t element) [inline]`

Sets the 'Group number' & 'Element number' of the given [Tag](#).

27.302.3.28 `void gdcm::Tag::SetElementTag (uint32_t tag) [inline]`

Sets the full tag value of the given [Tag](#).

27.302.3.29 `void gdcm::Tag::SetGroup (uint16_t group) [inline]`

Sets the 'Group number' of the given [Tag](#).

Referenced by `gdcm::operator>>()`.

27.302.3.30 `void gdcm::Tag::SetPrivateCreator (Tag const & t) [inline]`

Set private creator:

Examples:

[DuplicatePCDE.cxx](#).

References `GetElement()`, `GetGroup()`, and `IsPrivate()`.

27.302.3.31 `template<typename TSwap > const std::ostream& gdcm::Tag::Write (std::ostream & os) const [inline]`

Write a tag in binary rep.

Referenced by `gdcm::SequenceOfItems::Write()`, `gdcm::Item::Write()`, and `gdcm::SequenceOfFragments::Write()`.

27.302.4 Friends And Related Function Documentation

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

27.302.4.2 `std::istream& operator>> (std::istream & _is, Tag & _val) [friend]`

27.302.5 Member Data Documentation

27.302.5.1 `char gdcm::Tag::bytes[4]`

27.302.5.2 `uint32_t gdcm::Tag::tag`

Referenced by `operator!=()`, `operator<()`, `operator=()`, `operator==()`, and `Tag()`.

27.302.5.3 `uint16_t gdcm::Tag::tags[2]`

Referenced by `operator<()`.

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

- [gdcmTag.h](#)

27.303 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.

27.303.1 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)

27.303.2 Constructor & Destructor Documentation

27.303.2.1 `gdcm::TagPath::TagPath ()`

27.303.2.2 `gdcm::TagPath::~~TagPath ()`

27.303.3 Member Function Documentation

27.303.3.1 `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

27.303.3.2 `bool gdcm::TagPath::ConstructFromTagList (Tag const * l, unsigned int n)`

Construct from a list of tags.

27.303.3.3 `static bool gdcm::TagPath::IsValid (const char * path)` `[static]`

Return if path is valid or not.

27.303.3.4 void gdcm::TagPath::Print (std::ostream &) const

27.303.3.5 bool gdcm::TagPath::Push (Tag const & t)

27.303.3.6 bool gdcm::TagPath::Push (unsigned int itemnum)

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

- [gdcmTagPath.h](#)

27.304 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](#) ()
- [~Testing](#) ()
- 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, unsigned long 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=0)
- static const wchar_t * [GetTempDirectoryW](#) (const wchar_t *subdir=0)
NOT THREAD SAFE.
- static const char * [GetTempFilename](#) (const char *filename, const char *subdir=0)
NOT THREAD SAFE.
- static const wchar_t * [GetTempFilenameW](#) (const wchar_t *filename, const wchar_t *subdir=0)
NOT THREAD SAFE.

27.304.1 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

27.304.2 Member Typedef Documentation

27.304.2.1 `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

27.304.2.2 `typedef const char* const(* gdcm::Testing::MediaStorageDataFilesType)[2]`

return the table that map the media storage (as string) of a filename (gdcmData)

27.304.3 Constructor & Destructor Documentation

27.304.3.1 `gdcm::Testing::Testing ()` `[inline]`

27.304.3.2 `gdcm::Testing::~~Testing ()` `[inline]`

27.304.4 Member Function Documentation

27.304.4.1 `static bool gdcm::Testing::ComputeFileMD5 (const char * filename, char digest_str[33])` `[static]`

Examples:

[MetalImageMD5Activiz.cs](#).

27.304.4.2 static bool gdcmm::Testing::ComputeMD5 (const char * *buffer*, unsigned long *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 [gdcmm::MD5](#) API when doing md5 computation.

27.304.4.3 static const char* gdcmm::Testing::GetDataExtraRoot () [static]

Return the GDCM DATA EXTRA ROOT.

Examples:

[DiscriminateVolume.cxx](#), [reslicesphere.cxx](#), and [VolumeSorter.cxx](#).

27.304.4.4 static const char* gdcmm::Testing::GetDataRoot () [static]

Return the GDCM DATA ROOT.

Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), and [MagnifyFile.cxx](#).

27.304.4.5 static const char* gdcmm::Testing::GetFileName (unsigned int *file*) [static]

Examples:

[MetalImageMD5Activiz.cs](#).

27.304.4.6 static const char* const* gdcmm::Testing::GetFileNames () [static]

return the table of fullpath to gdcmmData DICOM files:

Examples:

[TestReader.cxx](#).

27.304.4.7 static int gdcmm::Testing::GetLossyFlagFromFile (const char * *filepath*) [static]

Return the lossy flag of the given filename -1 -> Error 0 -> Lossless 1 -> Lossy

27.304.4.8 static const char* const* gdcmm::Testing::GetMD5DataImage (unsigned int *file*) [static]

27.304.4.9 static MD5DataImagesType gdcmm::Testing::GetMD5DataImages () [static]

27.304.4.10 static const char* gdcmm::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.

27.304.4.11 `static const char* gdcmm::Testing::GetMD5FromFile (const char * filepath) [static]`

27.304.4.12 `static const char* const* gdcmm::Testing::GetMediaStorageDataFile (unsigned int file) [static]`

27.304.4.13 `static MediaStorageDataFilesType gdcmm::Testing::GetMediaStorageDataFiles () [static]`

27.304.4.14 `static const char* gdcmm::Testing::GetMediaStorageFromFile (const char * filepath) [static]`

Examples:

[MetaImageMD5Activiz.cs](#), and [TestReader.cxx](#).

27.304.4.15 `static unsigned int gdcmm::Testing::GetNumberOfFileNames () [static]`

Examples:

[MetaImageMD5Activiz.cs](#).

27.304.4.16 `static unsigned int gdcmm::Testing::GetNumberOfMD5DataImages () [static]`

27.304.4.17 `static unsigned int gdcmm::Testing::GetNumberOfMediaStorageDataFiles () [static]`

27.304.4.18 `static const char* gdcmm::Testing::GetPixelSpacingDataRoot () [static]`

Return the GDCM PIXEL SPACING DATA ROOT (See David Clunie website for dataset)

27.304.4.19 `static std::streamoff gdcmm::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

27.304.4.20 `static std::streamoff gdcmm::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

27.304.4.21 `static const char* gdcmm::Testing::GetSourceDirectory () [static]`

Examples:

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

27.304.4.22 `static std::streamoff gdcmm::Testing::GetStreamOffsetFromFile (const char * filepath) [static]`

Return the offset of the very first pixel cell in the PixelData -1 if not found

27.304.4.23 `static const char* gdcm::Testing::GetTempDirectory (const char * subdir = 0) [static]`

NOT THREAD SAFE Returns the temp directory as used in testing needing to output data:

Examples:

[MetImageMD5Activiz.cs](#).

27.304.4.24 `static const wchar_t* gdcm::Testing::GetTempDirectoryW (const wchar_t * subdir = 0) [static]`

NOT THREAD SAFE.

27.304.4.25 `static const char* gdcm::Testing::GetTempFilename (const char * filename, const char * subdir = 0) [static]`

NOT THREAD SAFE.

Examples:

[MetImageMD5Activiz.cs](#).

27.304.4.26 `static const wchar_t* gdcm::Testing::GetTempFilenameW (const wchar_t * filename, const wchar_t * subdir = 0) [static]`

NOT THREAD SAFE.

27.304.4.27 `void gdcm::Testing::Print (std::ostream & os = std::cout)`

Print.

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

- [gdcmTesting.h](#)

27.305 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](#) ()

27.305.1 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.

27.305.2 Constructor & Destructor Documentation

27.305.2.1 [gdcm::Trace::Trace](#) ()

27.305.2.2 [gdcm::Trace::~~Trace](#) ()

27.305.3 Member Function Documentation

27.305.3.1 static void gdcm::Trace::DebugOff () [static]

Examples:

[MetalImageMD5Activiz.cs](#), and [TestReader.cxx](#).

27.305.3.2 static void gdcm::Trace::DebugOn () [static]

Examples:

[CreateFakePET.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

27.305.3.3 static void gdcm::Trace::ErrorOff () [static]

Examples:

[MetalImageMD5Activiz.cs](#).

27.305.3.4 static void gdcm::Trace::ErrorOn () [static]

27.305.3.5 static bool gdcm::Trace::GetDebugFlag () [static]

27.305.3.6 static std::ostream& gdcm::Trace::GetDebugStream () [static]

27.305.3.7 static bool gdcm::Trace::GetErrorFlag () [static]

27.305.3.8 static std::ostream& gdcm::Trace::GetErrorStream () [static]

27.305.3.9 static std::ostream& gdcm::Trace::GetStream () [static]

27.305.3.10 static bool gdcm::Trace::GetWarningFlag () [static]

27.305.3.11 static std::ostream& gdcm::Trace::GetWarningStream () [static]

27.305.3.12 static void gdcm::Trace::SetDebug (bool *debug*) [static]

Turn debug messages on (default: false)

Examples:

[DumpToSQLITE3.cxx](#).

27.305.3.13 static void gdcm::Trace::SetDebugStream (std::ostream & *os*) [static]

Explicitly set the stream which receive Debug messages:

27.305.3.14 static void gdcm::Trace::SetError (bool *debug*) [static]

Turn error messages on (default: true)

27.305.3.15 `static void gdcm::Trace::SetErrorStream (std::ostream & os) [static]`

Explicitly set the stream which receive Error messages:

Examples:

[CStoreQtProgress.cxx](#).

27.305.3.16 `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:

27.305.3.17 `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)

27.305.3.18 `static void gdcm::Trace::SetWarning (bool debug) [static]`

Turn warning messages on (default: true)

Examples:

[DumpToSQLITE3.cxx](#).

27.305.3.19 `static void gdcm::Trace::SetWarningStream (std::ostream & os) [static]`

Explicitly set the stream which receive Warning messages:

27.305.3.20 `static void gdcm::Trace::WarningOff () [static]`

Examples:

[MetaImageMD5Activiz.cs](#), and [TestReader.cxx](#).

27.305.3.21 `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](#)

27.306 gdcm::TransferSyntax Class Reference

Class to manipulate Transfer Syntax.

```
#include <gdcmTransferSyntax.h>
```


Public Types

- enum [NegociatedType](#) {
 [Unknown](#) = 0,
 [Explicit](#),
 [Implicit](#) }
- enum [TSType](#) {
 [ImplicitVRLittleEndian](#) = 0,
 [ImplicitVRBigEndianPrivateGE](#),
 [ExplicitVRLittleEndian](#),
 [DeflatedExplicitVRLittleEndian](#),
 [ExplicitVRBigEndian](#),
 [JPEGBaselineProcess1](#),
 [JPEGExtendedProcess2_4](#),
 [JPEGExtendedProcess3_5](#),
 [JPEGSpectralSelectionProcess6_8](#),
 [JPEGFullProgressionProcess10_12](#),
 [JPEGLosslessProcess14](#),
 [JPEGLosslessProcess14_1](#),
 [JPEGLSLossless](#),
 [JPEGLSNearLossless](#),
 [JPEG2000Lossless](#),
 [JPEG2000](#),
 [JPEG2000Part2Lossless](#),
 [JPEG2000Part2](#),
 [RLELossless](#),
 [MPEG2MainProfile](#),
 [ImplicitVRBigEndianACRNEMA](#),
 [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)

27.306.1 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](#).

27.306.2 Member Enumeration Documentation

27.306.2.1 enum [gdcm::TransferSyntax::NegociatedType](#)

Enumerator

Unknown

Explicit

Implicit

27.306.2.2 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
CT_private_ELE
JPIPRendered
MPEG2MainProfileHighLevel
MPEG4AVCH264HighProfileLevel4_1
MPEG4AVCH264BDcompatibleHighProfileLevel4_1
TS_END

27.306.3 Constructor & Destructor Documentation

27.306.3.1 `gdcm::TransferSyntax::TransferSyntax (TSType type = ImplicitVRLittleEndian)` `[inline]`

27.306.4 Member Function Documentation

27.306.4.1 `bool gdcm::TransferSyntax::CanStoreLossy () const`

return true if TransFer Syntax Allow storing of Lossy Pixel Data

27.306.4.2 `NegotiatedType gdcm::TransferSyntax::GetNegociatedType () const`

27.306.4.3 `const char* gdcm::TransferSyntax::GetString () const` `[inline]`

References `GetTSString()`.

27.306.4.4 `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.

27.306.4.5 `static const char* gdcm::TransferSyntax::GetTSString (TSType ts) [static]`

Examples:

[LargeVRDSExplicit.cxx.](#)

Referenced by `GetString()`, and `gdcm::operator<<()`.

27.306.4.6 `static TSType gdcm::TransferSyntax::GetTSType (const char * str) [static]`

27.306.4.7 `bool gdcm::TransferSyntax::IsEncapsulated () const`

Examples:

[ExtractIconFromFile.cxx.](#)

27.306.4.8 `bool gdcm::TransferSyntax::IsEncoded () const`

27.306.4.9 `bool gdcm::TransferSyntax::IsExplicit () const`

27.306.4.10 `bool gdcm::TransferSyntax::IsImplicit () const`

27.306.4.11 `bool gdcm::TransferSyntax::IsLossless () const`

Return true if the transfer syntax algorithm is a lossless algorithm

27.306.4.12 `bool gdcm::TransferSyntax::IsLossy () const`

Return true if the transfer syntax algorithm is a lossy algorithm

27.306.4.13 `bool gdcm::TransferSyntax::IsValid () const [inline]`

27.306.4.14 `gdcm::TransferSyntax::operator TSType () const [inline]`

27.306.5 Friends And Related Function Documentation

27.306.5.1 `std::ostream& operator<< (std::ostream & os, const TransferSyntax & ts) [friend]`

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

- [gdcmTransferSyntax.h](#)

27.307 gdcm::network::TransferSyntaxSub Class Reference

[TransferSyntaxSub](#) Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS.

```
#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

27.307.1 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

27.307.2 Constructor & Destructor Documentation

27.307.2.1 `gdcm::network::TransferSyntaxSub::TransferSyntaxSub ()`

27.307.3 Member Function Documentation

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

27.307.3.2 `bool gdcm::network::TransferSyntaxSub::operator== (const TransferSyntaxSub & ts) const` `[inline]`

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

27.307.3.4 `std::istream& gdcm::network::TransferSyntaxSub::Read (std::istream & is)`

27.307.3.5 `void gdcm::network::TransferSyntaxSub::SetName (const char * name)`

27.307.3.6 `void gdcm::network::TransferSyntaxSub::SetNameFromUID (UIDs::TSName tsname)`

27.307.3.7 `size_t gdcm::network::TransferSyntaxSub::Size () const`

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

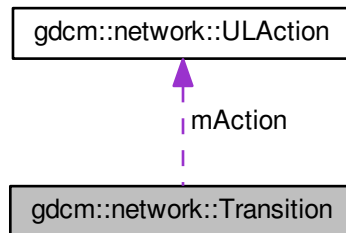
The documentation for this class was generated from the following file:

- [gdcmTransferSyntaxSub.h](#)

27.308 gdcm::network::Transition Struct Reference

```
#include <gdcmULTransitionTable.h>
```

Collaboration diagram for gdcn::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](#)

27.308.1 Constructor & Destructor Documentation

27.308.1.1 `gdcn::network::Transition::Transition ()` `[inline]`

References `gdcn::network::eStaDoesNotExist`.

Referenced by `MakeNew()`.

27.308.1.2 `gdcn::network::Transition::~~Transition ()` `[inline]`

References `mAction`.

27.308.1.3 `gdcn::network::Transition::Transition (int inEndState, ULAction * inAction)` `[inline]`

27.308.2 Member Function Documentation

27.308.2.1 `static Transition* gdcm::network::Transition::MakeNew (int inEndState, ULAction * inAction)` `[inline]`,
`[static]`

References `Transition()`.

27.308.3 Member Data Documentation

27.308.3.1 `ULAction* gdcm::network::Transition::mAction`

Referenced by `~Transition()`.

27.308.3.2 `int gdcm::network::Transition::mEnd`

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

- [gdcmULTransitionTable.h](#)

27.309 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)

27.309.1 Detailed Description

Type.

Note

PS 3.5 7.4 DATA ELEMENT TYPE 7.4.1 TYPE 1 REQUIRED DATA ELEMENTS 7.4.2 TYPE 1C CONDITIONAL DATA ELEMENTS 7.4.3 TYPE 2 REQUIRED DATA ELEMENTS 7.4.4 TYPE 2C CONDITIONAL DATA ELEMENTS 7.4.5 TYPE 3 OPTIONAL DATA ELEMENTS

The intent of Type 2 Data Elements is to allow a zero length to be conveyed when the operator or application does not know its value or has a specific reason for not specifying its value. It is the intent that the device should support these Data Elements.

Examples:

[TraverseModules.cxx](#).

27.309.2 Member Enumeration Documentation

27.309.2.1 enum gdcm::Type::TypeType

Enumerator

T1
T1C
T2
T2C
T3
UNKNOWN

27.309.3 Constructor & Destructor Documentation

27.309.3.1 `gdcm::Type::Type (TypeType type = UNKNOWN) [inline]`

27.309.4 Member Function Documentation

27.309.4.1 `static const char* gdcm::Type::GetTypeString (TypeType type) [static]`

Referenced by `gdcm::operator<<()`.

27.309.4.2 `static TypeType gdcm::Type::GetTypeType (const char * type) [static]`

Referenced by `gdcm::ModuleEntry::ModuleEntry()`.

27.309.4.3 `gdcm::Type::operator TypeType () const [inline]`

27.309.5 Friends And Related Function Documentation

27.309.5.1 `std::ostream& operator<< (std::ostream & os, const Type & vr)` [*friend*]

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

- [gdcmType.h](#)

27.310 gdcm::UI Struct Reference

```
#include <gdcmVR.h>
```

Public Attributes

- char [Internal](#) [64+1]

Friends

- `std::ostream & operator<< (std::ostream &_os, const UI &_val)`

27.310.1 Friends And Related Function Documentation

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

27.310.2 Member Data Documentation

27.310.2.1 `char gdcm::UI::Internal[64+1]`

Referenced by `gdcm::operator<<()`.

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

- [gdcmVR.h](#)

27.311 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)

27.311.1 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.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [StreamImageReaderTest.cxx](#), and [uid_unique.cxx](#).

27.311.2 Constructor & Destructor Documentation

27.311.2.1 `gdcm::UIDGenerator::UIDGenerator () [inline]`

By default the root of a UID is a GDCM Root...

27.311.3 Member Function Documentation

27.311.3.1 `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](#), and [uid_unique.cxx](#).

27.311.3.2 `static bool gdcm::UIDGenerator::GenerateUUID (unsigned char * uuid_data) [static], [protected]`

27.311.3.3 `static const char* gdcm::UIDGenerator::GetGDCMUID () [static]`

Return the default (GDCM) root UID:

27.311.3.4 static const char* gdcmm::UIDGenerator::GetRoot () [static]

Examples:

[ClinicalTrialIdentificationWorkflow.cs](#), [ReformatFile.cs](#), and [StandardizeFiles.cs](#).

27.311.3.5 static bool gdcmm::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)

27.311.3.6 static void gdcmm::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 responsibility 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:

- [gdcmmUIDGenerator.h](#)

27.312 gdcmm::UIDs Class Reference

all known uids

```
#include <gdcmmUIDs.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,  
JPEGsSpectralSelectionNonHierarchicalProcess68Retired = 9,  
JPEGsSpectralSelectionNonHierarchicalProcess79Retired = 10,  
JPEGFullProgressionNonHierarchicalProcess1012Retired = 11,  
JPEGFullProgressionNonHierarchicalProcess1113Retired = 12,  
JPEGLosslessNonHierarchicalProcess14 = 13,  
JPEGLosslessNonHierarchicalProcess15Retired = 14,  
JPEGExtendedHierarchicalProcess1618Retired = 15,  
JPEGExtendedHierarchicalProcess1719Retired = 16,  
JPEGsSpectralSelectionHierarchicalProcess2022Retired = 17,  
JPEGsSpectralSelectionHierarchicalProcess2123Retired = 18,  
JPEGFullProgressionHierarchicalProcess2426Retired = 19,  
JPEGFullProgressionHierarchicalProcess2527Retired = 20,  
JPEGLosslessHierarchicalProcess28Retired = 21,  
JPEGLosslessHierarchicalProcess29Retired = 22,  
JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLossless↵
```

JPEGImageCompression = 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,
SPM2CSFFFrameofReference = 48,
SPM2BRAINMASKFrameofReference = 49,
SPM2AVG305T1FrameofReference = 50,
SPM2AVG152T1FrameofReference = 51,
SPM2AVG152T2FrameofReference = 52,
SPM2AVG152PDFFrameofReference = 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,

[BreastTomosynthesisImageStorage](#) }

• 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_2_4_103 }
```

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)

27.312.1 Detailed Description

all known uids

Examples:

[GenerateStandardSOPClasses.cxx](#).

27.312.2 Member Typedef Documentation

27.312.2.1 `typedef const char* const(* gdcmm::UIDs::TransferSyntaxStringsType)[2]`

27.312.3 Member Enumeration Documentation

27.312.3.1 `enum gdcmm::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
JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLosslessJPEGImage

JPEGLSLosslessImageCompression
JPEGLSLossyNearLosslessImageCompression
JPEG2000ImageCompressionLosslessOnly
JPEG2000ImageCompression
JPEG2000Part2MulticomponentImageCompressionLosslessOnly
JPEG2000Part2MulticomponentImageCompression
JPIPIReferenced
JPIPIReferencedDeflate
MPEG2MainProfileMainLevel
RLELossless
RFC2557MIMEencapsulation
XMLEncoding
MediaStorageDirectoryStorage
TalairachBrainAtlasFrameofReference
SPM2T1FrameofReference
SPM2T2FrameofReference
SPM2PDFFrameofReference
SPM2EPIFrameofReference
SPM2FIL T1FrameofReference
SPM2PETFrameofReference
SPM2TRANSMFrameofReference
SPM2SPECTFrameofReference
SPM2GRAYFrameofReference
SPM2WHITEFrameofReference
SPM2CSFFFrameofReference
SPM2BRAINMASKFrameofReference
SPM2AVG305T1FrameofReference
SPM2AVG152T1FrameofReference
SPM2AVG152T2FrameofReference
SPM2AVG152PDFFrameofReference

SPM2SINGLESUBJT1FrameofReference
ICBM452T1FrameofReference
ICBMSingleSubjectMRIFrameofReference
BasicStudyContentNotificationSOPClassRetired
StorageCommitmentPushModelSOPClass
StorageCommitmentPushModelSOPInstance
StorageCommitmentPullModelSOPClassRetired
StorageCommitmentPullModelSOPInstanceRetired
ProceduralEventLoggingSOPClass
ProceduralEventLoggingSOPInstance
SubstanceAdministrationLoggingSOPClass
SubstanceAdministrationLoggingSOPInstance
DICOMUIDRegistry
DICOMControlledTerminology
DICOMApplicationContextName
DetachedPatientManagementSOPClassRetired
DetachedPatientManagementMetaSOPClassRetired
DetachedVisitManagementSOPClassRetired
DetachedStudyManagementSOPClassRetired
StudyComponentManagementSOPClassRetired
ModalityPerformedProcedureStepSOPClass
ModalityPerformedProcedureStepRetrieveSOPClass
ModalityPerformedProcedureStepNotificationSOPClass
DetachedResultsManagementSOPClassRetired
DetachedResultsManagementMetaSOPClassRetired
DetachedStudyManagementMetaSOPClassRetired
DetachedInterpretationManagementSOPClassRetired
StorageServiceClass
BasicFilmSessionSOPClass
BasicFilmBoxSOPClass
BasicGrayscaleImageBoxSOPClass
BasicColorImageBoxSOPClass
ReferencedImageBoxSOPClassRetired
BasicGrayscalePrintManagementMetaSOPClass
ReferencedGrayscalePrintManagementMetaSOPClassRetired
PrintJobSOPClass
BasicAnnotationBoxSOPClass
PrinterSOPClass
PrinterConfigurationRetrievalSOPClass
PrinterSOPInstance
PrinterConfigurationRetrievalSOPInstance
BasicColorPrintManagementMetaSOPClass

ReferencedColorPrintManagementMetaSOPClassRetired
VOILUTBoxSOPClass
PresentationLUTSOPClass
ImageOverlayBoxSOPClassRetired
BasicPrintImageOverlayBoxSOPClassRetired
PrintQueueSOPInstanceRetired
PrintQueueManagementSOPClassRetired
StoredPrintStorageSOPClassRetired
HardcopyGrayscaleImageStorageSOPClassRetired
HardcopyColorImageStorageSOPClassRetired
PullPrintRequestSOPClassRetired
PullStoredPrintManagementMetaSOPClassRetired
MediaCreationManagementSOPClassUID
ComputedRadiographyImageStorage
DigitalXRayImageStorageForPresentation
DigitalXRayImageStorageForProcessing
DigitalMammographyXRayImageStorageForPresentation
DigitalMammographyXRayImageStorageForProcessing
DigitalIntraoralXRayImageStorageForPresentation
DigitalIntraoralXRayImageStorageForProcessing
CTImageStorage
EnhancedCTImageStorage
UltrasoundMultiframeImageStorageRetired
UltrasoundMultiframeImageStorage
MRIImageStorage
EnhancedMRIImageStorage
MRSpectroscopyStorage
NuclearMedicineImageStorageRetired
UltrasoundImageStorageRetired
UltrasoundImageStorage
SecondaryCaptureImageStorage
MultiframeSingleBitSecondaryCaptureImageStorage
MultiframeGrayscaleByteSecondaryCaptureImageStorage
MultiframeGrayscaleWordSecondaryCaptureImageStorage
MultiframeTrueColorSecondaryCaptureImageStorage
StandaloneOverlayStorageRetired
StandaloneCurveStorageRetired
WaveformStorageTrialRetired
GeneralECGWaveformStorage
AmbulatoryECGWaveformStorage
HemodynamicWaveformStorage
CardiacElectrophysiologyWaveformStorage

BasicVoiceAudioWaveformStorage
StandaloneModalityLUTStorageRetired
StandaloneVOILUTStorageRetired
GrayscaleSoftcopyPresentationStateStorageSOPClass
ColorSoftcopyPresentationStateStorageSOPClass
PseudoColorSoftcopyPresentationStateStorageSOPClass
BlendingSoftcopyPresentationStateStorageSOPClass
XRayAngiographicImageStorage
EnhancedXAImageStorage
XRayRadiofluoroscopicImageStorage
EnhancedXRImageStorage
XRay3DAngiographicImageStorage
XRay3DCraniofacialImageStorage
XRayAngiographicBiPlaneImageStorageRetired
NuclearMedicineImageStorage
RawDataStorage
SpatialRegistrationStorage
SpatialFiducialsStorage
DeformableSpatialRegistrationStorage
SegmentationStorage
RealWorldValueMappingStorage
VLImageStorageTrialRetired
VLMultiframeImageStorageTrialRetired
VLEndoscopicImageStorage
VideoEndoscopicImageStorage
VLMicroscopicImageStorage
VideoMicroscopicImageStorage
VLSlideCoordinatesMicroscopicImageStorage
VLPhotographicImageStorage
VideoPhotographicImageStorage
OphthalmicPhotography8BitImageStorage
OphthalmicPhotography16BitImageStorage
StereometricRelationshipStorage
OphthalmicTomographyImageStorage
TextSRStorageTrialRetired
AudioSRStorageTrialRetired
DetailSRStorageTrialRetired
ComprehensiveSRStorageTrialRetired
BasicTextSRStorage
EnhancedSRStorage
ComprehensiveSRStorage
ProcedureLogStorage

MammographyCADSRStorage
KeyObjectSelectionDocumentStorage
ChestCADSRStorage
XRayRadiationDoseSRStorage
EncapsulatedPDFStorage
EncapsulatedCDASStorage
PositronEmissionTomographyImageStorage
StandalonePETCurveStorageRetired
RTImageStorage
RTDoseStorage
RTStructureSetStorage
RTBeamsTreatmentRecordStorage
RTPlanStorage
RTBrachyTreatmentRecordStorage
RTTreatmentSummaryRecordStorage
RTIonPlanStorage
RTIonBeamsTreatmentRecordStorage
PatientRootQueryRetrieveInformationModelFIND
PatientRootQueryRetrieveInformationModelMOVE
PatientRootQueryRetrieveInformationModelGET
StudyRootQueryRetrieveInformationModelFIND
StudyRootQueryRetrieveInformationModelMOVE
StudyRootQueryRetrieveInformationModelGET
PatientStudyOnlyQueryRetrieveInformationModelFINDRetired
PatientStudyOnlyQueryRetrieveInformationModelMOVERetired
PatientStudyOnlyQueryRetrieveInformationModelGETRetired
ModalityWorklistInformationModelFIND
GeneralPurposeWorklistInformationModelFIND
GeneralPurposeScheduledProcedureStepSOPClass
GeneralPurposePerformedProcedureStepSOPClass
GeneralPurposeWorklistManagementMetaSOPClass
InstanceAvailabilityNotificationSOPClass
RTBeamsDeliveryInstructionStorageSupplement74FrozenDraft
RTConventionalMachineVerificationSupplement74FrozenDraft
RTIonMachineVerificationSupplement74FrozenDraft
UnifiedWorklistandProcedureStepServiceClass
UnifiedProcedureStepPushSOPClass
UnifiedProcedureStepWatchSOPClass
UnifiedProcedureStepPullSOPClass
UnifiedProcedureStepEventSOPClass
UnifiedWorklistandProcedureStepSOPInstance
GeneralRelevantPatientInformationQuery

BreastImagingRelevantPatientInformationQuery
CardiacRelevantPatientInformationQuery
HangingProtocolStorage
HangingProtocolInformationModelFIND
HangingProtocolInformationModelMOVE
ProductCharacteristicsQuerySOPClass
SubstanceApprovalQuerySOPClass
dicomDeviceName
dicomDescription
dicomManufacturer
dicomManufacturerModelName
dicomSoftwareVersion
dicomVendorData
dicomAETitle
dicomNetworkConnectionReference
dicomApplicationCluster
dicomAssociationInitiator
dicomAssociationAcceptor
dicomHostname
dicomPort
dicomSOPClass
dicomTransferRole
dicomTransferSyntax
dicomPrimaryDeviceType
dicomRelatedDeviceReference
dicomPreferredCalledAETitle
dicomTLSCyphersuite
dicomAuthorizedNodeCertificateReference
dicomThisNodeCertificateReference
dicomInstalled
dicomStationName
dicomDeviceSerialNumber
dicomInstitutionName
dicomInstitutionAddress
dicomInstitutionDepartmentName
dicomIssuerOfPatientID
dicomPreferredCallingAETitle
dicomSupportedCharacterSet
dicomConfigurationRoot
dicomDevicesRoot
dicomUniqueAETitlesRegistryRoot
dicomDevice

dicomNetworkAE
dicomNetworkConnection
dicomUniqueAETitle
dicomTransferCapability
VLWholeSlideMicroscopyImageStorage
EnhancedUSVolumeStorage
SurfaceSegmentationStorage
BreastTomosynthesisImageStorage

27.312.3.2 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
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
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
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
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
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
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_1_2_4_101
uid_1_2_840_10008_1_2_4_102
uid_1_2_840_10008_1_2_4_103

```

27.312.4 Member Function Documentation

27.312.4.1 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<<()`.

27.312.4.2 `static unsigned int gdcmm::UIDs::GetNumberOfTransferSyntaxStrings () [static]`

27.312.4.3 `const char* gdcmm::UIDs::GetString () const`

When object is Initialize function return the uid return NULL when not initialized

Examples:

[GenerateStandardSOPClasses.cxx.](#)

Referenced by `gdcmm::operator<<()`.

27.312.4.4 `static const char* const* gdcmm::UIDs::GetTransferSyntaxString (unsigned int ts) [static]`

27.312.4.5 `static TransferSyntaxStringsType gdcmm::UIDs::GetTransferSyntaxStrings () [static]`

27.312.4.6 `static const char* gdcmm::UIDs::GetUIDName (unsigned int ts) [static]`

27.312.4.7 `static const char* gdcmm::UIDs::GetUIDString (unsigned int ts) [static]`

27.312.4.8 `gdcmm::UIDs::operator TSType () const [inline]`

27.312.4.9 `bool gdcmm::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:

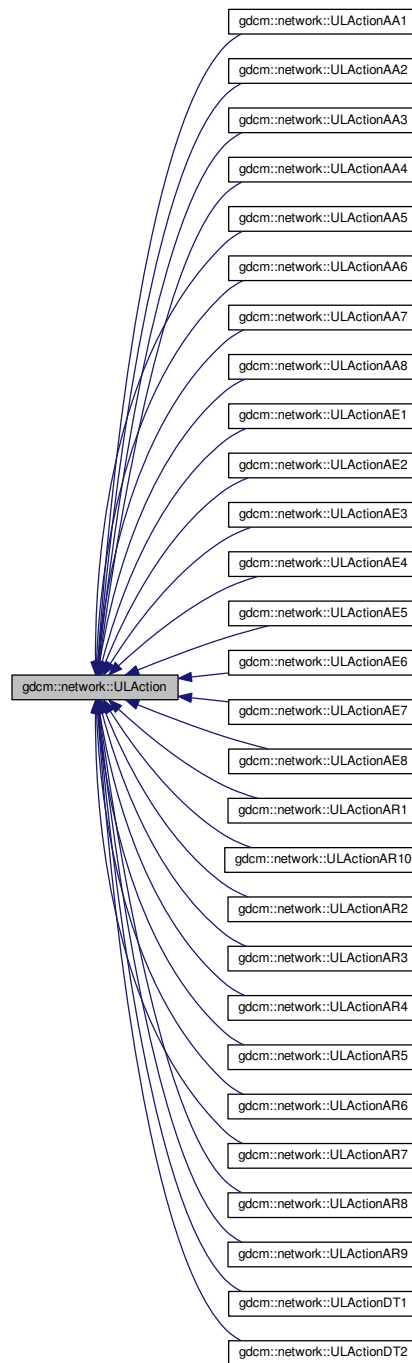
- [gdcmmUIDs.h](#)

27.313 gdcmm::network::ULAction Class Reference

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](#).

```
#include <gdcmmULAction.h>
```

Inheritance diagram for gdom::network::ULAction:



Public Member Functions

- [ULAction](#) ()
- virtual [~ULAction](#) ()

- virtual [EStateID PerformAction](#) ([Subject *s](#), [ULEvent &inEvent](#), [ULConnection &inConnection](#), bool &outWaiting←
ForEvent, [EEventID &outRaisedEvent](#))=0

27.313.1 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 [gdcm](#)-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.

27.313.2 Constructor & Destructor Documentation

27.313.2.1 `gdcm::network::ULAction::ULAction () [inline]`

27.313.2.2 `virtual gdcm::network::ULAction::~~ULAction () [inline],[virtual]`

27.313.3 Member Function Documentation

27.313.3.1 `virtual EStateID gdcm::network::ULAction::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [pure virtual]`

Implemented in [gdcm::network::ULActionAR10](#), [gdcm::network::ULActionAR9](#), [gdcm::network::ULActionAE8](#), [gdcm::network::ULActionAA8](#), [gdcm::network::ULActionAR8](#), [gdcm::network::ULActionAE7](#), [gdcm::network::ULActionA←A7](#), [gdcm::network::ULActionAR7](#), [gdcm::network::ULActionAE6](#), [gdcm::network::ULActionAA6](#), [gdcm::network::ULActionAR6](#), [gdcm::network::ULActionAA5](#), [gdcm::network::ULActionAE5](#), [gdcm::network::ULActionAR5](#), [gdcm::network::ULActionAA4](#), [gdcm::network::ULActionAE4](#), [gdcm::network::ULActionAR4](#), [gdcm::network::ULActionA←A3](#), [gdcm::network::ULActionAE3](#), [gdcm::network::ULActionAR3](#), [gdcm::network::ULActionAA2](#), [gdcm::network::ULActionAE2](#), [gdcm::network::ULActionAR2](#), [gdcm::network::ULActionDT2](#), [gdcm::network::ULActionAA1](#), [gdcm::network::ULActionAE1](#), [gdcm::network::ULActionAR1](#), and [gdcm::network::ULActionDT1](#).

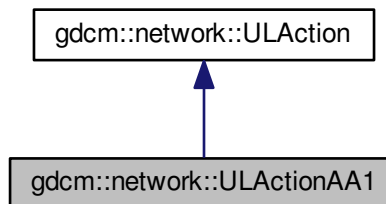
The documentation for this class was generated from the following file:

- [gdcmULAction.h](#)

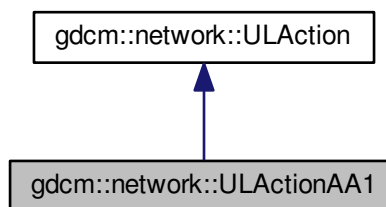
27.314 gdcm::network::ULActionAA1 Class Reference

```
#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 &outWaitingFor↵
Event, [EEventID](#) &outRaisedEvent)

27.314.1 Member Function Documentation

27.314.1.1 [EStateID](#) `gdcm::network::ULActionAA1::PerformAction` ([Subject](#) * s, [ULEvent](#) & *inEvent*, [ULConnection](#) & *inConnection*, bool & *outWaitingForEvent*, [EEventID](#) & *outRaisedEvent*) `[virtual]`

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

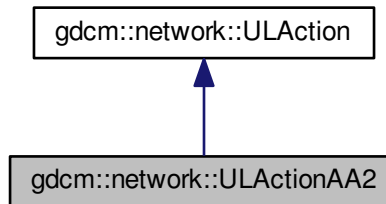
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

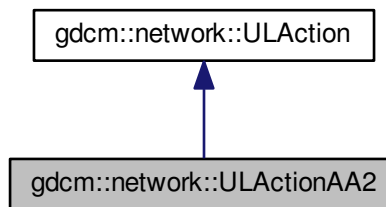
27.315 gdcm::network::ULActionAA2 Class Reference

```
#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 &outWaitingFor←
Event, [EEventID](#) &outRaisedEvent)

27.315.1 Member Function Documentation

27.315.1.1 `EStateID gdcm::network::ULActionAA2::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

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

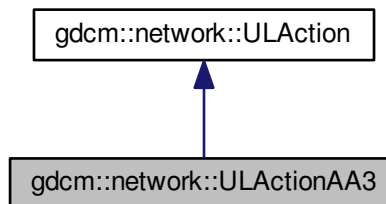
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

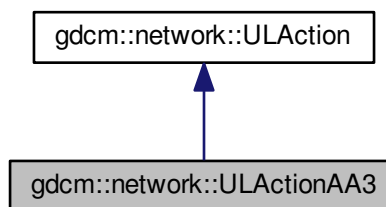
27.316 gdcm::network::ULActionAA3 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA3:



Collaboration diagram for gdcm::network::ULActionAA3:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingFor↵
Event, [EEventID](#) &outRaisedEvent)

27.316.1 Member Function Documentation

27.316.1.1 [EStateID](#) `gdcm::network::ULActionAA3::PerformAction` ([Subject](#) * s, [ULEvent](#) & *inEvent*, [ULConnection](#) & *inConnection*, bool & *outWaitingForEvent*, [EEventID](#) & *outRaisedEvent*) [virtual]

Implements [gdcm::network::ULAction](#).

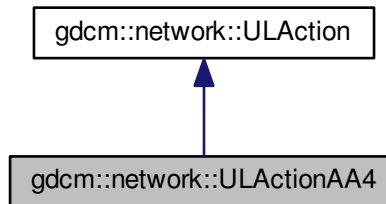
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

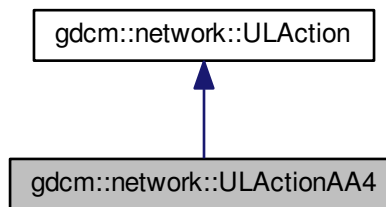
27.317 gdcm::network::ULActionAA4 Class Reference

```
#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 &outWaitingFor↵
Event, [EEventID](#) &outRaisedEvent)

27.317.1 Member Function Documentation

27.317.1.1 `EStateID gdcm::network::ULActionAA4::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements [gdcm::network::ULAction](#).

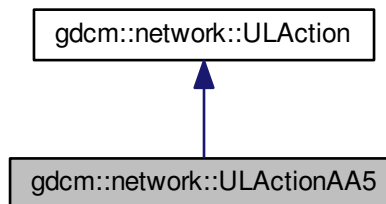
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

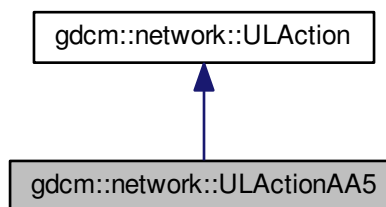
27.318 gdcmm::network::ULActionAA5 Class Reference

```
#include <gdcmmULActionAA.h>
```

Inheritance diagram for gdcmm::network::ULActionAA5:



Collaboration diagram for gdcmm::network::ULActionAA5:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingFor←
Event, [EEventID](#) &outRaisedEvent)

27.318.1 Member Function Documentation

27.318.1.1 [EStateID gdcmm::network::ULActionAA5::PerformAction](#) ([Subject](#) * s, [ULEvent](#) & *inEvent*, [ULConnection](#) & *inConnection*, bool & *outWaitingForEvent*, [EEventID](#) & *outRaisedEvent*) `[virtual]`

Implements [gdcmm::network::ULAction](#).

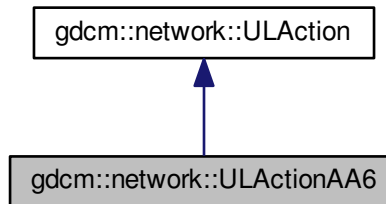
The documentation for this class was generated from the following file:

- [gdcmmULActionAA.h](#)

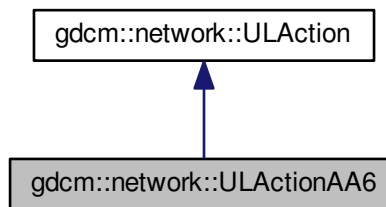
27.319 gdcm::network::ULActionAA6 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA6:



Collaboration diagram for gdcm::network::ULActionAA6:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingFor↵
Event, [EEventID](#) &outRaisedEvent)

27.319.1 Member Function Documentation

27.319.1.1 [EStateID](#) `gdcm::network::ULActionAA6::PerformAction` ([Subject](#) * s, [ULEvent](#) & *inEvent*, [ULConnection](#) & *inConnection*, bool & *outWaitingForEvent*, [EEventID](#) & *outRaisedEvent*) `[virtual]`

Implements [gdcm::network::ULAction](#).

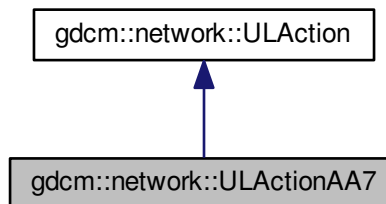
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

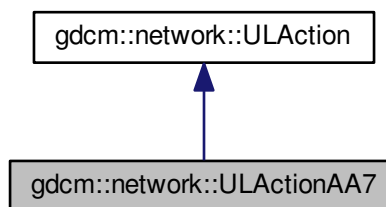
27.320 gdcm::network::ULActionAA7 Class Reference

```
#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 &outWaitingFor↵
Event, [EEventID](#) &outRaisedEvent)

27.320.1 Member Function Documentation

27.320.1.1 [EStateID](#) gdcm::network::ULActionAA7::PerformAction ([Subject](#) * s, [ULEvent](#) & *inEvent*, [ULConnection](#) & *inConnection*, bool & *outWaitingForEvent*, [EEventID](#) & *outRaisedEvent*) [\[virtual\]](#)

Implements [gdcm::network::ULAction](#).

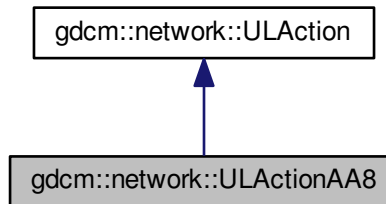
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

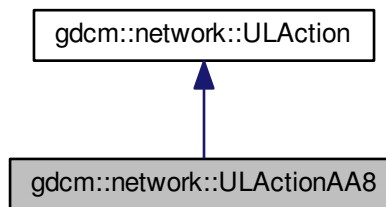
27.321 gdcmm::network::ULActionAA8 Class Reference

```
#include <gdcmmULActionAA.h>
```

Inheritance diagram for gdcmm::network::ULActionAA8:



Collaboration diagram for gdcmm::network::ULActionAA8:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingFor←
Event, [EEventID](#) &outRaisedEvent)

27.321.1 Member Function Documentation

27.321.1.1 `EStateID gdcmm::network::ULActionAA8::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements [gdcmm::network::ULAction](#).

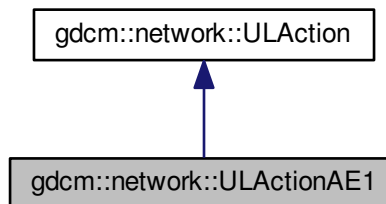
The documentation for this class was generated from the following file:

- [gdcmmULActionAA.h](#)

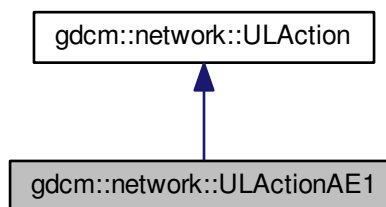
27.322 gdcmm::network::ULActionAE1 Class Reference

```
#include <gdcmmULActionAE.h>
```

Inheritance diagram for gdcmm::network::ULActionAE1:



Collaboration diagram for gdcmm::network::ULActionAE1:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingFor←
Event, [EEventID](#) &outRaisedEvent)

27.322.1 Member Function Documentation

27.322.1.1 `EStateID gdcmm::network::ULActionAE1::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements [gdcmm::network::ULAction](#).

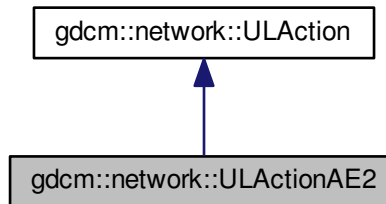
The documentation for this class was generated from the following file:

- [gdcmmULActionAE.h](#)

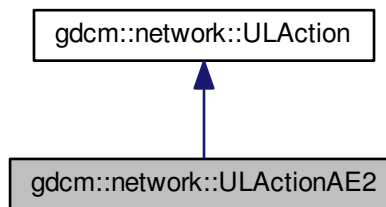
27.323 gdcm::network::ULActionAE2 Class Reference

```
#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 &outWaitingFor↵
Event, [EEventID](#) &outRaisedEvent)

27.323.1 Member Function Documentation

27.323.1.1 `EStateID gdcm::network::ULActionAE2::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements [gdcm::network::ULAction](#).

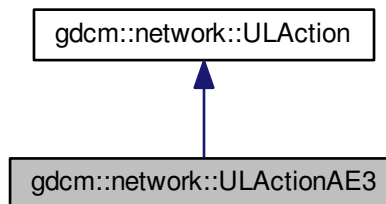
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

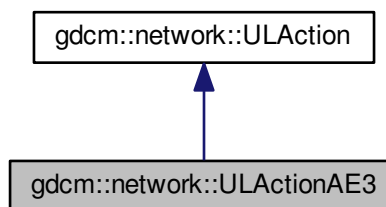
27.324 gdcm::network::ULActionAE3 Class Reference

```
#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 &outWaitingFor←
Event, [EEventID](#) &outRaisedEvent)

27.324.1 Member Function Documentation

27.324.1.1 [EStateID](#) `gdcm::network::ULActionAE3::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent)` [virtual]

Implements [gdcm::network::ULAction](#).

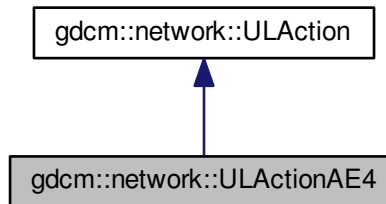
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

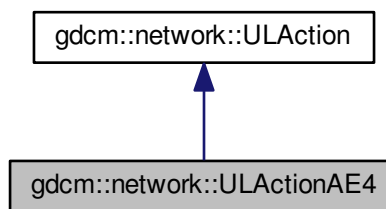
27.325 gdcm::network::ULActionAE4 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE4:



Collaboration diagram for gdcm::network::ULActionAE4:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingFor↵
Event, [EEventID](#) &outRaisedEvent)

27.325.1 Member Function Documentation

27.325.1.1 [EStateID](#) `gdcm::network::ULActionAE4::PerformAction` ([Subject](#) * s, [ULEvent](#) & *inEvent*, [ULConnection](#) & *inConnection*, bool & *outWaitingForEvent*, [EEventID](#) & *outRaisedEvent*) `[virtual]`

Implements [gdcm::network::ULAction](#).

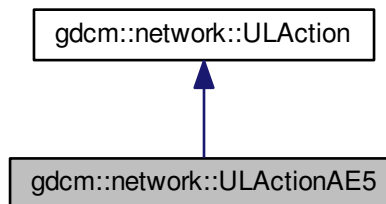
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

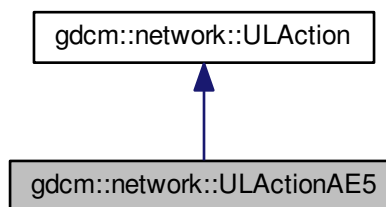
27.326 gdcm::network::ULActionAE5 Class Reference

```
#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 &outWaitingFor↵
Event, [EEventID](#) &outRaisedEvent)

27.326.1 Member Function Documentation

27.326.1.1 `EStateID gdcm::network::ULActionAE5::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements [gdcm::network::ULAction](#).

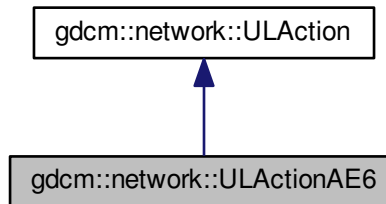
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

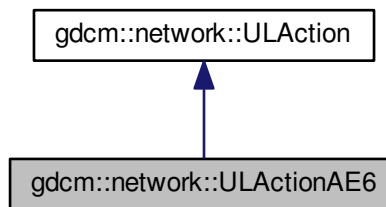
27.327 gdcm::network::ULActionAE6 Class Reference

```
#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 &outWaitingFor↵
Event, [EEventID](#) &outRaisedEvent)

27.327.1 Member Function Documentation

27.327.1.1 `EStateID gdcm::network::ULActionAE6::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements [gdcm::network::ULAction](#).

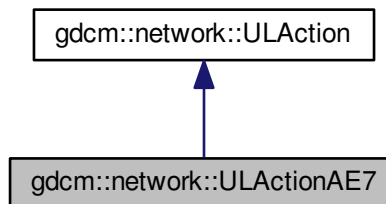
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

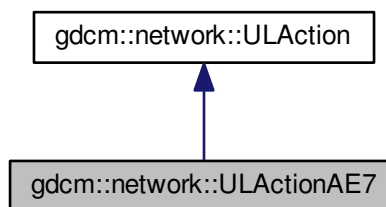
27.328 gdcmm::network::ULActionAE7 Class Reference

```
#include <gdcmmULActionAE.h>
```

Inheritance diagram for gdcmm::network::ULActionAE7:



Collaboration diagram for gdcmm::network::ULActionAE7:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingFor←
Event, [EEventID](#) &outRaisedEvent)

27.328.1 Member Function Documentation

27.328.1.1 [EStateID gdcmm::network::ULActionAE7::PerformAction](#) ([Subject](#) * s, [ULEvent](#) & *inEvent*, [ULConnection](#) & *inConnection*, bool & *outWaitingForEvent*, [EEventID](#) & *outRaisedEvent*) [[virtual](#)]

Implements [gdcmm::network::ULAction](#).

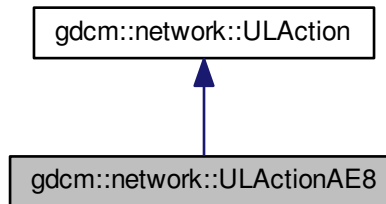
The documentation for this class was generated from the following file:

- [gdcmmULActionAE.h](#)

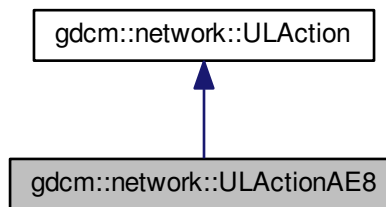
27.329 gdcm::network::ULActionAE8 Class Reference

```
#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 &outWaitingFor↵
Event, [EEventID](#) &outRaisedEvent)

27.329.1 Member Function Documentation

27.329.1.1 `EStateID gdcm::network::ULActionAE8::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements [gdcm::network::ULAction](#).

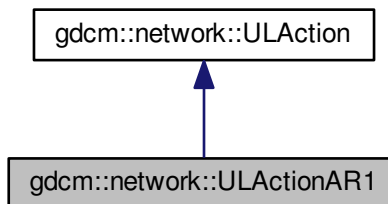
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

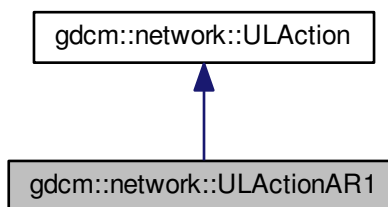
27.330 gdcm::network::ULActionAR1 Class Reference

```
#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 &outWaitingFor↵
Event, [EEventID](#) &outRaisedEvent)

27.330.1 Member Function Documentation

27.330.1.1 [EStateID](#) `gdcm::network::ULActionAR1::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent)` [virtual]

Implements [gdcm::network::ULAction](#).

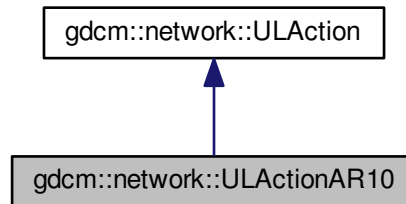
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

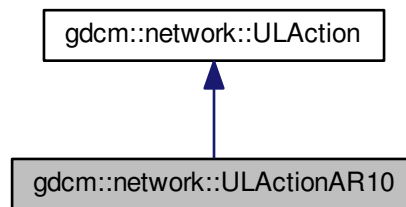
27.331 gdcm::network::ULActionAR10 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR10:



Collaboration diagram for gdcm::network::ULActionAR10:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingFor←
Event, [EEventID](#) &outRaisedEvent)

27.331.1 Member Function Documentation

27.331.1.1 **EStateID** `gdcm::network::ULActionAR10::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent)` [[virtual](#)]

Implements [gdcm::network::ULAction](#).

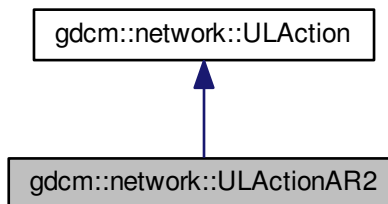
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

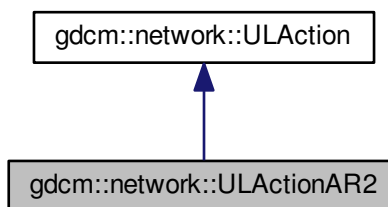
27.332 gdcm::network::ULActionAR2 Class Reference

```
#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)

27.332.1 Member Function Documentation

27.332.1.1 [EStateID](#) gdcm::network::ULActionAR2::PerformAction ([Subject](#) * s, [ULEvent](#) & *inEvent*, [ULConnection](#) & *inConnection*, bool & *outWaitingForEvent*, [EEventID](#) & *outRaisedEvent*) [virtual]

Implements [gdcm::network::ULAction](#).

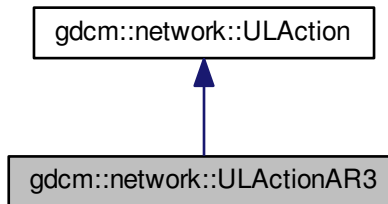
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

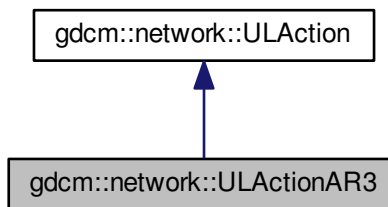
27.333 gdcmm::network::ULActionAR3 Class Reference

```
#include <gdcmmULActionAR.h>
```

Inheritance diagram for gdcmm::network::ULActionAR3:



Collaboration diagram for gdcmm::network::ULActionAR3:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingFor←
Event, [EEventID](#) &outRaisedEvent)

27.333.1 Member Function Documentation

27.333.1.1 `EStateID gdcmm::network::ULActionAR3::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements [gdcmm::network::ULAction](#).

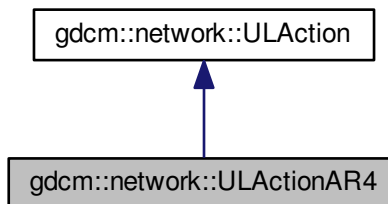
The documentation for this class was generated from the following file:

- [gdcmmULActionAR.h](#)

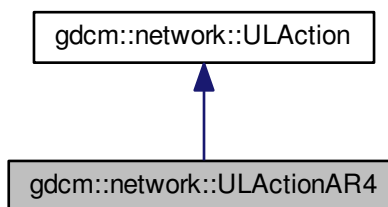
27.334 gdcm::network::ULActionAR4 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR4:



Collaboration diagram for gdcm::network::ULActionAR4:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingFor↵
Event, [EEventID](#) &outRaisedEvent)

27.334.1 Member Function Documentation

27.334.1.1 `EStateID gdcm::network::ULActionAR4::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements [gdcm::network::ULAction](#).

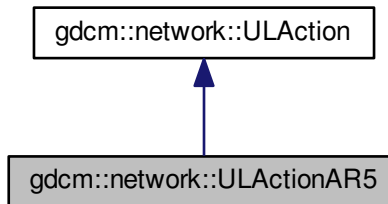
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

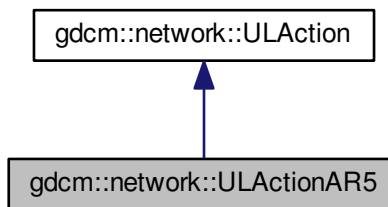
27.335 gdcmm::network::ULActionAR5 Class Reference

```
#include <gdcmmULActionAR.h>
```

Inheritance diagram for gdcmm::network::ULActionAR5:



Collaboration diagram for gdcmm::network::ULActionAR5:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingFor←
Event, [EEventID](#) &outRaisedEvent)

27.335.1 Member Function Documentation

27.335.1.1 `EStateID gdcmm::network::ULActionAR5::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements [gdcmm::network::ULAction](#).

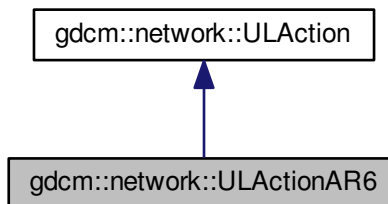
The documentation for this class was generated from the following file:

- [gdcmmULActionAR.h](#)

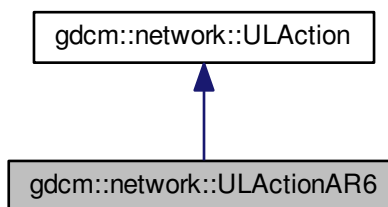
27.336 gdcm::network::ULActionAR6 Class Reference

```
#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 &outWaitingFor↵
Event, [EEventID](#) &outRaisedEvent)

27.336.1 Member Function Documentation

27.336.1.1 `EStateID gdcm::network::ULActionAR6::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements [gdcm::network::ULAction](#).

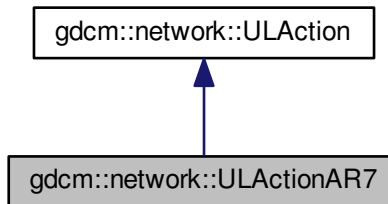
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

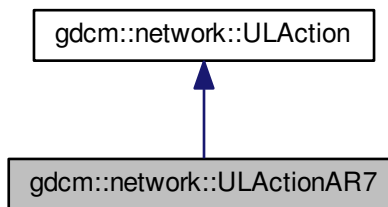
27.337 gdcmm::network::ULActionAR7 Class Reference

```
#include <gdcmmULActionAR.h>
```

Inheritance diagram for gdcmm::network::ULActionAR7:



Collaboration diagram for gdcmm::network::ULActionAR7:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingFor←
Event, [EEventID](#) &outRaisedEvent)

27.337.1 Member Function Documentation

27.337.1.1 [EStateID gdcmm::network::ULActionAR7::PerformAction](#) ([Subject](#) * s, [ULEvent](#) & *inEvent*, [ULConnection](#) & *inConnection*, bool & *outWaitingForEvent*, [EEventID](#) & *outRaisedEvent*) [[virtual](#)]

Implements [gdcmm::network::ULAction](#).

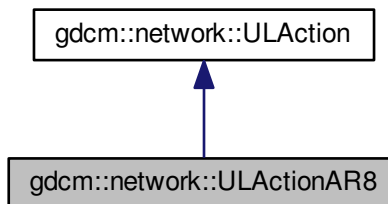
The documentation for this class was generated from the following file:

- [gdcmmULActionAR.h](#)

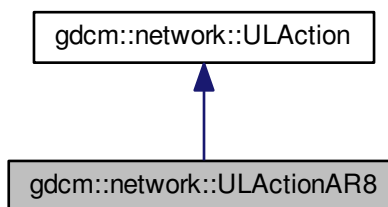
27.338 gdcm::network::ULActionAR8 Class Reference

```
#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 &outWaitingFor↵
Event, [EEventID](#) &outRaisedEvent)

27.338.1 Member Function Documentation

27.338.1.1 `EStateID gdcm::network::ULActionAR8::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements [gdcm::network::ULAction](#).

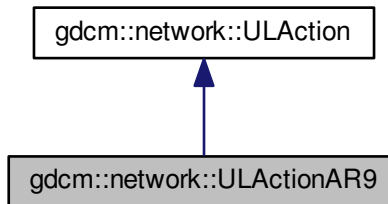
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

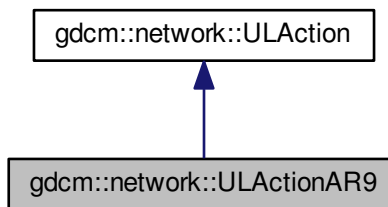
27.339 gdcmm::network::ULActionAR9 Class Reference

```
#include <gdcmmULActionAR.h>
```

Inheritance diagram for gdcmm::network::ULActionAR9:



Collaboration diagram for gdcmm::network::ULActionAR9:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingFor←
Event, [EEventID](#) &outRaisedEvent)

27.339.1 Member Function Documentation

27.339.1.1 [EStateID gdcmm::network::ULActionAR9::PerformAction](#) ([Subject](#) * s, [ULEvent](#) & *inEvent*, [ULConnection](#) & *inConnection*, bool & *outWaitingForEvent*, [EEventID](#) & *outRaisedEvent*) [[virtual](#)]

Implements [gdcmm::network::ULAction](#).

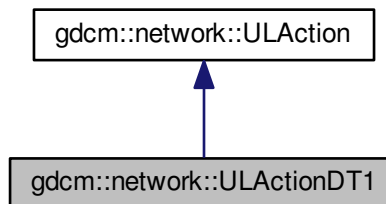
The documentation for this class was generated from the following file:

- [gdcmmULActionAR.h](#)

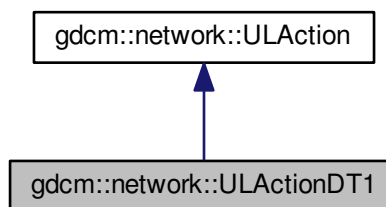
27.340 gdcm::network::ULActionDT1 Class Reference

```
#include <gdcmULActionDT.h>
```

Inheritance diagram for gdcm::network::ULActionDT1:



Collaboration diagram for gdcm::network::ULActionDT1:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingFor↵
Event, [EEventID](#) &outRaisedEvent)

27.340.1 Member Function Documentation

27.340.1.1 [EStateID](#) `gdcm::network::ULActionDT1::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent)` [[virtual](#)]

Implements [gdcm::network::ULAction](#).

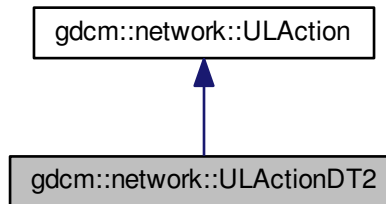
The documentation for this class was generated from the following file:

- [gdcmULActionDT.h](#)

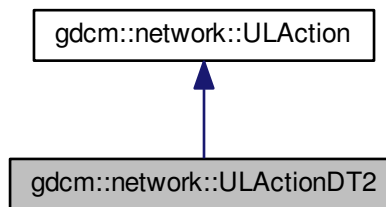
27.341 gdcmm::network::ULActionDT2 Class Reference

```
#include <gdcmmULActionDT.h>
```

Inheritance diagram for gdcmm::network::ULActionDT2:



Collaboration diagram for gdcmm::network::ULActionDT2:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingFor←
Event, [EEventID](#) &outRaisedEvent)

27.341.1 Member Function Documentation

27.341.1.1 [EStateID gdcmm::network::ULActionDT2::PerformAction](#) ([Subject](#) * s, [ULEvent](#) & *inEvent*, [ULConnection](#) & *inConnection*, bool & *outWaitingForEvent*, [EEventID](#) & *outRaisedEvent*) [[virtual](#)]

Implements [gdcmm::network::ULAction](#).

The documentation for this class was generated from the following file:

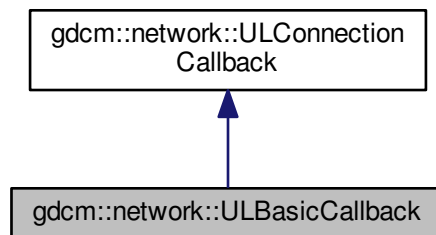
- [gdcmmULActionDT.h](#)

27.342 gdcm::network::ULBasicCallback Class Reference

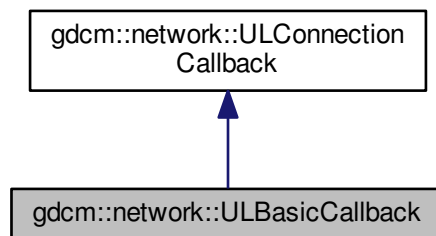
[ULBasicCallback](#) This is the most basic of callbacks for how the [ULConnectionManager](#) handles incoming datasets. DataSets are just concatenated to the mDataSets vector, and the result can be pulled out of the vector by later code. Alternatives to this method include progress updates, saving to disk, etc. This class is NOT THREAD SAFE. Access the dataset vector after the entire set of datasets has been returned by the [ULConnectionManager](#).

```
#include <gdcmULBasicCallback.h>
```

Inheritance diagram for gdcm::network::ULBasicCallback:



Collaboration diagram for gdcm::network::ULBasicCallback:



Public Member Functions

- [ULBasicCallback](#) ()
- virtual [~ULBasicCallback](#) ()
- std::vector< [DataSet](#) > const & [GetDataSets](#) () const
- std::vector< [DataSet](#) > const & [GetResponses](#) () const
- virtual void [HandleDataSet](#) (const [DataSet](#) &inDataSet)
- virtual void [HandleResponse](#) (const [DataSet](#) &inDataSet)

Additional Inherited Members

27.342.1 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](#).

27.342.2 Constructor & Destructor Documentation

27.342.2.1 `gdcm::network::ULBasicCallback::ULBasicCallback () [inline]`

27.342.2.2 `virtual gdcm::network::ULBasicCallback::~~ULBasicCallback () [inline],[virtual]`

27.342.3 Member Function Documentation

27.342.3.1 `std::vector<DataSet> const& gdcm::network::ULBasicCallback::GetDataSets () const`

27.342.3.2 `std::vector<DataSet> const& gdcm::network::ULBasicCallback::GetResponses () const`

27.342.3.3 `virtual void gdcm::network::ULBasicCallback::HandleDataSet (const DataSet & inDataSet) [virtual]`

Implements [gdcm::network::ULConnectionCallback](#).

27.342.3.4 `virtual void gdcm::network::ULBasicCallback::HandleResponse (const DataSet & inDataSet) [virtual]`

Implements [gdcm::network::ULConnectionCallback](#).

The documentation for this class was generated from the following file:

- [gdcmULBasicCallback.h](#)

27.343 gdcm::network::ULConnection Class Reference

[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.

```
#include <gdcmULConnection.h>
```

Public Member Functions

- [ULConnection](#) (const [ULConnectionInfo](#) &inUserInformation)
- virtual [~ULConnection](#) ()
- void [AddAcceptedPresentationContext](#) (const [PresentationContextAC](#) &inPC)
- [PresentationContextRQ FindContext](#) (const [DataElement](#) &de) const
- std::vector< [PresentationContextAC](#) > const & [GetAcceptedPresentationContexts](#) () const
- std::vector< [PresentationContextAC](#) > & [GetAcceptedPresentationContexts](#) ()
- 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 [SetMaxPDUSize](#) (uint32_t inSize)
- void [SetPresentationContexts](#) (const std::vector< [PresentationContextRQ](#) > &inContexts)
- void [SetPresentationContexts](#) (const std::vector< [PresentationContext](#) > &inContexts)
- void [SetState](#) (const [EStateID](#) &inState)
- void [StopProtocol](#) ()

Friends

- class [ULActionAE6](#)
- class [ULConnectionManager](#)

27.343.1 Detailed Description

[ULConnection](#) This is the class that contains the socket to another machine, and passes data through itself, as well as maintaining a sense of state.

The [ULConnectionManager](#) tells the [ULConnection](#) what data can actually be sent.

This class is done this way so that it can be eventually be replaced with a [ULSecureConnection](#), if such a protocol is warranted, so that all data that passes through can be managed through a secure connection. For now, this class provides a simple pass-through mechanism to the socket itself.

So, for instance, a gdcm object will be passes to this object, and it will then get passed along the connection, if that connection is in the proper state to do so.

For right now, this class is not directly intended to be inherited from, but the potential for future [ULSecureConnection](#) warrants the addition, rather than having everything be managed from within the [ULConnectionManager](#) (or this class) without a wrapper.

27.343.2 Constructor & Destructor Documentation

27.343.2.1 `gdcm::network::ULConnection::ULConnection (const ULConnectionInfo & inUserInformation)`

27.343.2.2 `virtual gdcm::network::ULConnection::~~ULConnection () [virtual]`

27.343.3 Member Function Documentation

27.343.3.1 `void gdcm::network::ULConnection::AddAcceptedPresentationContext (const PresentationContextAC & inPC)`

27.343.3.2 **PresentationContextRQ** gdcm::network::ULConnection::FindContext (const DataElement & *de*) const

27.343.3.3 std::vector<PresentationContextAC> const& gdcm::network::ULConnection::GetAcceptedPresentationContexts () const

27.343.3.4 std::vector<PresentationContextAC>& gdcm::network::ULConnection::GetAcceptedPresentationContexts ()

27.343.3.5 const ULConnectionInfo& gdcm::network::ULConnection::GetConnectionInfo () const

27.343.3.6 uint32_t gdcm::network::ULConnection::GetMaxPDUSize () const

27.343.3.7 const PresentationContextAC* gdcm::network::ULConnection::GetPresentationContextACByID (uint8_t *id*) const

27.343.3.8 uint8_t gdcm::network::ULConnection::GetPresentationContextIDFromPresentationContext (PresentationContextRQ const & *pc*) const

return 0 upon error

27.343.3.9 const PresentationContextRQ* gdcm::network::ULConnection::GetPresentationContextRQByID (uint8_t *id*) const

27.343.3.10 std::vector<PresentationContextRQ> const& gdcm::network::ULConnection::GetPresentationContexts () const

27.343.3.11 std::iostream* gdcm::network::ULConnection::GetProtocol ()

27.343.3.12 EStateID gdcm::network::ULConnection::GetState () const

27.343.3.13 ARTIMTimer& gdcm::network::ULConnection::GetTimer ()

27.343.3.14 bool gdcm::network::ULConnection::InitializeConnection ()

used to establish scu connections

27.343.3.15 bool gdcm::network::ULConnection::InitializeIncomingConnection ()

used to establish scp connections

27.343.3.16 void gdcm::network::ULConnection::SetMaxPDUSize (uint32_t *inSize*)

27.343.3.17 void gdcm::network::ULConnection::SetPresentationContexts (const std::vector< PresentationContextRQ > & *inContexts*)

27.343.3.18 void gdcm::network::ULConnection::SetPresentationContexts (const std::vector< PresentationContext > & *inContexts*)

27.343.3.19 void gdcm::network::ULConnection::SetState (const EStateID & *inState*)

27.343.3.20 void gdcm::network::ULConnection::StopProtocol ()

27.343.4 Friends And Related Function Documentation

27.343.4.1 friend class **ULActionAE6** [friend]

27.343.4.2 friend class **ULConnectionManager** [friend]

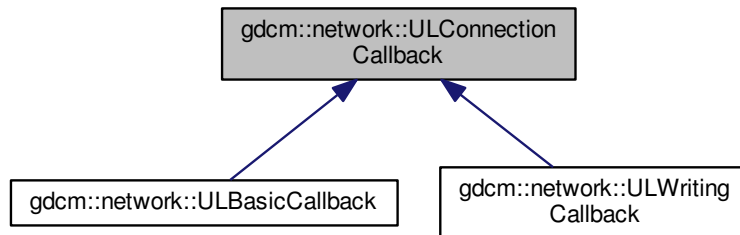
The documentation for this class was generated from the following file:

- [gdcmULConnection.h](#)

27.344 gdcm::network::ULConnectionCallback Class Reference

```
#include <gdcmULConnectionCallback.h>
```

Inheritance diagram for gdcm::network::ULConnectionCallback:



Public Member Functions

- [ULConnectionCallback](#) ()
- virtual [~ULConnectionCallback](#) ()
- 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](#)

27.344.1 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.

27.344.2 Constructor & Destructor Documentation

27.344.2.1 `gdcm::network::ULConnectionCallback::ULConnectionCallback ()` `[inline]`

27.344.2.2 `virtual gdcm::network::ULConnectionCallback::~~ULConnectionCallback ()` `[inline]`, `[virtual]`

27.344.3 Member Function Documentation

27.344.3.1 `void gdcm::network::ULConnectionCallback::DataSetHandled ()` `[inline]`, `[protected]`

27.344.3.2 `bool gdcm::network::ULConnectionCallback::DataSetHandles ()` `const` `[inline]`

27.344.3.3 `virtual void gdcm::network::ULConnectionCallback::HandleDataSet (const DataSet & inDataSet)` `[pure virtual]`

Implemented in [gdcm::network::ULBasicCallback](#), and [gdcm::network::ULWritingCallback](#).

27.344.3.4 `virtual void gdcm::network::ULConnectionCallback::HandleResponse (const DataSet & inDataSet)` `[pure virtual]`

Implemented in [gdcm::network::ULBasicCallback](#), and [gdcm::network::ULWritingCallback](#).

27.344.3.5 `void gdcm::network::ULConnectionCallback::ResetHandledDataSet ()` `[inline]`

27.344.3.6 `void gdcm::network::ULConnectionCallback::SetImplicitFlag (const bool imp)` `[inline]`

27.344.4 Member Data Documentation

27.344.4.1 `bool gdcm::network::ULConnectionCallback::mImplicit` `[protected]`

The documentation for this class was generated from the following file:

- [gdcmULConnectionCallback.h](#)

27.345 gdcm::network::ULConnectionInfo Class Reference

[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.

```
#include <gdcmULConnectionInfo.h>
```

Public Member Functions

- [ULConnectionInfo](#) ()
- const char * [GetCalledAETitle](#) () const
- std::string [GetCalledComputerName](#) () const
- unsigned long [GetCalledIPAddress](#) () const
- int [GetCalledIPPort](#) () const
- const char * [GetCallingAETitle](#) () const
- unsigned long [GetMaxPDULength](#) () const
- bool [Initialize](#) ([UserInfo](#) const &inUserInfo, const char *inCalledAETitle, const char *inCallingAETitle, unsigned long inCalledIPAddress, int inCalledIPPort, std::string inCalledComputerName)
- void [SetMaxPDULength](#) (unsigned long inMaxPDULength)

27.345.1 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.

27.345.2 Constructor & Destructor Documentation

27.345.2.1 `gdcm::network::ULConnectionInfo::ULConnectionInfo ()`

27.345.3 Member Function Documentation

27.345.3.1 `const char* gdcm::network::ULConnectionInfo::GetCalledAETitle () const`

27.345.3.2 `std::string gdcm::network::ULConnectionInfo::GetCalledComputerName () const`

27.345.3.3 `unsigned long gdcm::network::ULConnectionInfo::GetCalledIPAddress () const`

27.345.3.4 `int gdcm::network::ULConnectionInfo::GetCalledIPPort () const`

27.345.3.5 `const char* gdcm::network::ULConnectionInfo::GetCallingAETitle () const`

27.345.3.6 `unsigned long gdcm::network::ULConnectionInfo::GetMaxPDULength () const`

27.345.3.7 `bool gdcm::network::ULConnectionInfo::Initialize (UserInfo const & inUserInfo, const char * inCalledAETitle, const char * inCallingAETitle, unsigned long inCalledIPAddress, int inCalledIPPort, std::string inCalledComputerName)`

27.345.3.8 `void gdcm::network::ULConnectionInfo::SetMaxPDULength (unsigned long inMaxPDULength)`

The documentation for this class was generated from the following file:

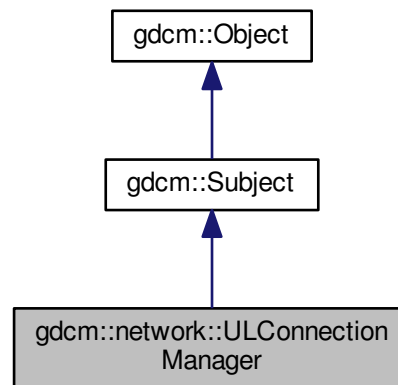
- [gdcmULConnectionInfo.h](#)

27.346 gdcmm::network::ULConnectionManager Class Reference

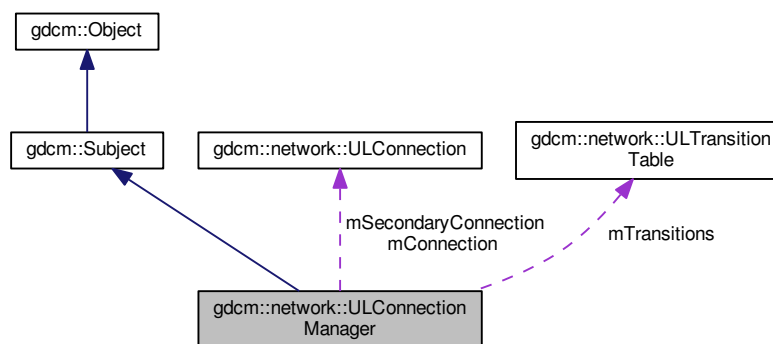
ULConnectionManager The **ULConnectionManager** performs actions on the **ULConnection** given inputs from the user and from the state of what's going on around the connection (ie, timeouts of the ARTIM timer, responses from the peer across the connection, etc).

```
#include <gdcmmULConnectionManager.h>
```

Inheritance diagram for gdcmm::network::ULConnectionManager:



Collaboration diagram for gdcmm::network::ULConnectionManager:



Public Member Functions

- [ULConnectionManager](#) ()
- virtual [~ULConnectionManager](#) ()

- 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=NULL, std::streampos dataSetOffset=0)
 - void [SendStore](#) (const [File](#) &file, [ULConnectionCallback](#) *inCallback, std::istream *pStream=NULL, 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

27.346.1 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](#).

27.346.2 Constructor & Destructor Documentation

27.346.2.1 `gdcm::network::ULConnectionManager::ULConnectionManager (const ULConnectionManager & inCM)`
[protected]

27.346.2.2 `gdcm::network::ULConnectionManager::ULConnectionManager ()`

27.346.2.3 `virtual gdcm::network::ULConnectionManager::~~ULConnectionManager ()` [virtual]

27.346.3 Member Function Documentation

27.346.3.1 `bool gdcm::network::ULConnectionManager::BreakConnection (const double & inTimeout)`

27.346.3.2 `void gdcm::network::ULConnectionManager::BreakConnectionNow ()`

27.346.3.3 `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.

27.346.3.4 `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

27.346.3.5 `EStateID gdcm::network::ULConnectionManager::RunEventLoop (ULEvent & inEvent, ULConnection * inWhichConnection, ULConnectionCallback * inCallback, const bool & startWaiting)` [protected]

27.346.3.6 `EStateID gdcm::network::ULConnectionManager::RunMoveEventLoop (ULEvent & inEvent, ULConnectionCallback * inCallback)` [protected]

27.346.3.7 `std::vector<PresentationDataValue> gdcm::network::ULConnectionManager::SendEcho ()`

27.346.3.8 `std::vector<DataSet> gdcm::network::ULConnectionManager::SendFind (const BaseRootQuery * inRootQuery)`

27.346.3.9 `void gdcm::network::ULConnectionManager::SendFind (const BaseRootQuery * inRootQuery, ULConnectionCallback * inCallback)`

27.346.3.10 `std::vector<DataSet> gdcm::network::ULConnectionManager::SendMove (const BaseRootQuery * inRootQuery)`

27.346.3.11 `bool gdcm::network::ULConnectionManager::SendMove (const BaseRootQuery * inRootQuery, ULConnectionCallback * inCallback)`

return false upon error

- 27.346.3.12 `std::vector<DataSet> gdcm::network::ULConnectionManager::SendNAction (const BaseQuery * inQuery)`
- 27.346.3.13 `void gdcm::network::ULConnectionManager::SendNAction (const BaseQuery * inQuery, ULConnectionCallback * inCallback)`
- 27.346.3.14 `std::vector<DataSet> gdcm::network::ULConnectionManager::SendNCreate (const BaseQuery * inQuery)`
- 27.346.3.15 `void gdcm::network::ULConnectionManager::SendNCreate (const BaseQuery * inQuery, ULConnectionCallback * inCallback)`
- 27.346.3.16 `std::vector<DataSet> gdcm::network::ULConnectionManager::SendNDelete (const BaseQuery * inQuery)`
- 27.346.3.17 `void gdcm::network::ULConnectionManager::SendNDelete (const BaseQuery * inQuery, ULConnectionCallback * inCallback)`
- 27.346.3.18 `std::vector<DataSet> gdcm::network::ULConnectionManager::SendNEventReport (const BaseQuery * inQuery)`
- 27.346.3.19 `void gdcm::network::ULConnectionManager::SendNEventReport (const BaseQuery * inQuery, ULConnectionCallback * inCallback)`
- 27.346.3.20 `std::vector<DataSet> gdcm::network::ULConnectionManager::SendNGet (const BaseQuery * inQuery)`
- 27.346.3.21 `void gdcm::network::ULConnectionManager::SendNGet (const BaseQuery * inQuery, ULConnectionCallback * inCallback)`
- 27.346.3.22 `std::vector<DataSet> gdcm::network::ULConnectionManager::SendNSet (const BaseQuery * inQuery)`
- 27.346.3.23 `void gdcm::network::ULConnectionManager::SendNSet (const BaseQuery * inQuery, ULConnectionCallback * inCallback)`
- 27.346.3.24 `std::vector<DataSet> gdcm::network::ULConnectionManager::SendStore (const File & file, std::istream * pStream = NULL, std::streampos dataSetOffset = 0)`
- 27.346.3.25 `void gdcm::network::ULConnectionManager::SendStore (const File & file, ULConnectionCallback * inCallback, std::istream * pStream = NULL, std::streampos dataSetOffset = 0)`

callback based API

27.346.4 Member Data Documentation

- 27.346.4.1 `ULConnection* gdcm::network::ULConnectionManager::mConnection` [protected]
- 27.346.4.2 `ULConnection* gdcm::network::ULConnectionManager::mSecondaryConnection` [protected]
- 27.346.4.3 `ULTransitionTable gdcm::network::ULConnectionManager::mTransitions` [protected]

The documentation for this class was generated from the following file:

- [gdcmULConnectionManager.h](#)

27.347 gdcmm::network::ULEvent Class Reference

[ULEvent](#) base class for network events.

```
#include <gdcmmULEvent.h>
```

Public Member Functions

- [ULEvent](#) (const [EEventID](#) &inEventID, std::vector< [BasePDU](#) * > inBasePDU, std::istream *iStream=NULL, std::streampos posDataSet=0)
- [ULEvent](#) (const [EEventID](#) &inEventID, [BasePDU](#) *inBasePDU, std::istream *iStream=NULL, std::streampos posDataSet=0)
- [~ULEvent](#) ()
- std::streampos [GetDataSetPos](#) () const
- [EEventID](#) [GetEvent](#) () const
- std::istream * [GetStream](#) () const
- std::vector< [BasePDU](#) * > const & [GetPDUs](#) () const
- void [SetEvent](#) (const [EEventID](#) &inEvent)
- void [SetPDU](#) (std::vector< [BasePDU](#) * > const &inPDU)

27.347.1 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!

27.347.2 Constructor & Destructor Documentation

27.347.2.1 `gdcmm::network::ULEvent (const EEventID & inEventID, std::vector< BasePDU * > inBasePDU, std::istream * iStream = NULL, std::streampos posDataSet = 0) [inline]`

27.347.2.2 `gdcmm::network::ULEvent (const EEventID & inEventID, BasePDU * inBasePDU, std::istream * iStream = NULL, std::streampos posDataSet = 0) [inline]`

27.347.2.3 `gdcmm::network::ULEvent::~~ULEvent () [inline]`

27.347.3 Member Function Documentation

27.347.3.1 `std::streampos gdcmm::network::ULEvent::GetDataSetPos () const [inline]`

27.347.3.2 `EEventID gdcmm::network::ULEvent::GetEvent () const [inline]`

27.347.3.3 `std::istream* gdcmm::network::ULEvent::GetStream () const [inline]`

27.347.3.4 `std::vector<BasePDU*> const& gdcmm::network::ULEvent::GetPDUs () const [inline]`

27.347.3.5 `void gdcmm::network::ULEvent::SetEvent (const EEventID & inEvent) [inline]`

27.347.3.6 void gdcm::network::ULEvent::SetPDU (std::vector< BasePDU * > const & *inPDU*) [inline]

The documentation for this class was generated from the following file:

- [gdcmULEvent.h](#)

27.348 gdcm::network::ULTransitionTable Class Reference

[ULTransitionTable](#) The transition table of all the ULEvents, new ULActions, and ULStates.

```
#include <gdcmULTransitionTable.h>
```

Public Member Functions

- [ULTransitionTable](#) ()
- void [HandleEvent](#) (Subject *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EventID](#) &outRaisedEvent) const
- void [PrintTable](#) () const

27.348.1 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.

27.348.2 Constructor & Destructor Documentation

27.348.2.1 gdcm::network::ULTransitionTable::ULTransitionTable ()

27.348.3 Member Function Documentation

27.348.3.1 void gdcm::network::ULTransitionTable::HandleEvent (Subject * s, [ULEvent](#) & *inEvent*, [ULConnection](#) & *inConnection*, bool & *outWaitingForEvent*, [EventID](#) & *outRaisedEvent*) const

27.348.3.2 void gdcm::network::ULTransitionTable::PrintTable () const

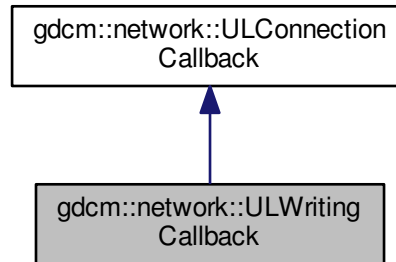
The documentation for this class was generated from the following file:

- [gdcmULTransitionTable.h](#)

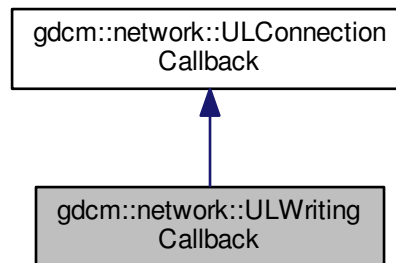
27.349 gdcm::network::ULWritingCallback Class Reference

```
#include <gdcmULWritingCallback.h>
```

Inheritance diagram for `gdcm::network::ULWritingCallback`:



Collaboration diagram for `gdcm::network::ULWritingCallback`:



Public Member Functions

- [ULWritingCallback](#) ()
- virtual [~ULWritingCallback](#) ()
- virtual void [HandleDataSet](#) (const [DataSet](#) &inDataSet)
- virtual void [HandleResponse](#) (const [DataSet](#) &inDataSet)
- void [SetDirectory](#) (const std::string &inDirectoryName)

provide the directory into which all files are written.

Additional Inherited Members

27.349.1 Constructor & Destructor Documentation

27.349.1.1 `gdcm::network::ULWritingCallback::ULWritingCallback ()` `[inline]`

27.349.1.2 `virtual gdcm::network::ULWritingCallback::~~ULWritingCallback ()` `[inline],[virtual]`

27.349.2 Member Function Documentation

27.349.2.1 `virtual void gdcm::network::ULWritingCallback::HandleDataSet (const DataSet & inDataSet)` `[virtual]`

Implements [gdcm::network::ULConnectionCallback](#).

27.349.2.2 `virtual void gdcm::network::ULWritingCallback::HandleResponse (const DataSet & inDataSet)` `[virtual]`

Implements [gdcm::network::ULConnectionCallback](#).

27.349.2.3 `void gdcm::network::ULWritingCallback::SetDirectory (const std::string & inDirectoryName)` `[inline]`

provide the directory into which all files are written.

The documentation for this class was generated from the following file:

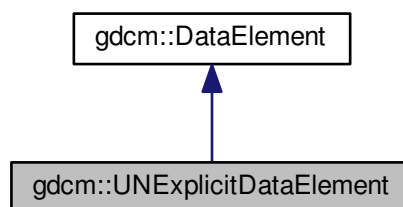
- [gdcmULWritingCallback.h](#)

27.350 gdcm::UNExplicitDataElement Class Reference

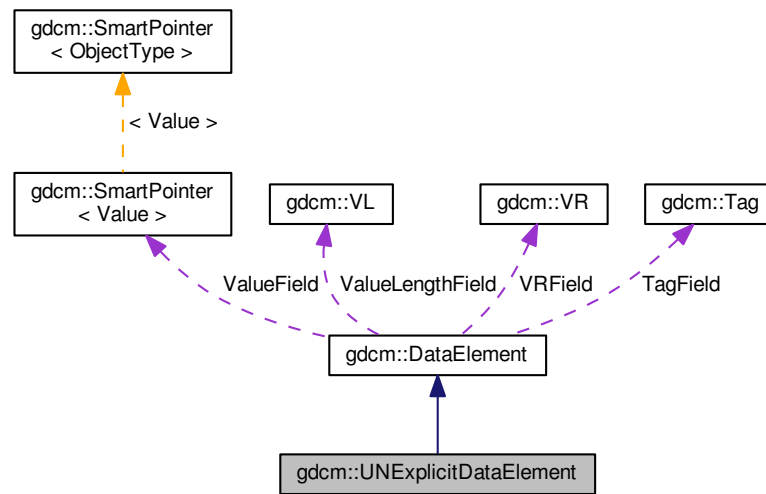
Class to read/write a [DataElement](#) as UNExplicit Data [Element](#).

```
#include <gdcmUNExplicitDataElement.h>
```

Inheritance diagram for `gdcm::UNExplicitDataElement`:



Collaboration diagram for `gdcmm::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

27.350.1 Detailed Description

Class to read/write a [DataElement](#) as UNExplicit Data [Element](#).

Note

bla

27.350.2 Member Function Documentation

27.350.2.1 VL `gdcmm::UNExplicitDataElement::GetLength` () const

27.350.2.2 `template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::Read (std::istream & is)`

27.350.2.3 `template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::ReadPreValue (std::istream & is)`

27.350.2.4 `template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::ReadValue (std::istream & is, bool readvalues = true)`

27.350.2.5 `template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::ReadWithLength (std::istream & is, VL & length)`

The documentation for this class was generated from the following file:

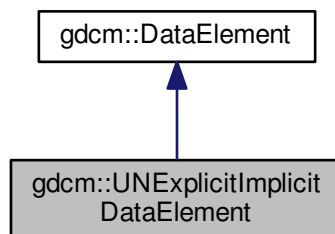
- [gdcmUNExplicitDataElement.h](#)

27.351 gdcm::UNExplicitImplicitDataElement Class Reference

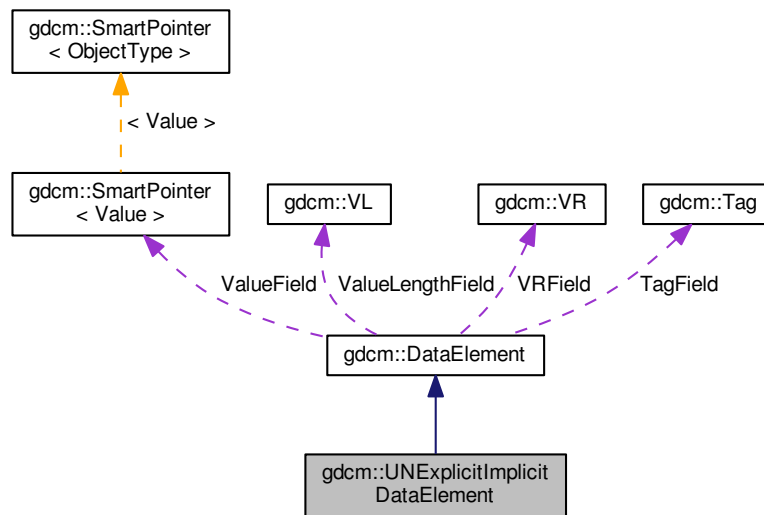
Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#) This class gather two known bugs:

```
#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

27.351.1 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/TherapysGDCM120Bug.dcm`

27.351.2 Member Function Documentation

27.351.2.1 VL `gdcM::UNExplicitImplicitDataElement::GetLength` () const

27.351.2.2 `template<typename TSwap > std::istream& gdcm::UNExplicitImplicitDataElement::Read (std::istream & is)`

27.351.2.3 `template<typename TSwap > std::istream& gdcm::UNExplicitImplicitDataElement::ReadPreValue (std::istream & is)`

27.351.2.4 `template<typename TSwap > std::istream& gdcm::UNExplicitImplicitDataElement::ReadValue (std::istream & is)`

The documentation for this class was generated from the following file:

- [gdcmUNExplicitImplicitDataElement.h](#)

27.352 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)

27.352.1 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](#)

27.352.2 Member Function Documentation

27.352.2.1 `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 $(n / 2) * 3$

27.352.2.2 `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](#)

27.353 gdcmm::Usage Class Reference

Usage.

```
#include <gdcmmUsage.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)

27.353.1 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.

27.353.2 Member Enumeration Documentation

27.353.2.1 enum gdcm::Usage::UsageType

Enumerator

Mandatory

Conditional

UserOption

Invalid

27.353.3 Constructor & Destructor Documentation

27.353.3.1 gdcm::Usage::Usage (UsageType type = Invalid) [inline]

27.353.4 Member Function Documentation

27.353.4.1 static const char* gdcm::Usage::GetUsageString (UsageType type) [static]

Referenced by gdcm::operator<<().

27.353.4.2 static UsageType gdcm::Usage::GetUsageType (const char * type) [static]

27.353.4.3 gdcm::Usage::operator UsageType () const [inline]

27.353.5 Friends And Related Function Documentation

27.353.5.1 std::ostream& operator<< (std::ostream & os, const Usage & vr) [friend]

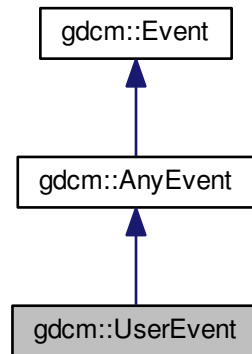
The documentation for this class was generated from the following file:

- [gdcmUsage.h](#)

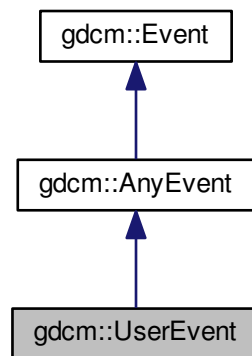
27.354 gdcm::UserEvent Class Reference

```
#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](#)

27.355 gdcm::network::UserInformation Class Reference

[UserInformation Table](#) 9-16 USER INFORMATION ITEM FIELDS.

```
#include <gdcmUserInformation.h>
```

Public Member Functions

- [UserInformation](#) ()
- [~UserInformation](#) ()
- void [AddRoleSelectionSub](#) ([RoleSelectionSub](#) const &r)
- void [AddSOPClassExtendedNegociationSub](#) ([SOPClassExtendedNegociationSub](#) const &s)
- const [MaximumLengthSub](#) & [GetMaximumLengthSub](#) () const
- [MaximumLengthSub](#) & [GetMaximumLengthSub](#) ()
- [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

27.355.1 Detailed Description

[UserInformation Table](#) 9-16 USER INFORMATION ITEM FIELDS.

TODO what is the goal of :

[Table](#) 9-20 USER INFORMATION ITEM FIELDS

27.355.2 Constructor & Destructor Documentation

27.355.2.1 `gdcm::network::UserInformation::UserInformation ()`

27.355.2.2 `gdcm::network::UserInformation::~~UserInformation ()`

27.355.3 Member Function Documentation

27.355.3.1 `void gdcm::network::UserInformation::AddRoleSelectionSub (RoleSelectionSub const & r)`

27.355.3.2 `void gdcm::network::UserInformation::AddSOPClassExtendedNegociationSub (SOPClassExtendedNegociationSub const & s)`

27.355.3.3 `const MaximumLengthSub& gdcm::network::UserInformation::GetMaximumLengthSub () const` `[inline]`

27.355.3.4 `MaximumLengthSub& gdcm::network::UserInformation::GetMaximumLengthSub ()` `[inline]`

27.355.3.5 `UserInformation& gdcm::network::UserInformation::operator= (const UserInformation &)`

27.355.3.6 `void gdcm::network::UserInformation::Print (std::ostream & os) const`

27.355.3.7 `std::istream& gdcm::network::UserInformation::Read (std::istream & is)`

27.355.3.8 `size_t gdcmm::network::UserInformation::Size () const`

27.355.3.9 `const std::ostream& gdcmm::network::UserInformation::Write (std::ostream & os) const`

The documentation for this class was generated from the following file:

- [gdcmmUserInformation.h](#)

27.356 gdcmm::UUIDGenerator Class Reference

Class for generating unique UUID generate DCE 1.1 uid.

```
#include <gdcmmUUIDGenerator.h>
```

Public Member Functions

- `const char * Generate ()`

Static Public Member Functions

- `static bool IsValid (const char *uid)`
Find out if the string is a valid UUID or not.

27.356.1 Detailed Description

Class for generating unique UUID generate DCE 1.1 uid.

27.356.2 Member Function Documentation

27.356.2.1 `const char* gdcmm::UUIDGenerator::Generate ()`

Return the generated uuid NOT THREAD SAFE

27.356.2.2 `static bool gdcmm::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:

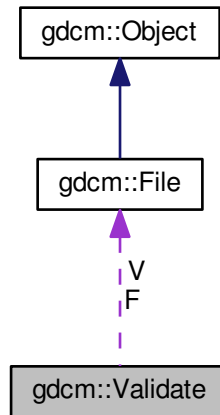
- [gdcmmUUIDGenerator.h](#)

27.357 gdcmm::Validate Class Reference

[Validate](#) class.

```
#include <gdcmmValidate.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](#)

27.357.1 Detailed Description

[Validate](#) class.

27.357.2 Constructor & Destructor Documentation

27.357.2.1 `gdcm::Validate::Validate ()`

27.357.2.2 `gdcm::Validate::~~Validate ()`

27.357.3 Member Function Documentation

27.357.3.1 `const File& gdcm::Validate::GetValidatedFile ()` `[inline]`

27.357.3.2 `void gdcM::Validate::SetFile (File const & f)` `[inline]`

27.357.3.3 `void gdcM::Validate::Validation ()`

27.357.4 Member Data Documentation

27.357.4.1 `const File* gdcM::Validate::F` `[protected]`

27.357.4.2 `File gdcM::Validate::V` `[protected]`

The documentation for this class was generated from the following file:

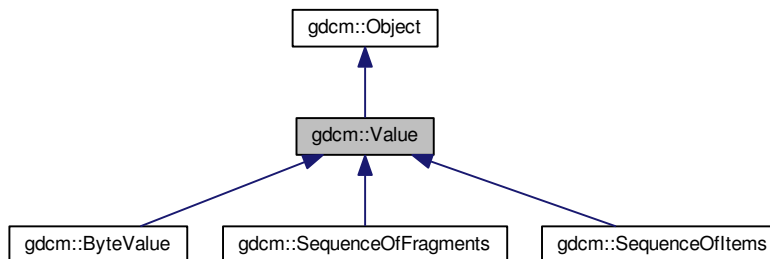
- [gdcMValidate.h](#)

27.358 gdcM::Value Class Reference

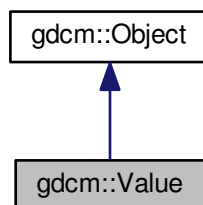
Class to represent the value of a Data [Element](#).

```
#include <gdcMValue.h>
```

Inheritance diagram for gdcM::Value:



Collaboration diagram for gdcM::Value:



Public Member Functions

- [Value](#) ()
- [~Value](#) ()
- 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](#)

27.358.1 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.

27.358.2 Constructor & Destructor Documentation

27.358.2.1 `gdcm::Value::Value ()` `[inline]`

27.358.2.2 `gdcm::Value::~~Value ()` `[inline]`

27.358.3 Member Function Documentation

27.358.3.1 `virtual void gdcm::Value::Clear ()` `[pure virtual]`

Implemented in [gdcm::ByteValue](#), [gdcm::SequenceOfItems](#), and [gdcm::SequenceOfFragments](#).

27.358.3.2 `virtual VL gdcm::Value::GetLength () const` `[pure virtual]`

Implemented in [gdcm::ByteValue](#), [gdcm::SequenceOfItems](#), and [gdcm::SequenceOfFragments](#).

Referenced by `gdcm::DataSet::InsertDataElement()`, and `gdcm::DataElement::SetValue()`.

27.358.3.3 `virtual bool gdcm::Value::operator== (const Value & val) const` `[pure virtual]`

Implemented in [gdcm::SequenceOfFragments](#), [gdcm::SequenceOfItems](#), and [gdcm::ByteValue](#).

27.358.3.4 virtual void `gdcm::Value::SetLength (VL /)` [pure virtual]

Implemented in [gdcm::ByteValue](#), [gdcm::SequenceOfItems](#), and [gdcm::SequenceOfFragments](#).

27.358.3.5 virtual void `gdcm::Value::SetLengthOnly (VL /)` [protected],[virtual]

Reimplemented in [gdcm::ByteValue](#).

27.358.4 Friends And Related Function Documentation

27.358.4.1 friend class `DataElement` [friend]

The documentation for this class was generated from the following file:

- [gdcmValue.h](#)

27.359 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)

27.359.1 Detailed Description

```
template<typename TDE, typename TSwap, typename TType = uint8_t> class gdcm::ValueIO< TDE, TSwap, TType >
```

Class to dispatch template calls.

27.359.2 Member Function Documentation

27.359.2.1 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]

27.359.2.2 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](#)

27.360 gdcm::Version Class Reference

major/minor and build version

```
#include <gdcmVersion.h>
```

Public Member Functions

- [Version](#) ()
- [~Version](#) ()
- 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)

27.360.1 Detailed Description

major/minor and build version

27.360.2 Constructor & Destructor Documentation

27.360.2.1 `gdcm::Version::Version ()` [inline]

27.360.2.2 `gdcm::Version::~~Version ()` [inline]

27.360.3 Member Function Documentation

27.360.3.1 `static int gdcm::Version::GetBuildVersion ()` [static]

27.360.3.2 `static int gdcm::Version::GetMajorVersion ()` [static]

27.360.3.3 `static int gdcm::Version::GetMinorVersion ()` [static]

27.360.3.4 `static const char* gdcm::Version::GetVersion ()` [static]

27.360.3.5 `void gdcm::Version::Print (std::ostream & os = std::cout) const`

Referenced by `gdcm::operator<<()`.

27.360.4 Friends And Related Function Documentation

27.360.4.1 `std::ostream& operator<< (std::ostream & _os, const Version & v)` [*friend*]

The documentation for this class was generated from the following file:

- [gdcVersion.h](#)

27.361 gdc::VL Class Reference

Value Length.

```
#include <gdcVL.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)

27.361.1 Detailed Description

Value Length.

Warning

this is a 4bytes value ! Do not try to use it for 2bytes value length

Examples:

[rle2img.cxx](#).

27.361.2 Member Typedef Documentation

27.361.2.1 `typedef uint32_t gdcm::VL::Type`

27.361.3 Constructor & Destructor Documentation

27.361.3.1 `gdcm::VL::VL (uint32_t v/ = 0) [inline]`

27.361.4 Member Function Documentation

27.361.4.1 `VL gdcm::VL::GetLength () const [inline]`

Referenced by `gdcm::FileMetaInformation::GetFullLength()`, and `gdcm::Item::Write()`.

27.361.4.2 `static uint16_t gdcm::VL::GetVL16Max () [inline], [static]`

27.361.4.3 `static uint32_t gdcm::VL::GetVL32Max () [inline], [static]`

27.361.4.4 `bool gdcm::VL::IsOdd () const [inline]`

Return whether or not the [VL](#) is odd or not.

27.361.4.5 `bool gdcm::VL::IsUndefined () const [inline]`

27.361.4.6 `gdcm::VL::operator uint32_t () const [inline]`

27.361.4.7 `VL& gdcm::VL::operator++ () [inline]`

27.361.4.8 `VL gdcm::VL::operator++ (int) [inline]`

27.361.4.9 `VL& gdcm::VL::operator+= (VL const & v/) [inline]`

`+=` operator

27.361.4.10 `template<typename TSwap> std::istream& gdcm::VL::Read (std::istream & is) [inline]`

27.361.4.11 `template<typename TSwap> std::istream& gdcm::VL::Read16 (std::istream & is) [inline]`

27.361.4.12 `void gdcm::VL::SetToUndefined () [inline]`

27.361.4.13 `template<typename TSwap > const std::ostream& gdcm::VL::Write (std::ostream & os) const [inline]`

Referenced by `gdcm::Fragment::Write()`, `gdcm::SequenceOfItems::Write()`, `gdcm::Item::Write()`, and `gdcm::SequenceOfFragments::Write()`.

27.361.4.14 `template<typename TSwap > const std::ostream& gdcm::VL::Write16 (std::ostream & os) const [inline]`

27.361.5 Friends And Related Function Documentation

27.361.5.1 `std::ostream& operator<< (std::ostream & os, const VL & vl) [friend]`

The documentation for this class was generated from the following file:

- [gdcmVL.h](#)

27.362 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,
 - [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 unsigned int [GetNumberOfElementsFromArray](#) (const char *array, unsigned int length)

- static const char * [GetVMString](#) ([VMType](#) vm)
- static [VMType](#) [GetVMType](#) (const char *vm)
- static [VMType](#) [GetVMTypeFromLength](#) (unsigned int 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)

27.362.1 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

27.362.2 Member Enumeration Documentation

27.362.2.1 enum `gdcm::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
VM7_7n
VM30_30n
VM47_47n
VM_END

27.362.3 Constructor & Destructor Documentation

27.362.3.1 `gdcm::VM::VM (VMType type = VM0) [inline]`

27.362.4 Member Function Documentation

27.362.4.1 `bool gdcm::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.

27.362.4.2 `static unsigned int gdcm::VM::GetIndex (VMType vm) [static], [protected]`

27.362.4.3 `unsigned int gdcm::VM::GetLength () const`

27.362.4.4 `static unsigned int gdcm::VM::GetNumberOfElementsFromArray (const char * array, unsigned int length) [static]`

27.362.4.5 `static const char* gdcm::VM::GetVMString (VMType vm) [static]`

Return the string as written in the official DICOM dict from a custom enum type

Referenced by `gdcm::operator<<()`.

27.362.4.6 static VMType gdcM::VM::GetVMType (const char * *vm*) [static]

27.362.4.7 static VMType gdcM::VM::GetVMTypeFromLength (unsigned int *length*, unsigned int *size*) [static]

27.362.4.8 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

27.362.4.9 gdcM::VM::operator VMType () const [inline]

27.362.5 Friends And Related Function Documentation

27.362.5.1 std::ostream& operator<< (std::ostream & *os*, const VM & *vm*) [friend]

The documentation for this class was generated from the following file:

- [gdcMVM.h](#)

27.363 gdcM::VMToLength< T > Struct Template Reference

```
#include <gdcMVM.h>
```

The documentation for this struct was generated from the following file:

- [gdcMVM.h](#)

27.364 gdcM::VR Class Reference

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.

```
#include <gdcMVR.h>
```


Public Types

- enum [VRType](#) {
 - [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,
 - [OW](#) = 16384,
 - [PN](#) = 32768,
 - [SH](#) = 65536,
 - [SL](#) = 131072,
 - [SQ](#) = 262144,
 - [SS](#) = 524288,
 - [ST](#) = 1048576,
 - [TM](#) = 2097152,
 - [UI](#) = 4194304,
 - [UL](#) = 8388608,
 - [UN](#) = 16777216,
 - [US](#) = 33554432,
 - [UT](#) = 67108864,
 - [OB_OW](#) = OB | OW,
 - [US_SS](#) = US | SS,
 - [US_SS_OW](#) = US | SS | 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 | SQ | UN | UT,
 - [VRASCII](#) = AE | AS | CS | DA | DS | DT | IS | LO | LT | PN | SH | ST | TM | UI | UT,
 - [VRBINARY](#) = AT | FL | FD | OB | OD | OF | OW | SL | SQ | SS | UL | UN | US,
 - [VR_VM1](#) = AS | LT | ST | UT | SQ | OF | OD | OW | OB | UN,
 - [VRALL](#) = VRASCII | VRBINARY,
 - [VR_END](#) = UT+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)`

27.364.1 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`.

27.364.2 Member Enumeration Documentation

27.364.2.1 enum `gdcm::VR::VRType`

Enumerator

INVALID
AE
AS
AT
CS

DA
DS
DT
FD
FL
IS
LO
LT
OB
OD
OF
OW
PN
SH
SL
SQ
SS
ST
TM
UI
UL
UN
US
UT
OB_OW
US_SS
US_SS_OW
VL16
VL32
VRASCII
VRBINARY
VR_VM1
VRALL
VR_END

Examples:

[NewSequence.cs](#).

27.364.3 Constructor & Destructor Documentation

27.364.3.1 `gdcm::VR::VR (VRType vr = INVALID) [inline]`

27.364.4 Member Function Documentation

27.364.4.1 `static bool gdcm::VR::CanDisplay (VRType vr) [static]`

27.364.4.2 `bool gdcm::VR::Compatible (VR const & vr) const`

27.364.4.3 `int gdcm::VR::GetLength () const [inline]`

27.364.4.4 `static uint32_t gdcm::VR::GetLength (VRType vr) [inline],[static]`

27.364.4.5 `unsigned int gdcm::VR::GetSize () const [inline]`

References AE, US_SS, and VRTypeTemplateCase.

27.364.4.6 `unsigned int gdcm::VR::GetSizeof () const`

27.364.4.7 `static const char* gdcm::VR::GetVRString (VRType vr) [static]`

Referenced by `gdcm::operator<<()`.

27.364.4.8 `static const char* gdcm::VR::GetVRStringFromFile (VRType vr) [static]`

27.364.4.9 `static VRType gdcm::VR::GetVRType (const char * vr) [static]`

27.364.4.10 `static VRType gdcm::VR::GetVRTypeFromFile (const char * vr) [static]`

27.364.4.11 `static bool gdcm::VR::IsASCII (VRType vr) [static]`

27.364.4.12 `static bool gdcm::VR::IsASCII2 (VRType vr) [static]`

27.364.4.13 `static bool gdcm::VR::IsBinary (VRType vr) [static]`

27.364.4.14 `static bool gdcm::VR::IsBinary2 (VRType vr) [static]`

27.364.4.15 `bool gdcm::VR::IsDual () const`

27.364.4.16 `static bool gdcm::VR::IsSwap (const char * vr) [static]`

27.364.4.17 `static bool gdcm::VR::IsValid (const char * vr) [static]`

27.364.4.18 `static bool gdcm::VR::IsValid (const char * vr1, VRType vr2) [static]`

27.364.4.19 `bool gdcm::VR::IsVRFile () const`

Referenced by `gdcm::DataElement::SetVR()`.

27.364.4.20 `gdcm::VR::operator VRType () const` `[inline]`

27.364.4.21 `std::istream& gdcm::VR::Read (std::istream & is)` `[inline]`

References `gdcmDebugMacro`, `INVALID`, and `VR_END`.

27.364.4.22 `const std::ostream& gdcm::VR::Write (std::ostream & os) const` `[inline]`

References `gdcmAssertAlwaysMacro`, and `INVALID`.

27.364.5 Friends And Related Function Documentation

27.364.5.1 `std::ostream& operator<< (std::ostream & os, const VR & vr)` `[friend]`

The documentation for this class was generated from the following file:

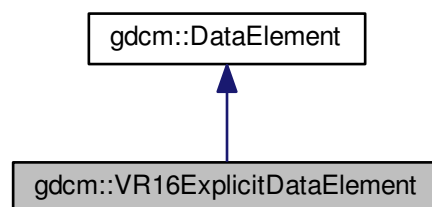
- [gdcmVR.h](#)

27.365 gdcm::VR16ExplicitDataElement Class Reference

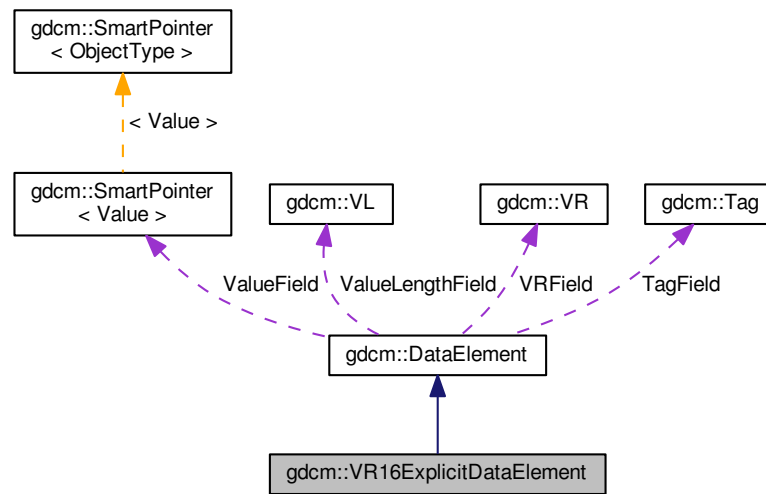
Class to read/write a [DataElement](#) as Explicit Data [Element](#).

```
#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

27.365.1 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

27.365.2 Member Function Documentation

27.365.2.1 VL `gdcm::VR16ExplicitDataElement::GetLength` () const

27.365.2.2 `template<typename TSwap > std::istream& gdcm::VR16ExplicitDataElement::Read (std::istream & is)`

27.365.2.3 `template<typename TSwap > std::istream& gdcm::VR16ExplicitDataElement::ReadPreValue (std::istream & is)`

27.365.2.4 `template<typename TSwap > std::istream& gdcm::VR16ExplicitDataElement::ReadValue (std::istream & is, bool readvalues = true)`

27.365.2.5 `template<typename TSwap > std::istream& gdcm::VR16ExplicitDataElement::ReadWithLength (std::istream & is, VL & length)`

The documentation for this class was generated from the following file:

- [gdcmVR16ExplicitDataElement.h](#)

27.366 `gdcm::VRToEncoding< T >` Struct Template Reference

```
#include <gdcmVR.h>
```

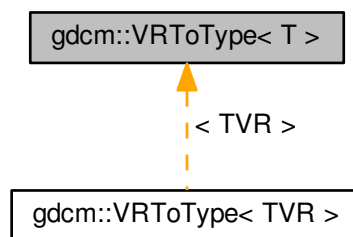
The documentation for this struct was generated from the following file:

- [gdcmVR.h](#)

27.367 `gdcm::VRToType< T >` Struct Template Reference

```
#include <gdcmVR.h>
```

Inheritance diagram for `gdcm::VRToType< T >`:



27.367.1 Detailed Description

```
template<int T>struct gdcm::VRToType< T >
```

Examples:

[DumpGEMSMovieGroup.cxx.](#)

The documentation for this struct was generated from the following file:

- [gdcmVR.h](#)

27.368 `gdcm::VRVLSize< T >` Class Template Reference

```
#include <gdcmAttribute.h>
```

The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

27.369 `gdcm::VRVLSize< 0 >` Class Template Reference

```
#include <gdcmAttribute.h>
```

Static Public Member Functions

- static uint16_t [Read](#) (std::istream &_is)
- static void [Write](#) (std::ostream &os)

27.369.1 Member Function Documentation

27.369.1.1 static uint16_t `gdcm::VRVLSize< 0 >::Read (std::istream &_is)` [inline], [static]

27.369.1.2 static void `gdcm::VRVLSize< 0 >::Write (std::ostream &os)` [inline], [static]

The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

27.370 `gdcm::VRVLSize< 1 >` Class Template Reference

```
#include <gdcmAttribute.h>
```

Static Public Member Functions

- static uint32_t [Read](#) (std::istream &_is)
- static void [Write](#) (std::ostream &os)

27.370.1 Member Function Documentation

27.370.1.1 static uint32_t `gdcm::VRVLSize< 1 >::Read (std::istream &_is)` [inline], [static]

27.370.1.2 static void `gdcm::VRVLSize< 1 >::Write (std::ostream &os)` [inline], [static]

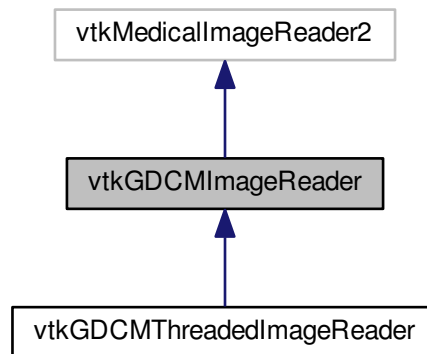
The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

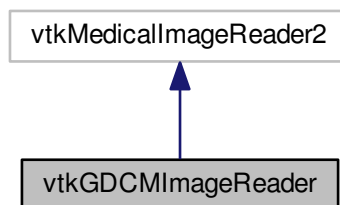
27.371 vtkGDCMImageReader Class Reference

```
#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](#) (LoadOverlays, int)
- [vtkBooleanMacro](#) (LoadIconImage, int)
- [vtkBooleanMacro](#) (LossyFlag, int)
- [vtkBooleanMacro](#) (ApplyLookupTable, int)
- int [vtkBooleanMacro](#) (ApplyYBRToRGB, int)
- [vtkGetMacro](#) (LoadOverlays, int)
- [vtkGetMacro](#) (LoadIconImage, int)
- [vtkGetMacro](#) (LossyFlag, int)
- [vtkGetMacro](#) (NumberOfOverlays, int)
- [vtkGetMacro](#) (NumberOfIconImages, int)
- [vtkGetMacro](#) (ApplyLookupTable, int)
- [vtkGetMacro](#) (ApplyYBRToRGB, int) [vtkSetMacro](#) (ApplyYBRToRGB, int)
- [vtkGetMacro](#) (ImageFormat, int)
- [vtkGetMacro](#) (PlanarConfiguration, int)
- [vtkGetMacro](#) (Shift, double)
- [vtkGetMacro](#) (Scale, double)
- [vtkGetObjectMacro](#) (DirectionCosines, vtkMatrix4x4)
- [vtkGetObjectMacro](#) (MedicalImageProperties, vtkMedicalImageProperties)
- [vtkGetObjectMacro](#) (FileNames, vtkStringArray)
- [vtkGetObjectMacro](#) (Curve, vtkPolyData)
- [vtkGetVector3Macro](#) (ImagePositionPatient, double)
- [vtkGetVector6Macro](#) (ImageOrientationPatient, double)
- [vtkSetMacro](#) (LoadOverlays, int)
- [vtkSetMacro](#) (LoadIconImage, int)
- [vtkSetMacro](#) (LossyFlag, int)
- [vtkSetMacro](#) (ApplyLookupTable, int)
- [vtkTypeRevisionMacro](#) (vtkGDCMImageReader, vtkMedicalImageReader2)

Static Public Member Functions

- static [vtkGDCMImageReader * New](#) ()

Protected Member Functions

- [vtkGDCMImageReader](#) ()
- [~vtkGDCMImageReader](#) ()
- void [ExecuteData](#) (vtkDataObject *out)
- void [ExecuteInformation](#) ()
- void [FillMedicalImageInformation](#) (const [gdcm::ImageReader](#) &reader)
- int [LoadSingleFile](#) (const char *filename, char *pointer, unsigned long &outlen)
- int [RequestDataCompat](#) ()
- int [RequestInformationCompat](#) ()
- void [SetFilePattern](#) (const char *)
- void [SetFilePrefix](#) (const char *)
- [vtkGetStringMacro](#) (FilePrefix)
- [vtkGetStringMacro](#) (FilePattern)
- [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](#)

27.371.1 Detailed Description

Examples:

[AWTMedical3.java](#), [Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [gdcmmorthoplanes.cxx](#), [gdcmmreslice.cxx](#), [gdcmmtexture.cxx](#), [gdcmmvolume.cxx](#), [HelloActiviz.cs](#), [HelloActiviz2.cs](#), [HelloActiviz3.cs](#), [HelloActiviz4.cs](#), [HelloActiviz5.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld.java](#), [MagnifyFile.cxx](#), [MetaImageMD5Activiz.cs](#), [MIPViewer.java](#), [MPRViewer.java](#), [MPRViewer2.java](#), [offscreenimage.cxx](#), [ReadSeriesIntoVTK.java](#), [RefCounting.cs](#), and [reslicesphere.cxx](#).

27.371.2 Constructor & Destructor Documentation

27.371.2.1 `vtkGDCMImageReader::vtkGDCMImageReader ()` [protected]

Examples:

[HelloActiviz2.cs](#).

27.371.2.2 `vtkGDCMImageReader::~~vtkGDCMImageReader ()` [protected]

27.371.3 Member Function Documentation

27.371.3.1 `virtual int vtkGDCMImageReader::CanReadFile (const char * fname)` [virtual]

Examples:

[MetalImageMD5Activiz.cs](#).

27.371.3.2 `void vtkGDCMImageReader::ExecuteData (vtkDataObject * out)` [protected]

27.371.3.3 `void vtkGDCMImageReader::ExecuteInformation ()` [protected]

27.371.3.4 `void vtkGDCMImageReader::FillMedicalImageInformation (const gdcm::ImageReader & reader)`
[protected]

27.371.3.5 `virtual const char* vtkGDCMImageReader::GetDescriptiveName ()` [inline],[virtual]

27.371.3.6 `virtual const char* vtkGDCMImageReader::GetFileExtensions ()` [inline],[virtual]

27.371.3.7 `vtkImageData* vtkGDCMImageReader::GetIconImage ()`

27.371.3.8 `vtkImageData* vtkGDCMImageReader::GetOverlay (int i)`

27.371.3.9 `int vtkGDCMImageReader::LoadSingleFile (const char * filename, char * pointer, unsigned long & outlen)`
[protected]

27.371.3.10 `static vtkGDCMImageReader* vtkGDCMImageReader::New ()` [static]

Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [gdcmmorphoplanes.cxx](#), [gdcmreslice.cxx](#), [gdcmttexture.cxx](#), [gdcmvolume.cxx](#), [HelloActiviz.cs](#), [HelloActiviz3.cs](#), [HelloActiviz4.cs](#), [HelloActiviz5.cs](#), [HelloVTKWorld.cs](#), [MagnifyFile.cxx](#), [MetalImageMD5Activiz.cs](#), [offscreenimage.cxx](#), [RefCounting.cs](#), and [reslicesphere.cxx](#).

27.371.3.11 `virtual void vtkGDCMImageReader::PrintSelf (ostream & os, vtkIndent indent)` [virtual]

Reimplemented in [vtkGDCMThreadedImageReader](#).

27.371.3.12 `int vtkGDCMImageReader::RequestDataCompat ()` [protected]

27.371.3.13 `int vtkGDCMImageReader::RequestInformationCompat ()` [protected]

27.371.3.14 `virtual void vtkGDCMImageReader::SetCurve (vtkPolyData * pd)` [virtual]

27.371.3.15 `virtual void vtkGDCMImageReader::SetFileNames (vtkStringArray *)` [virtual]

Examples:

[gdcmortoplanes.cxx](#), [HelloActiviz3.cs](#), [HelloActiviz4.cs](#), [HelloActiviz5.cs](#), [MIPViewer.java](#), [MPRViewer.java](#), [MPRViewer2.java](#), and [ReadSeriesIntoVTK.java](#).

27.371.3.16 `void vtkGDCMImageReader::SetFilePattern (const char *)` [inline],[protected]

27.371.3.17 `void vtkGDCMImageReader::SetFilePrefix (const char *)` [inline],[protected]

27.371.3.18 `virtual void vtkGDCMImageReader::SetMedicalImageProperties (vtkMedicalImageProperties * pd)` [virtual]

27.371.3.19 `vtkGDCMImageReader::vtkBooleanMacro (LoadOverlays , int)`

27.371.3.20 `vtkGDCMImageReader::vtkBooleanMacro (LoadIconImage , int)`

27.371.3.21 `vtkGDCMImageReader::vtkBooleanMacro (LossyFlag , int)`

27.371.3.22 `vtkGDCMImageReader::vtkBooleanMacro (ApplyLookupTable , int)`

27.371.3.23 `int vtkGDCMImageReader::vtkBooleanMacro (ApplyYBRToRGB , int)`

27.371.3.24 `vtkGDCMImageReader::vtkGetMacro (LoadOverlays , int)`

27.371.3.25 `vtkGDCMImageReader::vtkGetMacro (LoadIconImage , int)`

27.371.3.26 `vtkGDCMImageReader::vtkGetMacro (LossyFlag , int)`

27.371.3.27 `vtkGDCMImageReader::vtkGetMacro (NumberOfOverlays , int)`

27.371.3.28 `vtkGDCMImageReader::vtkGetMacro (NumberOfIconImages , int)`

27.371.3.29 `vtkGDCMImageReader::vtkGetMacro (ApplyLookupTable , int)`

27.371.3.30 `vtkGDCMImageReader::vtkGetMacro (ApplyYBRToRGB , int)`

27.371.3.31 `vtkGDCMImageReader::vtkGetMacro (ImageFormat , int)`

27.371.3.32 `vtkGDCMImageReader::vtkGetMacro (PlanarConfiguration , int)`

27.371.3.33 `vtkGDCMImageReader::vtkGetMacro (Shift , double)`

27.371.3.34 `vtkGDCMImageReader::vtkGetMacro (Scale , double)`

27.371.3.35 `vtkGDCMImageReader::vtkGetObjectMacro (DirectionCosines , vtkMatrix4x4)`

27.371.3.36 `vtkGDCMImageReader::vtkGetObjectMacro (MedicalImageProperties , vtkMedicalImageProperties)`

27.371.3.37 `vtkGDCMImageReader::vtkGetObjectMacro (FileNames , vtkStringArray)`

- 27.371.3.38 `vtkGDCMImageReader::vtkGetObjectMacro (Curve , vtkPolyData)`
- 27.371.3.39 `vtkGDCMImageReader::vtkGetStringMacro (FilePrefix)` [protected]
- 27.371.3.40 `vtkGDCMImageReader::vtkGetStringMacro (FilePattern)` [protected]
- 27.371.3.41 `vtkGDCMImageReader::vtkGetVector3Macro (ImagePositionPatient , double)`
- 27.371.3.42 `vtkGDCMImageReader::vtkGetVector6Macro (ImageOrientationPatient , double)`
- 27.371.3.43 `vtkGDCMImageReader::vtkSetMacro (LoadOverlays , int)`
- 27.371.3.44 `vtkGDCMImageReader::vtkSetMacro (LoadIconImage , int)`
- 27.371.3.45 `vtkGDCMImageReader::vtkSetMacro (LossyFlag , int)`
- 27.371.3.46 `vtkGDCMImageReader::vtkSetMacro (ApplyLookupTable , int)`
- 27.371.3.47 `vtkGDCMImageReader::vtkSetVector6Macro (ImageOrientationPatient , double)` [protected]
- 27.371.3.48 `vtkGDCMImageReader::vtkTypeRevisionMacro (vtkGDCMImageReader , vtkMedicalImageReader2)`

27.371.4 Member Data Documentation

- 27.371.4.1 `int vtkGDCMImageReader::ApplyInverseVideo` [protected]
- 27.371.4.2 `int vtkGDCMImageReader::ApplyLookupTable` [protected]
- 27.371.4.3 `int vtkGDCMImageReader::ApplyPlanarConfiguration` [protected]
- 27.371.4.4 `int vtkGDCMImageReader::ApplyShiftScale` [protected]
- 27.371.4.5 `int vtkGDCMImageReader::ApplyYBRToRGB` [protected]
- 27.371.4.6 `vtkPolyData* vtkGDCMImageReader::Curve` [protected]
- 27.371.4.7 `vtkMatrix4x4* vtkGDCMImageReader::DirectionCosines` [protected]
- 27.371.4.8 `vtkStringArray* vtkGDCMImageReader::FileNames` [protected]
- 27.371.4.9 `int vtkGDCMImageReader::ForceRescale` [protected]
- 27.371.4.10 `int vtkGDCMImageReader::IconDataScalarType` [protected]
- 27.371.4.11 `int vtkGDCMImageReader::IconImageDataExtent[6]` [protected]
- 27.371.4.12 `int vtkGDCMImageReader::IconNumberOfScalarComponents` [protected]
- 27.371.4.13 `int vtkGDCMImageReader::ImageFormat` [protected]
- 27.371.4.14 `double vtkGDCMImageReader::ImageOrientationPatient[6]` [protected]

- 27.371.4.15 `double vtkGDCMImageReader::ImagePositionPatient[3]` [protected]
- 27.371.4.16 `int vtkGDCMImageReader::LoadIconImage` [protected]
- 27.371.4.17 `int vtkGDCMImageReader::LoadOverlays` [protected]
- 27.371.4.18 `int vtkGDCMImageReader::LossyFlag` [protected]
- 27.371.4.19 `vtkMedicalImageProperties* vtkGDCMImageReader::MedicalImageProperties` [protected]
- 27.371.4.20 `int vtkGDCMImageReader::NumberOfIconImages` [protected]
- 27.371.4.21 `int vtkGDCMImageReader::NumberOfOverlays` [protected]
- 27.371.4.22 `int vtkGDCMImageReader::PlanarConfiguration` [protected]
- 27.371.4.23 `double vtkGDCMImageReader::Scale` [protected]
- 27.371.4.24 `double vtkGDCMImageReader::Shift` [protected]

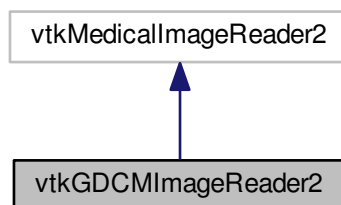
The documentation for this class was generated from the following file:

- [vtkGDCMImageReader.h](#)

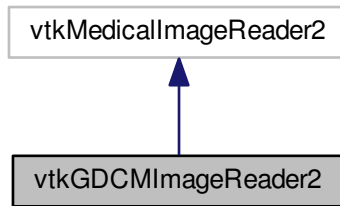
27.372 vtkGDCMImageReader2 Class Reference

```
#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](#) (LoadOverlays, int)
- [vtkBooleanMacro](#) (LoadIconImage, int)
- [vtkBooleanMacro](#) (LossyFlag, int)
- [vtkBooleanMacro](#) (ApplyLookupTable, int)
- int [vtkBooleanMacro](#) (ApplyYBRTToRGB, int)
- [vtkGetMacro](#) (LoadOverlays, int)
- [vtkGetMacro](#) (LoadIconImage, int)
- [vtkGetMacro](#) (LossyFlag, int)
- [vtkGetMacro](#) (NumberOfOverlays, int)
- [vtkGetMacro](#) (NumberOfIconImages, int)
- [vtkGetMacro](#) (ApplyLookupTable, int)
- [vtkGetMacro](#) (ApplyYBRTToRGB, int) [vtkSetMacro](#) (ApplyYBRTToRGB, int)
- [vtkGetMacro](#) (ImageFormat, int)
- [vtkGetMacro](#) (PlanarConfiguration, int)
- [vtkGetMacro](#) (Shift, double)
- [vtkGetMacro](#) (Scale, double)
- [vtkGetObjectMacro](#) (DirectionCosines, vtkMatrix4x4)
- [vtkGetObjectMacro](#) (Curve, vtkPolyData)
- [vtkGetVector3Macro](#) (ImagePositionPatient, double)
- [vtkGetVector6Macro](#) (ImageOrientationPatient, double)
- [vtkSetMacro](#) (LoadOverlays, int)
- [vtkSetMacro](#) (LoadIconImage, int)

- [vtkSetMacro](#) ([LossyFlag](#), int)
- [vtkSetMacro](#) ([ApplyLookupTable](#), int)
- [vtkTypeRevisionMacro](#) ([vtkGDCMImageReader2](#), [vtkMedicalImageReader2](#))

Static Public Member Functions

- static [vtkGDCMImageReader2](#) * [New](#) ()

Protected Member Functions

- [vtkGDCMImageReader2](#) ()
- [~vtkGDCMImageReader2](#) ()
- void [FillMedicalImageInformation](#) (const [gdcm::ImageReader](#) &reader)
- int [LoadSingleFile](#) (const char *filename, char *pointer, unsigned long &outlen)
- int [ProcessRequest](#) ([vtkInformation](#) *request, [vtkInformationVector](#) **inputVector, [vtkInformationVector](#) *outputVector)
- int [RequestData](#) ([vtkInformation](#) *request, [vtkInformationVector](#) **inputVector, [vtkInformationVector](#) *outputVector)
- int [RequestDataCompat](#) ()
- int [RequestInformation](#) ([vtkInformation](#) *request, [vtkInformationVector](#) **inputVector, [vtkInformationVector](#) *outputVector)
- int [RequestInformationCompat](#) ()
- void [SetFilePattern](#) (const char *)
- void [SetFilePrefix](#) (const char *)
- [vtkGetStringMacro](#) ([FilePrefix](#))
- [vtkGetStringMacro](#) ([FilePattern](#))
- [vtkSetVector6Macro](#) ([ImageOrientationPatient](#), double)

Protected Attributes

- int [ApplyInverseVideo](#)
- int [ApplyLookupTable](#)
- int [ApplyPlanarConfiguration](#)
- int [ApplyShiftScale](#)
- int [ApplyYBRToRGB](#)
- [vtkPolyData](#) * [Curve](#)
- [vtkMatrix4x4](#) * [DirectionCosines](#)
- int [ForceRescale](#)
- int [IconDataScalarType](#)
- int [IconImageDataExtent](#) [6]
- int [IconNumberOfScalarComponents](#)
- int [ImageFormat](#)
- double [ImageOrientationPatient](#) [6]
- double [ImagePositionPatient](#) [3]
- int [LoadIconImage](#)
- int [LoadOverlays](#)
- int [LossyFlag](#)
- int [NumberOfIconImages](#)
- int [NumberOfOverlays](#)
- int [PlanarConfiguration](#)
- double [Scale](#)
- double [Shift](#)

27.372.1 Detailed Description

Examples:

[Compute3DSpacing.cxx](#).

27.372.2 Constructor & Destructor Documentation

27.372.2.1 `vtkGDCMImageReader2::vtkGDCMImageReader2 ()` [protected]

27.372.2.2 `vtkGDCMImageReader2::~~vtkGDCMImageReader2 ()` [protected]

27.372.3 Member Function Documentation

27.372.3.1 `virtual int vtkGDCMImageReader2::CanReadFile (const char * fname)` [virtual]

27.372.3.2 `void vtkGDCMImageReader2::FillMedicalImageInformation (const gdcm::ImageReader & reader)`
[protected]

27.372.3.3 `virtual const char* vtkGDCMImageReader2::GetDescriptiveName ()` [inline],[virtual]

27.372.3.4 `virtual const char* vtkGDCMImageReader2::GetFileExtensions ()` [inline],[virtual]

27.372.3.5 `vtkImageData* vtkGDCMImageReader2::GetIconImage ()`

27.372.3.6 `vtkAlgorithmOutput* vtkGDCMImageReader2::GetIconImagePort ()`

27.372.3.7 `vtkImageData* vtkGDCMImageReader2::GetOverlay (int i)`

27.372.3.8 `vtkAlgorithmOutput* vtkGDCMImageReader2::GetOverlayPort (int index)`

27.372.3.9 `int vtkGDCMImageReader2::LoadSingleFile (const char * filename, char * pointer, unsigned long & outlen)`
[protected]

27.372.3.10 `static vtkGDCMImageReader2* vtkGDCMImageReader2::New ()` [static]

Examples:

[Compute3DSpacing.cxx](#).

27.372.3.11 `virtual void vtkGDCMImageReader2::PrintSelf (ostream & os, vtkIndent indent)` [virtual]

27.372.3.12 `int vtkGDCMImageReader2::ProcessRequest (vtkInformation * request, vtkInformationVector ** inputVector,
vtkInformationVector * outputVector)` [protected]

27.372.3.13 `int vtkGDCMImageReader2::RequestData (vtkInformation * request, vtkInformationVector ** inputVector,
vtkInformationVector * outputVector)` [protected]

27.372.3.14 `int vtkGDCMImageReader2::RequestDataCompat ()` [protected]

27.372.3.15 `int vtkGDCMImageReader2::RequestInformation (vtkInformation * request, vtkInformationVector ** inputVector,
vtkInformationVector * outputVector)` [protected]

- 27.372.3.16 int vtkGDCMImageReader2::RequestInformationCompat () [protected]
- 27.372.3.17 virtual void vtkGDCMImageReader2::SetCurve (vtkPolyData * *pd*) [virtual]
- 27.372.3.18 void vtkGDCMImageReader2::SetFilePattern (const char *) [inline],[protected]
- 27.372.3.19 void vtkGDCMImageReader2::SetFilePrefix (const char *) [inline],[protected]
- 27.372.3.20 virtual void vtkGDCMImageReader2::SetMedicalImageProperties (vtkMedicalImageProperties * *pd*) [virtual]
- 27.372.3.21 vtkGDCMImageReader2::vtkBooleanMacro (LoadOverlays , int)
- 27.372.3.22 vtkGDCMImageReader2::vtkBooleanMacro (LoadIconImage , int)
- 27.372.3.23 vtkGDCMImageReader2::vtkBooleanMacro (LossyFlag , int)
- 27.372.3.24 vtkGDCMImageReader2::vtkBooleanMacro (ApplyLookupTable , int)
- 27.372.3.25 int vtkGDCMImageReader2::vtkBooleanMacro (ApplyYBRToRGB , int)
- 27.372.3.26 vtkGDCMImageReader2::vtkGetMacro (LoadOverlays , int)
- 27.372.3.27 vtkGDCMImageReader2::vtkGetMacro (LoadIconImage , int)
- 27.372.3.28 vtkGDCMImageReader2::vtkGetMacro (LossyFlag , int)
- 27.372.3.29 vtkGDCMImageReader2::vtkGetMacro (NumberOfOverlays , int)
- 27.372.3.30 vtkGDCMImageReader2::vtkGetMacro (NumberOfIconImages , int)
- 27.372.3.31 vtkGDCMImageReader2::vtkGetMacro (ApplyLookupTable , int)
- 27.372.3.32 vtkGDCMImageReader2::vtkGetMacro (ApplyYBRToRGB , int)
- 27.372.3.33 vtkGDCMImageReader2::vtkGetMacro (ImageFormat , int)
- 27.372.3.34 vtkGDCMImageReader2::vtkGetMacro (PlanarConfiguration , int)
- 27.372.3.35 vtkGDCMImageReader2::vtkGetMacro (Shift , double)
- 27.372.3.36 vtkGDCMImageReader2::vtkGetMacro (Scale , double)
- 27.372.3.37 vtkGDCMImageReader2::vtkGetObjectMacro (DirectionCosines , vtkMatrix4x4)
- 27.372.3.38 vtkGDCMImageReader2::vtkGetObjectMacro (Curve , vtkPolyData)
- 27.372.3.39 vtkGDCMImageReader2::vtkGetStringMacro (FilePrefix) [protected]
- 27.372.3.40 vtkGDCMImageReader2::vtkGetStringMacro (FilePattern) [protected]
- 27.372.3.41 vtkGDCMImageReader2::vtkGetVector3Macro (ImagePositionPatient , double)

- 27.372.3.42 `vtkGDCMImageReader2::vtkGetVector6Macro (ImageOrientationPatient , double)`
- 27.372.3.43 `vtkGDCMImageReader2::vtkSetMacro (LoadOverlays , int)`
- 27.372.3.44 `vtkGDCMImageReader2::vtkSetMacro (LoadIconImage , int)`
- 27.372.3.45 `vtkGDCMImageReader2::vtkSetMacro (LossyFlag , int)`
- 27.372.3.46 `vtkGDCMImageReader2::vtkSetMacro (ApplyLookupTable , int)`
- 27.372.3.47 `vtkGDCMImageReader2::vtkSetVector6Macro (ImageOrientationPatient , double)` [protected]
- 27.372.3.48 `vtkGDCMImageReader2::vtkTypeRevisionMacro (vtkGDCMImageReader2 , vtkMedicalImageReader2)`

27.372.4 Member Data Documentation

- 27.372.4.1 `int vtkGDCMImageReader2::ApplyInverseVideo` [protected]
- 27.372.4.2 `int vtkGDCMImageReader2::ApplyLookupTable` [protected]
- 27.372.4.3 `int vtkGDCMImageReader2::ApplyPlanarConfiguration` [protected]
- 27.372.4.4 `int vtkGDCMImageReader2::ApplyShiftScale` [protected]
- 27.372.4.5 `int vtkGDCMImageReader2::ApplyYBRToRGB` [protected]
- 27.372.4.6 `vtkPolyData* vtkGDCMImageReader2::Curve` [protected]
- 27.372.4.7 `vtkMatrix4x4* vtkGDCMImageReader2::DirectionCosines` [protected]
- 27.372.4.8 `int vtkGDCMImageReader2::ForceRescale` [protected]
- 27.372.4.9 `int vtkGDCMImageReader2::IconDataScalarType` [protected]
- 27.372.4.10 `int vtkGDCMImageReader2::IconImageDataExtent[6]` [protected]
- 27.372.4.11 `int vtkGDCMImageReader2::IconNumberOfScalarComponents` [protected]
- 27.372.4.12 `int vtkGDCMImageReader2::ImageFormat` [protected]
- 27.372.4.13 `double vtkGDCMImageReader2::ImageOrientationPatient[6]` [protected]
- 27.372.4.14 `double vtkGDCMImageReader2::ImagePositionPatient[3]` [protected]
- 27.372.4.15 `int vtkGDCMImageReader2::LoadIconImage` [protected]
- 27.372.4.16 `int vtkGDCMImageReader2::LoadOverlays` [protected]
- 27.372.4.17 `int vtkGDCMImageReader2::LossyFlag` [protected]
- 27.372.4.18 `int vtkGDCMImageReader2::NumberOfIconImages` [protected]

27.372.4.19 int vtkGDCMImageReader2::NumberOfOverlays [protected]

27.372.4.20 int vtkGDCMImageReader2::PlanarConfiguration [protected]

27.372.4.21 double vtkGDCMImageReader2::Scale [protected]

27.372.4.22 double vtkGDCMImageReader2::Shift [protected]

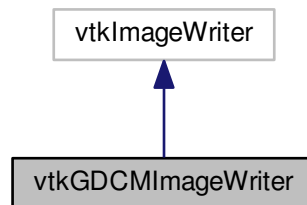
The documentation for this class was generated from the following file:

- [vtkGDCMImageReader2.h](#)

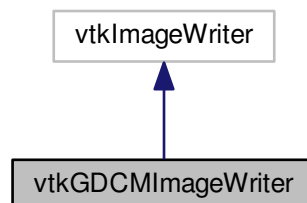
27.373 vtkGDCMImageWriter Class Reference

```
#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](#) (LossyFlag, int)
- [vtkBooleanMacro](#) (FileLowerLeft, int)
- [vtkGetMacro](#) (LossyFlag, int)
- [vtkGetMacro](#) (Shift, double)
- [vtkGetMacro](#) (Scale, double)
- [vtkGetMacro](#) (ImageFormat, int)
- [vtkGetMacro](#) (FileLowerLeft, int)
- [vtkGetMacro](#) (PlanarConfiguration, int)
- [vtkGetMacro](#) (CompressionType, int)
- [vtkGetObjectMacro](#) (MedicalImageProperties, vtkMedicalImageProperties)
- [vtkGetObjectMacro](#) (FileNames, vtkStringArray)
- [vtkGetObjectMacro](#) (DirectionCosines, vtkMatrix4x4)
- [vtkGetStringMacro](#) (StudyUID)
- [vtkGetStringMacro](#) (SeriesUID)
- [vtkSetMacro](#) (LossyFlag, int)
- [vtkSetMacro](#) (Shift, double)
- [vtkSetMacro](#) (Scale, double)
- [vtkSetMacro](#) (ImageFormat, int)
- [vtkSetMacro](#) (FileLowerLeft, int)
- [vtkSetMacro](#) (PlanarConfiguration, int)
- [vtkSetMacro](#) (CompressionType, int)
- [vtkSetStringMacro](#) (StudyUID)
- [vtkSetStringMacro](#) (SeriesUID)
- [vtkTypeRevisionMacro](#) (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)

27.373.1 Detailed Description

Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [CreateFakePET.cxx](#), [CreateFakeRTDOSE.cxx](#), [gdcmorphoplanes.cxx](#), [HelloActiviz.cs](#), [HelloActiviz2.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld.java](#), [HelloVTKWorld2.cs](#), [MagnifyFile.cxx](#), and [RefCounting.cs](#).

27.373.2 Member Enumeration Documentation

27.373.2.1 enum vtkGDCMImageWriter::CompressionTypes

Enumerator

NO_COMPRESSION
JPEG_COMPRESSION
JPEG2000_COMPRESSION
JPEGLS_COMPRESSION
RLE_COMPRESSION

27.373.3 Constructor & Destructor Documentation

27.373.3.1 `vtkGDCMImageWriter::vtkGDCMImageWriter ()` [protected]

27.373.3.2 `vtkGDCMImageWriter::~~vtkGDCMImageWriter ()` [protected]

27.373.4 Member Function Documentation

27.373.4.1 `virtual const char* vtkGDCMImageWriter::GetDescriptiveName ()` [inline],[virtual]

27.373.4.2 `virtual const char* vtkGDCMImageWriter::GetFileExtensions ()` [inline],[virtual]

27.373.4.3 `virtual char* vtkGDCMImageWriter::GetFileName ()` [protected],[virtual]

27.373.4.4 `static vtkGDCMImageWriter* vtkGDCMImageWriter::New ()` [static]

Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [CreateFakePET.cxx](#), [CreateFakeRTDOSE.cxx](#), [gdcmorphoplanes.cxx](#), [HelloActiviz.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld2.cs](#), [MagnifyFile.cxx](#), and [RefCounting.cs](#).

27.373.4.5 virtual void vtkGDCMImageWriter::PrintSelf (ostream & *os*, vtkIndent *indent*) [virtual]

27.373.4.6 virtual void vtkGDCMImageWriter::SetDirectionCosines (vtkMatrix4x4 * *matrix*) [virtual]

Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [gdcmorphoplanes.cxx](#), [HelloActiviz2.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld.java](#), and [MagnifyFile.cxx](#).

27.373.4.7 virtual void vtkGDCMImageWriter::SetDirectionCosinesFromImageOrientationPatient (const double *dircos*[6]) [virtual]

27.373.4.8 virtual void vtkGDCMImageWriter::SetFileNames (vtkStringArray *) [virtual]

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

27.373.4.9 virtual void vtkGDCMImageWriter::SetMedicalImageProperties (vtkMedicalImageProperties *) [virtual]

Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [gdcmorphoplanes.cxx](#), [HelloActiviz.cs](#), [HelloActiviz2.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld.java](#), and [MagnifyFile.cxx](#).

27.373.4.10 vtkGDCMImageWriter::vtkBooleanMacro (LossyFlag , int)

27.373.4.11 vtkGDCMImageWriter::vtkBooleanMacro (FileLowerLeft , int)

27.373.4.12 vtkGDCMImageWriter::vtkGetMacro (LossyFlag , int)

27.373.4.13 vtkGDCMImageWriter::vtkGetMacro (Shift , double)

27.373.4.14 vtkGDCMImageWriter::vtkGetMacro (Scale , double)

27.373.4.15 vtkGDCMImageWriter::vtkGetMacro (ImageFormat , int)

27.373.4.16 vtkGDCMImageWriter::vtkGetMacro (FileLowerLeft , int)

27.373.4.17 vtkGDCMImageWriter::vtkGetMacro (PlanarConfiguration , int)

27.373.4.18 vtkGDCMImageWriter::vtkGetMacro (CompressionType , int)

27.373.4.19 vtkGDCMImageWriter::vtkGetObjectMacro (MedicalImageProperties , vtkMedicalImageProperties)

27.373.4.20 vtkGDCMImageWriter::vtkGetObjectMacro (FileNames , vtkStringArray)

27.373.4.21 vtkGDCMImageWriter::vtkGetObjectMacro (DirectionCosines , vtkMatrix4x4)

27.373.4.22 vtkGDCMImageWriter::vtkGetStringMacro (StudyUID)

- 27.373.4.23 `vtkGDCMImageWriter::vtkGetStringMacro (SeriesUID)`
- 27.373.4.24 `vtkGDCMImageWriter::vtkSetMacro (LossyFlag , int)`
- 27.373.4.25 `vtkGDCMImageWriter::vtkSetMacro (Shift , double)`
- 27.373.4.26 `vtkGDCMImageWriter::vtkSetMacro (Scale , double)`
- 27.373.4.27 `vtkGDCMImageWriter::vtkSetMacro (ImageFormat , int)`
- 27.373.4.28 `vtkGDCMImageWriter::vtkSetMacro (FileLowerLeft , int)`
- 27.373.4.29 `vtkGDCMImageWriter::vtkSetMacro (PlanarConfiguration , int)`
- 27.373.4.30 `vtkGDCMImageWriter::vtkSetMacro (CompressionType , int)`
- 27.373.4.31 `vtkGDCMImageWriter::vtkSetStringMacro (StudyUID)`
- 27.373.4.32 `vtkGDCMImageWriter::vtkSetStringMacro (SeriesUID)`
- 27.373.4.33 `vtkGDCMImageWriter::vtkTypeRevisionMacro (vtkGDCMImageWriter , vtkImageWriter)`
- 27.373.4.34 `virtual void vtkGDCMImageWriter::Write () [virtual]`

Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [CreateFakePET.cxx](#), [CreateFakeRTDOSE.cxx](#), [gdcmmorthoplanes.cxx](#), [HelloActiviz.cs](#), [HelloActiviz2.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld.java](#), [HelloVTKWorld2.cs](#), and [MagnifyFile.cxx](#).

- 27.373.4.35 `int vtkGDCMImageWriter::WriteGDCMData (vtkImageData * data, int timeStep) [protected]`
- 27.373.4.36 `void vtkGDCMImageWriter::WriteSlice (vtkImageData * data) [protected]`

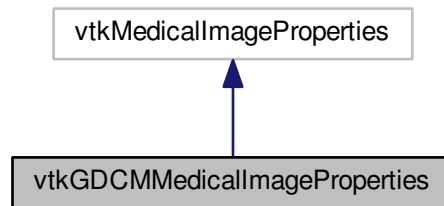
The documentation for this class was generated from the following file:

- [vtkGDCMImageWriter.h](#)

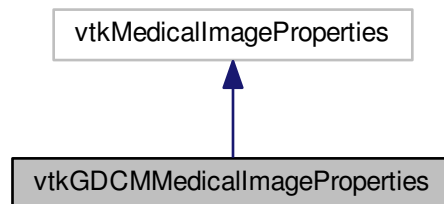
27.374 vtkGDCMMedicalImageProperties Class Reference

```
#include <vtkGDCMMedicalImageProperties.h>
```

Inheritance diagram for `vtkGDCMMedicalImageProperties`:



Collaboration diagram for `vtkGDCMMedicalImageProperties`:



Public Member Functions

- virtual void [Clear](#) ()
- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkTypeRevisionMacro](#) ([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](#)

27.374.1 Constructor & Destructor Documentation

27.374.1.1 `vtkGDCMMedicalImageProperties::vtkGDCMMedicalImageProperties ()` [protected]

27.374.1.2 `vtkGDCMMedicalImageProperties::~~vtkGDCMMedicalImageProperties ()` [protected]

27.374.2 Member Function Documentation

27.374.2.1 `virtual void vtkGDCMMedicalImageProperties::Clear ()` [virtual]

27.374.2.2 `gdcmm::File const& vtkGDCMMedicalImageProperties::GetFile (unsigned int t)` [protected]

27.374.2.3 `static vtkGDCMMedicalImageProperties* vtkGDCMMedicalImageProperties::New ()` [static]

27.374.2.4 `void vtkGDCMMedicalImageProperties::PrintSelf (ostream & os, vtkIndent indent)`

27.374.2.5 `void vtkGDCMMedicalImageProperties::PushBackFile (gdcmm::File const & f)` [protected]

27.374.2.6 `vtkGDCMMedicalImageProperties::vtkTypeRevisionMacro (vtkGDCMMedicalImageProperties ,
vtkMedicalImageProperties)`

27.374.3 Friends And Related Function Documentation

27.374.3.1 `friend class vtkGDCMImageReader` [friend]

27.374.3.2 `friend class vtkGDCMImageReader2` [friend]

27.374.3.3 `friend class vtkGDCMImageWriter` [friend]

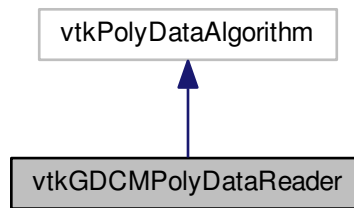
The documentation for this class was generated from the following file:

- [vtkGDCMMedicalImageProperties.h](#)

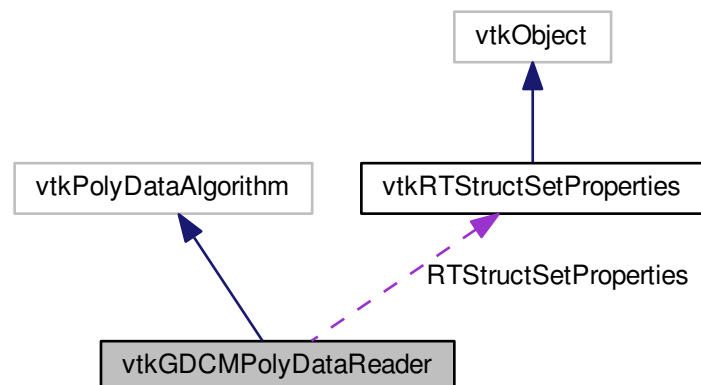
27.375 vtkGDCMPolyDataReader Class Reference

```
#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](#))
- [vtkTypeRevisionMacro](#) ([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](#)

27.375.1 Detailed Description

Examples:

[gdcmscene.cxx](#), [GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

27.375.2 Constructor & Destructor Documentation

27.375.2.1 `vtkGDCMPolyDataReader::vtkGDCMPolyDataReader ()` [protected]

27.375.2.2 `vtkGDCMPolyDataReader::~~vtkGDCMPolyDataReader ()` [protected]

27.375.3 Member Function Documentation

27.375.3.1 `void vtkGDCMPolyDataReader::FillMedicalImageInformation (const gdcmm::Reader & reader)` [protected]

27.375.3.2 `static vtkGDCMPolyDataReader* vtkGDCMPolyDataReader::New ()` [static]

Examples:

[gdcmscene.cxx](#), [GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

27.375.3.3 `virtual void vtkGDCMPolyDataReader::PrintSelf (ostream & os, vtkIndent indent)` [virtual]

27.375.3.4 `int vtkGDCMPolyDataReader::RequestData (vtkInformation *, vtkInformationVector **, vtkInformationVector *)` [protected]

27.375.3.5 `int vtkGDCMPolyDataReader::RequestData_HemodynamicWaveformStorage (gdcmm::Reader const & reader, vtkInformationVector * outputVector)` [protected]

- 27.375.3.6 `int vtkGDCMPolyDataReader::RequestData_RTStructureSetStorage (gdcm::Reader const & reader, vtkInformationVector * outputVector)` `[protected]`
- 27.375.3.7 `int vtkGDCMPolyDataReader::RequestInformation (vtkInformation * vtkNotUsedrequest, vtkInformationVector ** vtkNotUsedinputVector, vtkInformationVector * outputVector)` `[protected]`
- 27.375.3.8 `int vtkGDCMPolyDataReader::RequestInformation_HemodynamicWaveformStorage (gdcm::Reader const & reader)` `[protected]`
- 27.375.3.9 `int vtkGDCMPolyDataReader::RequestInformation_RTStructureSetStorage (gdcm::Reader const & reader)` `[protected]`
- 27.375.3.10 `vtkGDCMPolyDataReader::vtkGetObjectMacro (MedicalImageProperties , vtkMedicalImageProperties)`
- 27.375.3.11 `vtkGDCMPolyDataReader::vtkGetObjectMacro (RTStructSetProperties , vtkRTStructSetProperties)`
- 27.375.3.12 `vtkGDCMPolyDataReader::vtkGetStringMacro (FileName)`
- 27.375.3.13 `vtkGDCMPolyDataReader::vtkSetStringMacro (FileName)`
- 27.375.3.14 `vtkGDCMPolyDataReader::vtkTypeRevisionMacro (vtkGDCMPolyDataReader , vtkPolyDataAlgorithm)`

27.375.4 Member Data Documentation

- 27.375.4.1 `char* vtkGDCMPolyDataReader::FileName` `[protected]`
- 27.375.4.2 `vtkMedicalImageProperties* vtkGDCMPolyDataReader::MedicalImageProperties` `[protected]`
- 27.375.4.3 `vtkRTStructSetProperties* vtkGDCMPolyDataReader::RTStructSetProperties` `[protected]`

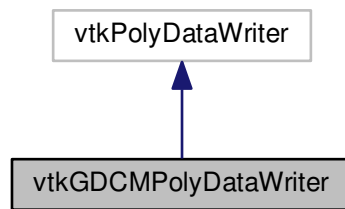
The documentation for this class was generated from the following file:

- [vtkGDCMPolyDataReader.h](#)

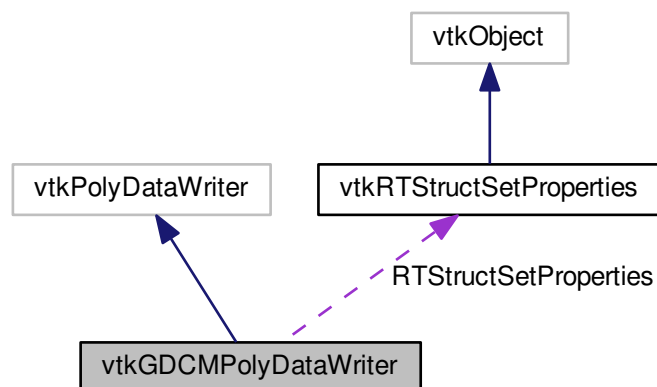
27.376 vtkGDCMPolyDataWriter Class Reference

```
#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)
- [vtkTypeRevisionMacro](#) (vtkGDCMPolyDataWriter, vtkPolyDataWriter)

Static Public Member Functions

- static [vtkGDCMPolyDataWriter * New](#) ()

Protected Member Functions

- [vtkGDCMPolyDataWriter](#) ()
- [~vtkGDCMPolyDataWriter](#) ()
- void [WriteData](#) ()
- void [WriteRTSTRUCTData](#) ([gdcmm::File](#) &file, int num)
- void [WriteRTSTRUCTInfo](#) ([gdcmm::File](#) &file)

Protected Attributes

- [vtkMedicalImageProperties](#) * [MedicalImageProperties](#)
- [vtkRTStructSetProperties](#) * [RTStructSetProperties](#)

27.376.1 Detailed Description

Examples:

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

27.376.2 Constructor & Destructor Documentation

27.376.2.1 `vtkGDCMPolyDataWriter::vtkGDCMPolyDataWriter ()` [protected]

27.376.2.2 `vtkGDCMPolyDataWriter::~~vtkGDCMPolyDataWriter ()` [protected]

27.376.3 Member Function Documentation

27.376.3.1 `void vtkGDCMPolyDataWriter::InitializeRTStructSet (vtkStdString inDirectory, vtkStdString inStructLabel, vtkStdString inStructName, vtkStringArray * inROINames, vtkStringArray * inROIAlgorithmName, vtkStringArray * inROIType)`

Examples:

[GenerateRTSTRUCT.cxx](#).

27.376.3.2 `static vtkGDCMPolyDataWriter* vtkGDCMPolyDataWriter::New ()` [static]

Examples:

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

27.376.3.3 `virtual void vtkGDCMPolyDataWriter::PrintSelf (ostream & os, vtkIndent indent)` [virtual]

27.376.3.4 `virtual void vtkGDCMPolyDataWriter::SetMedicalImageProperties (vtkMedicalImageProperties * pd)` [virtual]

Examples:

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

27.376.3.5 void vtkGDCMPolyDataWriter::SetNumberOfInputPorts (int *n*)

Examples:

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

27.376.3.6 virtual void vtkGDCMPolyDataWriter::SetRTStructSetProperties (vtkRTStructSetProperties * *pd*) [virtual]

Examples:

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

27.376.3.7 vtkGDCMPolyDataWriter::vtkTypeRevisionMacro (vtkGDCMPolyDataWriter , vtkPolyDataWriter)

27.376.3.8 void vtkGDCMPolyDataWriter::WriteData () [protected]

27.376.3.9 void vtkGDCMPolyDataWriter::WriteRTSTRUCTData (gdcm::File & *file*, int *num*) [protected]

27.376.3.10 void vtkGDCMPolyDataWriter::WriteRTSTRUCTInfo (gdcm::File & *file*) [protected]

27.376.4 Member Data Documentation

27.376.4.1 vtkMedicalImageProperties* vtkGDCMPolyDataWriter::MedicalImageProperties [protected]

27.376.4.2 vtkRTStructSetProperties* vtkGDCMPolyDataWriter::RTStructSetProperties [protected]

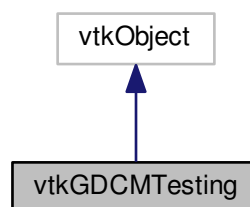
The documentation for this class was generated from the following file:

- [vtkGDCMPolyDataWriter.h](#)

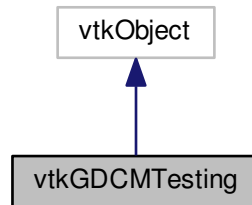
27.377 vtkGDCMTesting Class Reference

```
#include <vtkGDCMTesting.h>
```

Inheritance diagram for vtkGDCMTesting:



Collaboration diagram for vtkGDCMTesting:



Public Types

- typedef const char *const (* [MD5MetalImagesType](#))[3]

Public Member Functions

- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkTypeRevisionMacro](#) (vtkGDCMTesting, vtkObject)

Static Public Member Functions

- static const char * [GetGDCMDataRoot](#) ()
- static const char *const * [GetMD5MetalImage](#) (unsigned int file)
- static const char * [GetMHDMD5FromFile](#) (const char *filepath)
- static unsigned int [GetNumberOfMD5MetalImages](#) ()
- static const char * [GetRAWMD5FromFile](#) (const char *filepath)
- static const char * [GetVTKDataRoot](#) ()
- static [vtkGDCMTesting](#) * [New](#) ()

Protected Member Functions

- [vtkGDCMTesting](#) ()
- [~vtkGDCMTesting](#) ()

27.377.1 Detailed Description

Examples:

[HelloActiviz5.cs](#), [HelloVTKWorld2.cs](#), [MetalImageMD5Activiz.cs](#), [ReadSeriesIntoVTK.java](#), and [RefCounting.cs](#).

27.377.2 Member Typedef Documentation

27.377.2.1 `typedef const char* const(* vtkGDCMTesting::MD5MetalmagesType)[3]`

27.377.3 Constructor & Destructor Documentation

27.377.3.1 `vtkGDCMTesting::vtkGDCMTesting ()` [protected]

27.377.3.2 `vtkGDCMTesting::~~vtkGDCMTesting ()` [protected]

27.377.4 Member Function Documentation

27.377.4.1 `static const char* vtkGDCMTesting::GetGDCMDataRoot ()` [static]

Examples:

[HelloActiviz5.cs](#), and [ReadSeriesIntoVTK.java](#).

27.377.4.2 `static const char* const* vtkGDCMTesting::GetMD5Metalmage (unsigned int file)` [static]

27.377.4.3 `static const char* vtkGDCMTesting::GetMHDM5FromFile (const char * filepath)` [static]

Examples:

[MetalmageMD5Activiz.cs](#).

27.377.4.4 `static unsigned int vtkGDCMTesting::GetNumberOfMD5Metalmages ()` [static]

27.377.4.5 `static const char* vtkGDCMTesting::GetRAWMD5FromFile (const char * filepath)` [static]

Examples:

[MetalmageMD5Activiz.cs](#).

27.377.4.6 `static const char* vtkGDCMTesting::GetVTKDataRoot ()` [static]

Examples:

[HelloActiviz5.cs](#), and [HelloVTKWorld2.cs](#).

27.377.4.7 `static vtkGDCMTesting* vtkGDCMTesting::New ()` [static]

Examples:

[RefCounting.cs](#).

27.377.4.8 void vtkGDCMTesting::PrintSelf (ostream & *os*, vtkIndent *indent*)

27.377.4.9 vtkGDCMTesting::vtkTypeRevisionMacro (vtkGDCMTesting , vtkObject)

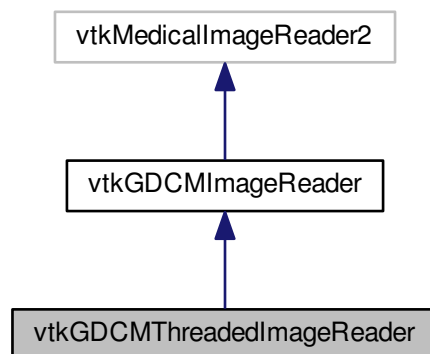
The documentation for this class was generated from the following file:

- [vtkGDCMTesting.h](#)

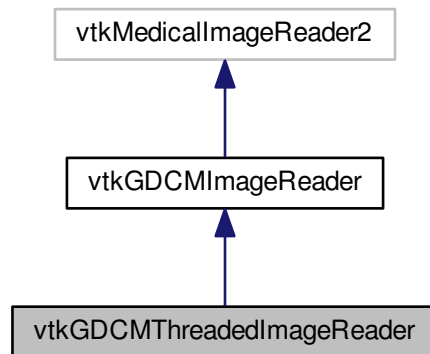
27.378 vtkGDCMThreadedImageReader Class Reference

```
#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](#) (Shift, double)
- [vtkSetMacro](#) (Scale, double)
- [vtkSetMacro](#) (UseShiftScale, int)
- [vtkTypeRevisionMacro](#) (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

27.378.1 Constructor & Destructor Documentation

27.378.1.1 [vtkGDCMThreadedImageReader::vtkGDCMThreadedImageReader](#) () [protected]

27.378.1.2 `vtkGDCMThreadedImageReader::~~vtkGDCMThreadedImageReader ()` [protected]

27.378.2 Member Function Documentation

27.378.2.1 `void vtkGDCMThreadedImageReader::ExecuteData (vtkDataObject * out)` [protected]

27.378.2.2 `void vtkGDCMThreadedImageReader::ExecuteInformation ()` [protected]

27.378.2.3 `static vtkGDCMThreadedImageReader* vtkGDCMThreadedImageReader::New ()` [static]

27.378.2.4 `virtual void vtkGDCMThreadedImageReader::PrintSelf (ostream & os, vtkIndent indent)` [virtual]

Reimplemented from [vtkGDCMImageReader](#).

27.378.2.5 `void vtkGDCMThreadedImageReader::ReadFiles (unsigned int nfiles, const char * filenames[])` [protected]

27.378.2.6 `void vtkGDCMThreadedImageReader::RequestDataCompat ()` [protected]

27.378.2.7 `vtkGDCMThreadedImageReader::vtkBooleanMacro (UseShiftScale , int)`

27.378.2.8 `vtkGDCMThreadedImageReader::vtkGetMacro (UseShiftScale , int)`

27.378.2.9 `vtkGDCMThreadedImageReader::vtkSetMacro (Shift , double)`

27.378.2.10 `vtkGDCMThreadedImageReader::vtkSetMacro (Scale , double)`

27.378.2.11 `vtkGDCMThreadedImageReader::vtkSetMacro (UseShiftScale , int)`

27.378.2.12 `vtkGDCMThreadedImageReader::vtkTypeRevisionMacro (vtkGDCMThreadedImageReader ,
vtkGDCMImageReader)`

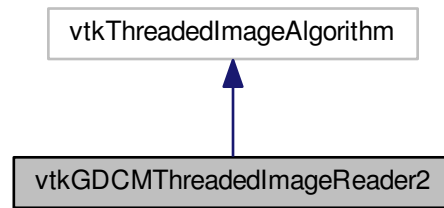
The documentation for this class was generated from the following file:

- [vtkGDCMThreadedImageReader.h](#)

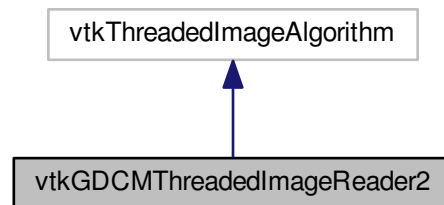
27.379 vtkGDCMThreadedImageReader2 Class Reference

```
#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](#) (FileLowerLeft, int)
- [vtkGetMacro](#) (NumberOfOverlays, int)
- [vtkGetMacro](#) (DataScalarType, int)
- [vtkGetMacro](#) (NumberOfScalarComponents, int)
- [vtkGetMacro](#) (LoadOverlays, int)
- [vtkGetMacro](#) (Shift, double)
- [vtkGetMacro](#) (Scale, double)
- [vtkGetMacro](#) (UseShiftScale, int)

- [vtkGetObjectMacro](#) (FileNames, vtkStringArray)
- [vtkGetVector3Macro](#) (DataOrigin, double)
- [vtkGetVector3Macro](#) (DataSpacing, double)
- [vtkGetVector6Macro](#) (DataExtent, int)
- [vtkSetMacro](#) (FileLowerLeft, int)
- [vtkSetMacro](#) (DataScalarType, int)
- [vtkSetMacro](#) (NumberOfScalarComponents, int)
- [vtkSetMacro](#) (LoadOverlays, int)
- [vtkSetMacro](#) (Shift, double)
- [vtkSetMacro](#) (Scale, double)
- [vtkSetMacro](#) (UseShiftScale, int)
- [vtkSetVector3Macro](#) (DataOrigin, double)
- [vtkSetVector3Macro](#) (DataSpacing, double)
- [vtkSetVector6Macro](#) (DataExtent, int)
- [vtkTypeRevisionMacro](#) ([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)

27.379.1 Constructor & Destructor Documentation

27.379.1.1 [vtkGDCMThreadedImageReader2::vtkGDCMThreadedImageReader2](#) () [protected]

27.379.1.2 [vtkGDCMThreadedImageReader2::~~vtkGDCMThreadedImageReader2](#) () [protected]

27.379.2 Member Function Documentation

27.379.2.1 virtual const char* [vtkGDCMThreadedImageReader2::GetFileName](#) (int *i* = 0) [virtual]

27.379.2.2 static [vtkGDCMThreadedImageReader2](#)* [vtkGDCMThreadedImageReader2::New](#) () [static]

27.379.2.3 virtual void [vtkGDCMThreadedImageReader2::PrintSelf](#) (ostream & *os*, vtkIndent *indent*) [virtual]

27.379.2.4 int [vtkGDCMThreadedImageReader2::RequestInformation](#) (vtkInformation * *request*, vtkInformationVector ** *inputVector*, vtkInformationVector * *outputVector*) [protected]

27.379.2.5 virtual void [vtkGDCMThreadedImageReader2::SetFileName](#) (const char * *filename*) [virtual]

27.379.2.6 virtual void [vtkGDCMThreadedImageReader2::SetFileNames](#) (vtkStringArray *) [virtual]

- 27.379.2.7 `int vtkGDCMThreadedImageReader2::SplitExtent (int splitExt[6], int startExt[6], int num, int total)`
- 27.379.2.8 `void vtkGDCMThreadedImageReader2::ThreadedRequestData (vtkInformation * request, vtkInformationVector ** inputVector, vtkInformationVector * outputVector, vtkImageData *** inData, vtkImageData ** outData, int outExt[6], int id) [protected]`
- 27.379.2.9 `vtkGDCMThreadedImageReader2::vtkBooleanMacro (FileLowerLeft , int)`
- 27.379.2.10 `vtkGDCMThreadedImageReader2::vtkBooleanMacro (LoadOverlays , int)`
- 27.379.2.11 `vtkGDCMThreadedImageReader2::vtkBooleanMacro (UseShiftScale , int)`
- 27.379.2.12 `vtkGDCMThreadedImageReader2::vtkGetMacro (FileLowerLeft , int)`
- 27.379.2.13 `vtkGDCMThreadedImageReader2::vtkGetMacro (NumberOfOverlays , int)`
- 27.379.2.14 `vtkGDCMThreadedImageReader2::vtkGetMacro (DataScalarType , int)`
- 27.379.2.15 `vtkGDCMThreadedImageReader2::vtkGetMacro (NumberOfScalarComponents , int)`
- 27.379.2.16 `vtkGDCMThreadedImageReader2::vtkGetMacro (LoadOverlays , int)`
- 27.379.2.17 `vtkGDCMThreadedImageReader2::vtkGetMacro (Shift , double)`
- 27.379.2.18 `vtkGDCMThreadedImageReader2::vtkGetMacro (Scale , double)`
- 27.379.2.19 `vtkGDCMThreadedImageReader2::vtkGetMacro (UseShiftScale , int)`
- 27.379.2.20 `vtkGDCMThreadedImageReader2::vtkGetObjectMacro (FileNames , vtkStringArray)`
- 27.379.2.21 `vtkGDCMThreadedImageReader2::vtkGetVector3Macro (DataOrigin , double)`
- 27.379.2.22 `vtkGDCMThreadedImageReader2::vtkGetVector3Macro (DataSpacing , double)`
- 27.379.2.23 `vtkGDCMThreadedImageReader2::vtkGetVector6Macro (DataExtent , int)`
- 27.379.2.24 `vtkGDCMThreadedImageReader2::vtkSetMacro (FileLowerLeft , int)`
- 27.379.2.25 `vtkGDCMThreadedImageReader2::vtkSetMacro (DataScalarType , int)`
- 27.379.2.26 `vtkGDCMThreadedImageReader2::vtkSetMacro (NumberOfScalarComponents , int)`
- 27.379.2.27 `vtkGDCMThreadedImageReader2::vtkSetMacro (LoadOverlays , int)`
- 27.379.2.28 `vtkGDCMThreadedImageReader2::vtkSetMacro (Shift , double)`
- 27.379.2.29 `vtkGDCMThreadedImageReader2::vtkSetMacro (Scale , double)`
- 27.379.2.30 `vtkGDCMThreadedImageReader2::vtkSetMacro (UseShiftScale , int)`
- 27.379.2.31 `vtkGDCMThreadedImageReader2::vtkSetVector3Macro (DataOrigin , double)`

27.379.2.32 `vtkGDCMThreadedImageReader2::vtkSetVector3Macro (DataSpacing , double)`

27.379.2.33 `vtkGDCMThreadedImageReader2::vtkSetVector6Macro (DataExtent , int)`

27.379.2.34 `vtkGDCMThreadedImageReader2::vtkTypeRevisionMacro (vtkGDCMThreadedImageReader2 ,
vtkThreadedImageAlgorithm)`

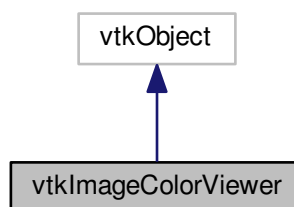
The documentation for this class was generated from the following file:

- [vtkGDCMThreadedImageReader2.h](#)

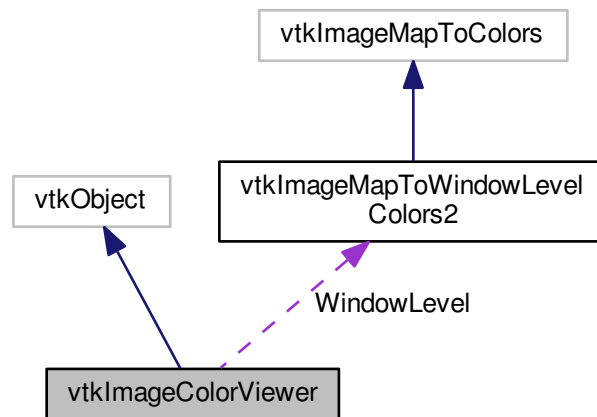
27.380 vtkImageColorViewer Class Reference

```
#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 void [GetSliceRange](#) (int range[2])
- virtual void [GetSliceRange](#) (int &min, int &max)
- virtual int * [GetSliceRange](#) ()
- 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 GetWholeZMin())
- [VTK_LEGACY](#) (int GetWholeZMax())
- [VTK_LEGACY](#) (int GetZSlice())
- [VTK_LEGACY](#) (void SetZSlice(int))
- [vtkBooleanMacro](#) (OffScreenRendering, int)
- [vtkGetMacro](#) (SliceOrientation, int)
- [vtkGetMacro](#) (Slice, int)
- [vtkGetObjectMacro](#) (RenderWindow, vtkRenderWindow)
- [vtkGetObjectMacro](#) (Renderer, vtkRenderer)
- [vtkGetObjectMacro](#) (ImageActor, vtkImageActor)
- [vtkGetObjectMacro](#) (WindowLevel, vtkImageMapToWindowLevelColors2)
- [vtkGetObjectMacro](#) (InteractorStyle, vtkInteractorStyleImage)
- [vtkTypeRevisionMacro](#) (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](#)

27.380.1 Detailed Description

Examples:

[gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

27.380.2 Member Enumeration Documentation

27.380.2.1 anonymous enum

Enumerator

SLICE_ORIENTATION_YZ
SLICE_ORIENTATION_XZ
SLICE_ORIENTATION_XY

27.380.3 Constructor & Destructor Documentation

27.380.3.1 `vtkImageColorViewer::vtkImageColorViewer ()` [protected]

27.380.3.2 `vtkImageColorViewer::~~vtkImageColorViewer ()` [protected]

27.380.4 Member Function Documentation

27.380.4.1 `virtual void vtkImageColorViewer::AddInput (vtkImageData * input)` [virtual]

27.380.4.2 `virtual void vtkImageColorViewer::AddInputConnection (vtkAlgorithmOutput * input)` [virtual]

27.380.4.3 `virtual double vtkImageColorViewer::GetColorLevel ()` [virtual]

27.380.4.4 `virtual double vtkImageColorViewer::GetColorWindow ()` [virtual]

27.380.4.5 `virtual vtkImageData* vtkImageColorViewer::GetInput ()` [virtual]

```

27.380.4.6  virtual int vtkImageColorViewer::GetOffScreenRendering ( ) [virtual]

27.380.4.7  double vtkImageColorViewer::GetOverlayVisibility ( )

27.380.4.8  virtual int* vtkImageColorViewer::GetPosition ( ) [virtual]

27.380.4.9  virtual int* vtkImageColorViewer::GetSize ( ) [virtual]

27.380.4.10 virtual int vtkImageColorViewer::GetSliceMax ( ) [virtual]

27.380.4.11 virtual int vtkImageColorViewer::GetSliceMin ( ) [virtual]

27.380.4.12 virtual void vtkImageColorViewer::GetSliceRange ( int range[2] ) [inline],[virtual]

27.380.4.13 virtual void vtkImageColorViewer::GetSliceRange ( int & min, int & max ) [virtual]

27.380.4.14 virtual int* vtkImageColorViewer::GetSliceRange ( ) [virtual]

27.380.4.15 virtual const char* vtkImageColorViewer::GetWindowName ( ) [virtual]

27.380.4.16 virtual void vtkImageColorViewer::InstallPipeline ( ) [protected],[virtual]

27.380.4.17 static vtkImageColorViewer* vtkImageColorViewer::New ( ) [static]

```

Examples:

[gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

```

27.380.4.18 void vtkImageColorViewer::PrintSelf ( ostream & os, vtkIndent indent )

27.380.4.19 virtual void vtkImageColorViewer::Render ( void ) [virtual]

```

Examples:

[gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

```

27.380.4.20 virtual void vtkImageColorViewer::SetColorLevel ( double s ) [virtual]

27.380.4.21 virtual void vtkImageColorViewer::SetColorWindow ( double s ) [virtual]

27.380.4.22 virtual void vtkImageColorViewer::SetDisplayId ( void * a ) [virtual]

27.380.4.23 virtual void vtkImageColorViewer::SetInput ( vtkImageData * in ) [virtual]

```

Examples:

[gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

27.380.4.24 virtual void vtkImageColorViewer::SetInputConnection (vtkAlgorithmOutput * *input*) [virtual]

27.380.4.25 virtual void vtkImageColorViewer::SetOffScreenRendering (int) [virtual]

27.380.4.26 void vtkImageColorViewer::SetOverlayVisibility (double *vis*)

27.380.4.27 virtual void vtkImageColorViewer::SetParentId (void * *a*) [virtual]

27.380.4.28 virtual void vtkImageColorViewer::SetPosition (int *a*, int *b*) [virtual]

27.380.4.29 virtual void vtkImageColorViewer::SetPosition (int *a[2]*) [inline],[virtual]

References SetPosition().

Referenced by SetPosition().

27.380.4.30 virtual void vtkImageColorViewer::SetRenderer (vtkRenderer * *arg*) [virtual]

27.380.4.31 virtual void vtkImageColorViewer::SetRenderWindow (vtkRenderWindow * *arg*) [virtual]

27.380.4.32 virtual void vtkImageColorViewer::SetSize (int *a*, int *b*) [virtual]

Examples:

[gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

27.380.4.33 virtual void vtkImageColorViewer::SetSize (int *a[2]*) [inline],[virtual]

References SetSize().

Referenced by SetSize().

27.380.4.34 virtual void vtkImageColorViewer::SetSlice (int *s*) [virtual]

27.380.4.35 virtual void vtkImageColorViewer::SetSliceOrientation (int *orientation*) [virtual]

27.380.4.36 virtual void vtkImageColorViewer::SetSliceOrientationToXY () [inline],[virtual]

References SLICE_ORIENTATION_XY.

27.380.4.37 virtual void vtkImageColorViewer::SetSliceOrientationToXZ () [inline],[virtual]

References SLICE_ORIENTATION_XZ.

27.380.4.38 virtual void vtkImageColorViewer::SetSliceOrientationToYZ () [inline],[virtual]

References SLICE_ORIENTATION_YZ.

27.380.4.39 `virtual void vtkImageColorViewer::SetupInteractor (vtkRenderWindowInteractor *)` [virtual]

Examples:

[gdcmrtonplan.cxx](#), and [gdcmrtpplan.cxx](#).

27.380.4.40 `virtual void vtkImageColorViewer::SetWindowId (void * a)` [virtual]

27.380.4.41 `virtual void vtkImageColorViewer::UnInstallPipeline ()` [protected],[virtual]

27.380.4.42 `virtual void vtkImageColorViewer::UpdateDisplayExtent ()` [virtual]

27.380.4.43 `virtual void vtkImageColorViewer::UpdateOrientation ()` [protected],[virtual]

27.380.4.44 `vtkImageColorViewer::VTK_LEGACY (int GetWholeZMin())`

27.380.4.45 `vtkImageColorViewer::VTK_LEGACY (int GetWholeZMax())`

27.380.4.46 `vtkImageColorViewer::VTK_LEGACY (int GetZSlice())`

27.380.4.47 `vtkImageColorViewer::VTK_LEGACY (void SetZSliceint)`

27.380.4.48 `vtkImageColorViewer::vtkBooleanMacro (OffScreenRendering , int)`

27.380.4.49 `vtkImageColorViewer::vtkGetMacro (SliceOrientation , int)`

27.380.4.50 `vtkImageColorViewer::vtkGetMacro (Slice , int)`

27.380.4.51 `vtkImageColorViewer::vtkGetObjectMacro (RenderWindow , vtkRenderWindow)`

27.380.4.52 `vtkImageColorViewer::vtkGetObjectMacro (Renderer , vtkRenderer)`

27.380.4.53 `vtkImageColorViewer::vtkGetObjectMacro (ImageActor , vtkImageActor)`

27.380.4.54 `vtkImageColorViewer::vtkGetObjectMacro (WindowLevel , vtkImageMapToWindowLevelColors2)`

27.380.4.55 `vtkImageColorViewer::vtkGetObjectMacro (InteractorStyle , vtkInteractorStyleImage)`

27.380.4.56 `vtkImageColorViewer::vtkTypeRevisionMacro (vtkImageColorViewer , vtkObject)`

27.380.5 Friends And Related Function Documentation

27.380.5.1 `friend class vtkImageColorViewerCallback` [friend]

27.380.6 Member Data Documentation

27.380.6.1 `int vtkImageColorViewer::FirstRender` [protected]

27.380.6.2 `vtkImageActor* vtkImageColorViewer::ImageActor` [protected]

27.380.6.3 `vtkRenderWindowInteractor* vtkImageColorViewer::Interactor` [protected]

27.380.6.4 `vtkInteractorStyleImage*` `vtkImageColorViewer::InteractorStyle` `[protected]`

27.380.6.5 `vtkImageActor*` `vtkImageColorViewer::OverlayImageActor` `[protected]`

27.380.6.6 `vtkRenderer*` `vtkImageColorViewer::Renderer` `[protected]`

27.380.6.7 `vtkRenderWindow*` `vtkImageColorViewer::RenderWindow` `[protected]`

27.380.6.8 `int` `vtkImageColorViewer::Slice` `[protected]`

27.380.6.9 `int` `vtkImageColorViewer::SliceOrientation` `[protected]`

27.380.6.10 `vtkImageMapToWindowLevelColors2*` `vtkImageColorViewer::WindowLevel` `[protected]`

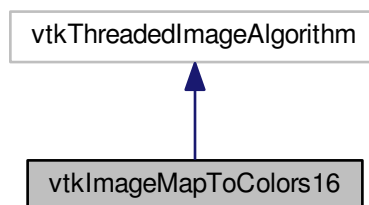
The documentation for this class was generated from the following file:

- [vtkImageColorViewer.h](#)

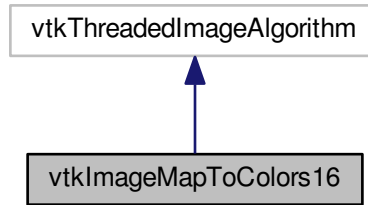
27.381 vtkImageMapToColors16 Class Reference

```
#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](#) (OutputFormat, int)
- [vtkGetMacro](#) (ActiveComponent, int)
- [vtkGetMacro](#) (PassAlphaToOutput, int)
- [vtkGetObjectMacro](#) (LookupTable, vtkScalarsToColors)
- [vtkSetMacro](#) (OutputFormat, int)
- [vtkSetMacro](#) (ActiveComponent, int)
- [vtkSetMacro](#) (PassAlphaToOutput, int)
- [vtkTypeRevisionMacro](#) (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](#)

27.381.1 Constructor & Destructor Documentation

27.381.1.1 `vtkImageMapToColors16::vtkImageMapToColors16 ()` `[protected]`

27.381.1.2 `vtkImageMapToColors16::~~vtkImageMapToColors16 ()` `[protected]`

27.381.2 Member Function Documentation

27.381.2.1 `virtual unsigned long vtkImageMapToColors16::GetMTime ()` `[virtual]`

27.381.2.2 `static vtkImageMapToColors16* vtkImageMapToColors16::New ()` `[static]`

27.381.2.3 `void vtkImageMapToColors16::PrintSelf (ostream & os, vtkIndent indent)`

27.381.2.4 `virtual int vtkImageMapToColors16::RequestData (vtkInformation * request, vtkInformationVector ** inputVector, vtkInformationVector * outputVector)` `[protected]`, `[virtual]`

27.381.2.5 `virtual int vtkImageMapToColors16::RequestInformation (vtkInformation * , vtkInformationVector ** , vtkInformationVector *)` `[protected]`, `[virtual]`

27.381.2.6 `virtual void vtkImageMapToColors16::SetLookupTable (vtkScalarsToColors *)` `[virtual]`

27.381.2.7 `void vtkImageMapToColors16::SetOutputFormatToLuminance ()` `[inline]`

27.381.2.8 `void vtkImageMapToColors16::SetOutputFormatToLuminanceAlpha ()` `[inline]`

27.381.2.9 `void vtkImageMapToColors16::SetOutputFormatToRGB ()` `[inline]`

27.381.2.10 `void vtkImageMapToColors16::SetOutputFormatToRGBA ()` `[inline]`

27.381.2.11 `void vtkImageMapToColors16::ThreadedRequestData (vtkInformation * request, vtkInformationVector ** inputVector, vtkInformationVector * outputVector, vtkImageData *** inData, vtkImageData ** outData, int extent[6], int id)` `[protected]`

27.381.2.12 `vtkImageMapToColors16::vtkBooleanMacro (PassAlphaToOutput , int)`

27.381.2.13 `vtkImageMapToColors16::vtkGetMacro (OutputFormat , int)`

27.381.2.14 `vtkImageMapToColors16::vtkGetMacro (ActiveComponent , int)`

27.381.2.15 `vtkImageMapToColors16::vtkGetMacro (PassAlphaToOutput , int)`

27.381.2.16 `vtkImageMapToColors16::vtkGetObjectMacro (LookupTable , vtkScalarsToColors)`

27.381.2.17 `vtkImageMapToColors16::vtkSetMacro (OutputFormat , int)`

27.381.2.18 `vtkImageMapToColors16::vtkSetMacro (ActiveComponent , int)`

27.381.2.19 `vtkImageMapToColors16::vtkSetMacro (PassAlphaToOutput , int)`

27.381.2.20 `vtkImageMapToColors16::vtkTypeRevisionMacro (vtkImageMapToColors16 , vtkThreadedImageAlgorithm)`

27.381.3 Member Data Documentation

27.381.3.1 `int vtkImageMapToColors16::ActiveComponent` [protected]

27.381.3.2 `int vtkImageMapToColors16::DataWasPassed` [protected]

27.381.3.3 `vtkScalarsToColors* vtkImageMapToColors16::LookupTable` [protected]

27.381.3.4 `int vtkImageMapToColors16::OutputFormat` [protected]

27.381.3.5 `int vtkImageMapToColors16::PassAlphaToOutput` [protected]

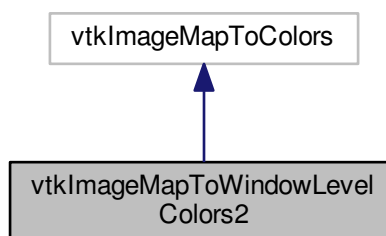
The documentation for this class was generated from the following file:

- [vtkImageMapToColors16.h](#)

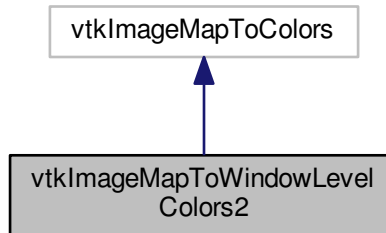
27.382 vtkImageMapToWindowLevelColors2 Class Reference

```
#include <vtkImageMapToWindowLevelColors2.h>
```

Inheritance diagram for `vtkImageMapToWindowLevelColors2`:



Collaboration diagram for vtkImageMapToWindowLevelColors2:



Public Member Functions

- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkGetMacro](#) (Window, double)
- [vtkGetMacro](#) (Level, double)
- [vtkSetMacro](#) (Window, double)
- [vtkSetMacro](#) (Level, double)
- [vtkTypeRevisionMacro](#) (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](#)

27.382.1 Constructor & Destructor Documentation

27.382.1.1 `vtkImageMapToWindowLevelColors2::vtkImageMapToWindowLevelColors2 ()` [protected]

27.382.1.2 `vtkImageMapToWindowLevelColors2::~~vtkImageMapToWindowLevelColors2 ()` [protected]

27.382.2 Member Function Documentation

27.382.2.1 `static vtkImageMapToWindowLevelColors2* vtkImageMapToWindowLevelColors2::New ()` [static]

27.382.2.2 `void vtkImageMapToWindowLevelColors2::PrintSelf (ostream & os, vtkIndent indent)`

27.382.2.3 `virtual int vtkImageMapToWindowLevelColors2::RequestData (vtkInformation * request, vtkInformationVector ** inputVector, vtkInformationVector * outputVector)` [protected],[virtual]

27.382.2.4 `virtual int vtkImageMapToWindowLevelColors2::RequestInformation (vtkInformation * , vtkInformationVector ** , vtkInformationVector *)` [protected],[virtual]

27.382.2.5 `void vtkImageMapToWindowLevelColors2::ThreadedRequestData (vtkInformation * request, vtkInformationVector ** inputVector, vtkInformationVector * outputVector, vtkImageData *** inData, vtkImageData ** outData, int extent[6], int id)` [protected]

27.382.2.6 `vtkImageMapToWindowLevelColors2::vtkGetMacro (Window , double)`

27.382.2.7 `vtkImageMapToWindowLevelColors2::vtkGetMacro (Level , double)`

27.382.2.8 `vtkImageMapToWindowLevelColors2::vtkSetMacro (Window , double)`

27.382.2.9 `vtkImageMapToWindowLevelColors2::vtkSetMacro (Level , double)`

27.382.2.10 `vtkImageMapToWindowLevelColors2::vtkTypeRevisionMacro (vtkImageMapToWindowLevelColors2 , vtkImageMapToColors)`

27.382.3 Member Data Documentation

27.382.3.1 `double vtkImageMapToWindowLevelColors2::Level` [protected]

27.382.3.2 `double vtkImageMapToWindowLevelColors2::Window` [protected]

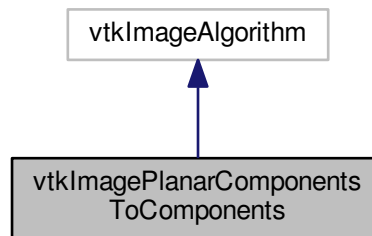
The documentation for this class was generated from the following file:

- [vtkImageMapToWindowLevelColors2.h](#)

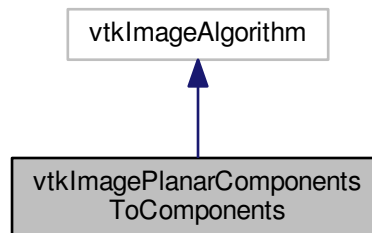
27.383 vtkImagePlanarComponentsToComponents Class Reference

```
#include <vtkImagePlanarComponentsToComponents.h>
```

Inheritance diagram for vtkImagePlanarComponentsToComponents:



Collaboration diagram for vtkImagePlanarComponentsToComponents:



Public Member Functions

- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkTypeRevisionMacro](#) ([vtkImagePlanarComponentsToComponents](#), vtkImageAlgorithm)

Static Public Member Functions

- static [vtkImagePlanarComponentsToComponents](#) * [New](#) ()

Protected Member Functions

- [vtkImagePlanarComponentsToComponents](#) ()
- [~vtkImagePlanarComponentsToComponents](#) ()
- virtual int [RequestData](#) (vtkInformation *, vtkInformationVector **, vtkInformationVector *)

27.383.1 Constructor & Destructor Documentation

27.383.1.1 `vtkImagePlanarComponentsToComponents::vtkImagePlanarComponentsToComponents ()` [protected]

27.383.1.2 `vtkImagePlanarComponentsToComponents::~~vtkImagePlanarComponentsToComponents ()` [inline], [protected]

27.383.2 Member Function Documentation

27.383.2.1 `static vtkImagePlanarComponentsToComponents* vtkImagePlanarComponentsToComponents::New ()` [static]

27.383.2.2 `void vtkImagePlanarComponentsToComponents::PrintSelf (ostream & os, vtkIndent indent)`

27.383.2.3 `virtual int vtkImagePlanarComponentsToComponents::RequestData (vtkInformation *, vtkInformationVector **, vtkInformationVector *)` [protected], [virtual]

27.383.2.4 `vtkImagePlanarComponentsToComponents::vtkTypeRevisionMacro (vtkImagePlanarComponentsToComponents, vtkImageAlgorithm)`

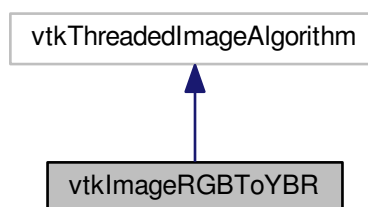
The documentation for this class was generated from the following file:

- [vtkImagePlanarComponentsToComponents.h](#)

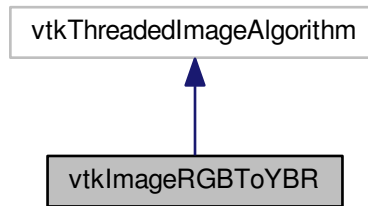
27.384 vtkImageRGBToYBR Class Reference

```
#include <vtkImageRGBToYBR.h>
```

Inheritance diagram for `vtkImageRGBToYBR`:



Collaboration diagram for vtkImageRGBToYBR:



Public Member Functions

- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkTypeRevisionMacro](#) ([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)

27.384.1 Constructor & Destructor Documentation

27.384.1.1 [vtkImageRGBToYBR::vtkImageRGBToYBR \(\)](#) [protected]

27.384.1.2 [vtkImageRGBToYBR::~~vtkImageRGBToYBR \(\)](#) [inline],[protected]

27.384.2 Member Function Documentation

27.384.2.1 static [vtkImageRGBToYBR*](#) [vtkImageRGBToYBR::New \(\)](#) [static]

27.384.2.2 void [vtkImageRGBToYBR::PrintSelf \(ostream & os, vtkIndent indent \)](#)

27.384.2.3 void [vtkImageRGBToYBR::ThreadedExecute \(vtkImageData * inData, vtkImageData * outData, int ext\[6\], int id \)](#)
[protected]

27.384.2.4 [vtkImageRGBToYBR::vtkTypeRevisionMacro \(vtkImageRGBToYBR , vtkThreadedImageAlgorithm \)](#)

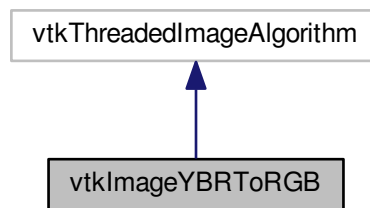
The documentation for this class was generated from the following file:

- [vtkImageRGBToYBR.h](#)

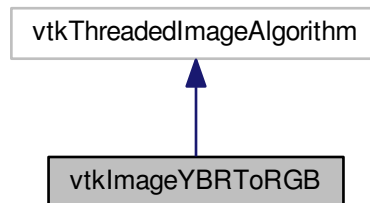
27.385 vtkImageYBRToRGB Class Reference

```
#include <vtkImageYBRToRGB.h>
```

Inheritance diagram for vtkImageYBRToRGB:



Collaboration diagram for vtkImageYBRToRGB:



Public Member Functions

- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkTypeRevisionMacro](#) (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)

27.385.1 Constructor & Destructor Documentation

27.385.1.1 `vtkImageYBRToRGB::vtkImageYBRToRGB ()` [protected]

27.385.1.2 `vtkImageYBRToRGB::~~vtkImageYBRToRGB ()` [inline], [protected]

27.385.2 Member Function Documentation

27.385.2.1 `static vtkImageYBRToRGB* vtkImageYBRToRGB::New ()` [static]

27.385.2.2 `void vtkImageYBRToRGB::PrintSelf (ostream & os, vtkIndent indent)`

27.385.2.3 `void vtkImageYBRToRGB::ThreadedExecute (vtkImageData * inData, vtkImageData * outData, int ext[6], int id)`
[protected]

27.385.2.4 `vtkImageYBRToRGB::vtkTypeRevisionMacro (vtkImageYBRToRGB , vtkThreadedImageAlgorithm)`

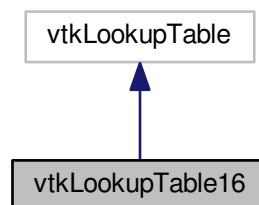
The documentation for this class was generated from the following file:

- [vtkImageYBRToRGB.h](#)

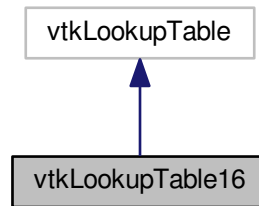
27.386 vtkLookupTable16 Class Reference

```
#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)
- [vtkTypeRevisionMacro](#) ([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](#)

27.386.1 Constructor & Destructor Documentation

27.386.1.1 `vtkLookupTable16::vtkLookupTable16 (int size = 256, int ext = 256)` [protected]

27.386.1.2 `vtkLookupTable16::~~vtkLookupTable16 ()` [protected]

27.386.2 Member Function Documentation

- 27.386.2.1 void vtkLookupTable16::Build ()
- 27.386.2.2 unsigned short* vtkLookupTable16::GetPointer (const vtkIdType *id*) [inline]
- 27.386.2.3 void vtkLookupTable16::MapScalarsThroughTable2 (void * *input*, unsigned char * *output*, int *inputDataType*, int *numberOfValues*, int *inputIncrement*, int *outputFormat*) [protected]
- 27.386.2.4 static vtkLookupTable16* vtkLookupTable16::New () [static]
- 27.386.2.5 void vtkLookupTable16::PrintSelf (ostream & *os*, vtkIndent *indent*)
- 27.386.2.6 void vtkLookupTable16::SetNumberOfTableValues (vtkIdType *number*)
- 27.386.2.7 vtkLookupTable16::vtkTypeRevisionMacro (vtkLookupTable16 , vtkLookupTable)
- 27.386.2.8 unsigned char * vtkLookupTable16::WritePointer (const vtkIdType *id*, const int *number*) [inline]

References Table16.

27.386.3 Member Data Documentation

- 27.386.3.1 vtkUnsignedShortArray* vtkLookupTable16::Table16 [protected]

Referenced by WritePointer().

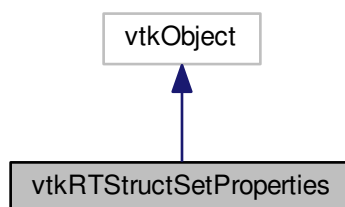
The documentation for this class was generated from the following file:

- [vtkLookupTable16.h](#)

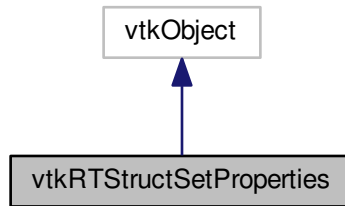
27.387 vtkRTStructSetProperties Class Reference

```
#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 *reframerefid, const char *roiname, const char *ROI←
GenerationAlgorithm, 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](#) (StructureSetLabel)
- [vtkGetStringMacro](#) (StructureSetName)
- [vtkGetStringMacro](#) (StructureSetDate)
- [vtkGetStringMacro](#) (StructureSetTime)
- [vtkGetStringMacro](#) (SOPInstanceUID)
- [vtkGetStringMacro](#) (StudyInstanceUID)
- [vtkGetStringMacro](#) (SeriesInstanceUID)

- [vtkGetStringMacro \(ReferenceSeriesInstanceUID\)](#)
- [vtkGetStringMacro \(ReferenceFrameOfReferenceUID\)](#)
- [vtkSetStringMacro \(StructureSetLabel\)](#)
- [vtkSetStringMacro \(StructureSetName\)](#)
- [vtkSetStringMacro \(StructureSetDate\)](#)
- [vtkSetStringMacro \(StructureSetTime\)](#)
- [vtkSetStringMacro \(SOPInstanceUID\)](#)
- [vtkSetStringMacro \(StudyInstanceUID\)](#)
- [vtkSetStringMacro \(SeriesInstanceUID\)](#)
- [vtkSetStringMacro \(ReferenceSeriesInstanceUID\)](#)
- [vtkSetStringMacro \(ReferenceFrameOfReferenceUID\)](#)
- [vtkTypeRevisionMacro \(vtkRTStructSetProperties, vtkObject\)](#)

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](#)

27.387.1 Detailed Description

Examples:

[GenerateRTSTRUCT.cxx](#).

27.387.2 Constructor & Destructor Documentation

27.387.2.1 [vtkRTStructSetProperties::vtkRTStructSetProperties \(\)](#) [protected]

27.387.2.2 [vtkRTStructSetProperties::~~vtkRTStructSetProperties \(\)](#) [protected]

27.387.3 Member Function Documentation

- 27.387.3.1 void vtkRTStructSetProperties::AddContourReferencedFrameOfReference (vtkIdType *pdnum*, const char * *classuid*, const char * *instanceuid*)
- 27.387.3.2 void vtkRTStructSetProperties::AddReferencedFrameOfReference (const char * *classuid*, const char * *instanceuid*)
- 27.387.3.3 void vtkRTStructSetProperties::AddStructureSetROI (int *roinumber*, const char * *refframerefid*, const char * *roiname*, const char * *ROIGenerationAlgorithm*, const char * *ROIDescription* = 0)
- 27.387.3.4 void vtkRTStructSetProperties::AddStructureSetROIObservation (int *refnumber*, int *observationnumber*, const char * *rtroiinterpretedtype*, const char * *roiinterpreter*, const char * *roiobservationlabel* = 0)
- 27.387.3.5 virtual void vtkRTStructSetProperties::Clear () [virtual]
- 27.387.3.6 virtual void vtkRTStructSetProperties::DeepCopy (vtkRTStructSetProperties * *p*) [virtual]
- 27.387.3.7 const char* vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceClassUID (vtkIdType *pdnum*, vtkIdType *id*)
- 27.387.3.8 const char* vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceInstanceUID (vtkIdType *pdnum*, vtkIdType *id*)
- 27.387.3.9 vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences ()
- 27.387.3.10 vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences (vtkIdType *pdnum*)
- 27.387.3.11 vtkIdType vtkRTStructSetProperties::GetNumberOfReferencedFrameOfReferences ()
- 27.387.3.12 vtkIdType vtkRTStructSetProperties::GetNumberOfStructureSetROIs ()
- 27.387.3.13 const char* vtkRTStructSetProperties::GetReferencedFrameOfReferenceClassUID (vtkIdType *id*)
- 27.387.3.14 const char* vtkRTStructSetProperties::GetReferencedFrameOfReferenceInstanceUID (vtkIdType *id*)
- 27.387.3.15 int vtkRTStructSetProperties::GetStructureSetObservationNumber (vtkIdType *id*)
- 27.387.3.16 const char* vtkRTStructSetProperties::GetStructureSetROIDescription (vtkIdType *id*)
- 27.387.3.17 const char* vtkRTStructSetProperties::GetStructureSetROIGenerationAlgorithm (vtkIdType)
- 27.387.3.18 const char* vtkRTStructSetProperties::GetStructureSetROIName (vtkIdType)
- 27.387.3.19 int vtkRTStructSetProperties::GetStructureSetROINumber (vtkIdType *id*)
- 27.387.3.20 const char* vtkRTStructSetProperties::GetStructureSetROIObservationLabel (vtkIdType *id*)
- 27.387.3.21 const char* vtkRTStructSetProperties::GetStructureSetROIRefFrameRefUID (vtkIdType)
- 27.387.3.22 const char* vtkRTStructSetProperties::GetStructureSetRTROIInterpretedType (vtkIdType *id*)

27.387.3.23 `static vtkRTStructSetProperties* vtkRTStructSetProperties::New ()` `[static]`

Examples:

[GenerateRTSTRUCT.cxx](#).

27.387.3.24 `void vtkRTStructSetProperties::PrintSelf (ostream & os, vtkIndent indent)`

27.387.3.25 `vtkRTStructSetProperties::vtkGetStringMacro (StructureSetLabel)`

27.387.3.26 `vtkRTStructSetProperties::vtkGetStringMacro (StructureSetName)`

27.387.3.27 `vtkRTStructSetProperties::vtkGetStringMacro (StructureSetDate)`

27.387.3.28 `vtkRTStructSetProperties::vtkGetStringMacro (StructureSetTime)`

27.387.3.29 `vtkRTStructSetProperties::vtkGetStringMacro (SOPInstanceUID)`

27.387.3.30 `vtkRTStructSetProperties::vtkGetStringMacro (StudyInstanceUID)`

27.387.3.31 `vtkRTStructSetProperties::vtkGetStringMacro (SeriesInstanceUID)`

27.387.3.32 `vtkRTStructSetProperties::vtkGetStringMacro (ReferenceSeriesInstanceUID)`

27.387.3.33 `vtkRTStructSetProperties::vtkGetStringMacro (ReferenceFrameOfReferenceUID)`

27.387.3.34 `vtkRTStructSetProperties::vtkSetStringMacro (StructureSetLabel)`

27.387.3.35 `vtkRTStructSetProperties::vtkSetStringMacro (StructureSetName)`

27.387.3.36 `vtkRTStructSetProperties::vtkSetStringMacro (StructureSetDate)`

27.387.3.37 `vtkRTStructSetProperties::vtkSetStringMacro (StructureSetTime)`

27.387.3.38 `vtkRTStructSetProperties::vtkSetStringMacro (SOPInstanceUID)`

27.387.3.39 `vtkRTStructSetProperties::vtkSetStringMacro (StudyInstanceUID)`

27.387.3.40 `vtkRTStructSetProperties::vtkSetStringMacro (SeriesInstanceUID)`

27.387.3.41 `vtkRTStructSetProperties::vtkSetStringMacro (ReferenceSeriesInstanceUID)`

27.387.3.42 `vtkRTStructSetProperties::vtkSetStringMacro (ReferenceFrameOfReferenceUID)`

27.387.3.43 `vtkRTStructSetProperties::vtkTypeRevisionMacro (vtkRTStructSetProperties , vtkObject)`

27.387.4 Member Data Documentation

27.387.4.1 `vtkRTStructSetPropertiesInternals* vtkRTStructSetProperties::Internals` `[protected]`

27.387.4.2 `char* vtkRTStructSetProperties::ReferenceFrameOfReferenceUID` `[protected]`

27.387.4.3 `char* vtkRTStructSetProperties::ReferenceSeriesInstanceUID` [protected]

27.387.4.4 `char* vtkRTStructSetProperties::SeriesInstanceUID` [protected]

27.387.4.5 `char* vtkRTStructSetProperties::SOPInstanceUID` [protected]

27.387.4.6 `char* vtkRTStructSetProperties::StructureSetDate` [protected]

27.387.4.7 `char* vtkRTStructSetProperties::StructureSetLabel` [protected]

27.387.4.8 `char* vtkRTStructSetProperties::StructureSetName` [protected]

27.387.4.9 `char* vtkRTStructSetProperties::StructureSetTime` [protected]

27.387.4.10 `char* vtkRTStructSetProperties::StudyInstanceUID` [protected]

The documentation for this class was generated from the following file:

- [vtkRTStructSetProperties.h](#)

27.388 gdcm::Waveform Class Reference

[Waveform](#) class.

```
#include <gdcmWaveform.h>
```

Public Member Functions

- [Waveform](#) ()

27.388.1 Detailed Description

[Waveform](#) class.

27.388.2 Constructor & Destructor Documentation

27.388.2.1 `gdcm::Waveform::Waveform ()` [inline]

The documentation for this class was generated from the following file:

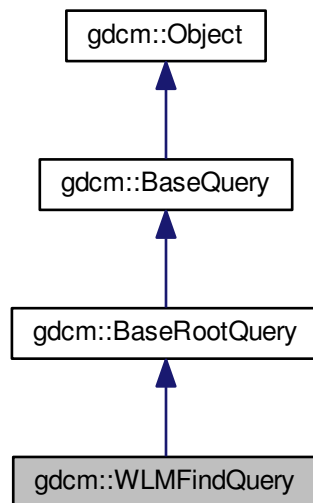
- [gdcmWaveform.h](#)

27.389 gdcm::WLMFindQuery Class Reference

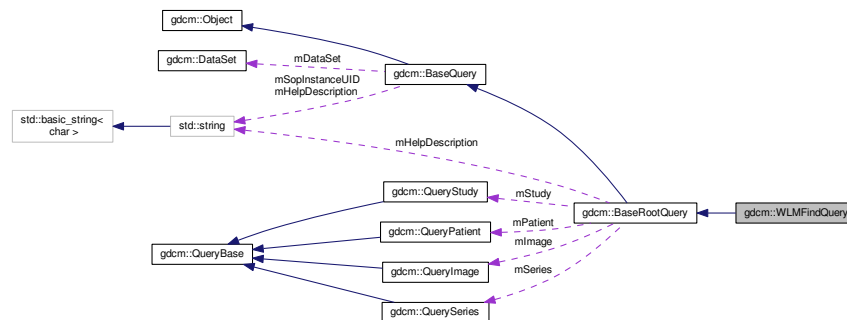
PatientRootQuery contains: the class which will produce a dataset for c-find with patient root.

```
#include <gdcmWLMFindQuery.h>
```

Inheritance diagram for gdcm::WLMFindQuery:



Collaboration diagram for gdcm::WLMFindQuery:



Public Member Functions

- [WLMFindQuery \(\)](#)
- [UIDs::TSName GetAbstractSyntaxUID \(\)](#) const
- `std::vector< Tag >` [GetTagListByLevel](#) (const [EQueryLevel](#) &inQueryLevel)
- void [InitializeDataSet](#) (const [EQueryLevel](#) &inQueryLevel)
- bool [ValidateQuery](#) (bool inStrict=true) const

Protected Member Functions

- [DataSet GetValidDataSet \(\)](#) const

Friends

- class [QueryFactory](#)

Additional Inherited Members

27.389.1 Detailed Description

PatientRootQuery contains: the class which will produce a dataset for c-find with patient root.

27.389.2 Constructor & Destructor Documentation

27.389.2.1 `gdcm::WLMFindQuery::WLMFindQuery ()`

27.389.3 Member Function Documentation

27.389.3.1 `UIDs::TSName gdcm::WLMFindQuery::GetAbstractSyntaxUID ()` const [virtual]

Implements [gdcm::BaseQuery](#).

27.389.3.2 `std::vector<Tag> gdcm::WLMFindQuery::GetTagListByLevel (const EQueryLevel & inQueryLevel)` [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](#).

27.389.3.3 `DataSet gdcm::WLMFindQuery::GetValidDataSet ()` const [protected]

27.389.3.4 `void gdcm::WLMFindQuery::InitializeDataSet (const EQueryLevel & inQueryLevel)` [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](#).

27.389.3.5 `bool gdcm::WLMFindQuery::ValidateQuery (bool inStrict=true)` const [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](#).

27.389.4 Friends And Related Function Documentation

27.389.4.1 friend class QueryFactory [friend]

The documentation for this class was generated from the following file:

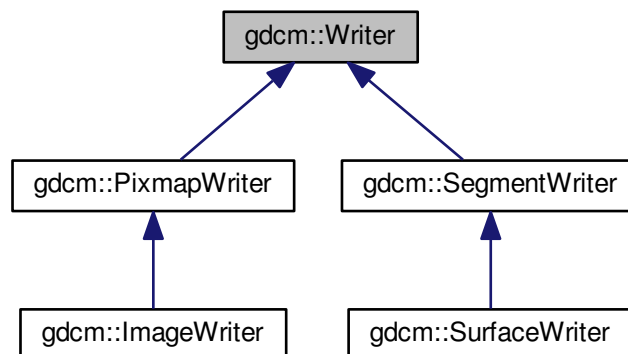
- [gdcmWLMFindQuery.h](#)

27.390 gdcm::Writer Class Reference

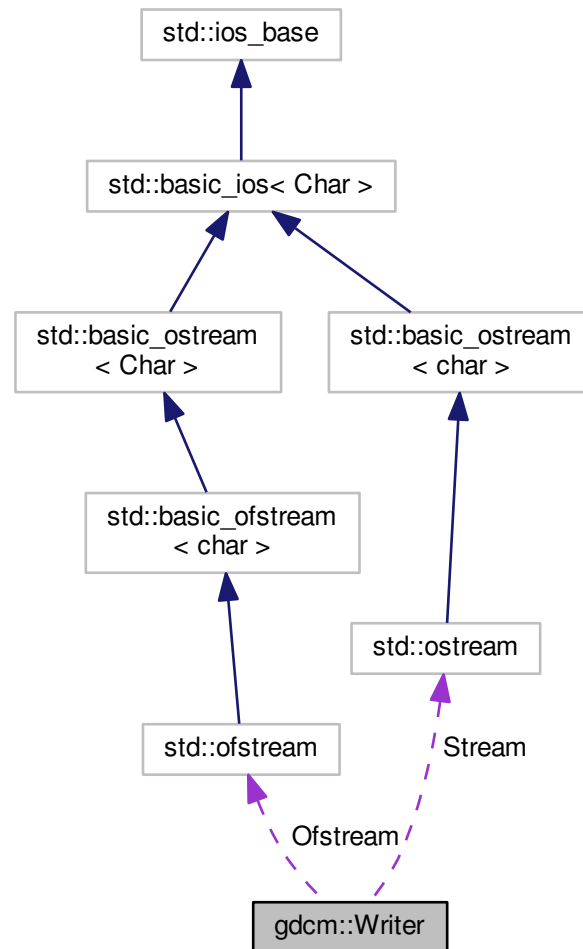
[Writer](#) ala DOM (Document [Object](#) Model) This class is a non-validating writer, it will only performs well- formedness check only.

```
#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

- std::ostream * [GetStreamPtr](#) () const
- void [SetWriteDataSetOnly](#) (bool b)

Protected Attributes

- std::ofstream * [Ofstream](#)
- std::ostream * [Stream](#)

Friends

- class [StreamImageWriter](#)

27.390.1 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](#), [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](#), [pmsct_rgb1.cxx](#), [QIDO-RS.cxx](#), [rle2img.cxx](#), and [StreamImageReaderTest.cxx](#).

27.390.2 Constructor & Destructor Documentation

27.390.2.1 `gdcm::Writer::Writer ()`

27.390.2.2 `virtual gdcm::Writer::~~Writer () [virtual]`

27.390.3 Member Function Documentation

27.390.3.1 `void gdcm::Writer::CheckFileMetaInformationOff () [inline]`

Examples:

[CreateFakeRTDOSE.cxx](#), [FixBrokenJ2K.cxx](#), and [HelloWorld.cxx](#).

27.390.3.2 `void gdcm::Writer::CheckFileMetaInformationOn () [inline]`

27.390.3.3 `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](#), [iU22tomultisc.cxx](#), [pmsct_rgb1.cxx](#), [QIDO-RS.cxx](#), [rle2img.cxx](#), and [StreamImageReaderTest.cxx](#).

27.390.3.4 `std::ostream* gdcm::Writer::GetStreamPtr () const [inline],[protected]`

27.390.3.5 `void gdcm::Writer::SetCheckFileMetaInformation (bool b) [inline]`

Undocumented function, do not use (= leave default)

Examples:

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), and [PatchFile.cxx](#).

27.390.3.6 `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](#), [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](#).

27.390.3.7 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](#), [CreateJIPIDataSet.cxx](#), [csa2img.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](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [MergeTwoFiles.cxx](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [QIDO-RS.cxx](#), and [rle2img.cxx](#).

27.390.3.8 void gdcm::Writer::SetStream (std::ostream & *output_stream*) [inline]

Set user ostream buffer.

27.390.3.9 void gdcm::Writer::SetWriteDataSetOnly (bool *b*) [inline], [protected]

27.390.3.10 virtual bool gdcm::Writer::Write () [virtual]

Main function to tell the writer to write.

Reimplemented in [gdcm::PixmapWriter](#), [gdcm::ImageWriter](#), [gdcm::SurfaceWriter](#), and [gdcm::SegmentWriter](#).

Examples:

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CreateFakeRTDOSE.cxx](#), [CreateJIPIDataSet.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](#), [pmsct_rgb1.cxx](#), [QIDO-RS.cxx](#), and [rle2img.cxx](#).

27.390.4 Friends And Related Function Documentation

27.390.4.1 friend class StreamImageWriter [friend]

27.390.5 Member Data Documentation

27.390.5.1 std::ofstream* gdcm::Writer::Ofstream [protected]

27.390.5.2 std::ostream* gdcm::Writer::Stream [protected]

The documentation for this class was generated from the following file:

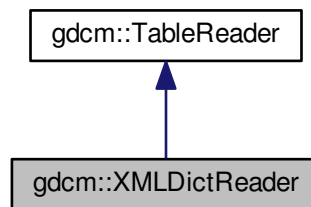
- [gdcmWriter.h](#)

27.391 gdcM::XMLDictReader Class Reference

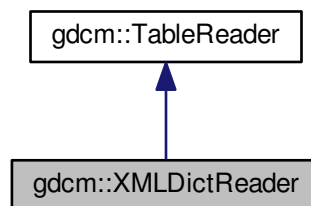
Class for representing a [XMLDictReader](#).

```
#include <gdcMXMLDictReader.h>
```

Inheritance diagram for gdcM::XMLDictReader:



Collaboration diagram for gdcM::XMLDictReader:



Public Member Functions

- [XMLDictReader](#) ()
- [~XMLDictReader](#) ()
- void [CharacterDataHandler](#) (const char *data, int length)
- void [EndElement](#) (const char *name)
- const [Dict](#) & [GetDict](#) ()
- void [StartElement](#) (const char *name, const char **atts)

Protected Member Functions

- void [HandleDescription](#) (const char **atts)
- void [HandleEntry](#) (const char **atts)

27.391.1 Detailed Description

Class for representing a [XMLDictReader](#).

Note

bla Will read the DICOMV3.xml file

27.391.2 Constructor & Destructor Documentation

27.391.2.1 `gdcm::XMLDictReader::XMLDictReader ()`

27.391.2.2 `gdcm::XMLDictReader::~XMLDictReader ()` `[inline]`

27.391.3 Member Function Documentation

27.391.3.1 `void gdcm::XMLDictReader::CharacterDataHandler (const char * data, int length)` `[virtual]`

Reimplemented from [gdcm::TableReader](#).

27.391.3.2 `void gdcm::XMLDictReader::EndElement (const char * name)` `[virtual]`

Reimplemented from [gdcm::TableReader](#).

27.391.3.3 `const Dict& gdcm::XMLDictReader::GetDict ()` `[inline]`

27.391.3.4 `void gdcm::XMLDictReader::HandleDescription (const char ** atts)` `[protected]`

27.391.3.5 `void gdcm::XMLDictReader::HandleEntry (const char ** atts)` `[protected]`

27.391.3.6 `void gdcm::XMLDictReader::StartElement (const char * name, const char ** atts)` `[virtual]`

Reimplemented from [gdcm::TableReader](#).

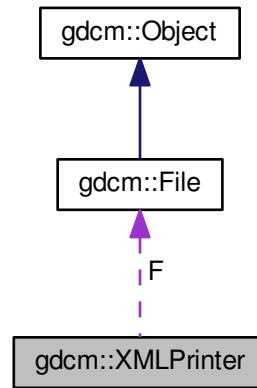
The documentation for this class was generated from the following file:

- [gdcmXMLDictReader.h](#)

27.392 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`

27.392.1 Member Enumeration Documentation

27.392.1.1 enum gdcm::XMLPrinter::PrintStyles

Enumerator

OnlyUUID

LOADBULKDATA

27.392.2 Constructor & Destructor Documentation

27.392.2.1 gdcm::XMLPrinter::XMLPrinter ()

27.392.2.2 virtual gdcm::XMLPrinter::~~XMLPrinter () [virtual]

27.392.3 Member Function Documentation

27.392.3.1 PrintStyles gdcm::XMLPrinter::GetPrintStyle () const [inline]

27.392.3.2 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

27.392.3.3 void gdcm::XMLPrinter::Print (std::ostream & *os*)

27.392.3.4 VR gdcm::XMLPrinter::PrintDataElement (std::ostream & *os*, const Dicts & *dicts*, const DataSet & *ds*, const DataElement & *de*, const TransferSyntax & *ts*) [protected]

27.392.3.5 void gdcm::XMLPrinter::PrintDataSet (const DataSet & *ds*, const TransferSyntax & *ts*, std::ostream & *os*)

27.392.3.6 void gdcm::XMLPrinter::PrintSQ (const SequenceOfItems * *sqi*, const TransferSyntax & *ts*, std::ostream & *os*) [protected]

27.392.3.7 void gdcm::XMLPrinter::SetFile (File const & *f*) [inline]

27.392.3.8 void gdcm::XMLPrinter::SetStyle (PrintStyles *ps*) [inline]

27.392.4 Member Data Documentation

27.392.4.1 const File* gdcm::XMLPrinter::F [protected]

27.392.4.2 PrintStyles gdcm::XMLPrinter::PrintStyle [protected]

The documentation for this class was generated from the following file:

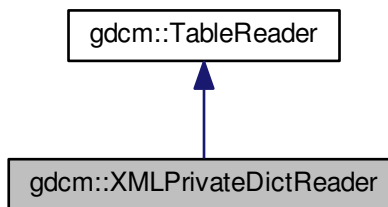
- [gdcmXMLPrinter.h](#)

27.393 gdcM::XMLPrivateDictReader Class Reference

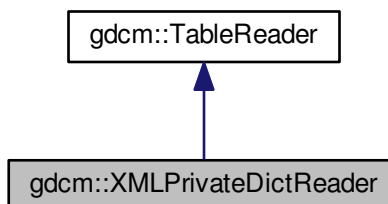
Class for representing a [XMLPrivateDictReader](#).

```
#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)

27.393.1 Detailed Description

Class for representing a [XMLPrivateDictReader](#).

Note

bla Will read the Private.xml file

27.393.2 Constructor & Destructor Documentation

27.393.2.1 `gdcm::XMLPrivateDictReader::XMLPrivateDictReader ()`

27.393.2.2 `gdcm::XMLPrivateDictReader::~XMLPrivateDictReader ()` `[inline]`

27.393.3 Member Function Documentation

27.393.3.1 `void gdcm::XMLPrivateDictReader::CharacterDataHandler (const char * data, int length)` `[virtual]`

Reimplemented from [gdcm::TableReader](#).

27.393.3.2 `void gdcm::XMLPrivateDictReader::EndElement (const char * name)` `[virtual]`

Reimplemented from [gdcm::TableReader](#).

27.393.3.3 `const PrivateDict& gdcm::XMLPrivateDictReader::GetPrivateDict ()` `[inline]`

27.393.3.4 `void gdcm::XMLPrivateDictReader::HandleDescription (const char ** atts)` `[protected]`

27.393.3.5 `void gdcm::XMLPrivateDictReader::HandleEntry (const char ** atts)` `[protected]`

27.393.3.6 `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 28

File Documentation

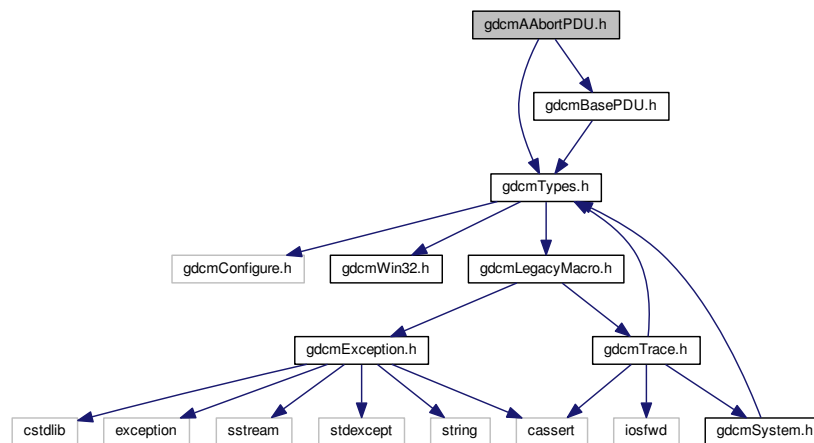
28.1 gdc2pnm.dox File Reference

28.2 gdc2vtk.dox File Reference

28.3 gdcmAAbortPDU.h File Reference

```
#include "gdcTypes.h"  
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAAbortPDU.h:



Classes

- class `gdcm::network::AAbortPDU`

AAbortPDU Table 9-26 A-ABORT PDU FIELDS.

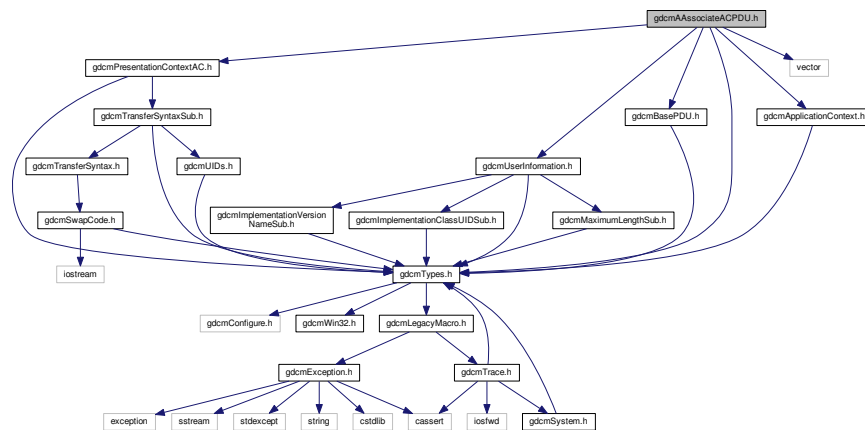
Namespaces

- [gdcm](#)
- [gdcm::network](#)

28.4 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](#) Table 9-17 ASSOCIATE-AC PDU fields.

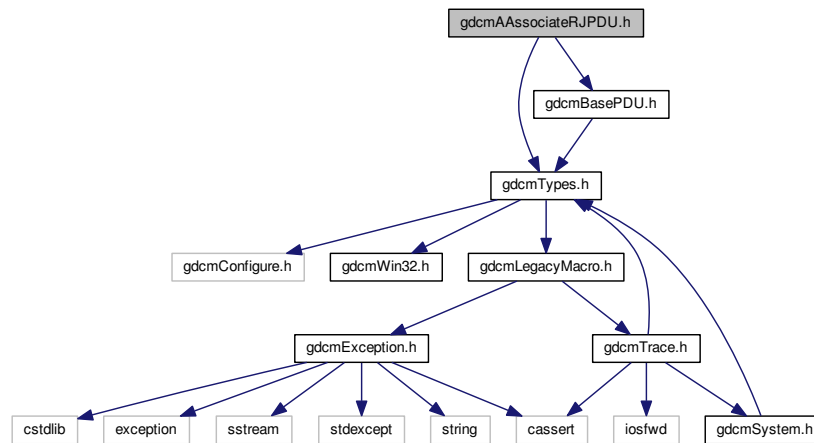
Namespaces

- [gdcm](#)
- [gdcm::network](#)

28.5 gdcmAAssociateRJPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAAssociateRJPDU.h:



Classes

- class `gdcm::network::AAssociateRJPDU`

[AAssociateRJPDU](#) Table 9-21 ASSOCIATE-RJ PDU FIELDS.

Namespaces

- `gdcm`
- `gdcm::network`

28.6 gdcmAAssociateRQPDU.h File Reference

```

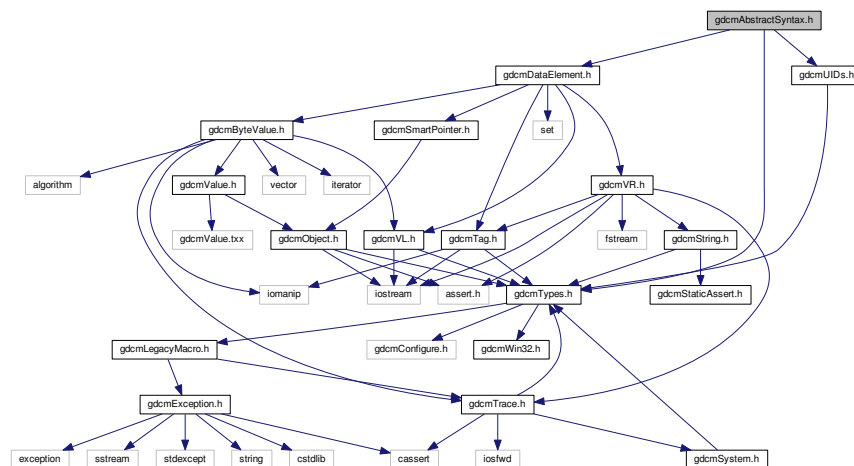
#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmApplicationContext.h"
#include "gdcmPresentationContextRQ.h"
#include "gdcmUserInfo.h"
#include "gdcmBasePDU.h"

```

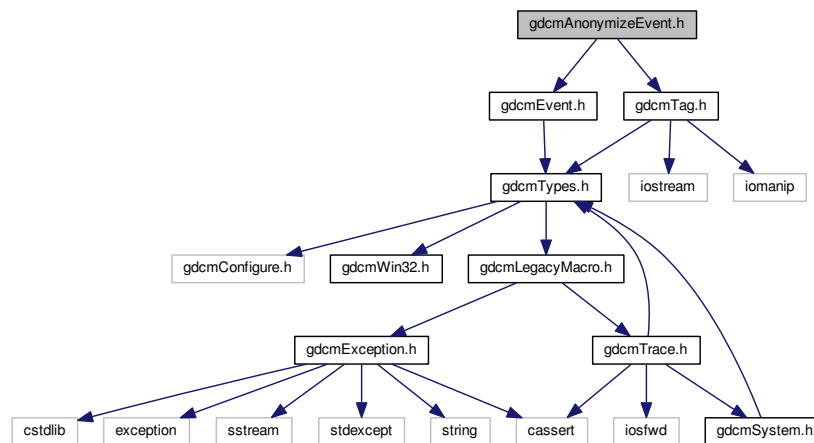
- class `gdcn::network::AAssociateRQPDU`
AAssociateRQPDU Table 9-11 ASSOCIATE-RQ PDU fields.

- gdc
- gdc::network

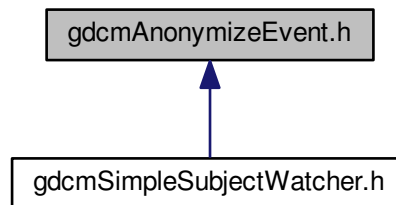
```
#include "gdcmTypes.h"
#include "gdcmUIDs.h"
#include "gdcmDataElement.h"
Include dependency graph for gdcmAbstractSyntax.h:
```



Include dependency graph for `gdcmAnonymizeEvent.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::AnonymizeEvent](#)
AnonymizeEvent Special type of event triggered during the Anonymization process.

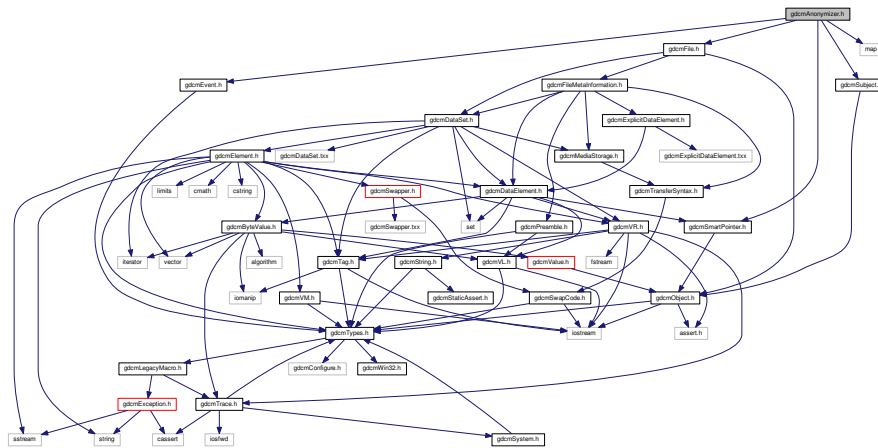
Namespaces

- [gdcm](#)

28.10 gdcmAnonymizer.h File Reference

```
#include "gdcmFile.h"
```

```
#include "gdcmSubject.h"
#include "gdcmEvent.h"
#include "gdcmSmartPointer.h"
#include <map>
Include dependency graph for gdcmAnonymizer.h:
```



Classes

- class `gdcm::Anonymizer`

Anonymizer This class is a multi purpose anonymizer. It can work in 2 mode:

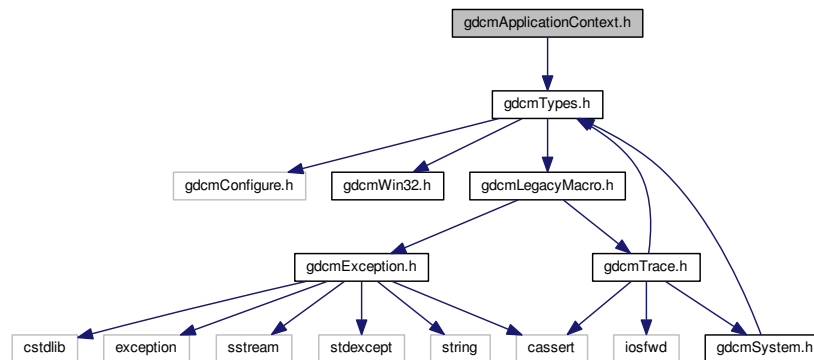
Namespaces

- **gdcm**

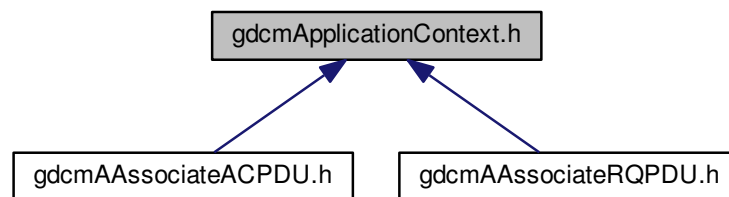
28.11 gdcmApplicationContext.h File Reference

```
#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 Table 9-12 APPLICATION CONTEXT ITEM FIELDS.

Namespaces

- `gdcm`
- `gdcm::network`

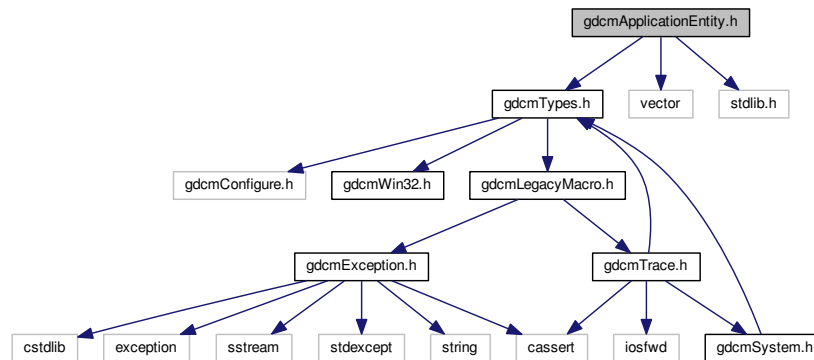
28.12 gdcmApplicationEntity.h File Reference

```

#include "gdcmTypes.h"
#include <vector>
#include <stdlib.h>

```


Include dependency graph for gdcmApplicationEntity.h:



Classes

- class [gdcm::ApplicationEntity](#)

ApplicationEntity.

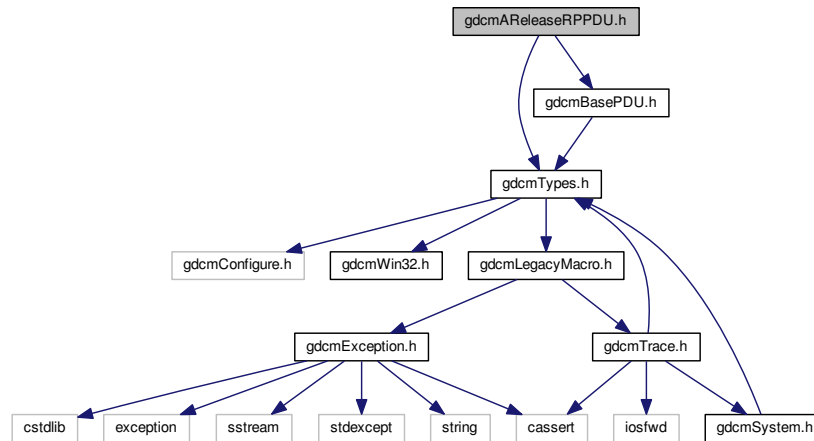
Namespaces

- [gdcm](#)

28.13 gdcmAReleaseRPPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for `gdcmAReleaseRPPDU.h`:



Classes

- class `gdcm::network::AReleaseRPPDU`

AReleaseRPPDU Table 9-25 A-RELEASE-RP PDU fields.

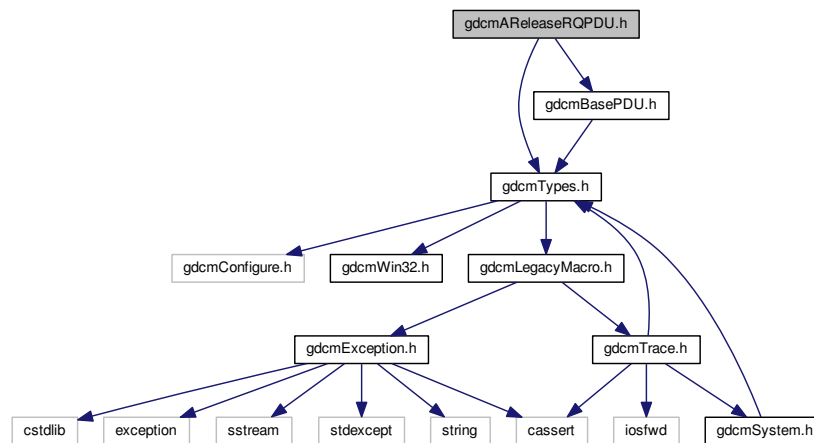
Namespaces

- `gdcm`
- `gdcm::network`

28.14 gdcmAReleaseRQPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcMAReleaseRQPDU.h:



Classes

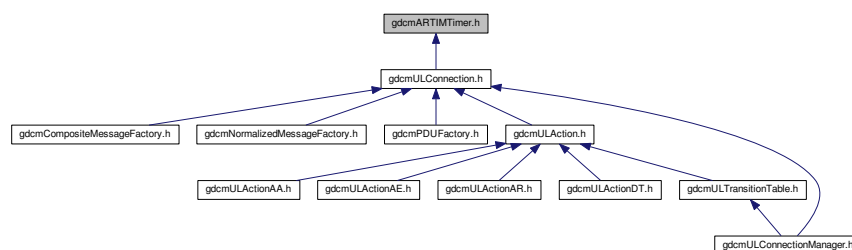
- class [gdcM::network::AReleaseRQPDU](#)
AReleaseRQPDU Table 9-24 A-RELEASE-RQ PDU FIELDS.

Namespaces

- [gdcM](#)
- [gdcM::network](#)

28.15 gdcMARTIMTimer.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [gdcM::network::ARTIMTimer](#)

ARTIMTimer This file contains the code for the ARTIM timer.

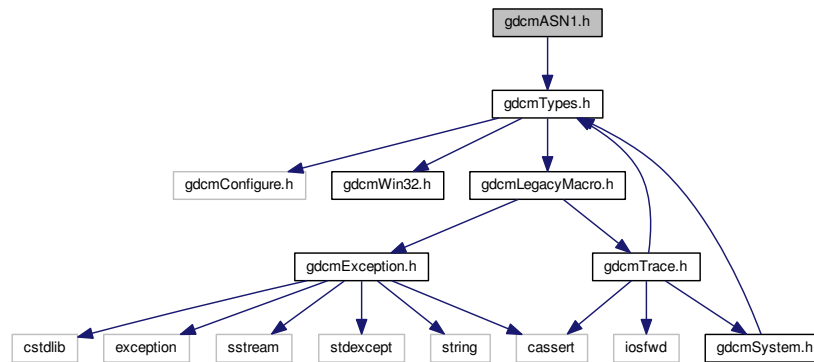
Namespaces

- [gdcm](#)
- [gdcm::network](#)

28.16 gdcmASN1.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmASN1.h:



Classes

- class [gdcm::ASN1](#)
Class for [ASN1](#).

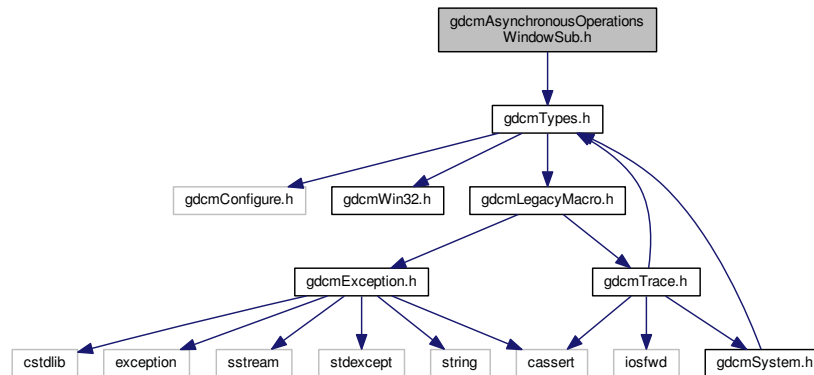
Namespaces

- [gdcm](#)

28.17 gdcmAsynchronousOperationsWindowSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmAsynchronousOperationsWindowSub.h:



Classes

- class `gdcm::network::AsynchronousOperationsWindowSub`

AsynchronousOperationsWindowSub PS 3.7 Table D.3-7 ASYNCHRONOUS OPERATIONS WINDOW SUB-ITEM FILE↔
LDS (A-ASSOCIATE-RQ)

Namespaces

- `gdcm`
- `gdcm::network`

28.18 gdcmAttribute.h File Reference

```

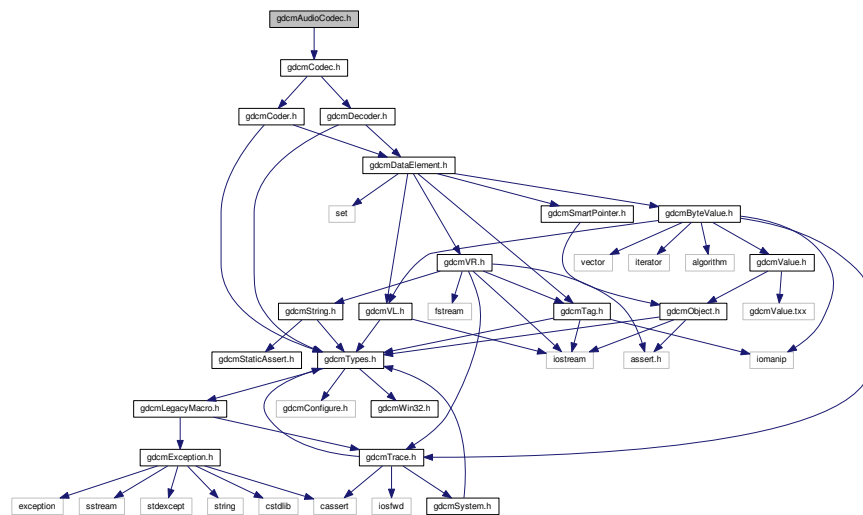
#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmTagToType.h"
#include "gdcmVM.h"
#include "gdcmElement.h"
#include "gdcmDataSet.h"
#include "gdcmStaticAssert.h"
#include <string>
#include <vector>
#include <sstream>

```


28.19 gdcmAudioCodec.h File Reference

```
#include "gdcmCodec.h"
```

Include dependency graph for gdcmAudioCodec.h:



Classes

- class [gdcm::AudioCodec](#)

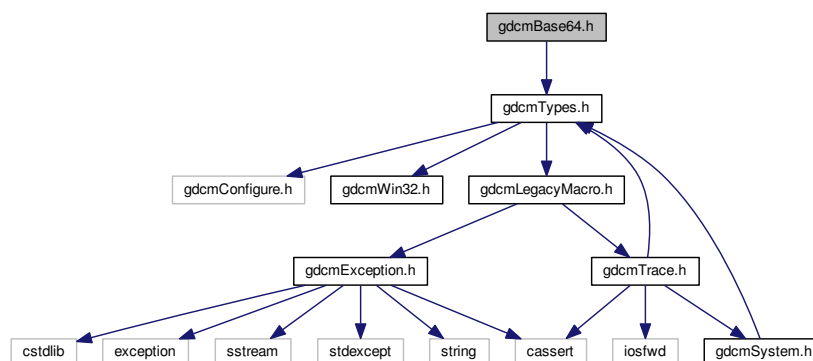
AudioCodec.

Namespaces

- [gdcm](#)

28.20 gdcmBase64.h File Reference

```
#include "gdcmTypes.h"
```



[illegible]

```

graph BT
    gdcBase[gdcmBaseCompositeMessage.h]
    gdcEcho[gdcmCEchoMessages.h]
    gdcFind[gdcmCFindMessages.h]
    gdcMove[gdcmCMoveMessages.h]
    gdcStore[gdcmCStoreMessages.h]
    gdcEcho --> gdcBase
    gdcFind --> gdcBase
    gdcMove --> gdcBase
    gdcStore --> gdcBase

```

- class `gdcm::network::BaseCompositeMessage`

- `gdcm`
- `gdcm::network`

```
#include "gdcmPresentationDataValue.h"
#include "gdcmBaseQuery.h"
#include <vector>
```

```

graph TD
    Base[gdc:BaseNormalizedMessage.h]
    NA[gdc:NAActionMessages.h]
    NC[gdc:NCCreateMessages.h]
    ND[gdc:NCDeleteMessages.h]
    NE[gdc:NEventReportMessages.h]
    NG[gdc:NGetMessages.h]
    NS[gdc:NSetMessages.h]
    NA --> Base
    NC --> Base
    ND --> Base
    NE --> Base
    NG --> Base
    NS --> Base
  
```

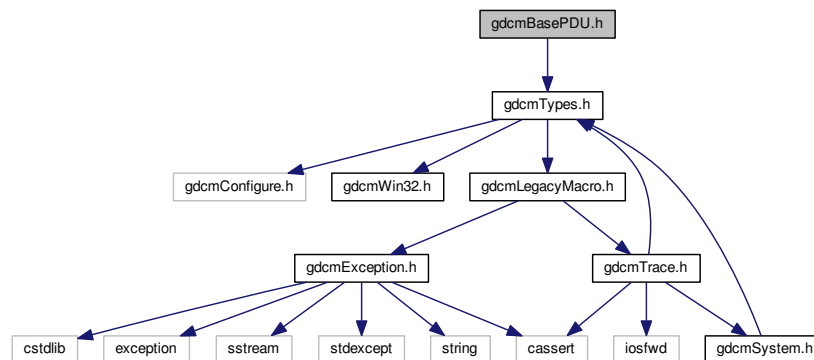
- class [gdcm::network::BaseNormalizedMessage](#)

[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.

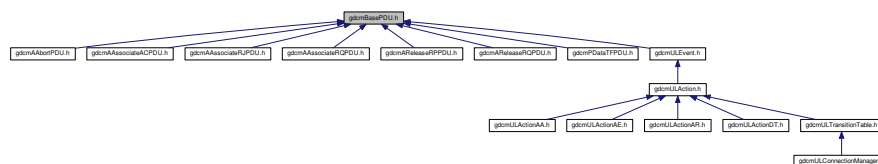
- `gdc`
- `gdc::network`

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmBasePDU.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::network::BasePDU](#)
BasePDU base class for PDUs.

Namespaces

- [gdcm](#)
- [gdcm::network](#)

28.24 gdcmBaseQuery.h File Reference

```

#include "gdcmDataSet.h"
#include "gdcmUIDs.h"
#include "gdcmObject.h"

```

[illegible][illegible]

- class `gdc::BaseQuery`
BaseQuery contains: a baseclass which will produce a dataset for all dimse messages.

- **gdcm**

- enum `gdcm::ENQueryType` {
`gdcm::eCreateMMPS` = 0,
`gdcm::eSetMMPS` }

```
#include "gdcmDataSet.h"
#include "gdcmUIDs.h"
#include "gdcmBaseQuery.h"
#include "gdcmQueryPatient.h"
#include "gdcmQueryStudy.h"
#include "gdcmQuerySeries.h"
#include "gdcmQueryImage.h"
```

- class `gdcm::BaseRootQuery`

Namespaces

- **gdcm**

- enum `gdcmm::EQueryLevel` {
 `gdcmm::ePatient` = 0,
 `gdcmm::eStudy` = 1,
 `gdcmm::eSeries` = 2,
 `gdcmm::eImage` = 3 }
- enum `gdcmm::EQueryType` {
 `gdcmm::eFind` = 0,
 `gdcmm::eMove`,
 `gdcmm::eWLMFind` }

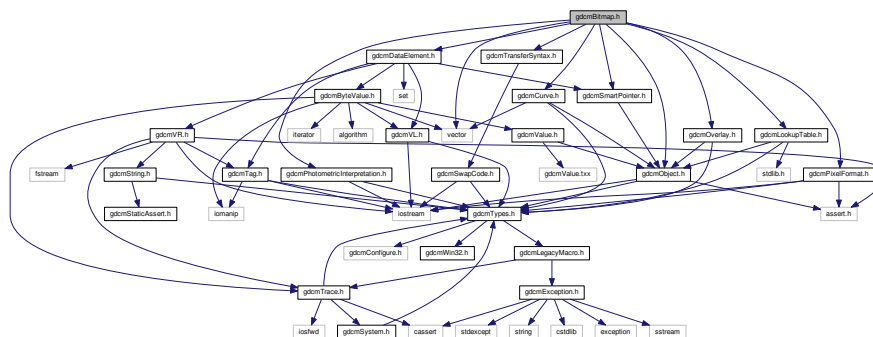
Functions

- `std::ostream & gdcmm::operator<< (std::ostream &os, const BasicOffsetTable &val)`

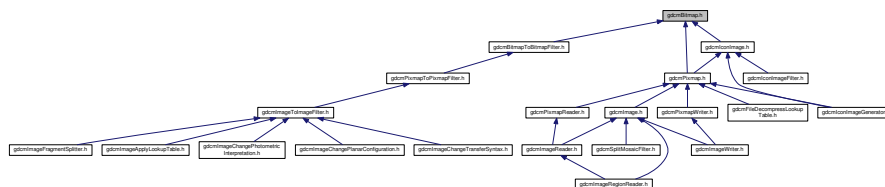
28.27 gdcmBitmap.h File Reference

```
#include "gdcmObject.h"
#include "gdcmCurve.h"
#include "gdcmDataElement.h"
#include "gdcmLookupTable.h"
#include "gdcmOverlay.h"
#include "gdcmPhotometricInterpretation.h"
#include "gdcmPixelFormat.h"
#include "gdcmSmartPointer.h"
#include "gdcmTransferSyntax.h"
#include <vector>
```

Include dependency graph for gdcMBitmap.h:



This graph shows which files directly or indirectly include this file:



Classes

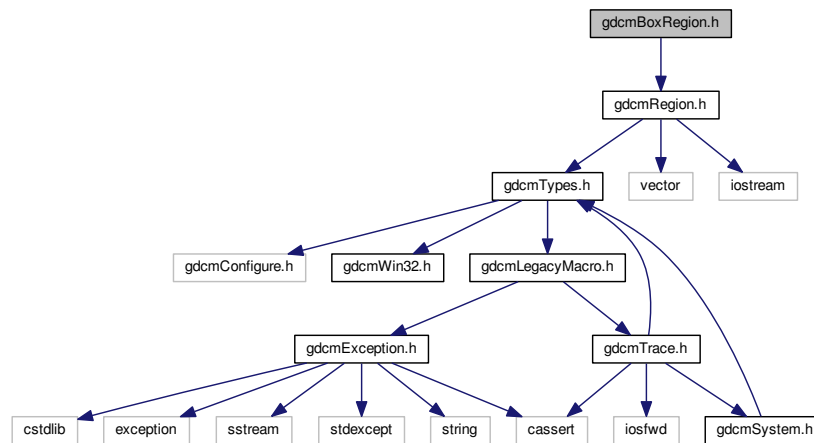
- class `gdcm::Bitmap`

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)

28.29 gdcmBoxRegion.h File Reference

```
#include "gdcmRegion.h"
```

Include dependency graph for gdcmBoxRegion.h:



Classes

- class [gdcm::BoxRegion](#)

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)

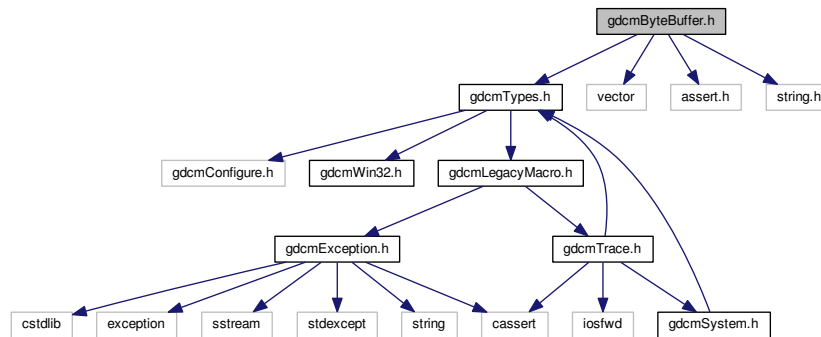
Namespaces

- [gdcm](#)

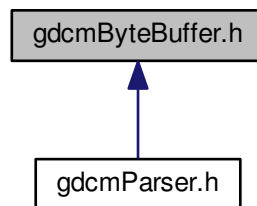
28.30 gdcmByteBuffer.h File Reference

```
#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`

28.31 gdcmByteSwap.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmSwapCode.h"
#include "gdcmByteSwap.txx"

```

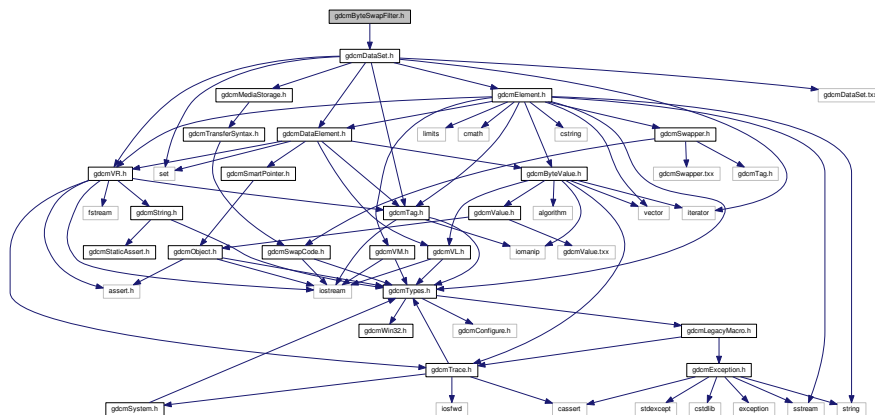
```

graph TD
    gdcmByteSwap_h[gdcmByteSwap.h] --> gdcmByteSwap_h
    gdcmByteSwap_h --> gdcmSwapCode_h[gdcmSwapCode.h]
    gdcmByteSwap_h --> gdcmByteSwap_txx[gdcmByteSwap.txx]
    gdcmByteSwap_h --> stdlib_h[stdlib.h]
    gdcmSwapCode_h --> gdcmTypes_h[gdcmTypes.h]
    gdcmByteSwap_txx --> gdcmTypes_h
    gdcmByteSwap_txx --> iostream
    stdlib_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
    gdcmException_h --> exception
    gdcmException_h --> sstream
    gdcmException_h --> stdexcept
    gdcmException_h --> string
    gdcmException_h --> cassert
    gdcmException_h --> gdcmSystem_h[gdcmSystem.h]
    gdcmTrace_h --> iostream
    gdcmTrace_h --> gdcmSystem_h
  
```

- `class gdcm::ByteSwap< T >`
`ByteSwap.`

- gdcm

```
#include "gdcmDataSet.h"
Include dependency graph for gdcmByteSwapFilter.h:
```



Classes

- class [gdcm::ByteSwapFilter](#)

ByteSwapFilter In place byte-swapping of a dataset *FIXME: FL status ??*

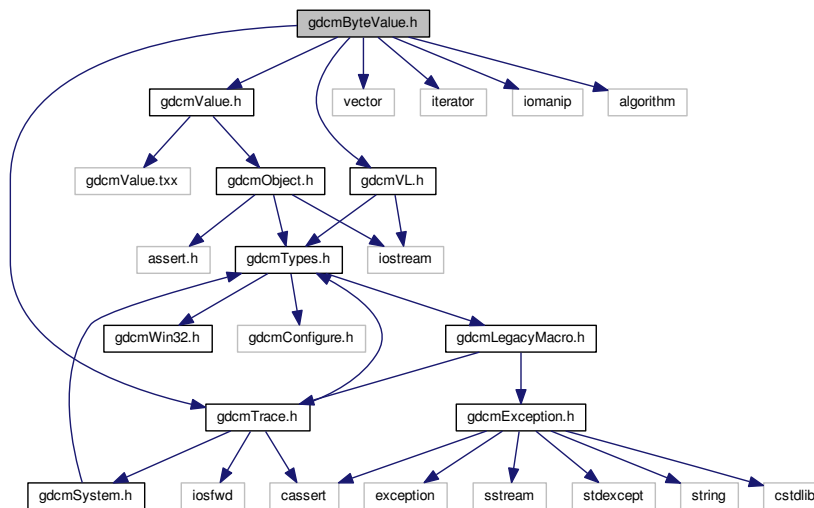
Namespaces

- [gdcm](#)

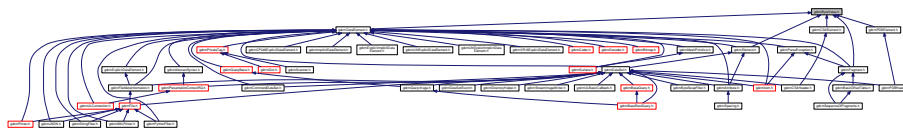
28.33 gdcmByteValue.h File Reference

```
#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)

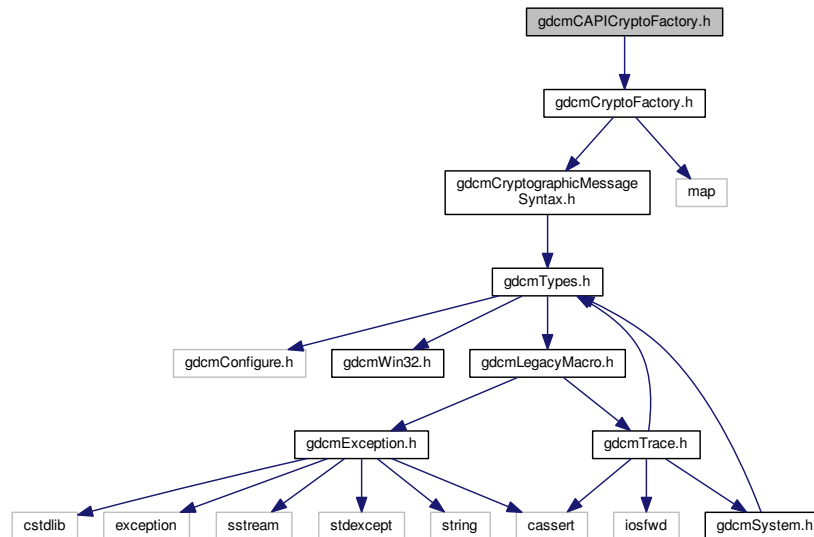
Namespaces

- [gdcm](#)

28.34 gdcmCAPICryptoFactory.h File Reference

```
#include "gdcmCryptoFactory.h"
```

Include dependency graph for gdcmCAPICryptoFactory.h:



Classes

- class [gdcm::CAPICryptoFactory](#)

Namespaces

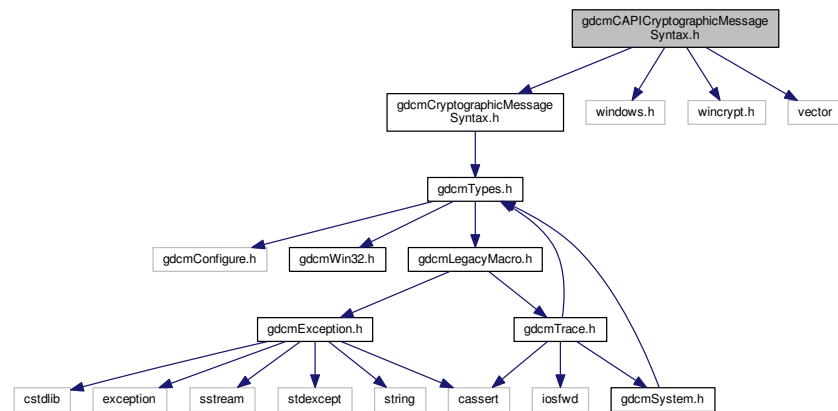
- [gdcm](#)

28.35 gdcmCAPICryptographicMessageSyntax.h File Reference

```
#include "gdcmCryptographicMessageSyntax.h"
```

```
#include <windows.h>
#include <wincrypt.h>
#include <vector>
```

Include dependency graph for `gdcmlCAPICryptographicMessageSyntax.h`:



Classes

- class [gdcml::CAPICryptographicMessageSyntax](#)

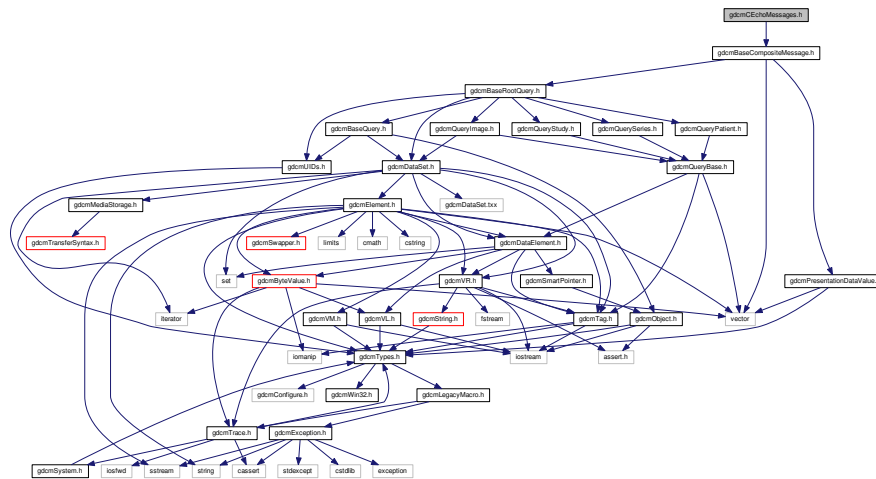
Namespaces

- [gdcml](#)

28.36 gdcmlCEchoMessages.h File Reference

```
#include "gdcmlBaseCompositeMessage.h"
```

Include dependency graph for gdcmCEchoMessages.h:



Classes

- class [gdcm::network::CEchoRQ](#)

[CEchoRQ](#) this file defines the messages for the cecho action.

- class [gdcm::network::CEchoRSP](#)

[CEchoRSP](#) this file defines the messages for the cecho action.

Namespaces

- [gdcm](#)
- [gdcm::network](#)

28.37 gdcmCFindMessages.h File Reference

```
#include "gdcmBaseCompositeMessage.h"
#include "gdcmBaseRootQuery.h"
```


- class `gdcm::network::CMoveCancelRq`
- class `gdcm::network::CMoveRQ`
`CMoveRQ` this file defines the messages for the `cmove` action.
- class `gdcm::network::CMoveRSP`
`CMoveRSP` this file defines the messages for the `cmove` action.

- `gdcm`
- `gdcm::network`

```
#include "gdcmCoder.h"
#include "gdcmDecoder.h"
```

```

graph TD
    gstormCodec_h[gstormCodec.h] --> gstormAudioCodec_h[gstormAudioCodec.h]
    gstormCodec_h --> gstormImageCodec_h[gstormImageCodec.h]
    gstormCodec_h --> gstormPOFCodec_h[gstormPOFCodec.h]
    gstormImageCodec_h --> gstormDetailCodingCodec_h[gstormDetailCodingCodec.h]
    gstormImageCodec_h --> gstormJPEG2000Codec_h[gstormJPEG2000Codec.h]
    gstormImageCodec_h --> gstormJPEGLSCodec_h[gstormJPEGLSCodec.h]
    gstormImageCodec_h --> gstormAKADCodec_h[gstormAKADCodec.h]
    gstormImageCodec_h --> gstormGstCodec_h[gstormGstCodec.h]
    gstormImageCodec_h --> gstormPBMCodec_h[gstormPBMCodec.h]
    gstormImageCodec_h --> gstormPNGCodec_h[gstormPNGCodec.h]
    gstormImageCodec_h --> gstormRAWCodec_h[gstormRAWCodec.h]
    gstormImageCodec_h --> gstormRLECodec_h[gstormRLECodec.h]
    gstormJPEG2000Codec_h --> gstormJPEG12Codec_h[gstormJPEG12Codec.h]
    gstormJPEG2000Codec_h --> gstormJPEG8Codec_h[gstormJPEG8Codec.h]
    gstormJPEG8Codec_h --> gstormJPEG8Codec_h
  
```

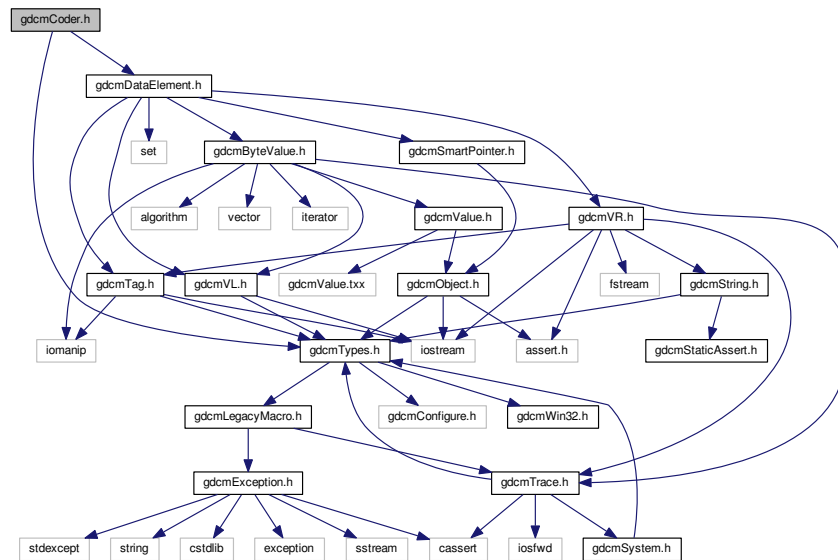
- class `gdcm::Codec`

Namespaces

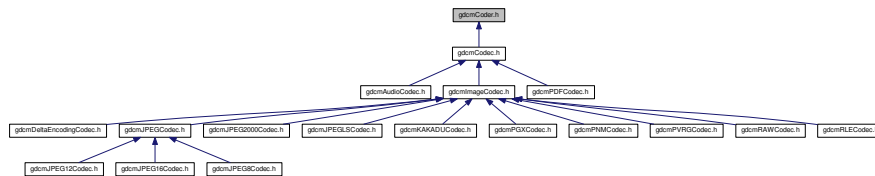
- **gdcm**

```
#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.

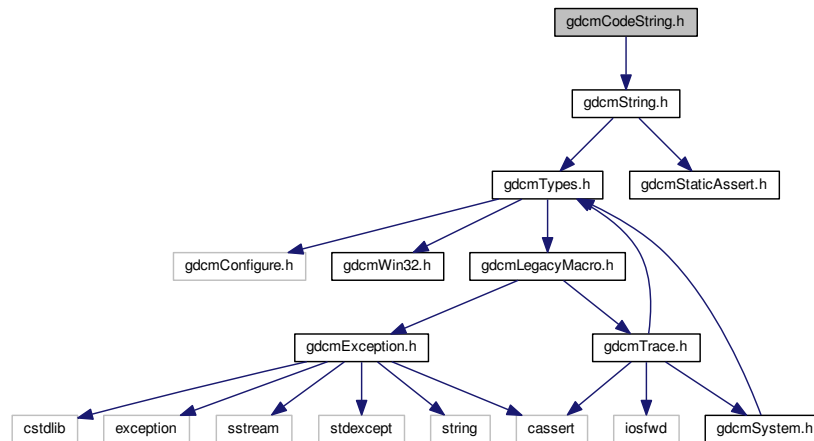
Namespaces

- [gdcm](#)

28.41 gdcmCodeString.h File Reference

```
#include "gdcmString.h"
```

Include dependency graph for `gdcmCodeString.h`:



Classes

- class `gdcm::CodeString`

`CodeString` This is an implementation of DICOM [VR](#): CS The ctor will properly Trim so that operator== is correct.

Namespaces

- `gdcm`

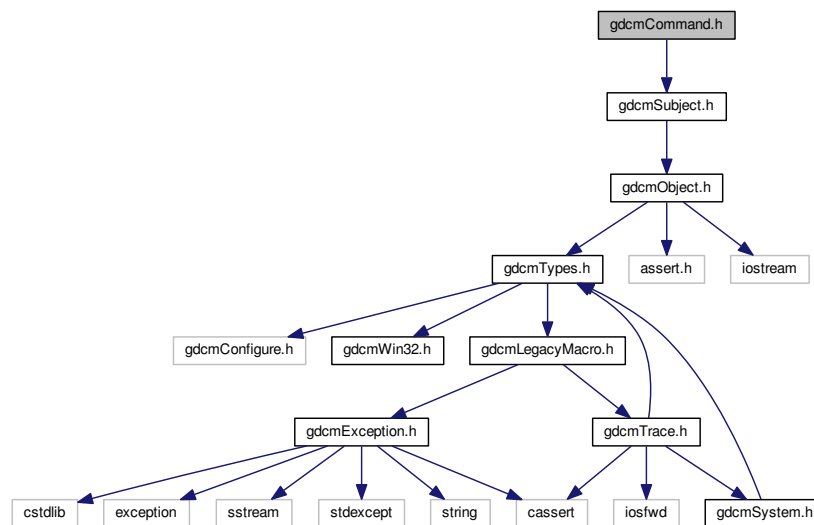
Functions

- 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)

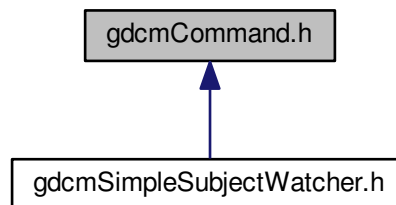
28.42 gdcmCommand.h File Reference

```
#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.

- class `gdcm::network::CompositeMessageFactory`

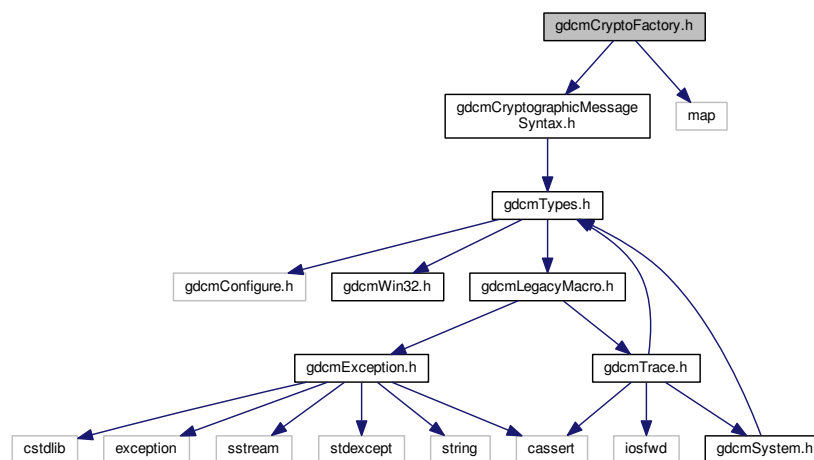
Namespaces

- `gdc`
- `gdc::network`

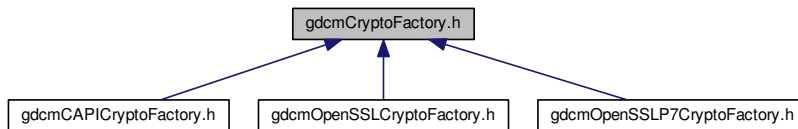
```
#include "gdcDirectory.h"
#include "gdcmBaseRootQuery.h"
#include <vector>
#include <string>
```


28.48 gdcmCP246ExplicitDataElement.h File Reference

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.

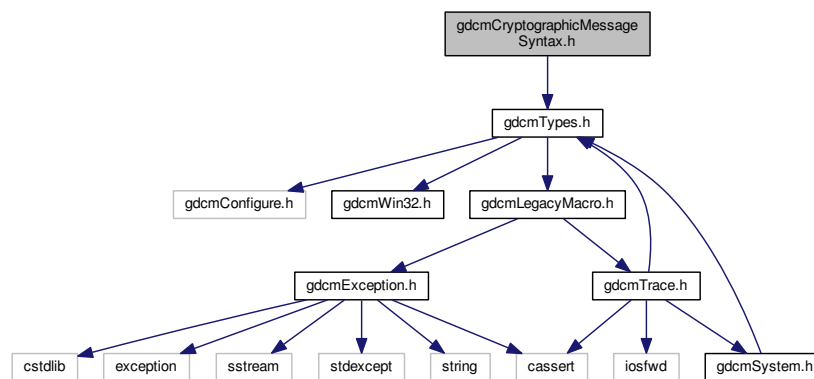
Namespaces

- [gdcm](#)

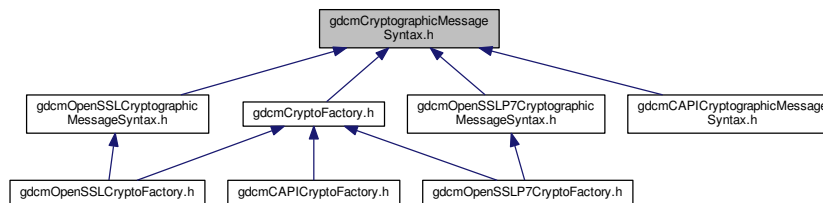
28.50 gdcmCryptographicMessageSyntax.h File Reference

```
#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](#)

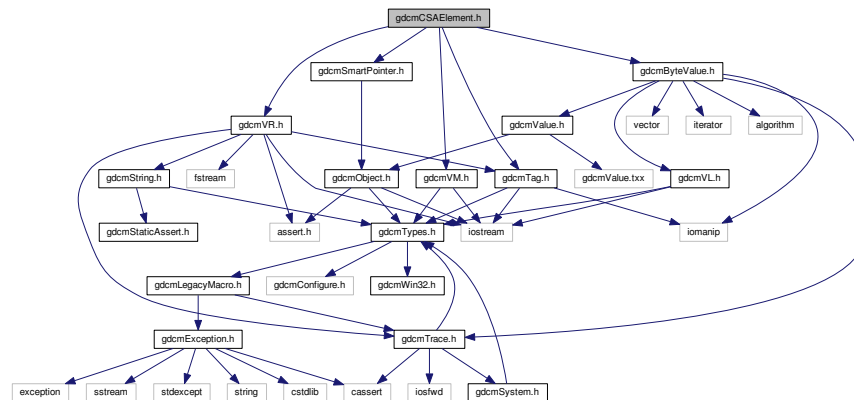
28.51 gdcmCSAElement.h File Reference

```

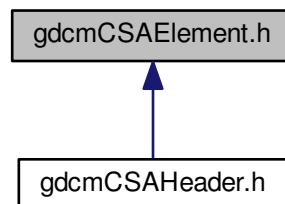
#include "gdcmTag.h"
#include "gdcmVM.h"
#include "gdcmVR.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"

```

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

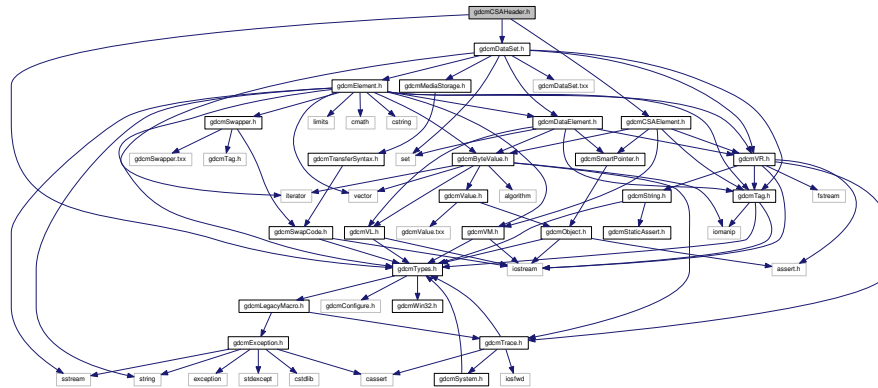
- `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const CSAElement &val)`

28.52 gdcmCSAHeader.h File Reference

```
#include "gdcTypes.h"
#include "gdcDataSet.h"
#include "gdcCSAElement.h"
Include dependency graph for gdcCSAHeader.h:
```



Classes

- class `gdcm::CSAHeader`
Class for CSAHeader.

Namespaces

- **gdcm**

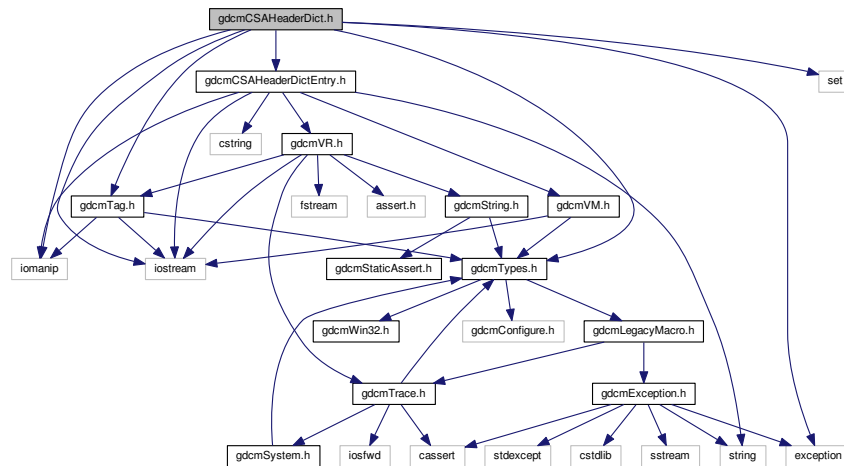
Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeader &d)`

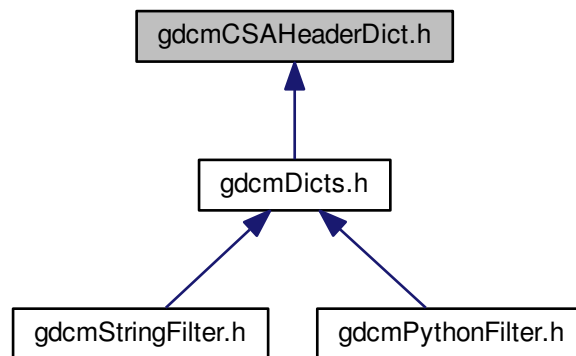
28.53 gdcmCSAHeaderDict.h File Reference

```
#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`

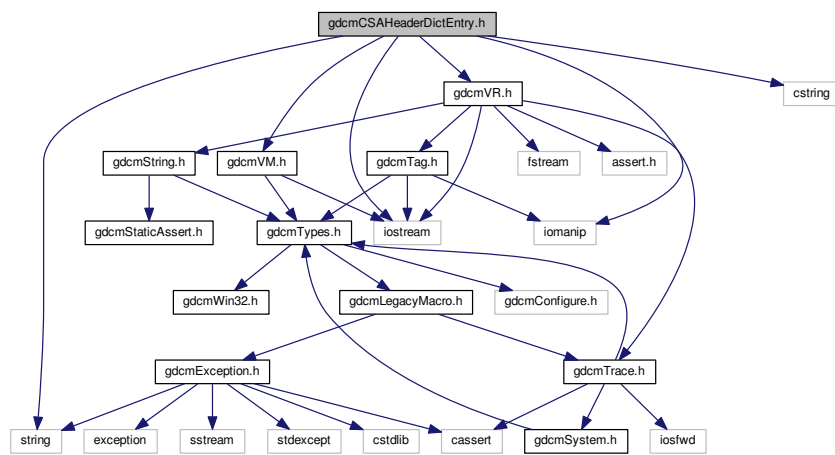
Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeaderDict &val)`

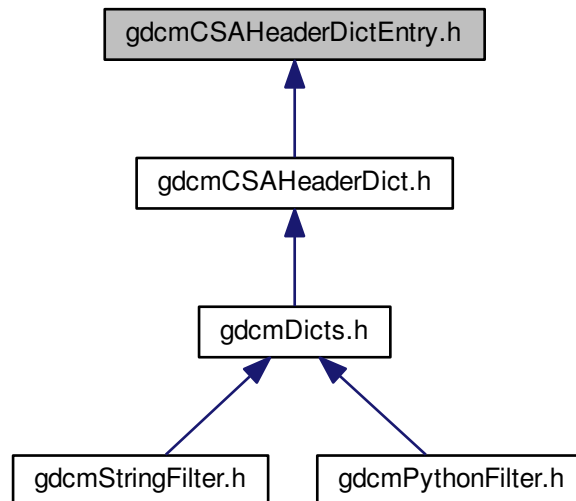
28.54 gdcmCSAHeaderDictEntry.h File Reference

```
#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](#) 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.

Namespaces

- [gdcm](#)

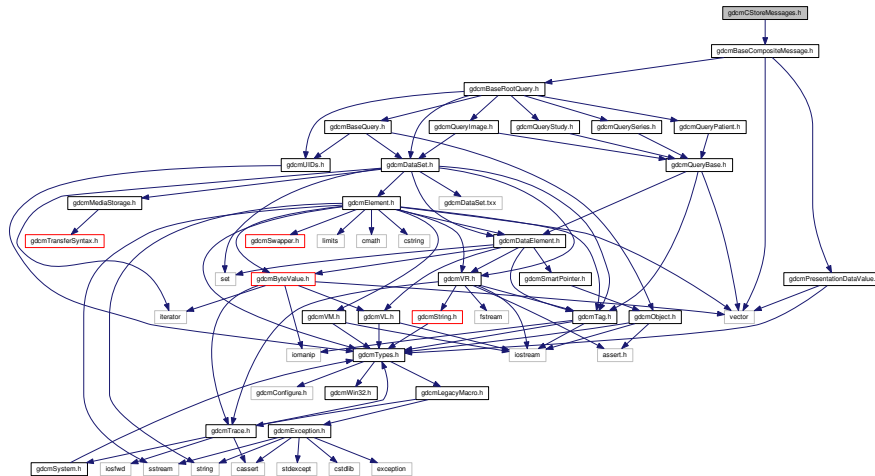
Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeaderDictEntry &val)`

28.55 gdcmCStoreMessages.h File Reference

```
#include "gdcmBaseCompositeMessage.h"
```


Include dependency graph for gdcmCStoreMessages.h:



Classes

- class [gdcm::network::CStoreRQ](#)

CStoreRQ this file defines the messages for the cecho action.

- class [gdcm::network::CStoreRSP](#)

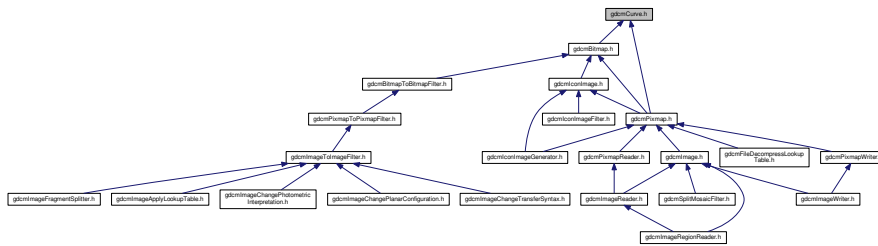
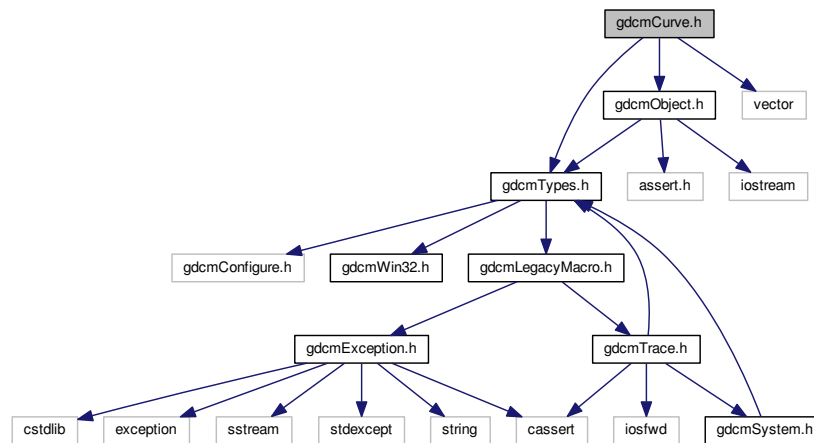
CStoreRSP this file defines the messages for the cecho action.

Namespaces

- [gdcm](#)
- [gdcm::network](#)

28.56 gdcmCurve.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmObject.h"
#include <vector>
```



- class `gdcm::Curve`

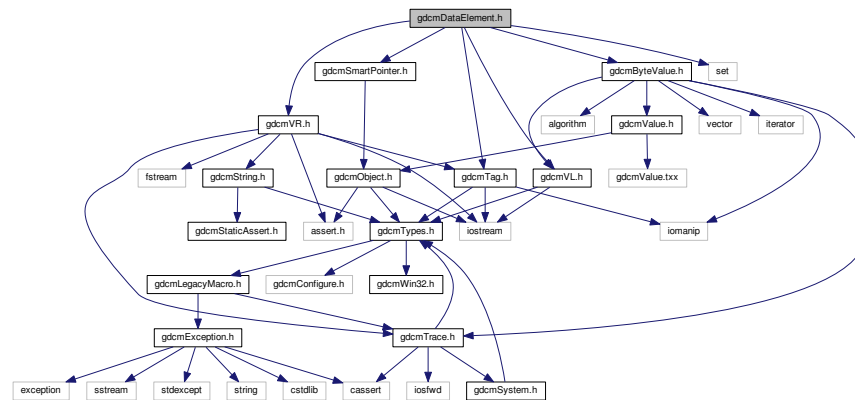
- Curve* class to handle element 50xx,3000 *Curve* Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004.

Curve class to handle element 50xx,3000 *Curve* Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004.

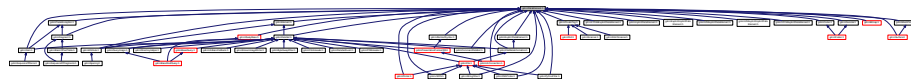
- **gdc**

```
#include "gdcmVL.h"
#include "gdcmVR.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"
#include <set>
```

Include dependency graph for gdcmDataElement.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::DataElement](#)
Class to represent a Data [Element](#) either Implicit or Explicit.

Namespaces

- [gdcm](#)

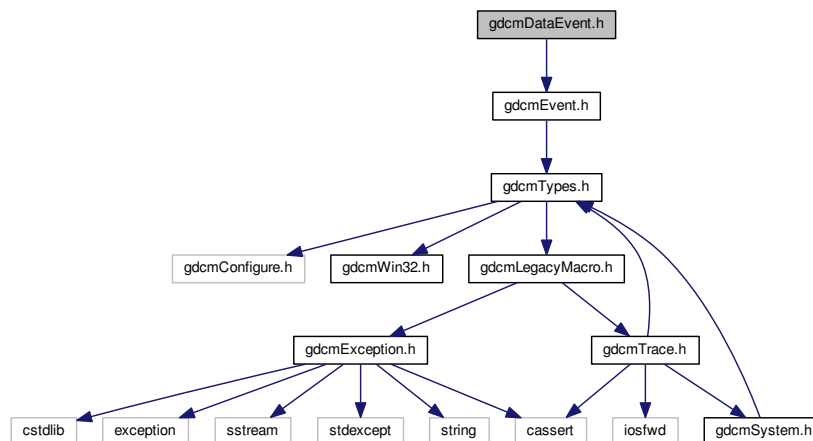
Functions

- bool [gdcm::operator!=](#) (const DataElement &lhs, const DataElement &rhs)
- std::ostream & [gdcm::operator<<](#) (std::ostream &os, const DataElement &val)

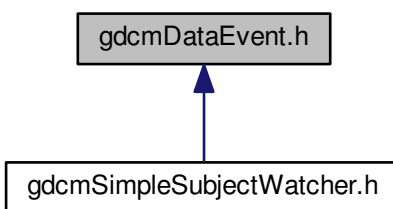
28.58 gdcmDataEvent.h File Reference

```
#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`

28.59 gdcmDataSet.h File Reference

```
#include "gdcmDataElement.h"
```

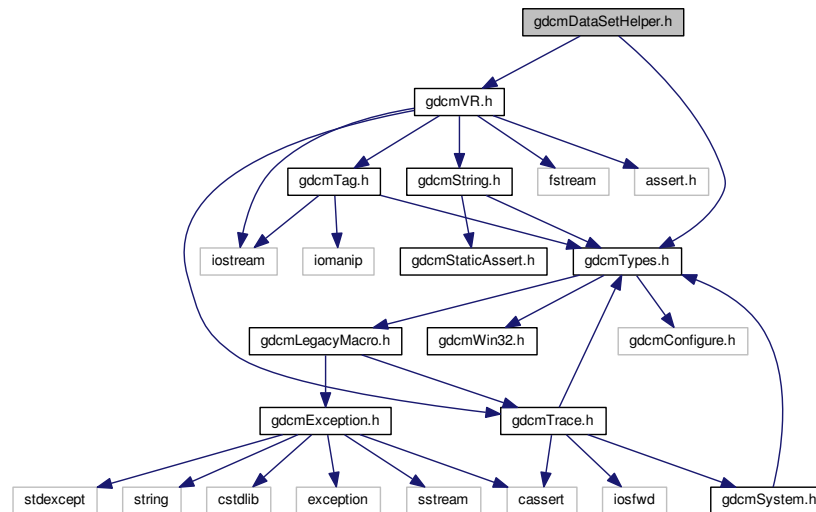
- class `gdcm::DataElementException`
- class `gdcm::DataSet`

- `gdcm`

- `std::ostream & gdcm::operator<< (std::ostream &os, const DataSet &val)`

```
#include "gdcmEvent.h"
```


Include dependency graph for gdcmDataSetHelper.h:



Classes

- class [gdcm::DataSetHelper](#)

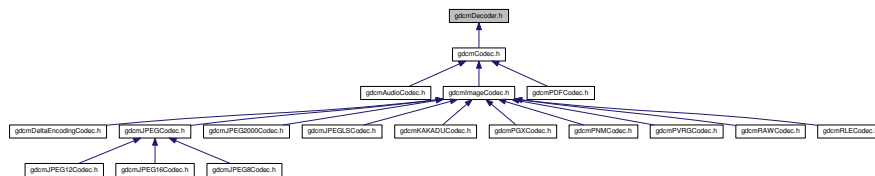
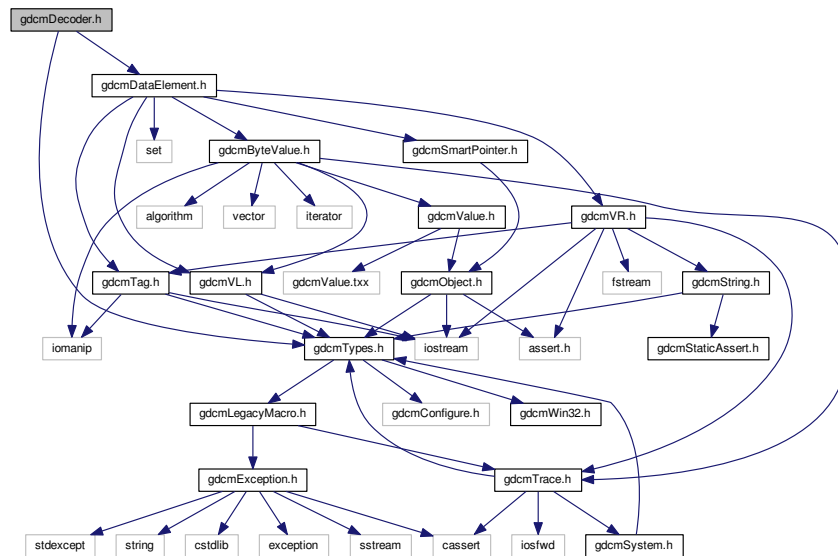
DataSetHelper (internal class, not intended for user level)

Namespaces

- [gdcm](#)

28.62 gdcmDecoder.h File Reference

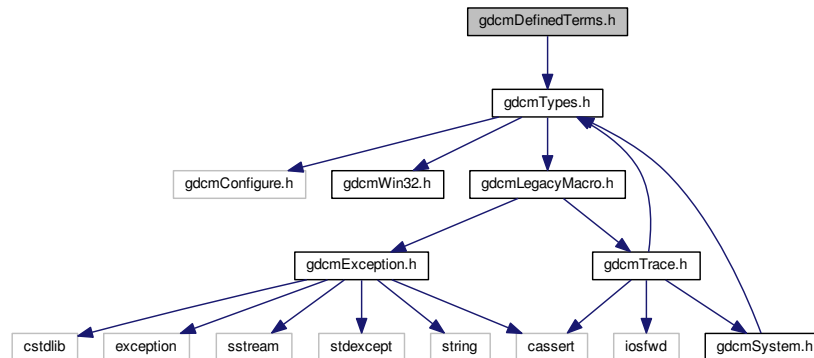
```
#include "gdcmTypes.h"
#include "gdcmDataElement.h"
```



- class `gdcm::Decoder`
Decoder.

- **gdcm**

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.

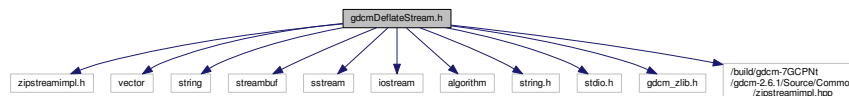
Namespaces

- [gdcm](#)

28.64 gdcmDeflateStream.h File Reference

```
#include "zipstreamimpl.h"
```

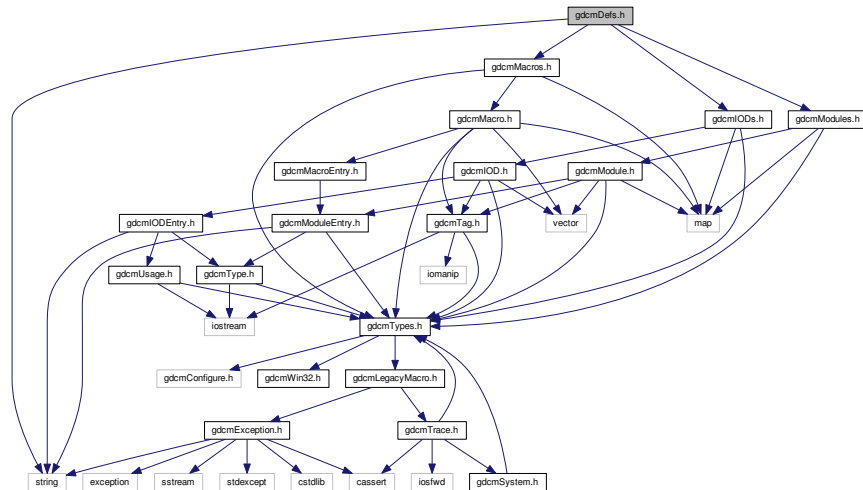
Include dependency graph for gdcmDeflateStream.h:



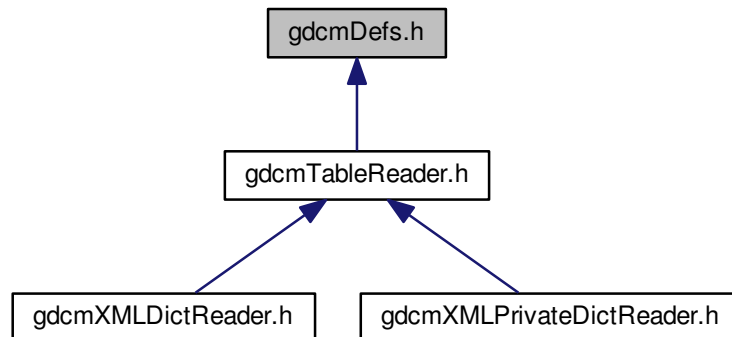
28.65 gdcmDefs.h File Reference

```
#include "gdcmModules.h"
```

```
#include "gdcmMacros.h"
#include "gdcmIODs.h"
#include <string>
Include dependency graph for gdcmDefs.h:
```



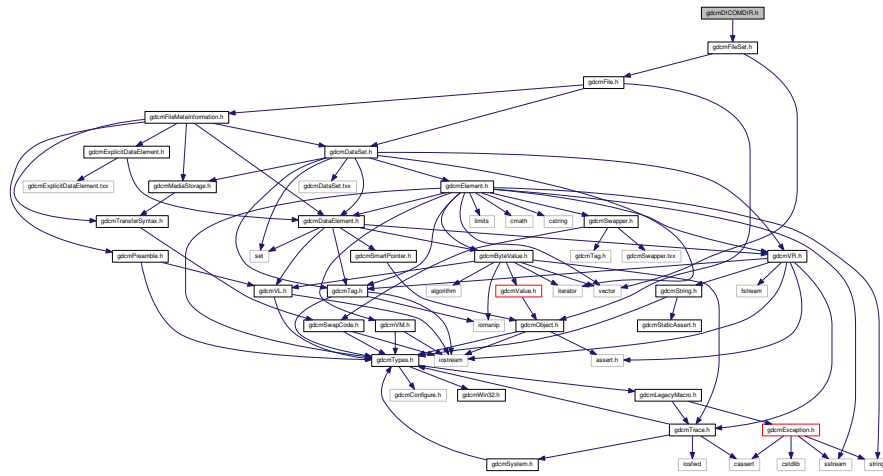
This graph shows which files directly or indirectly include this file:



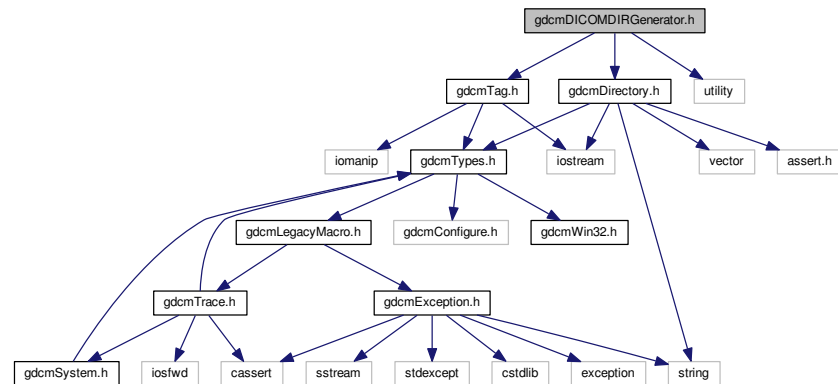
Classes

- class `gdcm::Defs`

FIXME I do not like the name 'Defs'.



Include dependency graph for gdcmDICOmdirGenerator.h:



Classes

- class [gdcm::DICOmdirGenerator](#)

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.

Namespaces

- [gdcm](#)

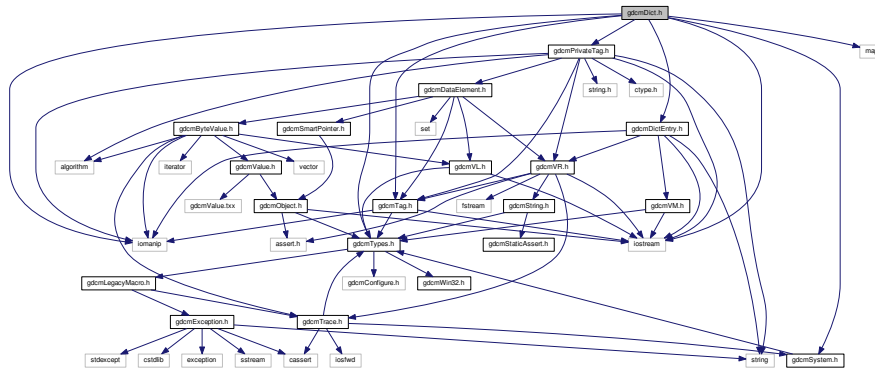
28.69 gdcmDict.h File Reference

```

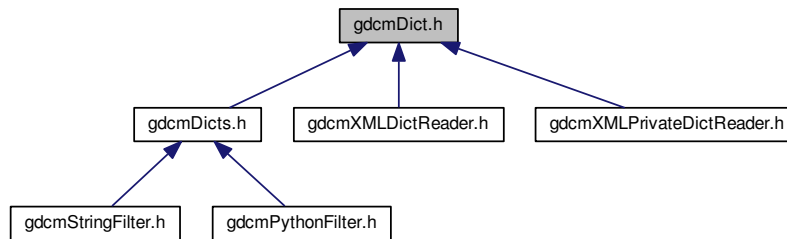
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmDictEntry.h"
#include "gdcmSystem.h"
#include <iostream>
#include <iomanip>
#include <map>

```

Include dependency graph for `gdcmDict.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::Dict`
Class to represent a map of `DictEntry`.
- class `gdcm::PrivateDict`
Private `Dict`.

Namespaces

- `gdcm`

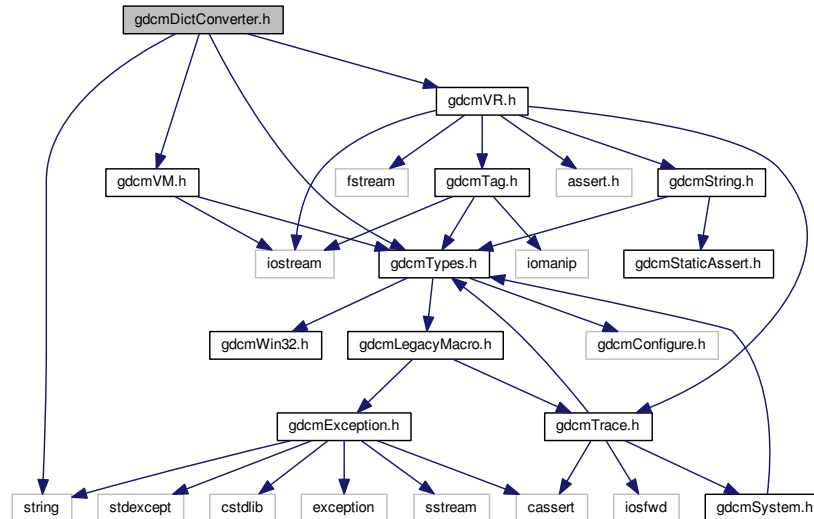
Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Dict &val)`
- `std::ostream & gdcm::operator<< (std::ostream &os, const PrivateDict &val)`

28.70 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:

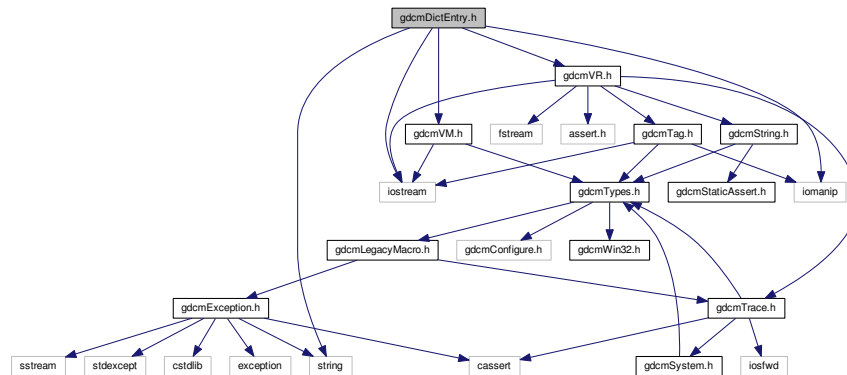
Namespaces

- [gdcm](#)

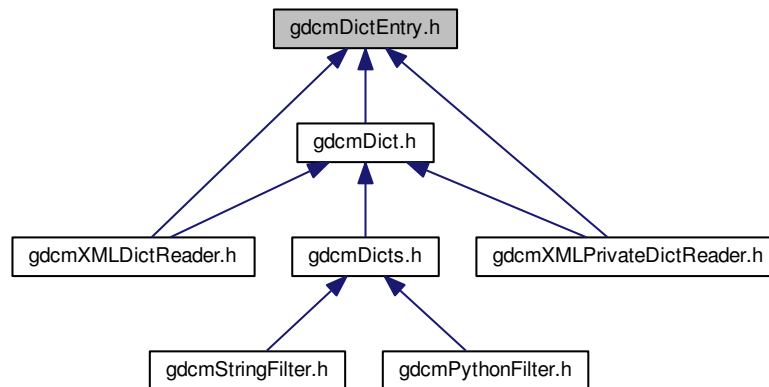
28.71 gdcmDictEntry.h File Reference

```
#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](#). 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.

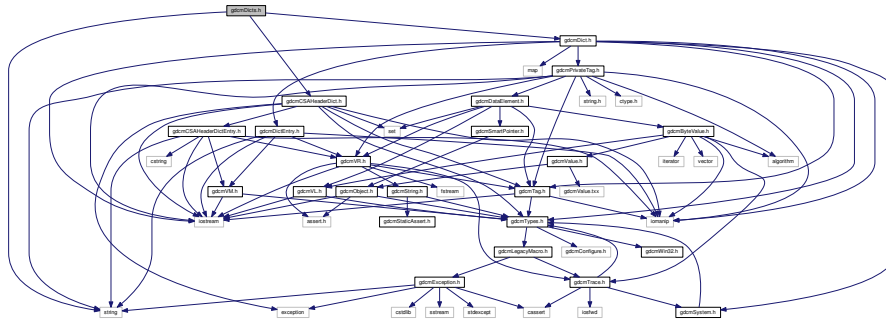
Namespaces

- [gdcm](#)

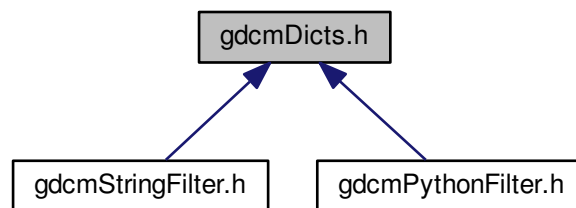
Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const DictEntry &val)`

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

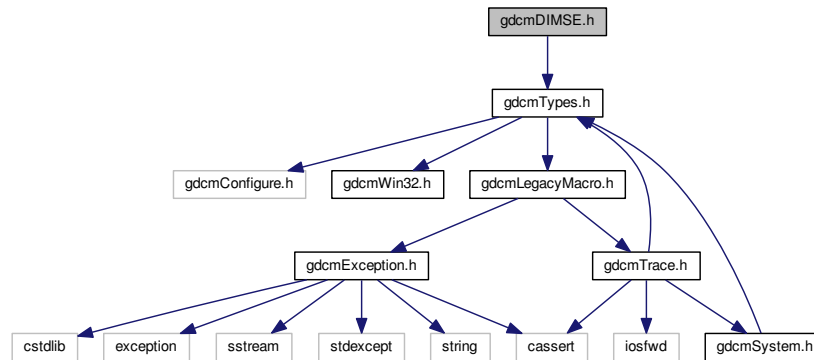
- `std::ostream & gdcm::operator<< (std::ostream &os, const Dicts &d)`

28.74 gdcmDiff.dox File Reference

28.75 gdcmDIMSE.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmDIMSE.h:



Classes

- class [gdcm::network::CEchoRQ](#)

[CEchoRQ](#) this file defines the messages for the cecho action.

- class [gdcm::network::CEchoRSP](#)

[CEchoRSP](#) this file defines the messages for the cecho action.

- class [gdcm::network::CFind](#)

- class [gdcm::network::DIMSE](#)

*[DIMSE](#) PS 3.7 - 2009 Annex E [Command Dictionary](#) (Normative) E.1 REGISTRY OF DICOM COMMAND ELEMENTS
[Table E.1-1 COMMAND FIELDS \(PART 1\)](#)*

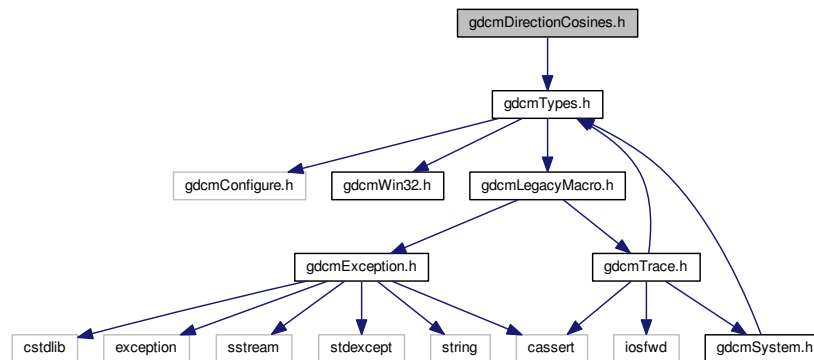
Namespaces

- [gdcm](#)
- [gdcm::network](#)

28.76 gdcmDirectionCosines.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for `gdcmdirDirectionCosines.h`:



Classes

- class `gdcmdir::DirectionCosines`
class to handle *DirectionCosines*

Namespaces

- `gdcmdir`

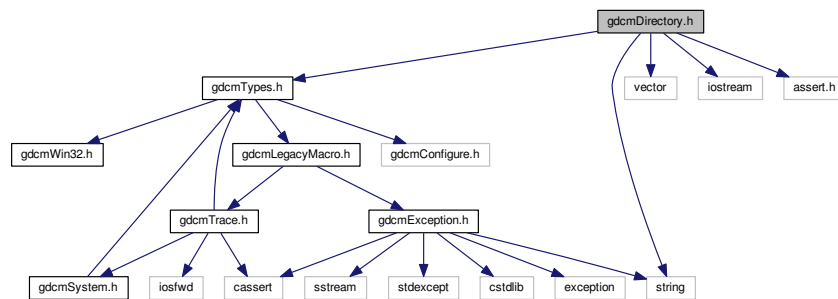
28.77 gdcmdirDirectory.h File Reference

```

#include "gdcmdirTypes.h"
#include <string>
#include <vector>
#include <iostream>
#include <assert.h>

```

Include dependency graph for `gdcmdirDirectory.h`:



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.

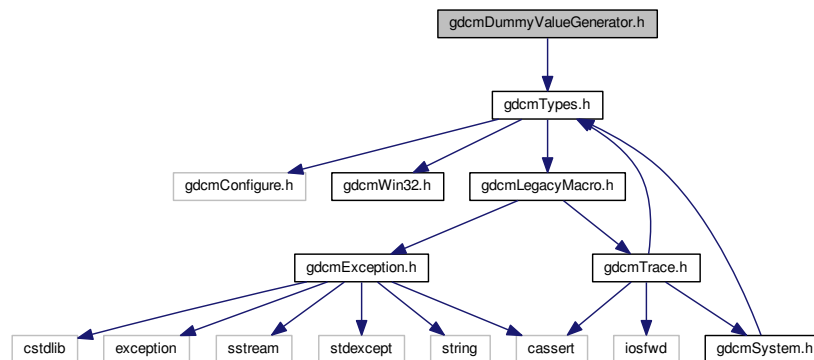
Namespaces

- [gdcm](#)

28.79 gdcmDummyValueGenerator.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmDummyValueGenerator.h:



Classes

- class [gdcm::DummyValueGenerator](#)
Class for generating dummy value.

Namespaces

- [gdcm](#)

28.80 gdcmdump.dox File Reference

28.81 gdcmDumper.h File Reference

```
#include "gdcmPrinter.h"
```

- class `gdcm::Dumper`
Codec class.

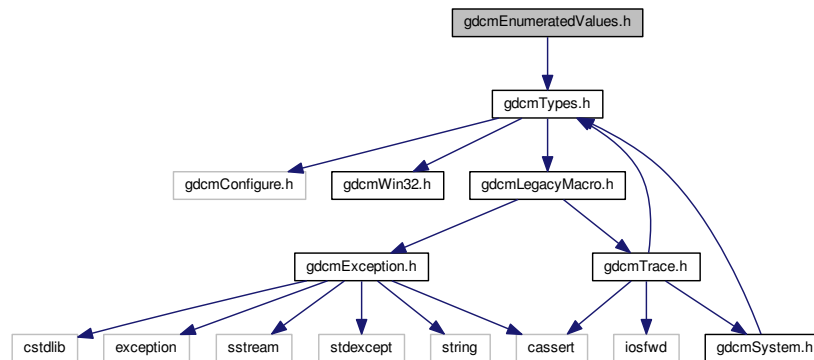
- gdc

```
#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmTag.h"
#include "gdcmVM.h"
#include "gdcmByteValue.h"
#include "gdcmDataElement.h"
#include "gdcmSwapper.h"
#include <string>
#include <vector>
#include <sstream>
#include <limits>
#include <cmath>
#include <cstring>
```


28.84 gdcmEnumeratedValues.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmEnumeratedValues.h:



Classes

- class [gdcm::EnumeratedValues](#)

***Element.** A Data [Element](#) with Enumerated Values that does not have a [Value](#) equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:*

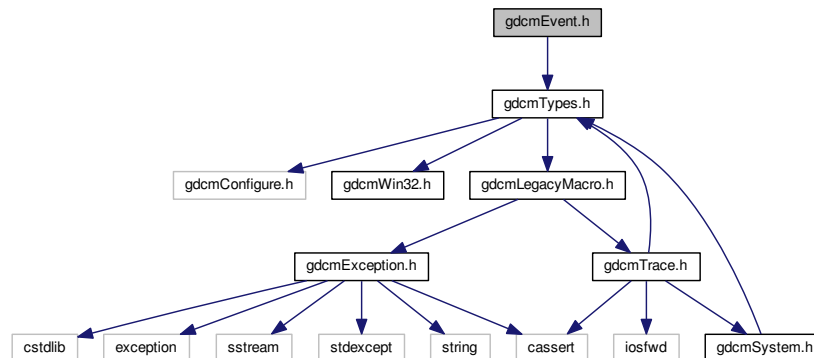
Namespaces

- [gdcm](#)

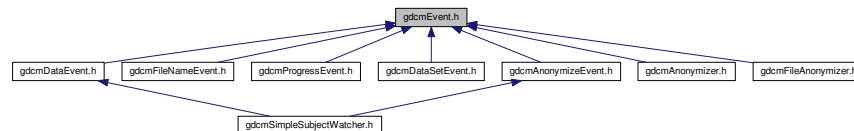
28.85 gdcmEvent.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcEvent.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

- [gdc](#)

Macros

- `#define gdcEventMacro(classname, super)`

Functions

- `std::ostream & gdcmm::operator<< (std::ostream &os, Event &e)`

Generic inserter operator for [Event](#) and its subclasses.

28.85.1 Macro Definition Documentation

28.85.1.1 `#define gdcmmEventMacro(classname, super)`

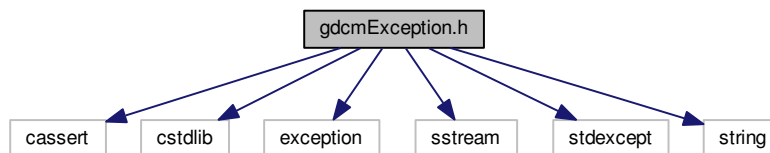
Value:

```
\
class classname : public super { \
public: \
    typedef classname Self; \
    typedef super Superclass; \
    classname() {} \
    virtual ~classname() {} \
    virtual const char * GetEventName() const { return #classname; } \
    virtual bool CheckEvent(const ::gdcmm::Event* e) const \
    { return dynamic_cast<const Self*>(e) ? true : false; } \
    virtual ::gdcmm::Event* MakeObject() const \
    { return new Self; } \
    classname(const Self&s) : super(s){}; \
private: \
    void operator=(const Self&); \
}
```

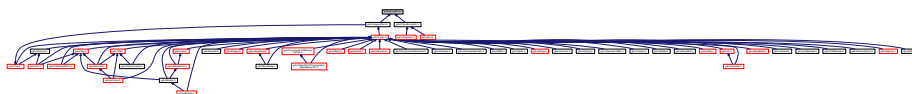
28.86 gdcmmException.h File Reference

```
#include <cassert>
#include <cstdlib>
#include <exception>
#include <sstream>
#include <stdexcept>
#include <string>
```

Include dependency graph for `gdcmmException.h`:



This graph shows which files directly or indirectly include this file:



Classes

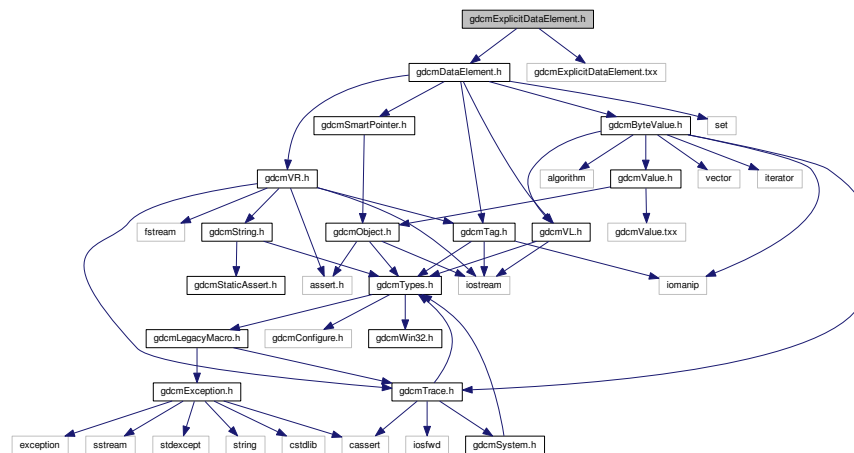
- class [gdcm::Exception](#)
Exception.

Namespaces

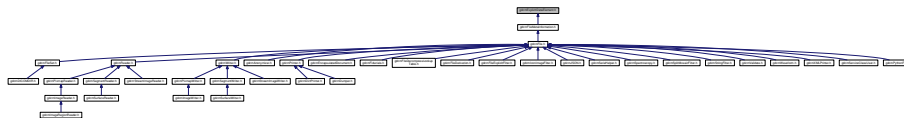
- [gdcm](#)

28.87 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](#)

Classes

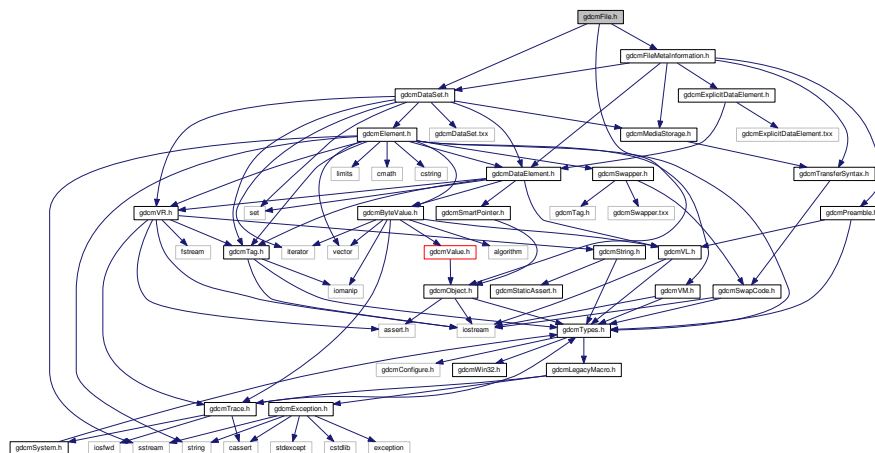
- class `gdc::Fiducials`
Fiducials.

Namespaces

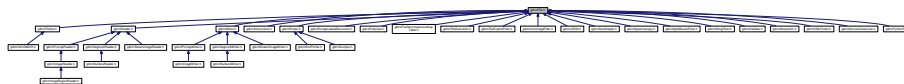
- `gdcm`

28.90 gdcMFile.h File Reference

```
#include "gdcmObject.h"
#include "gdcmDataSet.h"
#include "gdcmFileMetaInformation.h"
Include dependency graph for gdcmFile.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::File`

a DICOM [File](#) 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.

Namespaces

- [gdcm](#)

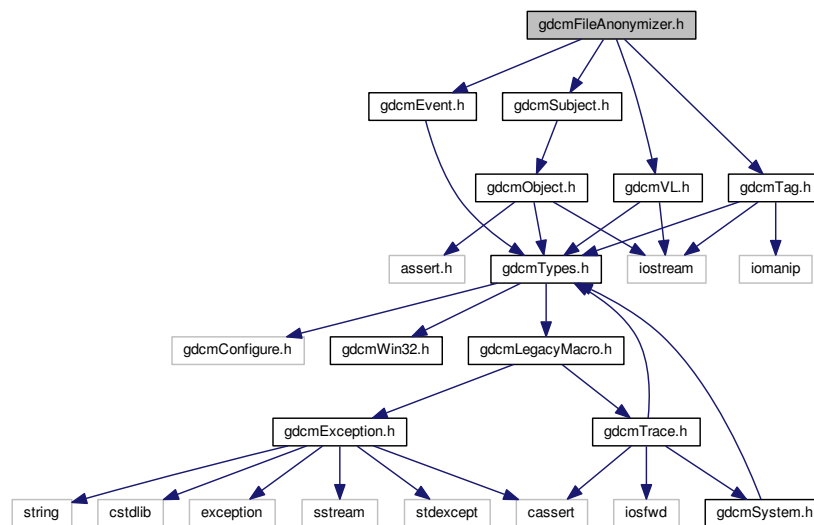
Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const File &val)`

28.91 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

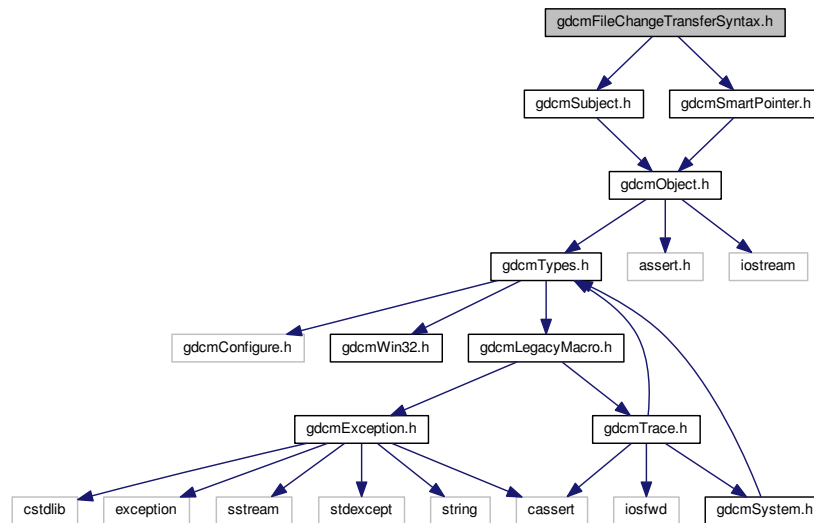
- [gdcm](#)

28.92 gdcmFileChangeTransferSyntax.h File Reference

```
#include "gdcmSubject.h"
```

```
#include "gdcmSmartPointer.h"
```

Include dependency graph for gdcmFileChangeTransferSyntax.h:



Classes

- class [gdcm::FileChangeTransferSyntax](#)

FileChangeTransferSyntax.

Namespaces

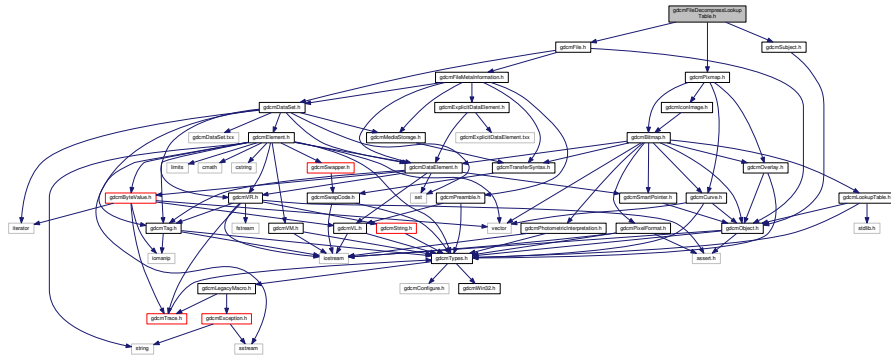
- [gdcm](#)

28.93 gdcmFileDecompressLookupTable.h File Reference

```
#include "gdcmSubject.h"
```

```
#include "gdcmFile.h"
```

```
#include "gdcmPixmap.h"
```

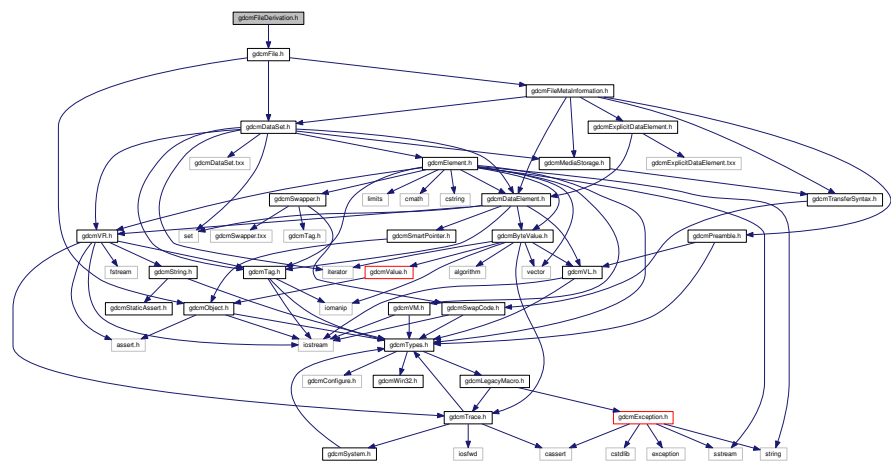


- class `gdcm::FileDecompressLookupTable`

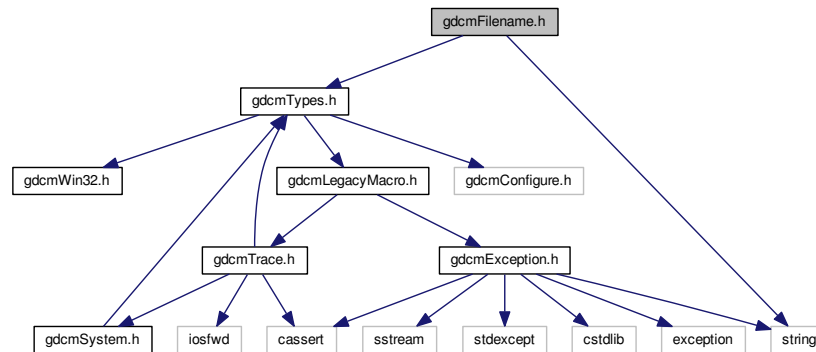
FileDecompressLookupTable class It decompress the segmented LUT into linearized one (only PALETTE_COLOR images) Output will be a *PhotometricInterpretation*=RGB image.

- **gdcm**

```
#include "gdcmFile.h"
```



Include dependency graph for gdcmFilename.h:



Classes

- class [gdcm::Filename](#)

Class to manipulate file name's.

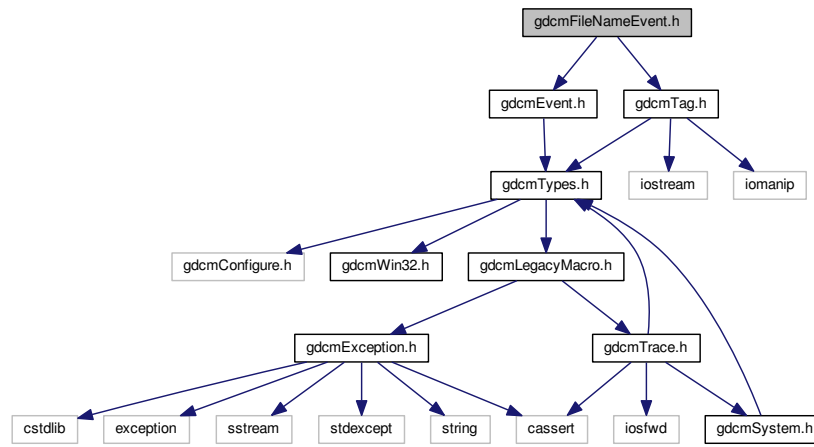
Namespaces

- [gdcm](#)

28.98 gdcmFileNameEvent.h File Reference

```
#include "gdcmEvent.h"
#include "gdcmTag.h"
```

Include dependency graph for `gdcmFileNameEvent.h`:



Classes

- class [gdcm::FileNameEvent](#)

[FileNameEvent](#) Special type of event triggered during processing of [FileSet](#).

Namespaces

- [gdcm](#)

28.99 gdcmFilenameGenerator.h File Reference

```

#include "gdcmTypes.h"
#include <string>
#include <vector>

```

```

graph TD
    gdcmFilenameGenerator.h --> gdcmTypes.h
    gdcmFilenameGenerator.h --> vector
    gdcmTypes.h --> gdcmWin32.h
    gdcmTypes.h --> gdcmLegacyMacro.h
    gdcmTypes.h --> gdcmConfigure.h
    gdcmLegacyMacro.h --> gdcmTrace.h
    gdcmLegacyMacro.h --> gdcmException.h
    gdcmTrace.h --> gdcmSystem.h
    gdcmTrace.h --> iosfwd
    gdcmTrace.h --> cassert
    gdcmException.h --> sstream
    gdcmException.h --> stdexcept
    gdcmException.h --> cstdlib
    gdcmException.h --> exception
    gdcmException.h --> string
  
```

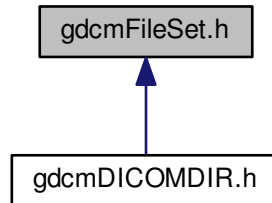
- class `gdcm::FilenameGenerator`
FilenameGenerator.

- **gdcm**

```
#include "gdcmFile.h"
#include <vector>
```

[illegible]

This graph shows which files directly or indirectly include this file:



Classes

- class [gdcmm::FileSet](#)

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.

Namespaces

- [gdcmm](#)

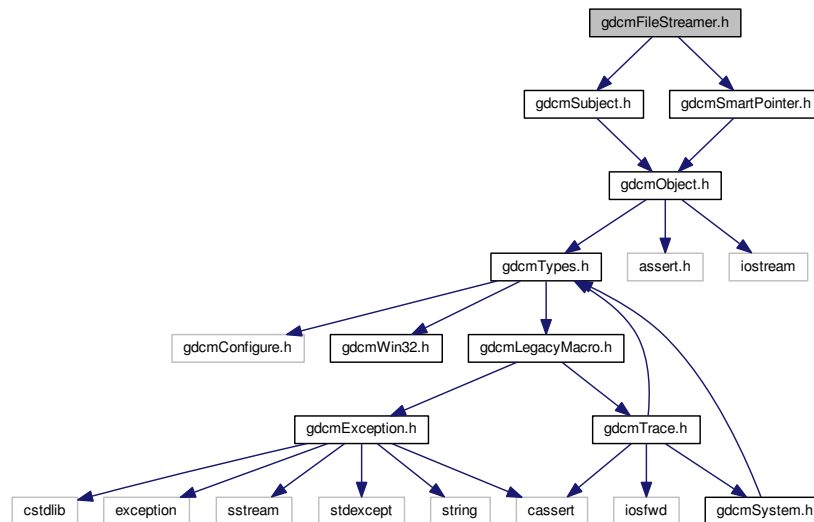
Functions

- `std::ostream & gdcmm::operator<< (std::ostream &os, const FileSet &f)`

28.101 gdcmFileStreamer.h File Reference

```
#include "gdcmSubject.h"  
#include "gdcmSmartPointer.h"
```


Include dependency graph for gdcmFileStreamer.h:



Classes

- class [gdcm::FileStreamer](#)

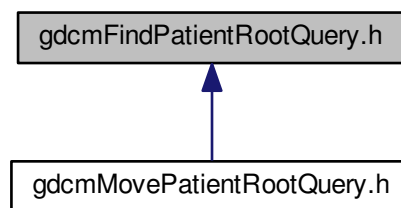
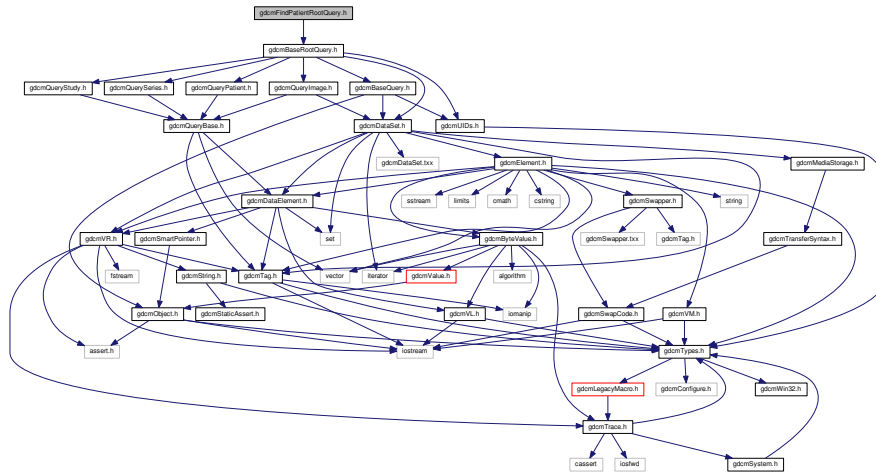
[FileStreamer](#) This class let a user create a massive DICOM [DataSet](#) from a template DICOM file, by appending chunks of data.

Namespaces

- [gdcm](#)

28.102 gdcmFindPatientRootQuery.h File Reference

```
#include "gdcmBaseRootQuery.h"
```



- class `gdcmm::FindPatientRootQuery`
PatientRootQuery contains: the class which will produce a dataset for c-find with patient root.

- **gdcm**

```
#include "gdcmBaseRootQuery.h"
```

[illegible]

- class `gdcm::FindStudyRootQuery`

Namespaces

- **gdcm**

```
#include "gdcmDataElement.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"
#include "gdcmParseException.h"
```


Functions

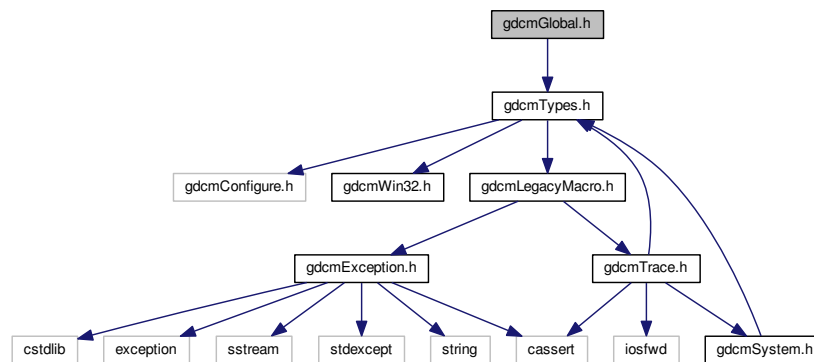
- `std::ostream & gdcmm::operator<<` (`std::ostream &os`, `const Fragment &val`)

28.105 gdcmgendir.dox File Reference

28.106 gdcmmGlobal.h File Reference

```
#include "gdcmmTypes.h"
```

Include dependency graph for gdcmmGlobal.h:



Classes

- class `gdcmm::Global`
Global.

Namespaces

- `gdcmm`

Functions

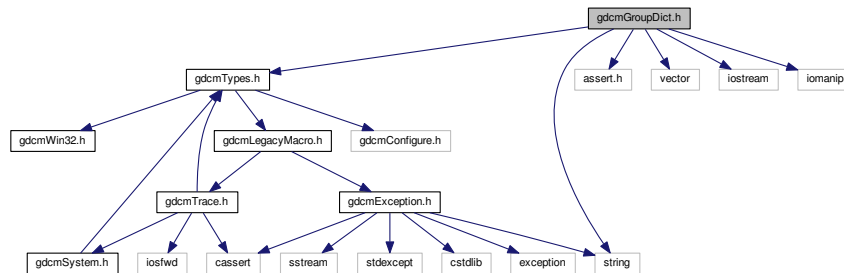
- `std::ostream & gdcmm::operator<<` (`std::ostream &os`, `const Global &g`)

Variables

- static Global `gdcmm::GlobalInstance`

28.107 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)`

28.108 gdcmIconImage.h File Reference

```
#include "gdcmBitmap.h"
```

```

classDiagram
    class gdcmConImage_h["gdcmConImage.h"]
    class gdcmPixmap_h["gdcmPixmap.h"]
    class gdcmConImageFilter_h["gdcmConImageFilter.h"]
    class gdcmConImageGenerator_h["gdcmConImageGenerator.h"]
    class gdcmFileDecompressLookupTable_h["gdcmFileDecompressLookupTable.h"]
    class gdcmPixmapReader_h["gdcmPixmapReader.h"]
    class gdcmImage_h["gdcmImage.h"]
    class gdcmPixmapWriter_h["gdcmPixmapWriter.h"]
    class gdcmImageReader_h["gdcmImageReader.h"]
    class gdcmImageRegionReader_h["gdcmImageRegionReader.h"]
    class gdcmSplitMosaicFilter_h["gdcmSplitMosaicFilter.h"]
    class gdcmImageWriter_h["gdcmImageWriter.h"]

    gdcmConImage_h --> gdcmConImageGenerator_h
    gdcmConImage_h --> gdcmFileDecompressLookupTable_h
    gdcmConImage_h --> gdcmPixmapReader_h
    gdcmConImage_h --> gdcmImage_h
    gdcmConImage_h --> gdcmPixmapWriter_h
    gdcmPixmap_h --> gdcmConImageGenerator_h
    gdcmPixmap_h --> gdcmFileDecompressLookupTable_h
    gdcmPixmap_h --> gdcmPixmapReader_h
    gdcmPixmap_h --> gdcmImage_h
    gdcmPixmap_h --> gdcmPixmapWriter_h
    gdcmConImageFilter_h --> gdcmImage_h
    gdcmImageReader_h --> gdcmPixmapReader_h
    gdcmImageRegionReader_h --> gdcmImageReader_h
    gdcmImageRegionReader_h --> gdcmImage_h
    gdcmSplitMosaicFilter_h --> gdcmImage_h
    gdcmImageWriter_h --> gdcmPixmapWriter_h
  
```

- **gdcm**

- typedef Bitmap **gdcmm::IconImage**

```
#include "gdcmFile.h"
#include "gdcmIconImage.h"
```

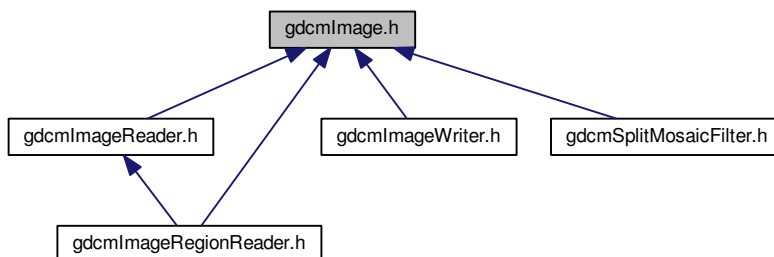

- class `gdcm::IconImageGenerator`

Namespaces

- gdc

```
#include <vector>
```

This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::Image`

Image This is the container for an *Image* in the general sense. From this container you should be able to request information like:

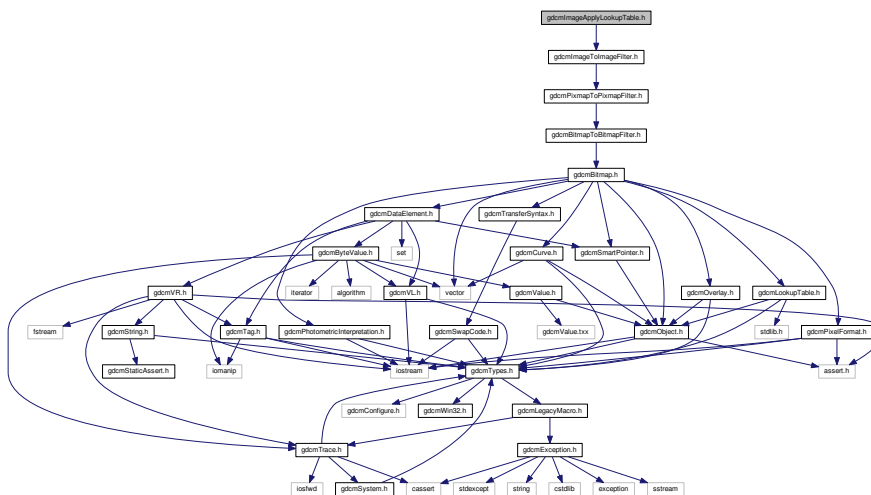
Namespaces

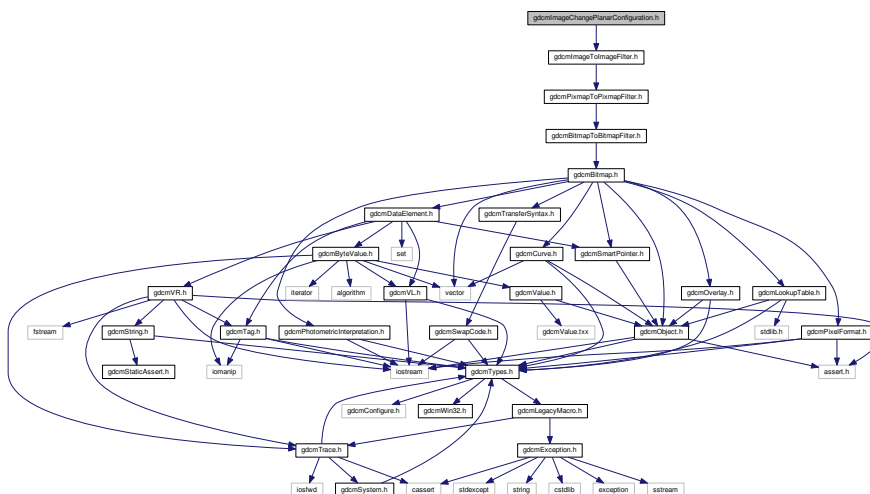
- **gdcm**

28.112 gdcmImageApplyLookupTable.h File Reference

```
#include "gdcmImageToImageFilter.h"
```

Include dependency graph for gdcmlImageApplyLookupTable.h:



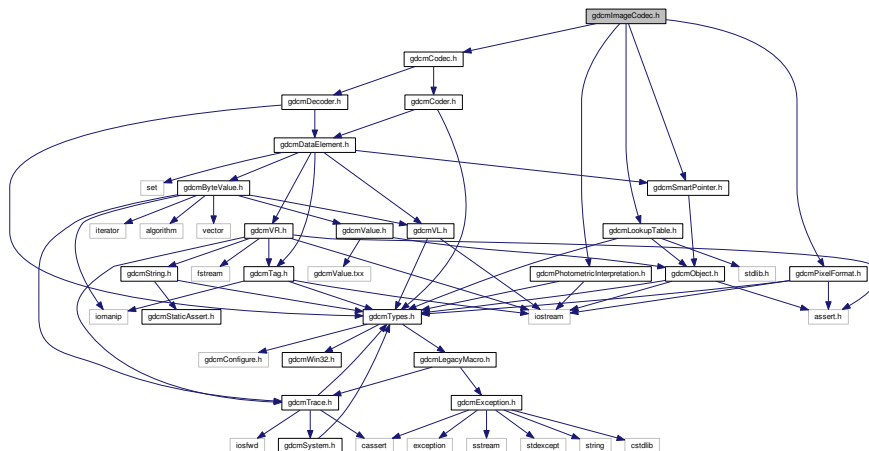


- class `gdcm::ImageChangeTransferSyntax`

Namespaces

- **gdcm**

```
#include "gdcmCodec.h"
#include "gdcmPhotometricInterpretation.h"
#include "gdcmLookupTable.h"
#include "gdcmSmartPointer.h"
#include "gdcmPixelFormat.h"
```

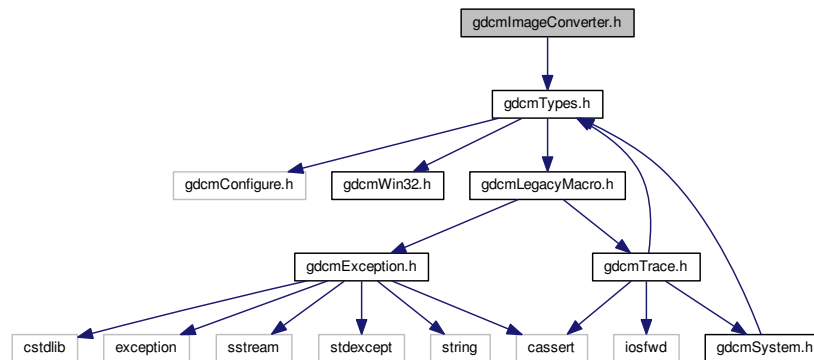


- class `gdcm::ImageCodec`
ImageCodec.

- [gdcm](#)

```
#include "gdcmTypes.h"
```

Include dependency graph for `gdcmlImageConverter.h`:



Classes

- class `gdcm::ImageConverter`
Image Converter.

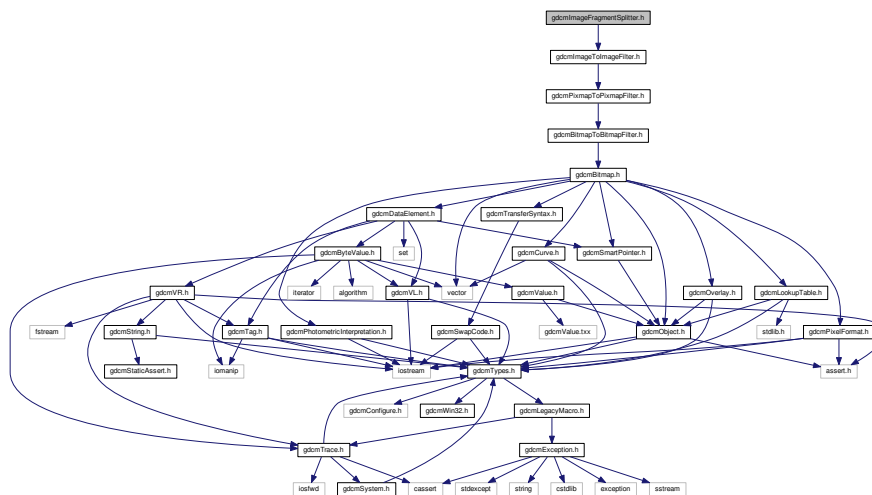
Namespaces

- **gdcm**

28.118 gdcmImageFragmentSplitter.h File Reference

```
#include "gdcmImageToImageFilter.h"
```

Include dependency graph for `gdcImageFragmentSplitter.h`:



Classes

- class [gdcm::ImageFragmentSplitter](#)
ImageFragmentSplitter class For single frame image, DICOM standard allow splitting the frame into multiple fragments.

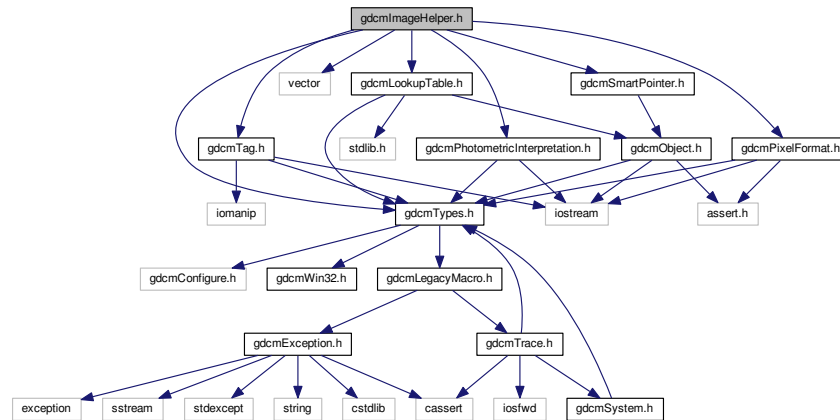
Namespaces

- [gdcm](#)

28.119 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

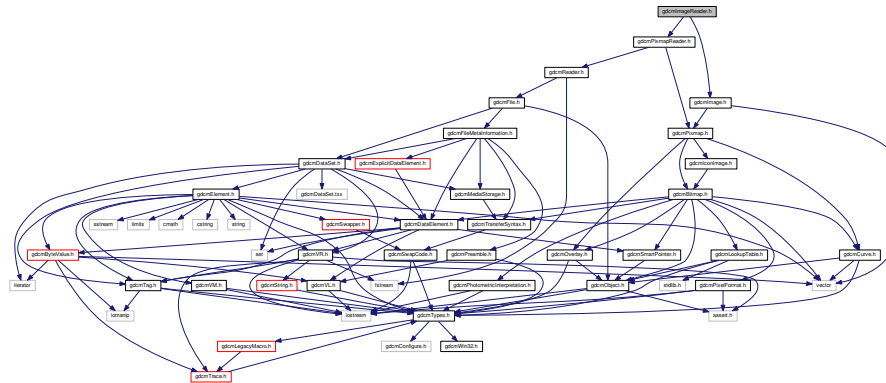
- class [gdcm::ImageHelper](#)
ImageHelper (internal class, not intended for user level)
- struct [gdcm::RealWorldValueMappingContent](#)

Namespaces

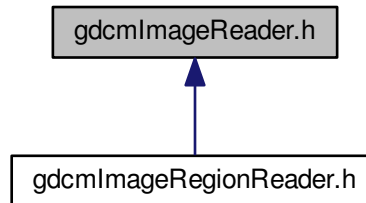
- [gdcm](#)

28.120 gdcmImageReader.h File Reference

```
#include "gdcmPixmapReader.h"
#include "gdcmImage.h"
Include dependency graph for gdcmImageReader.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::ImageReader](#)
ImageReader.

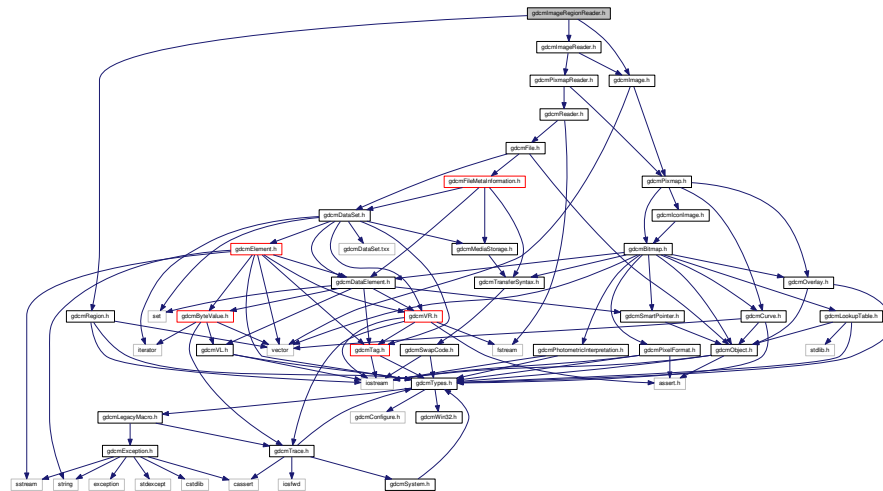
Namespaces

- [gdcm](#)

28.121 gdcmImageRegionReader.h File Reference

```
#include "gdcmImageReader.h"
```

```
#include "gdcmImage.h"
#include "gdcmRegion.h"
```



Classes

- class `gdcm::ImageRegionReader`

ImageRegionReader.

Namespaces

- **gdcm**

28.122 gdcmImageToImageFilter.h File Reference

```
#include "gdcmPixmapToPixmapFilter.h"
```

[illegible]

```

graph BT
    gdcimageToImageFilter.h --> gdcimageApplyLookupTable.h
    gdcimageToImageFilter.h --> gdcimageChangePhotometricInterpretation.h
    gdcimageToImageFilter.h --> gdcimageChangePlaneConfiguration.h
    gdcimageToImageFilter.h --> gdcimageChangeTransferSyntax.h
    gdcimageToImageFilter.h --> gdcimageFragmentSplitter.h

```

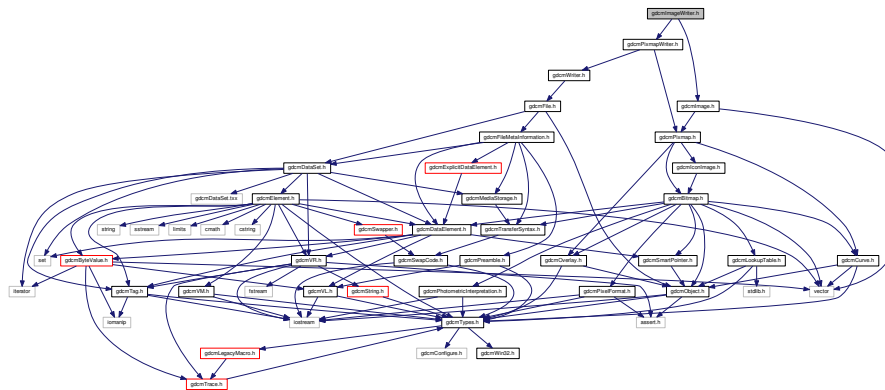
- class `gdcm::ImageToImageFilter`

Namespaces

- **gdcm**

```
#include "gdcmPixmapWriter.h"
#include "gdcmImage.h"
```

Include dependency graph for `gdcmImageWriter.h`:



Classes

- class `gdcm::ImageWriter`
ImageWriter.

Namespaces

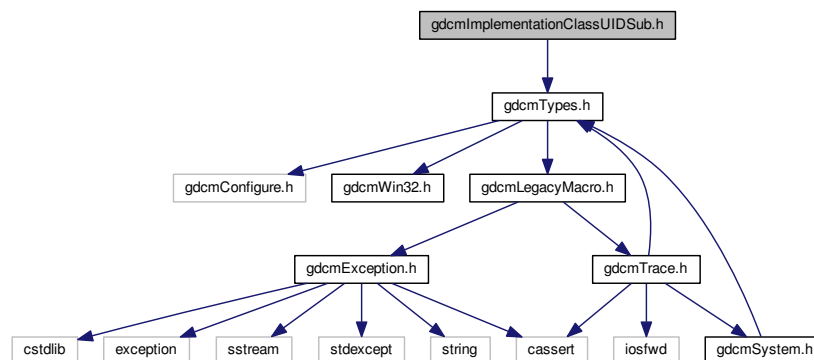
- `gdcm`

28.124 gdcming.dox File Reference

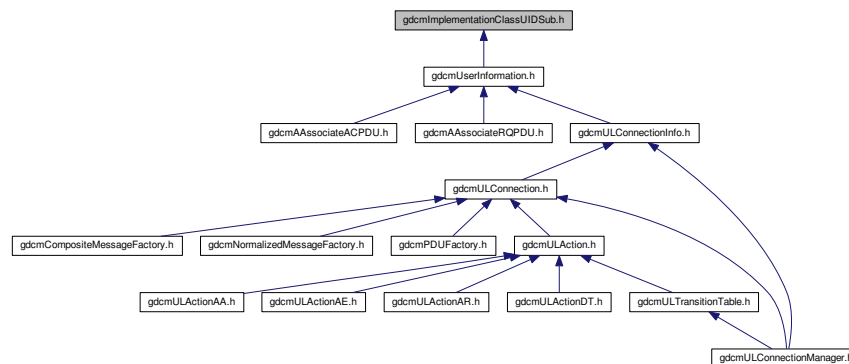
28.125 gdcmlImplementationClassUIDSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for `gdcmImplementationClassUIDSub.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::network::ImplementationClassUIDSub](#)

ImplementationClassUIDSub PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATED ↔ E-RQ)

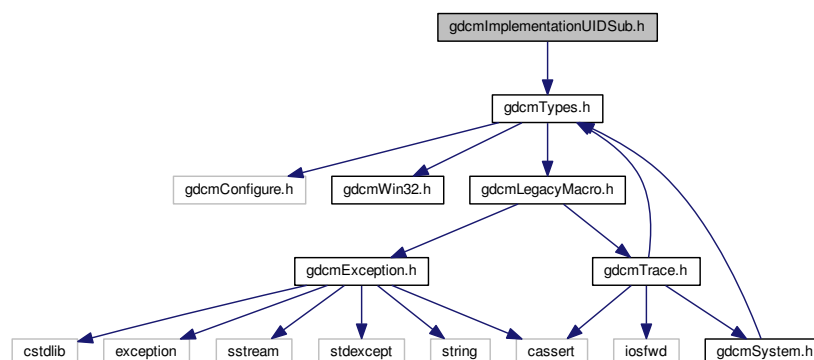
Namespaces

- [gdcm](#)
- [gdcm::network](#)

28.126 gdcmImplementationUIDSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmImplementationUIDSub.h:



Classes

- class [gdcm::network::ImplementationUIDSub](#)

[ImplementationUIDSub Table](#) *D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)*

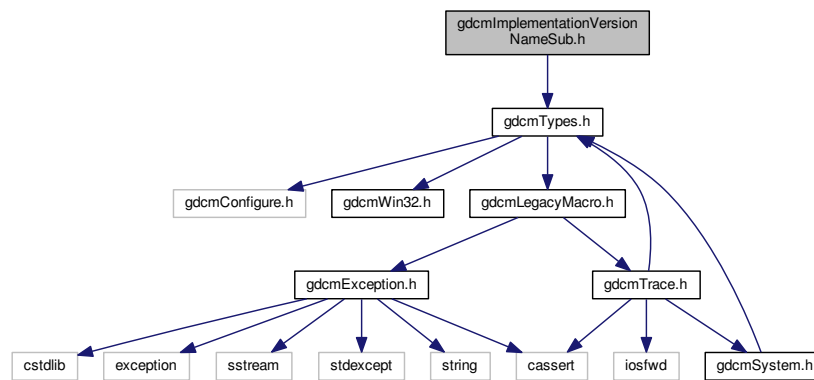
Namespaces

- [gdcm](#)
- [gdcm::network](#)

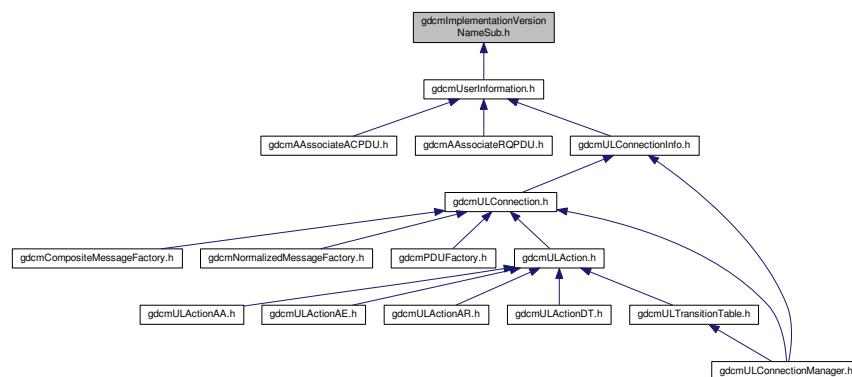
28.127 gdcmImplementationVersionNameSub.h File Reference

```
#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`

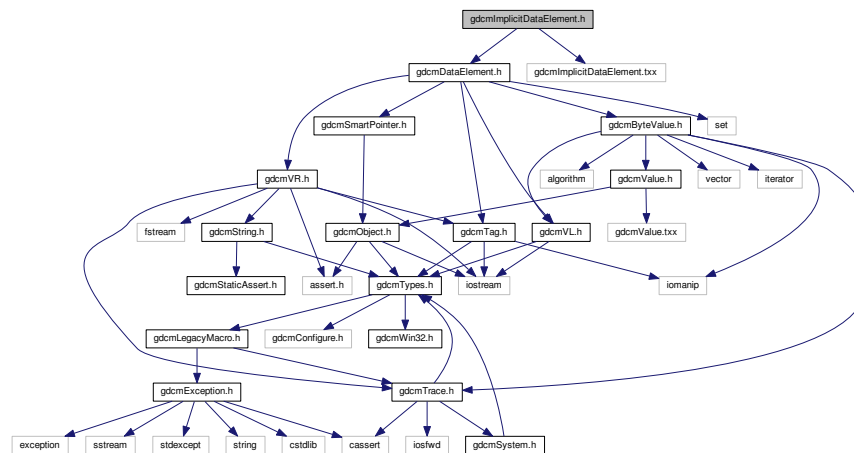
ImplementationVersionNameSub Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

Namespaces

- `gdcm`
- `gdcm::network`

28.128 gdcmlImplicitDataElement.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

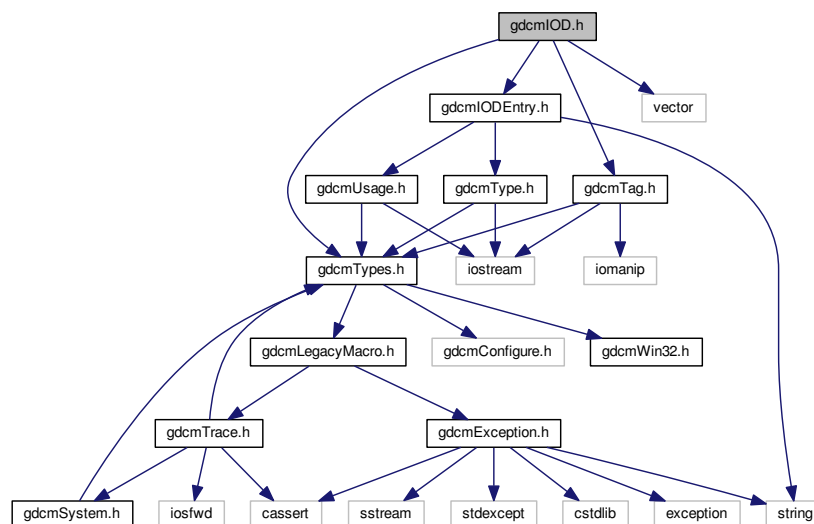
- **gdcm**

28.129 gdcminfo.dox File Reference

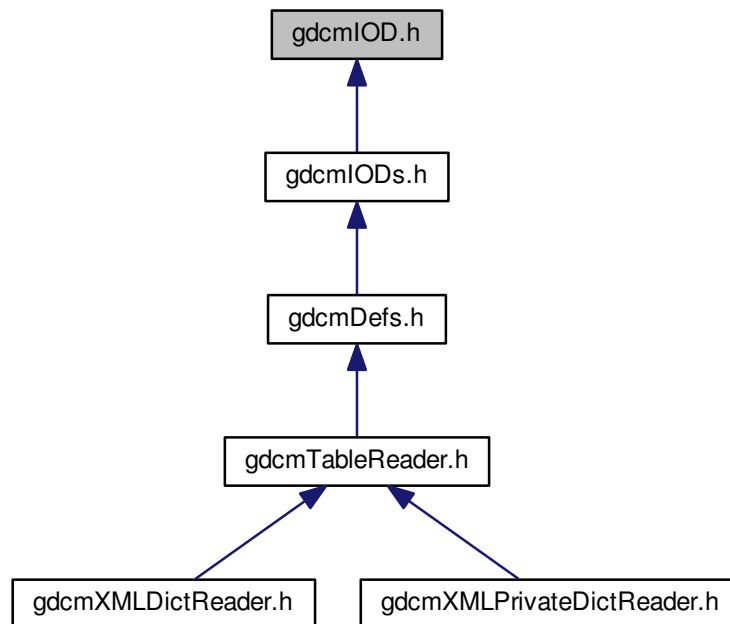
28.130 gdcmIOD.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmIODEntry.h"
#include <vector>
```

Include dependency graph for gdcmIOD.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::IOD](#)
Class for representing a [IOD](#).

Namespaces

- [gdcm](#)

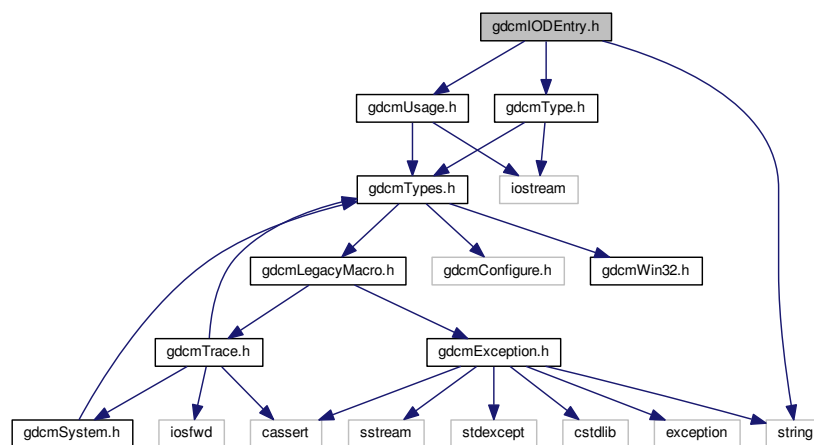
Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const IOD &_val)`

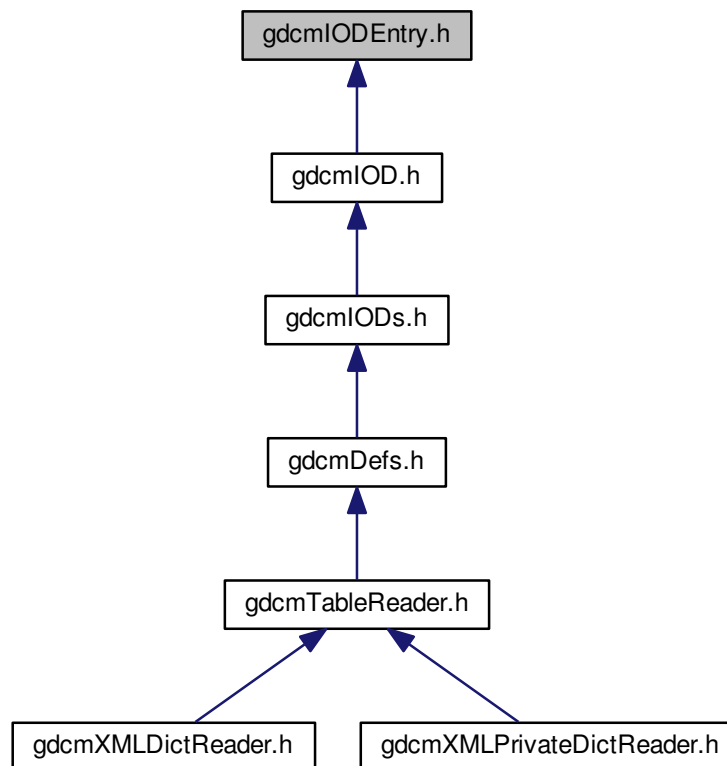
28.131 gdcmIODEntry.h File Reference

```
#include "gdcmUsage.h"
#include "gdcmType.h"
#include <string>
```

Include dependency graph for `gdcmIODEntry.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::IODEntry](#)
Class for representing a [IODEntry](#).

Namespaces

- [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const IODEntry &_val)`

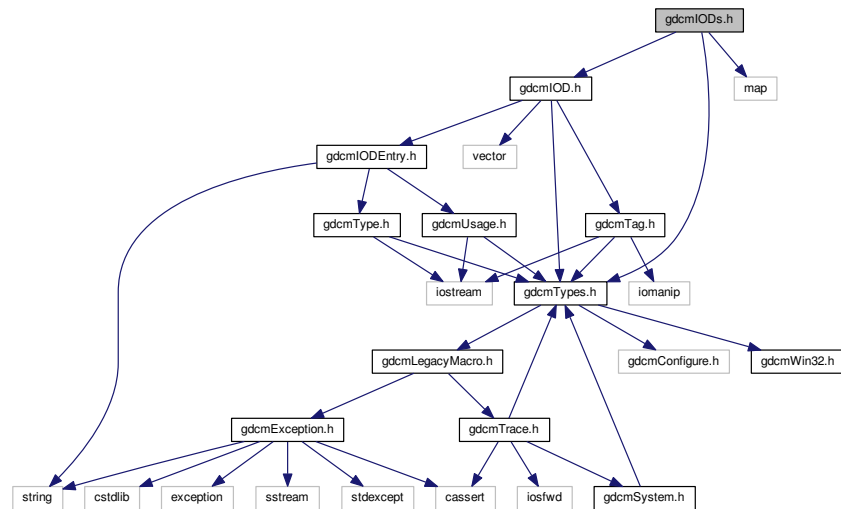
28.132 gdcmIODs.h File Reference

```
#include "gdcmTypes.h"
```

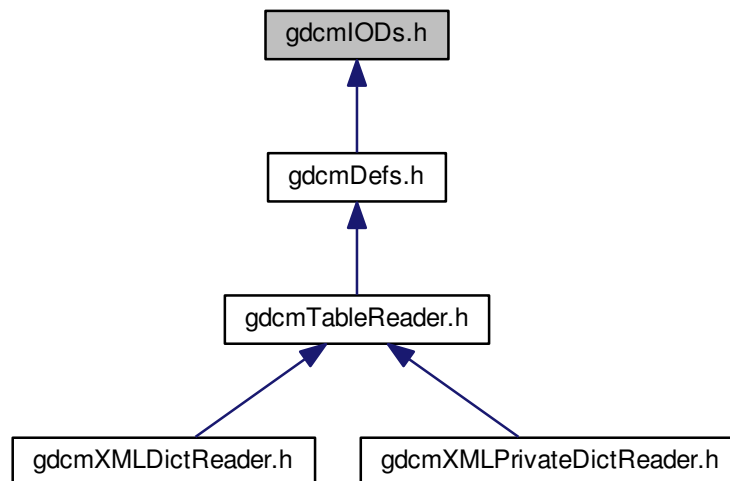
```
#include "gdcmIOD.h"
```

```
#include <map>
```

Include dependency graph for gdcmIODs.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::IODs](#)

Class for representing a [IODs](#).

Namespaces

- [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const IODs &_val)`

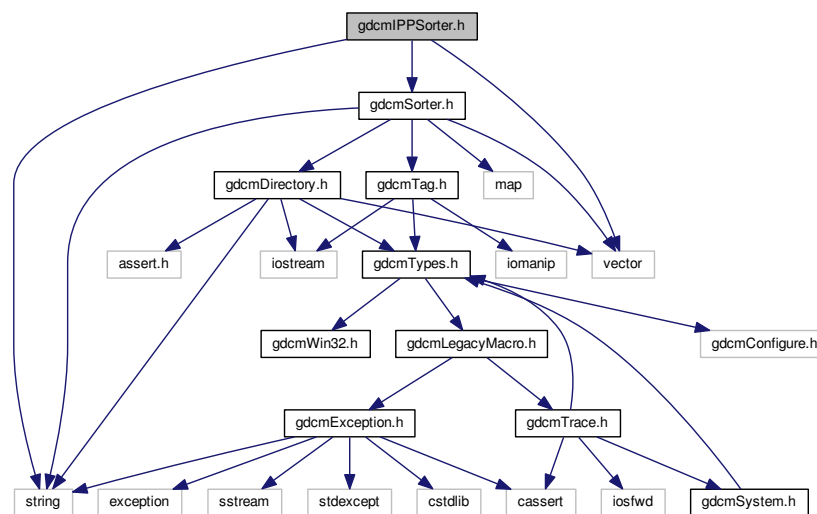
28.133 gdcmIPPSorter.h File Reference

```
#include "gdcmSorter.h"
```

```
#include <vector>
```

```
#include <string>
```

Include dependency graph for gdcmIPPSorter.h:



Classes

- class [gdcm::IPPSorter](#)

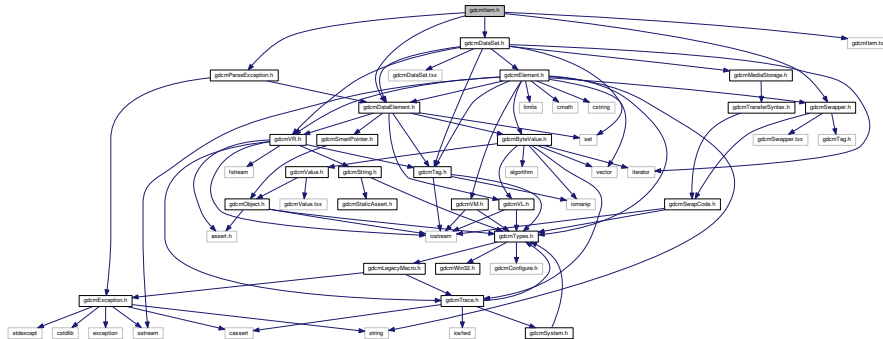
IPPSorter Implement a simple *Image* Position (*Patient*) sorter, along the *Image Orientation* (*Patient*) direction. This algorithm does NOT support duplicate and will FAIL in case of duplicate IPP.

Namespaces

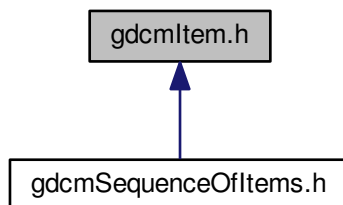
- [gdcm](#)

28.134 gdcmItem.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmDataSet.h"
#include "gdcmParseException.h"
#include "gdcmSwapper.h"
#include "gdcmItem.txx"
Include dependency graph for gdcmItem.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::Item](#)

Class to represent an [Item](#) A component of the value of a Data [Element](#) that is of [Value Representation Sequence of Items](#). An [Item](#) contains a Data Set . See PS 3.5 7.5.1 [Item Encoding Rules](#) Each [Item](#) of a Data [Element](#) of [VR SQ](#) shall be encoded as a DICOM Standard Data [Element](#) with a specific Data [Element Tag](#) of [Value](#) (FFFE,E000). The [Item Tag](#) is followed by a 4 byte [Item Length](#) field encoded in one of the following two ways *Explicit/ Implicit*.

Namespaces

- [gdcm](#)

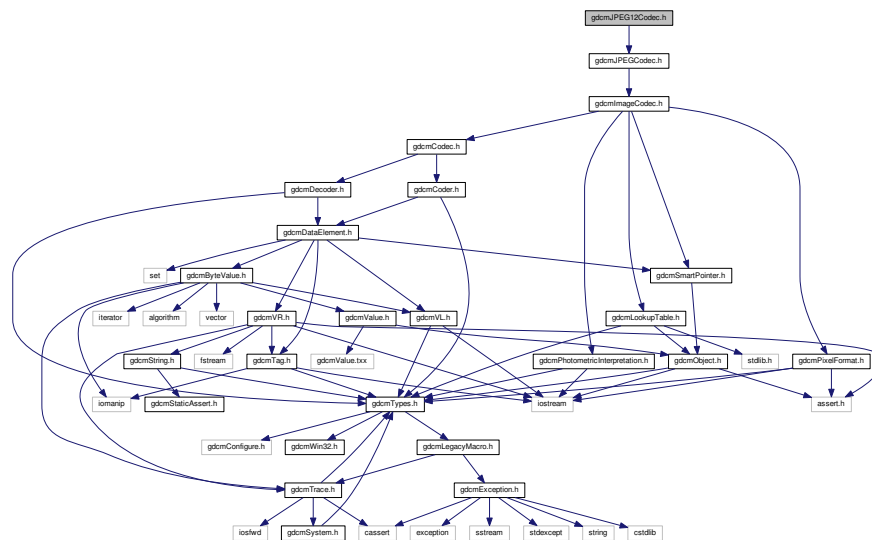
Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Item &val)`

28.135 gdcmJPEG12Codec.h File Reference

```
#include "gdcmJPEGCodec.h"
```

Include dependency graph for gdcmJPEG12Codec.h:



Classes

- class `gdcm::JPEG12Codec`
Class to do JPEG 12bits (lossy & lossless)

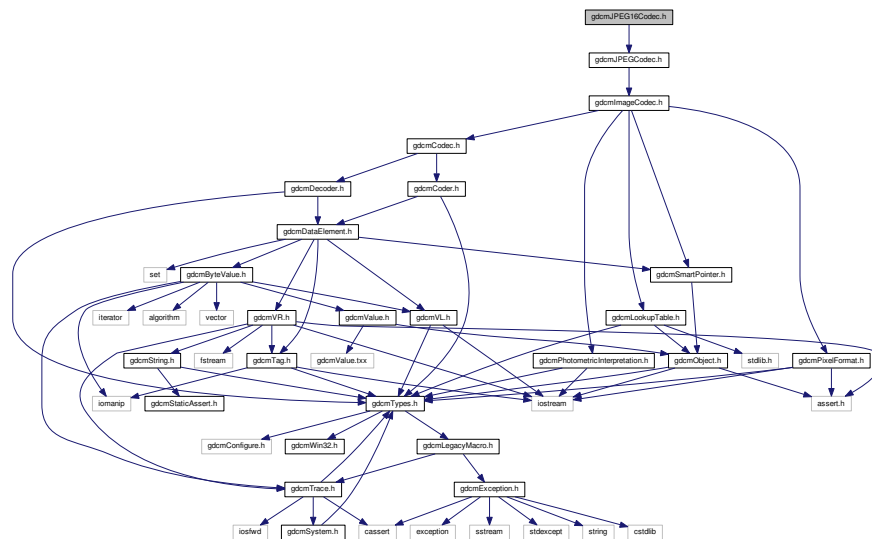
Namespaces

- `gdcm`

28.136 gdcmJPEG16Codec.h File Reference

```
#include "gdcmJPEGCodec.h"
```

Include dependency graph for `gdcmJPEG16Codec.h`:



Classes

- class [gdcm::JPEG16Codec](#)

Class to do JPEG 16bits (lossless)

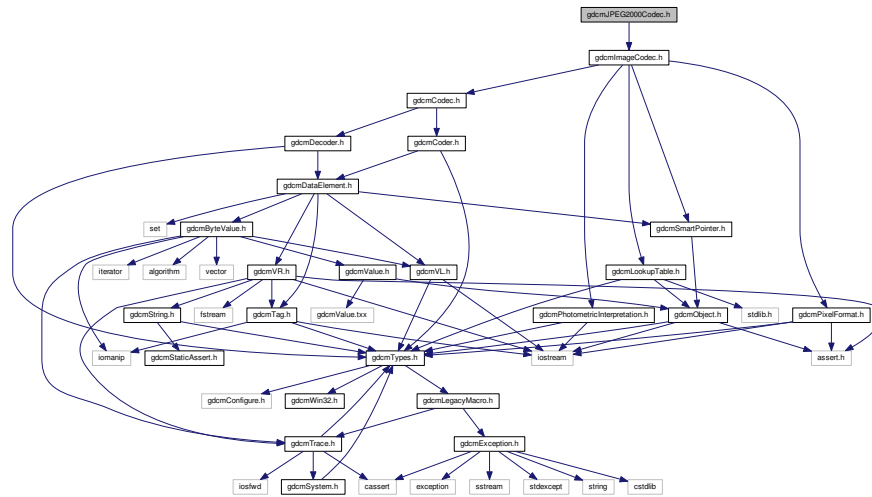
Namespaces

- [gdcm](#)

28.137 gdcmJPEG2000Codec.h File Reference

```
#include "gdcmImageCodec.h"
```


Include dependency graph for gdcmJPEG2000Codec.h:



Classes

- class [gdcm::JPEG2000Codec](#)

Class to do JPEG 2000.

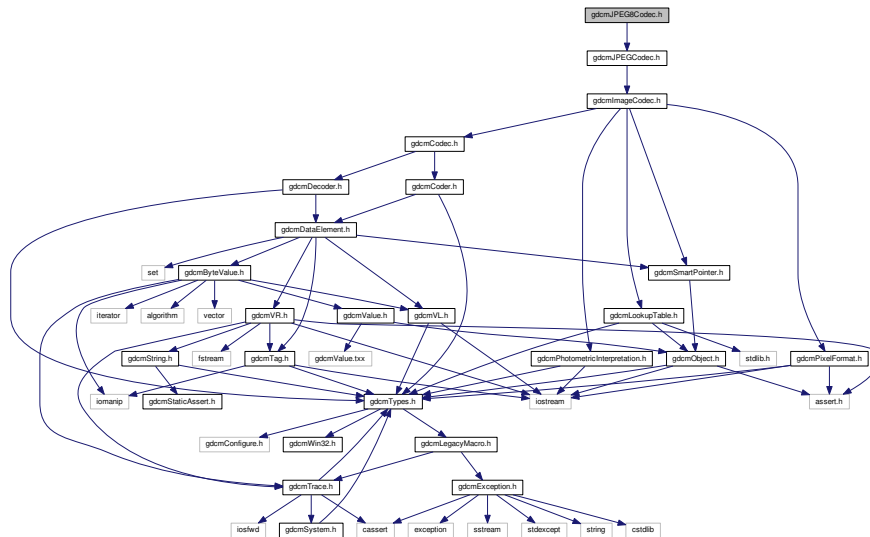
Namespaces

- [gdcm](#)

28.138 gdcmJPEG8Codec.h File Reference

```
#include "gdcmJPEGCodec.h"
```

Include dependency graph for `gdcmJPEG8Codec.h`:



Classes

- class [gdcm::JPEG8Codec](#)

Class to do JPEG 8bits (lossy & lossless)

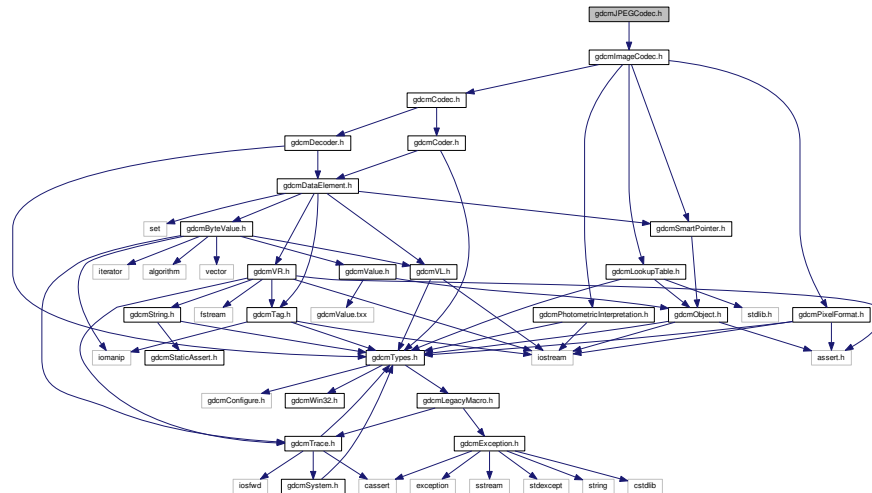
Namespaces

- [gdcm](#)

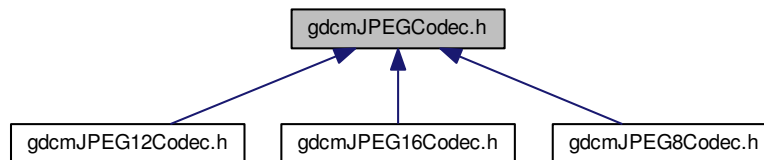
28.139 gdcmJPEGCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmJPEGCodec.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::JPEGCodec](#)

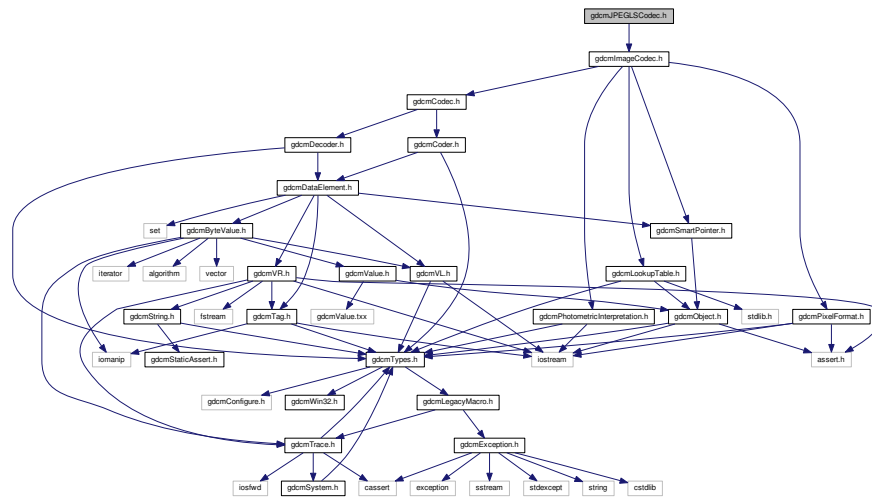
JPEG codec Class to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispach in between the different codec implementation: [JPEG8Codec](#), [JPEG12Codec](#) & [JPEG16Codec](#) It also support inconsistency in between DICOM header and JPEG compressed stream [ImageCodec](#) implementation for the JPEG case.

Namespaces

- [gdcm](#)

28.140 gdcmJPEGCodec.h File Reference

```
#include "gdcmImageCodec.h"
```



- class `gdcm::JPEGLSCodec`

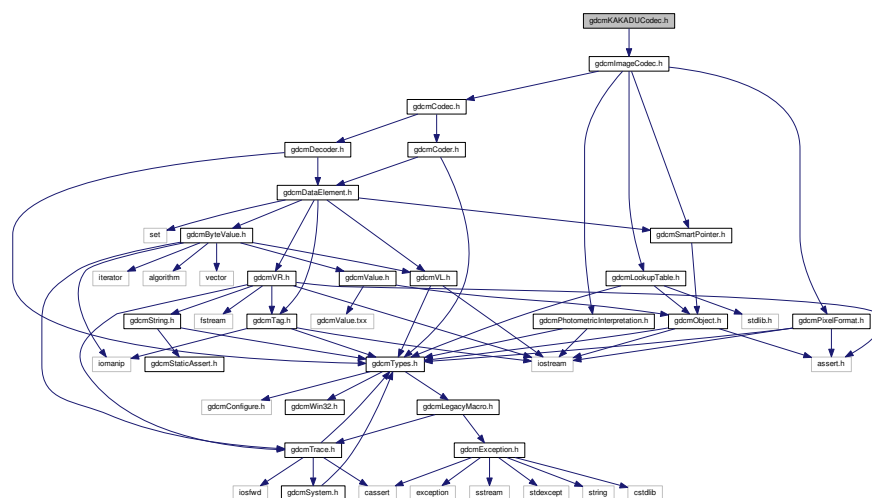
- **gdcm**

```
#include "gdcmFile.h"
#include "gdcmDataElement.h"
```

- class `gdcm::JSON`

- **gdcm**

```
#include "gdcmImageCodec.h"
Include dependency graph for gdcmKAKADUCodec.h:
```



Classes

- class [gdcm::KAKADUCodec](#)
KAKADUCodec.

Namespaces

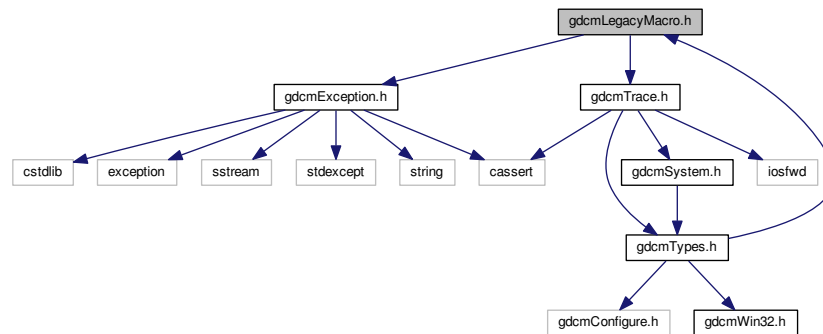
- [gdcm](#)

28.143 gdcmLegacyMacro.h File Reference

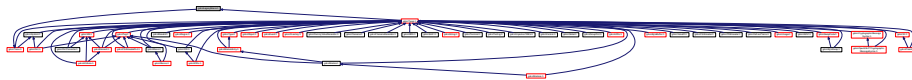
```
#include "gdcmException.h"
```

```
#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.")`

28.143.1 Macro Definition Documentation

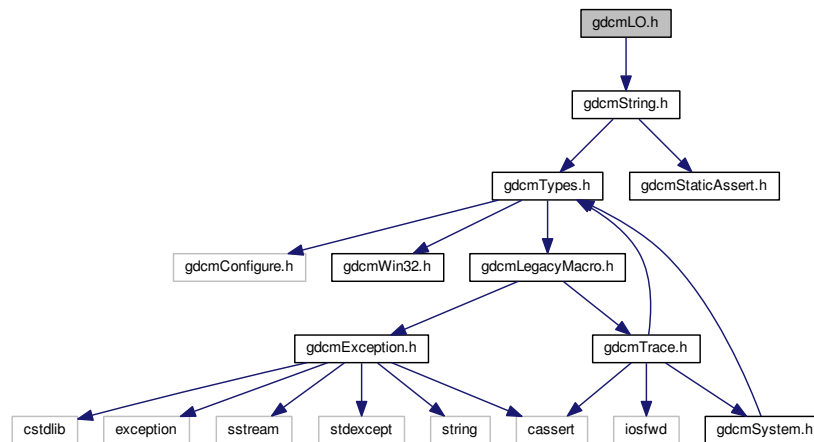
28.143.1.1 `#define GDCM_LEGACY(method) method;`

28.143.1.2 `#define GDCM_LEGACY_BODY(method, version) gdcmWarningMacro(#method " was deprecated for " version " and will be removed in a future version.")`

28.143.1.3 `#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.")`

28.144 gdcmLO.h File Reference

`#include "gdcmString.h"`
 Include dependency graph for gdcmLO.h:



Classes

- class `gdcm::LO`

Namespaces

- `gdcm`

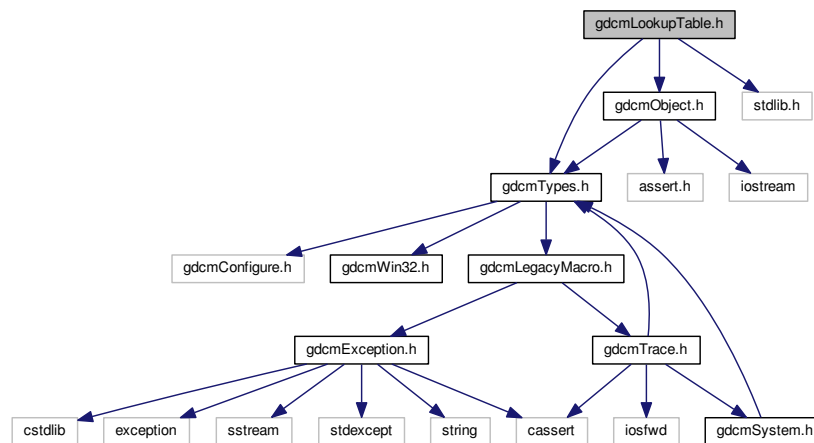
28.145 gdcmLookupTable.h File Reference

```

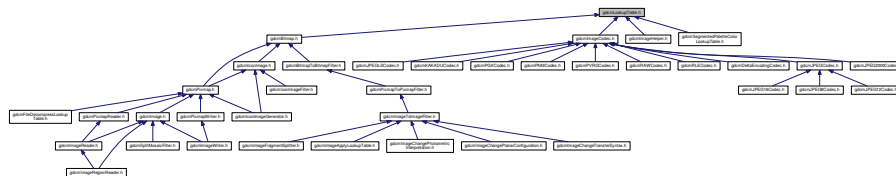
#include "gdcmTypes.h"
#include "gdcmObject.h"
#include <stdlib.h>

```

Include dependency graph for `gdcmlLookupTable.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcml::LookupTable](#)
LookupTable class.

Namespaces

- [gdcml](#)

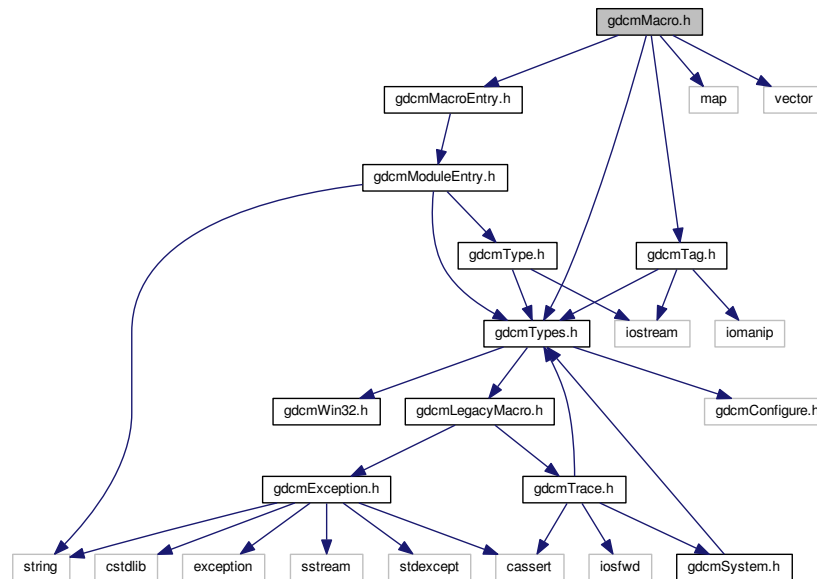
28.146 gdcmlMacro.h File Reference

```

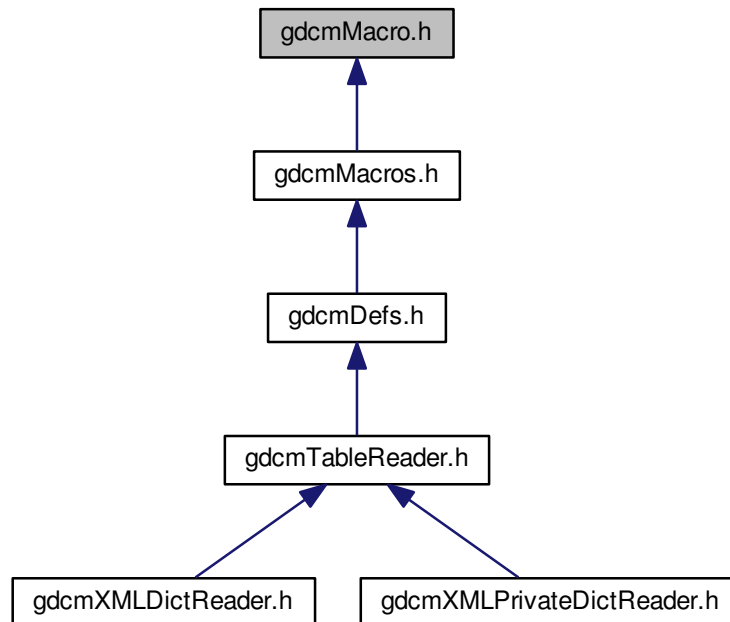
#include "gdcmlTypes.h"
#include "gdcmlTag.h"
#include "gdcmlMacroEntry.h"
#include <map>
#include <vector>

```


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](#)

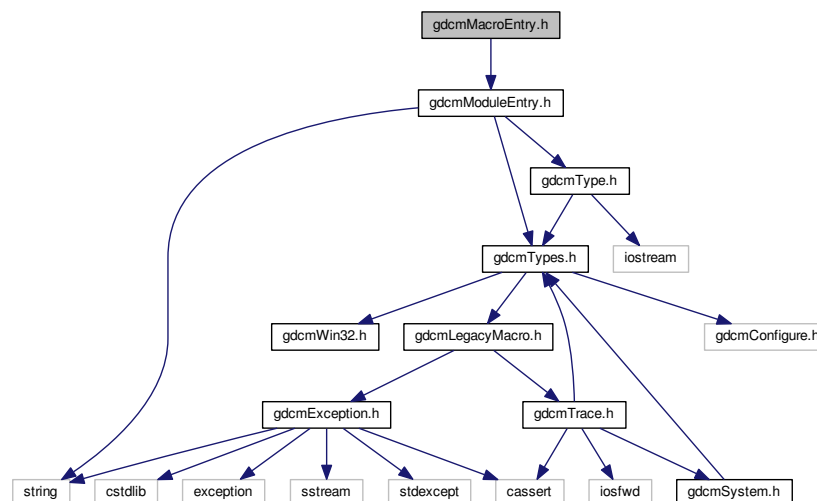
Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Macro &_val)`

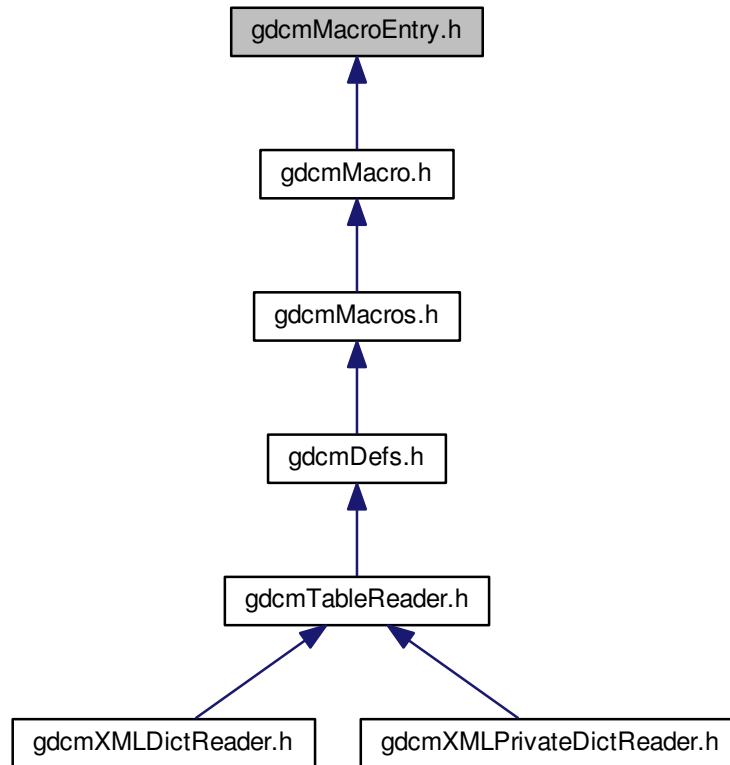
28.147 gdcmMacroEntry.h File Reference

```
#include "gdcmModuleEntry.h"
```

Include dependency graph for `gdcmMacroEntry.h`:



This graph shows which files directly or indirectly include this file:



Macros

- `#define GDCMMACROENTRY_H`

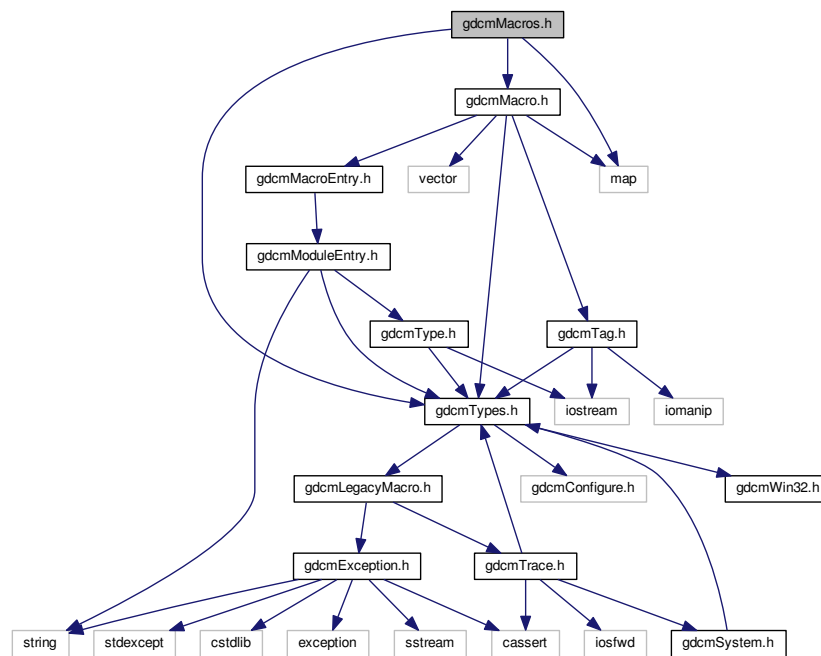
28.147.1 Macro Definition Documentation

28.147.1.1 `#define GDCMMACROENTRY_H`

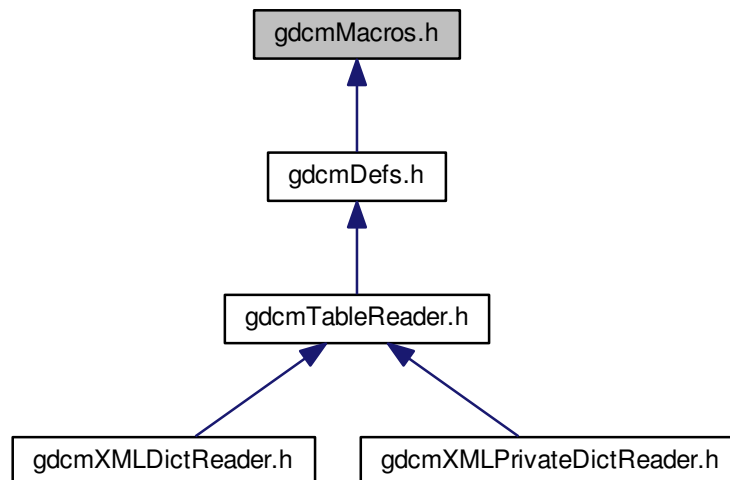
28.148 gdcmMacros.h File Reference

```
#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](#)

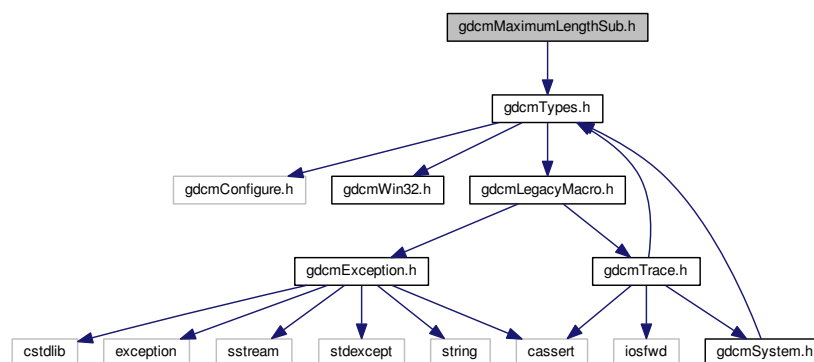
Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Macros &_val)`

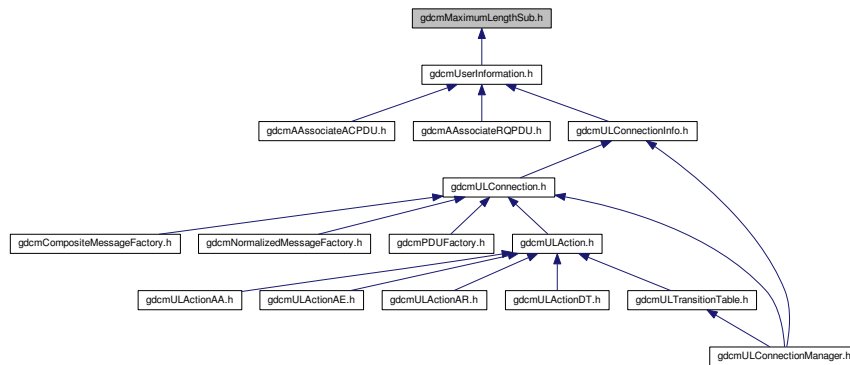
28.149 gdcmMaximumLengthSub.h File Reference

```
#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 Annex D Table D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

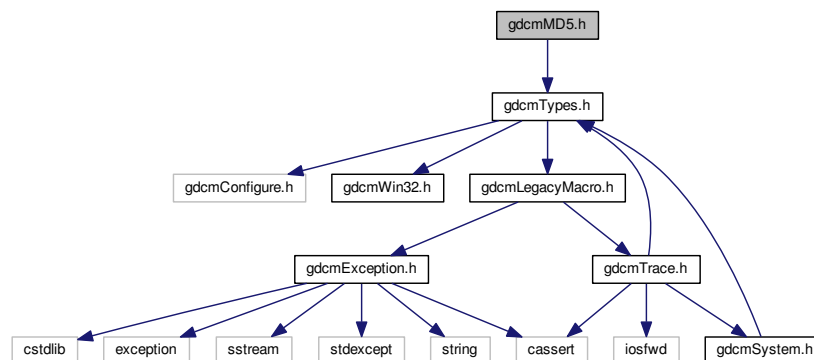
Namespaces

- [gdcm](#)
- [gdcm::network](#)

28.150 gdcmMD5.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmMD5.h:



Classes

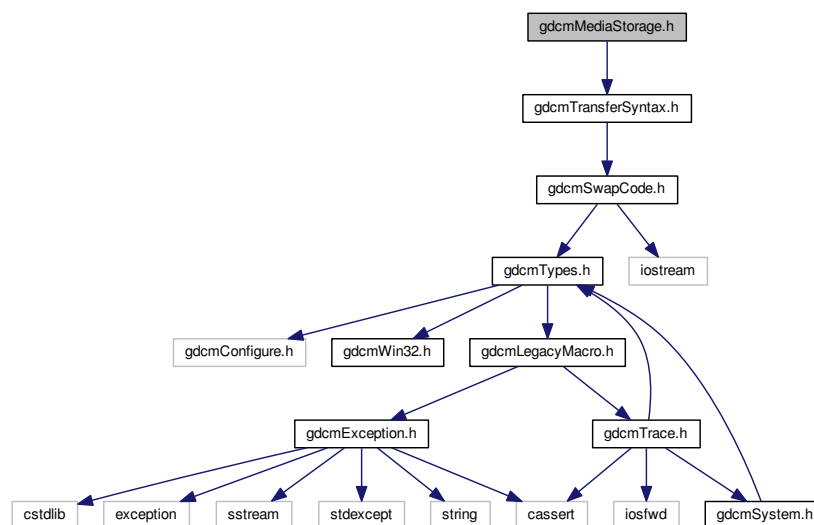
- class [gdcm::MD5](#)
Class for [MD5](#).

Namespaces

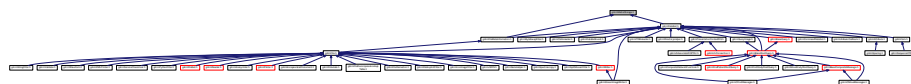
- [gdcm](#)

28.151 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

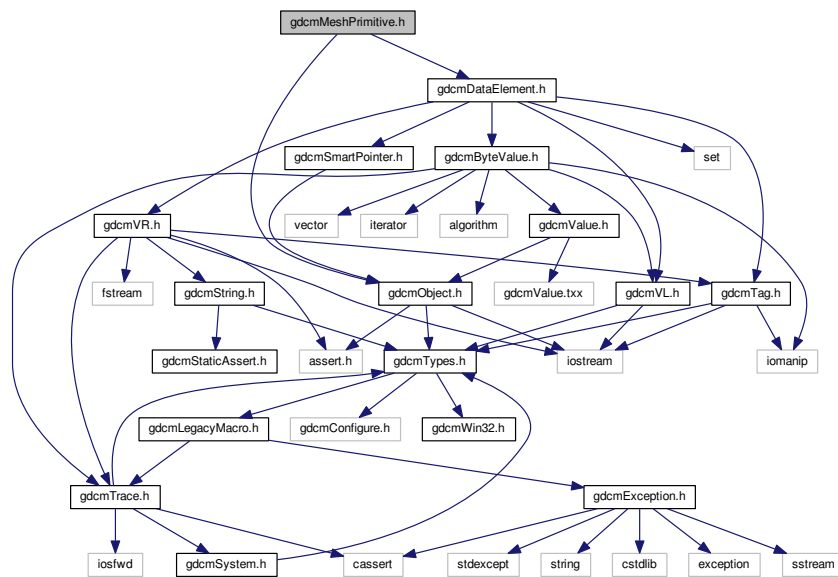
- `std::ostream & gdcm::operator<< (std::ostream &_os, const MediaStorage &ms)`

28.152 gdcmMeshPrimitive.h File Reference

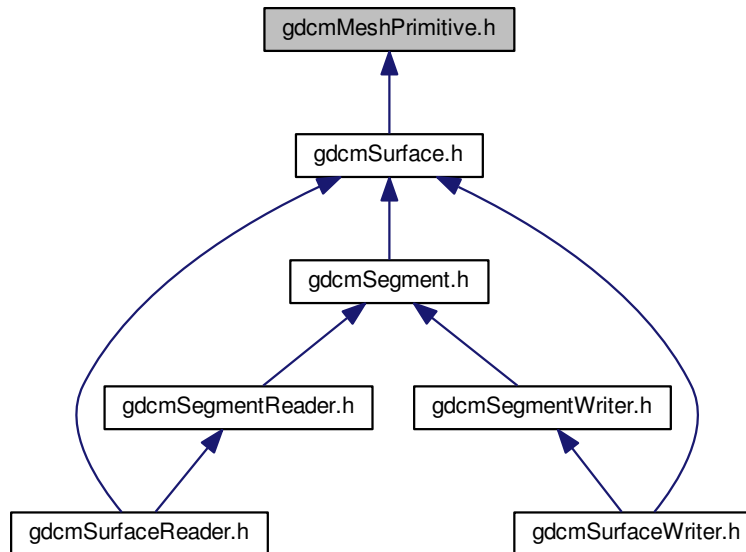
```
#include <gdcmObject.h>
```

```
#include <gdcmDataElement.h>
```

Include dependency graph for `gdcmMeshPrimitive.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::MeshPrimitive](#)

This class defines surface mesh primitives. It is designed from surface mesh primitives macro.

Namespaces

- [gdcm](#)

28.153 gdcmModalityPerformedProcedureStepCreateQuery.h File Reference

```
#include "gdcmBaseQuery.h"
```


Classes

- class [gdcm::ModalityPerformedProcedureStepSetQuery](#)

[ModalityPerformedProcedureStepSetQuery](#) contains: the class which will produce a dataset for n-set for Modality Performed Procedure Step sop class.

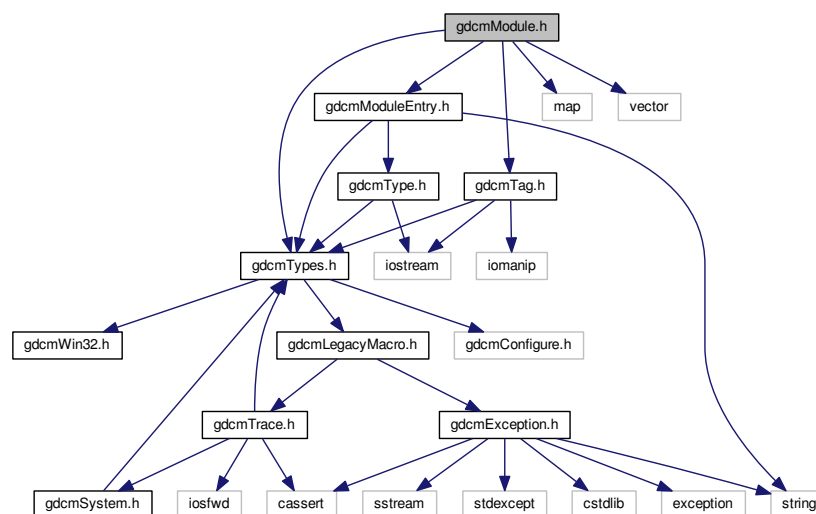
Namespaces

- [gdcm](#)

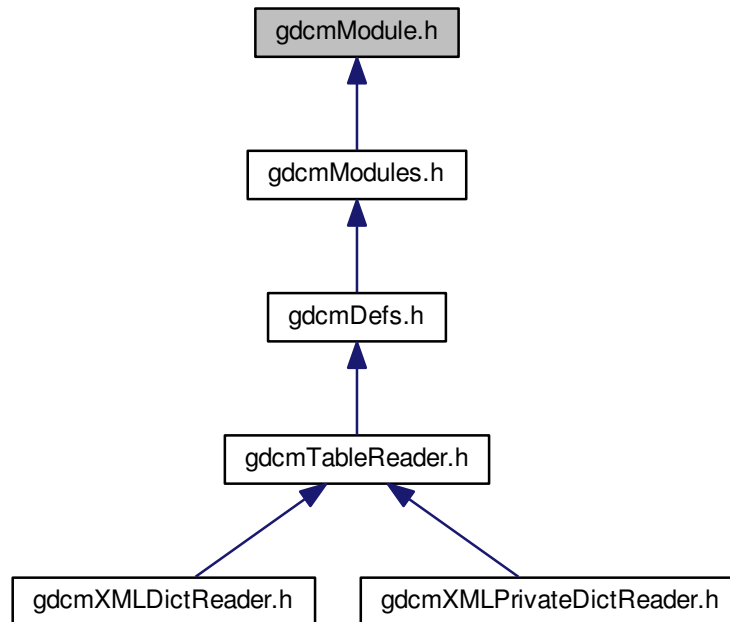
28.155 gdcmModule.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmModuleEntry.h"
#include <map>
#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](#)

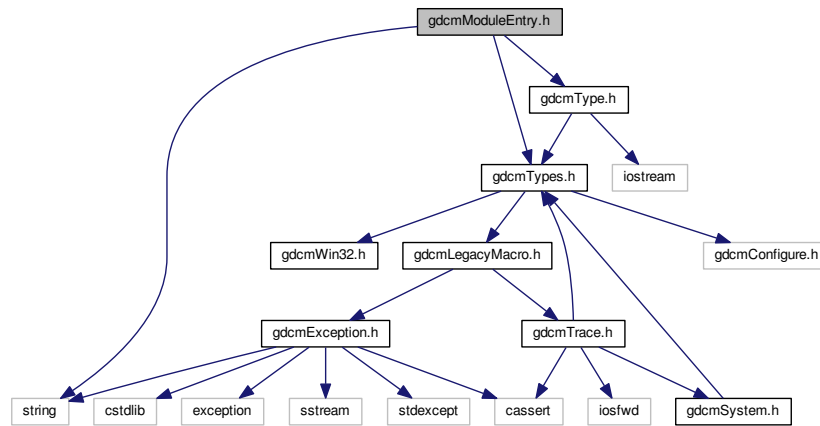
Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Module &_val)`

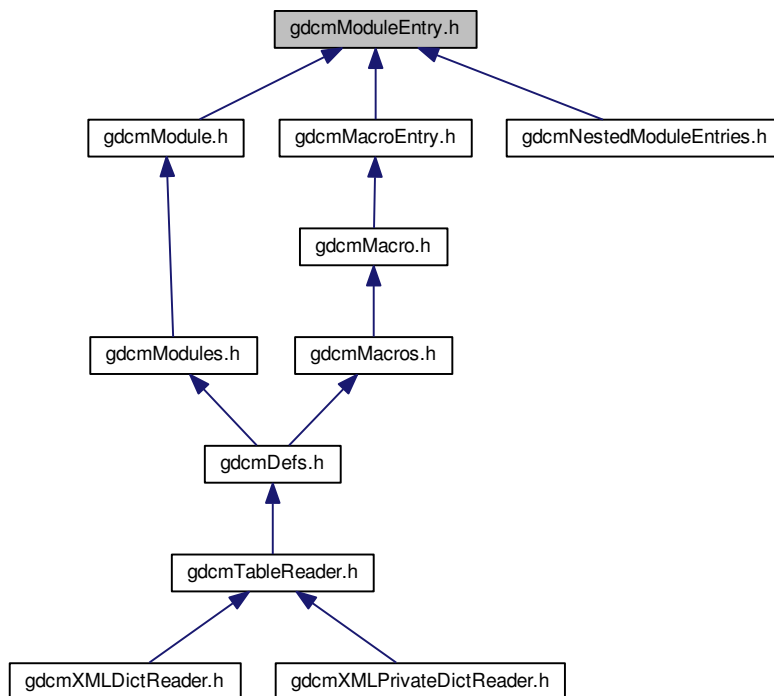
28.156 gdcModuleEntry.h File Reference

```
#include "gdcmTypes.h"  
#include "gdcmType.h"  
#include <string>
```

Include dependency graph for gdcModuleEntry.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::ModuleEntry](#)
Class for representing a [ModuleEntry](#).

Namespaces

- [gdcm](#)

Typedefs

- typedef ModuleEntry [gdcm::MacroEntry](#)

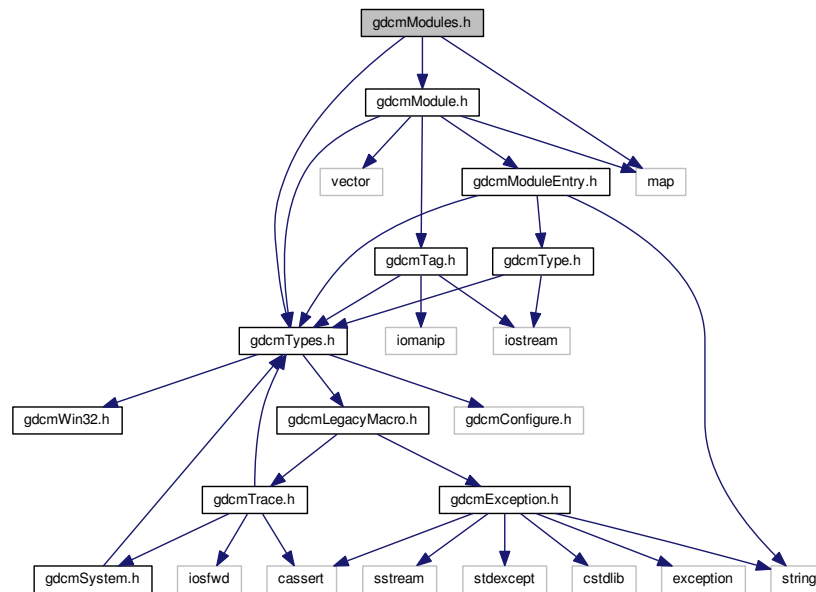
Functions

- std::ostream & [gdcm::operator<<](#) (std::ostream &_os, const ModuleEntry &_val)

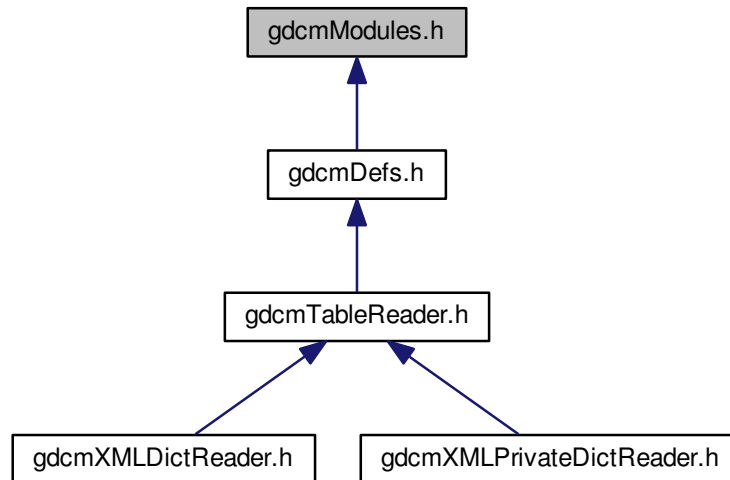
28.157 gdcmModules.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmModule.h"
#include <map>
```

Include dependency graph for gdcmModules.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::Modules](#)

Class for representing a [Modules](#).

Namespaces

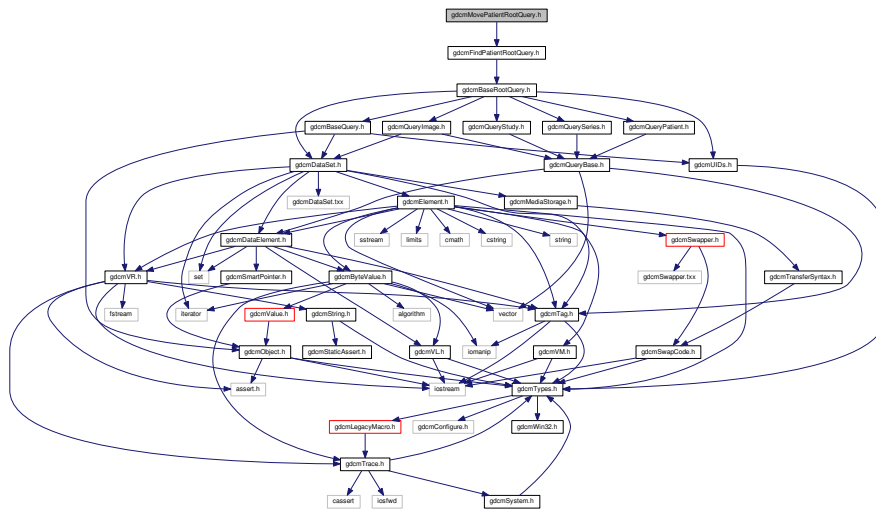
- [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Modules &_val)`

28.158 gdcmMovePatientRootQuery.h File Reference

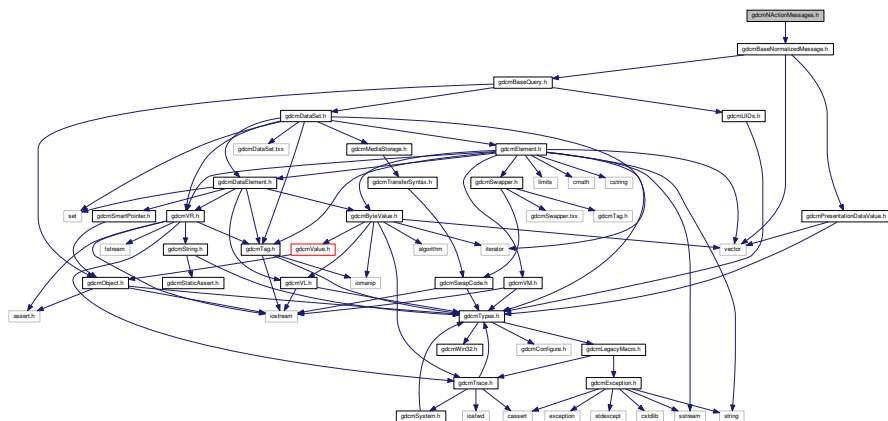
```
#include "gdcmFindPatientRootQuery.h"
```



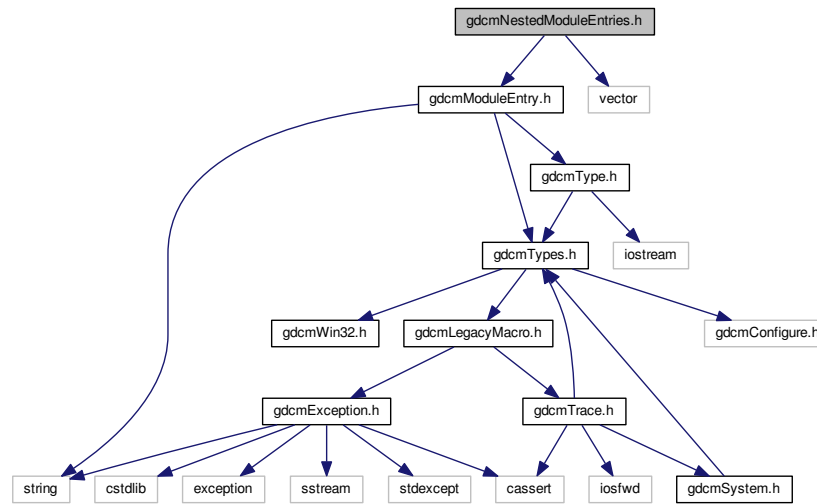
- class `gdcmm::MoveStudyRootQuery`
MoveStudyRootQuery contains: the class which will produce a dataset for C-MOVE with study root.

- **gdcm**

```
#include "gdcmBaseNormalizedMessage.h"
Include dependency graph for gdcmNActionMessages.h:
```



Include dependency graph for `gdcMNestedModuleEntries.h`:



Classes

- class [gdcM::NestedModuleEntries](#)

Class for representing a [NestedModuleEntries](#).

Namespaces

- [gdcM](#)

Typedefs

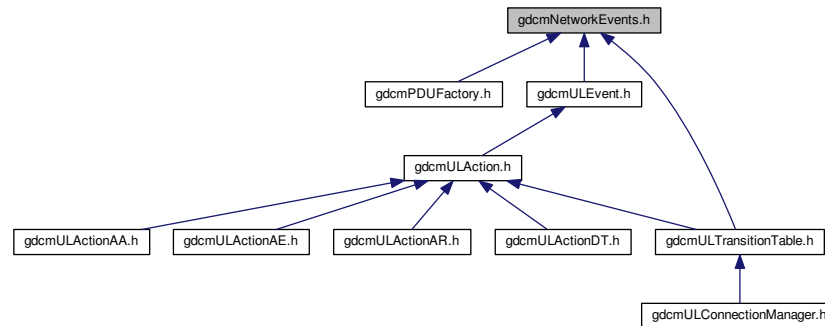
- typedef `NestedModuleEntries` [gdcM::NestedMacroEntries](#)

Functions

- `std::ostream & gdcM::operator<< (std::ostream &_os, const NestedModuleEntries &_val)`

28.164 gdcNetworkEvents.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- [gdc](#)
- [gdc::network](#)

Enumerations

- `enum gdc::network::EEventID {`
`gdc::network::eAASSOCIATERequestLocalUser = 0,`
`gdc::network::eTransportConnConfirmLocal,`
`gdc::network::eASSOCIATE_ACPDUreceived,`
`gdc::network::eASSOCIATE_RJPDUreceived,`
`gdc::network::eTransportConnIndicLocal,`
`gdc::network::eAASSOCIATE_RQPDUreceived,`
`gdc::network::eAASSOCIATEresponseAccept,`
`gdc::network::eAASSOCIATEresponseReject,`
`gdc::network::ePDATArequest,`
`gdc::network::ePDATATFPDU,`
`gdc::network::eARELEASERequest,`
`gdc::network::eARELEASE_RQPDUReceivedOpen,`
`gdc::network::eARELEASE_RPPDUReceived,`
`gdc::network::eARELEASEResponse,`
`gdc::network::eAABORTRequest,`
`gdc::network::eAABORTPDUReceivedOpen,`
`gdc::network::eTransportConnectionClosed,`
`gdc::network::eARTIMTimerExpired,`
`gdc::network::eUnrecognizedPDUReceived,`
`gdc::network::eEventDoesNotExist }`

Variables

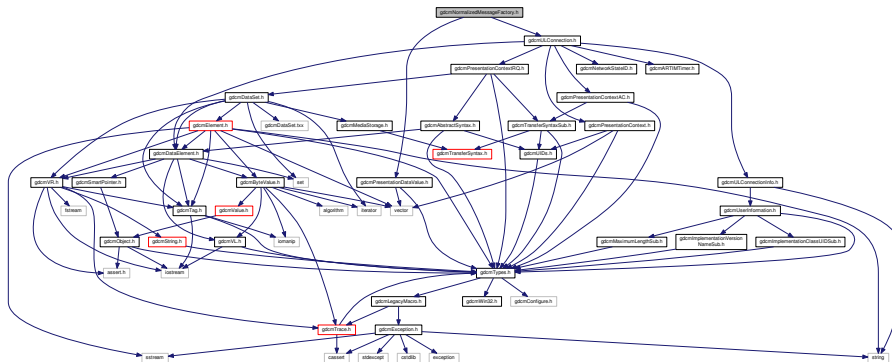
- `const int gdc::network::cMaxEventID = eEventDoesNotExist`

[illegible]

- class `gdcmm::network::NGetRQ`
NGetRQ this file defines the messages for the *nget* action.
- class `gdcmm::network::NGetRSP`
NGetRSP this file defines the messages for the *nget* action.

- `gdcm`
- `gdcm::network`

```
#include "gdcmPresentationDataValue.h"
#include "gdcmULConnection.h"
Include dependency graph for gdcmNormalizedMessageFactory.h:
```

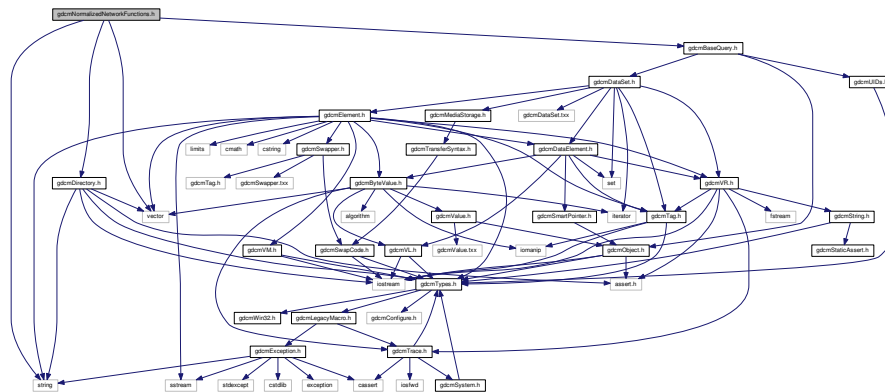


- class `gdcm::network::NormalizedMessageFactory`

- gdc
- gdc::network

```
#include "gdcmlDirectory.h"
#include "gdcmlBaseQuery.h"
#include <vector>
#include <string>
```

Include dependency graph for gdcmlNormalizedNetworkFunctions.h:

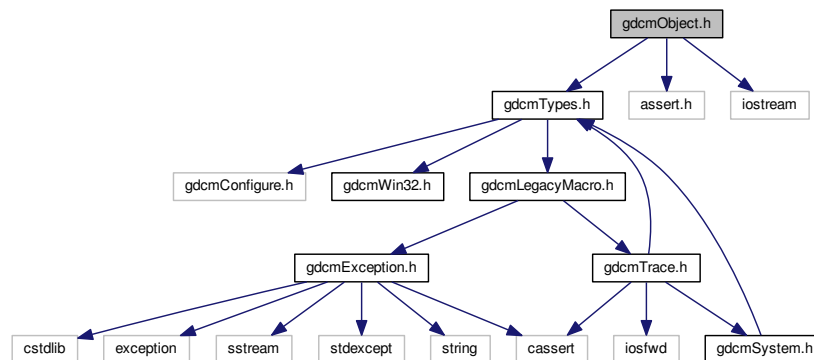


- class `gdc::NormalizedNetworkFunctions`

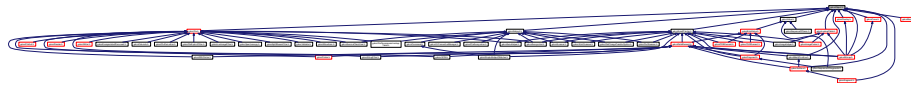
Normalized Network Functions These functions provide a generic API to the DICOM functions implemented in GDCM. Advanced users can use this code as a template for building their own versions of these functions (for instance, to provide progress bars or some other way of handling returned query information), 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:

- **gdcm**

Include dependency graph for gdcmObject.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::Object](#)
Object.
- class [gdcm::SmartPointer< ObjectType >](#)
Class for Smart Pointer.

Namespaces

- [gdcm](#)

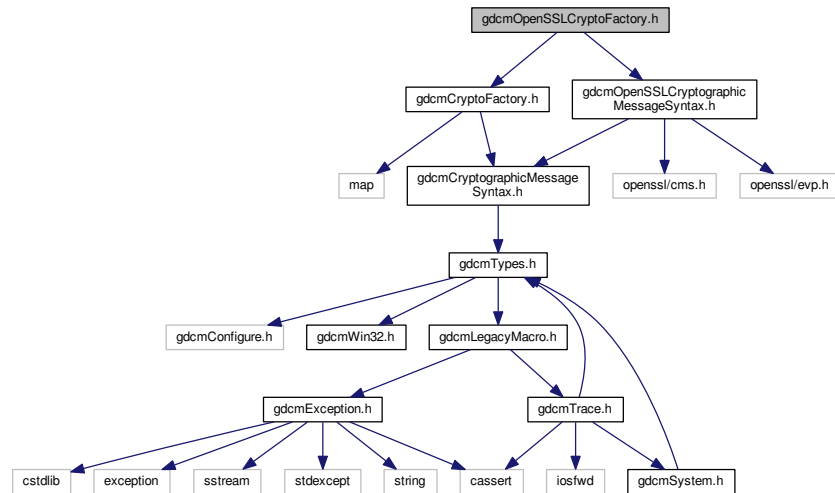
Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Object &obj)`

28.172 gdcmOpenSSLCryptoFactory.h File Reference

```
#include "gdcmCryptoFactory.h"
#include "gdcmOpenSSLCryptographicMessageSyntax.h"
```

Include dependency graph for `gdcOpenSSLCryptoFactory.h`:



Classes

- class [gdc::OpenSSLCryptoFactory](#)

Namespaces

- [gdc](#)

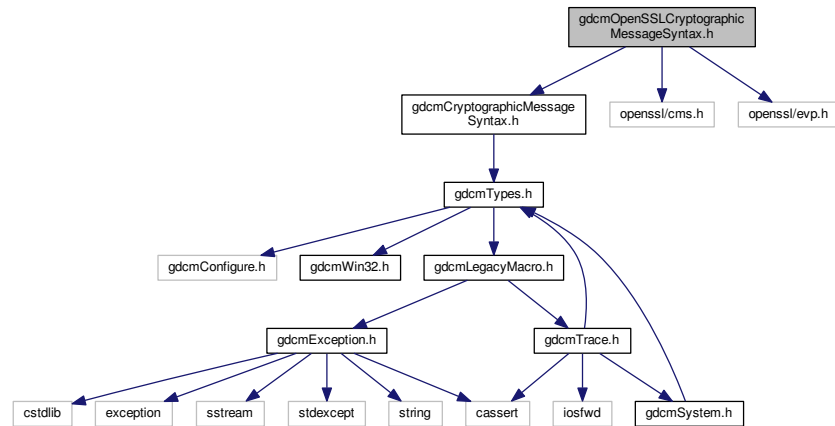
28.173 gdcOpenSSLCryptographicMessageSyntax.h File Reference

```

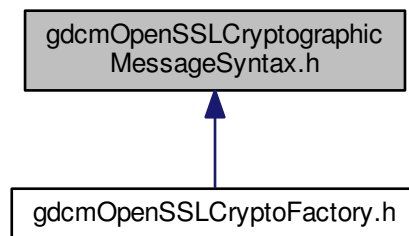
#include "gdcCryptographicMessageSyntax.h"
#include <openssl/cms.h>
#include <openssl/evp.h>

```

Include dependency graph for gdcmOpenSSLCryptographicMessageSyntax.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::OpenSSLCryptographicMessageSyntax](#)

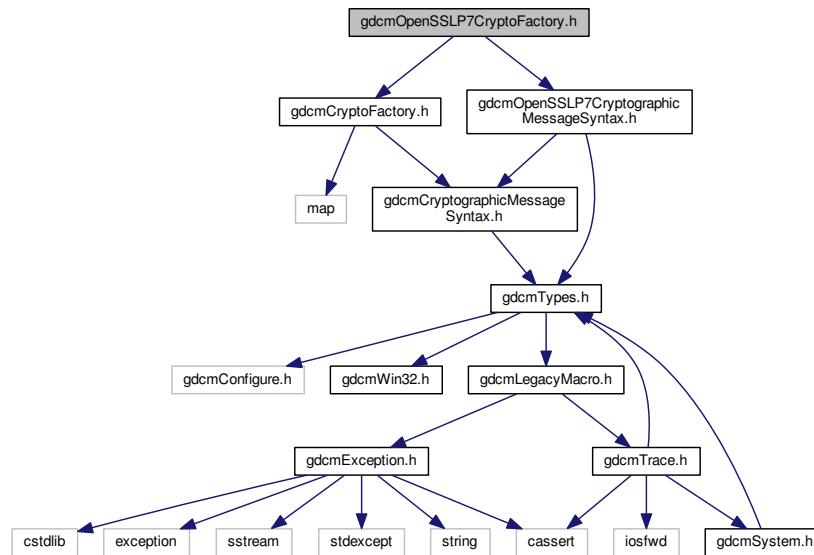
Namespaces

- [gdcm](#)

28.174 gdcmOpenSSLP7CryptoFactory.h File Reference

```
#include "gdcmCryptoFactory.h"
#include "gdcmOpenSSLP7CryptographicMessageSyntax.h"
```

Include dependency graph for `gdcOpenSSLP7CryptoFactory.h`:



Classes

- class `gdc::OpenSSLP7CryptoFactory`

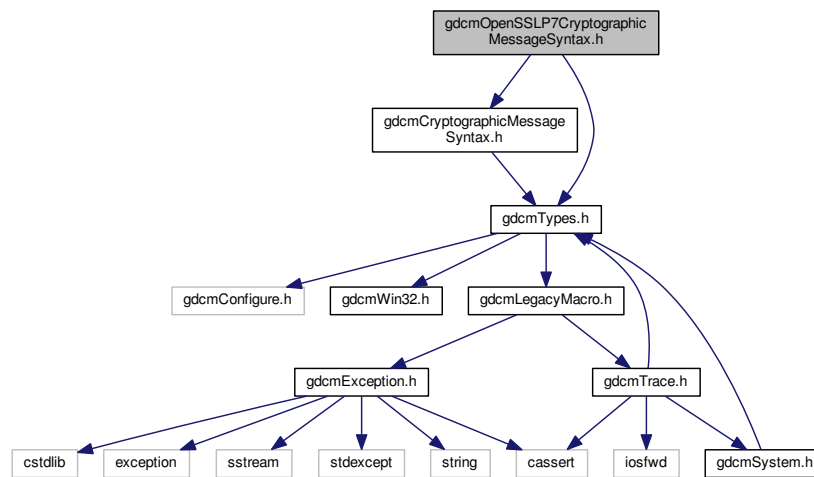
Namespaces

- `gdc`

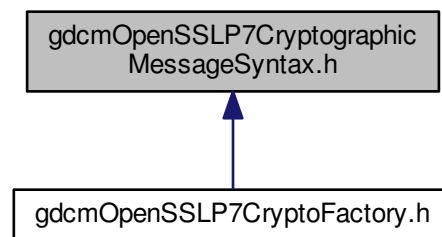
28.175 gdcOpenSSLP7CryptographicMessageSyntax.h File Reference

```
#include "gdcCryptographicMessageSyntax.h"
#include "gdcTypes.h"
```

Include dependency graph for gdcmOpenSSLP7CryptographicMessageSyntax.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::OpenSSLP7CryptographicMessageSyntax](#)

Class for [CryptographicMessageSyntax](#) encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities.

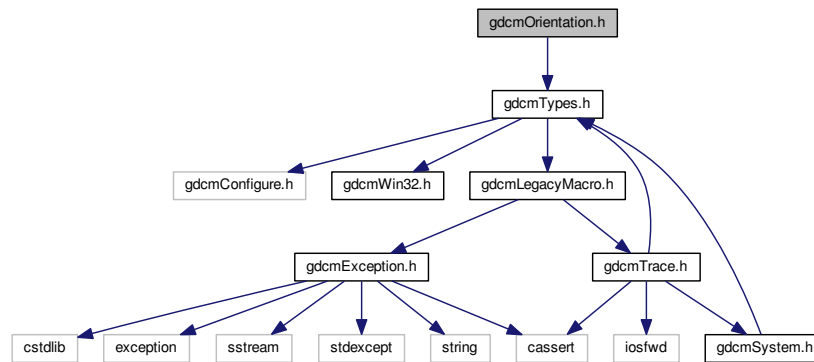
Namespaces

- [gdcm](#)

28.176 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)`

28.177 gdcmOverlay.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmObject.h"
```



```

graph TD
    gdcmOverlay.h[gdcmOverlay.h] --> gdcmObject.h[gdcmObject.h]
    gdcmOverlay.h --> gdcmTypes.h[gdcmTypes.h]
    gdcmOverlay.h --> assert.h[assert.h]
    gdcmOverlay.h --> iostream[iostream]
    gdcmTypes.h --> gdcmConfigure.h[gdcmConfigure.h]
    gdcmTypes.h --> gdcmWin32.h[gdcmWin32.h]
    gdcmTypes.h --> gdcmLegacyMacro.h[gdcmLegacyMacro.h]
    gdcmTypes.h --> gdcmException.h[gdcmException.h]
    gdcmTypes.h --> gdcmTrace.h[gdcmTrace.h]
    gdcmLegacyMacro.h --> gdcmException.h
    gdcmLegacyMacro.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]
    gdcmTrace.h --> iosfwd
    gdcmTrace.h --> gdcmSystem.h[gdcmSystem.h]
    gdcmSystem.h --> gdcmTrace.h
  
```

[illegible]

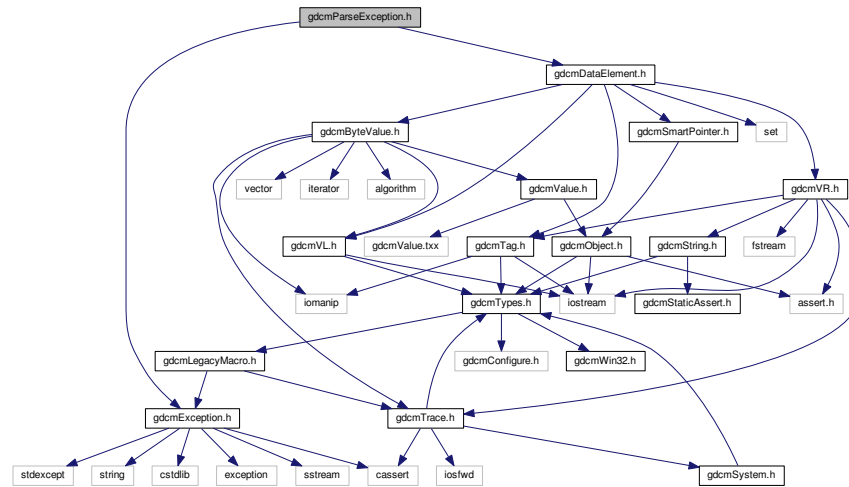
- class `gdcm::Overlay`
Overlay class.

- **gdcm**

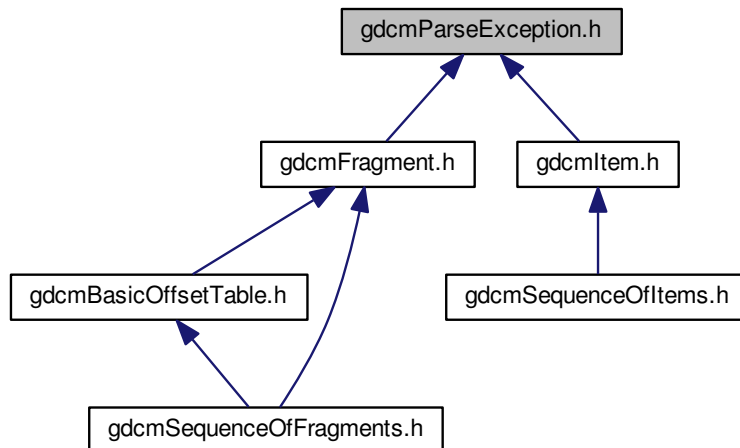
28.179 gdcmParseException.h File Reference

Generated on Mon Dec 21 2015 23:27:54 for GDCM by Doxygen

Include dependency graph for `gdcmParseException.h`:



This graph shows which files directly or indirectly include this file:



Classes

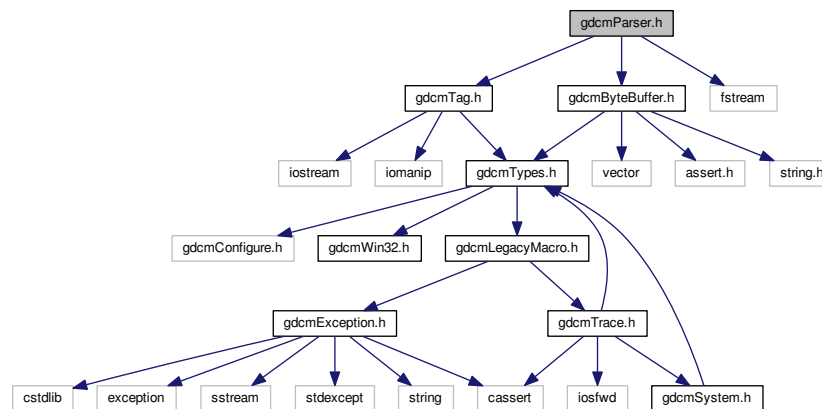
- class [gdcm::ParseException](#)
ParseException Standard exception handling object.

Namespaces

- [gdcm](#)

28.180 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)

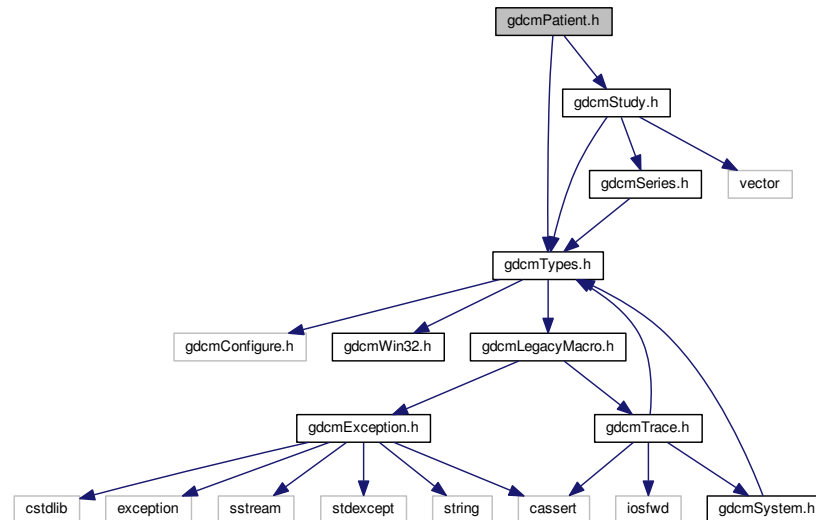
Namespaces

- [gdcm](#)

28.181 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](#)

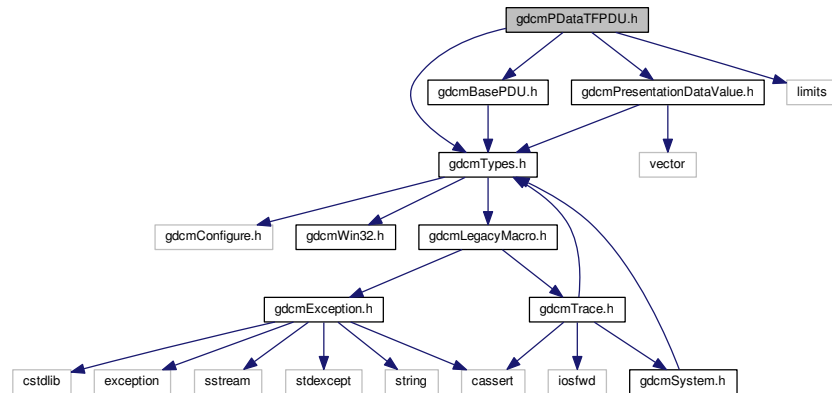
28.182 gdcmPDataTFPDU.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmPresentationDataValue.h"
#include "gdcmBasePDU.h"
#include <limits>

```

Include dependency graph for gdcmPDataTFPDU.h:



Classes

- class [gdcm::network::PDataTFPDU](#)

PDataTFPDU Table 9-22 P-DATA-TF PDU FIELDS.

Namespaces

- [gdcm](#)
- [gdcm::network](#)

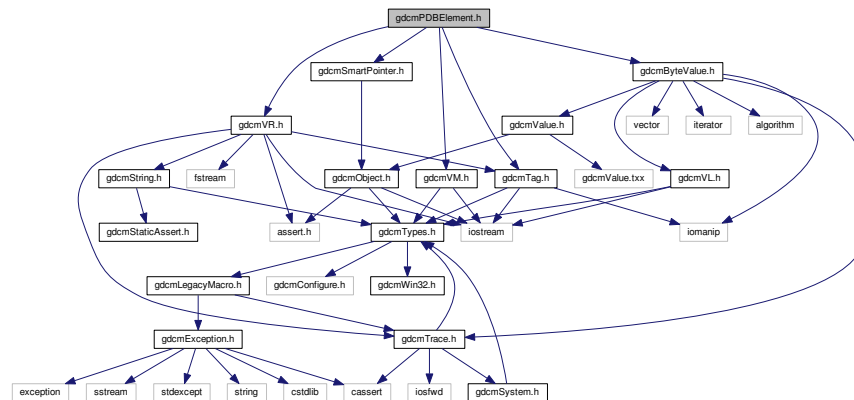
28.183 gdcmPDBelement.h File Reference

```

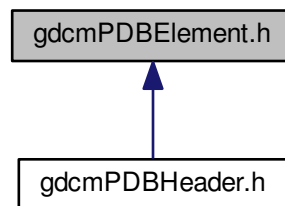
#include "gdcmTag.h"
#include "gdcmVM.h"
#include "gdcmVR.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"

```

Include dependency graph for `gdcmPDBElement.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::PDBElement`
Class to represent a PDB [Element](#).

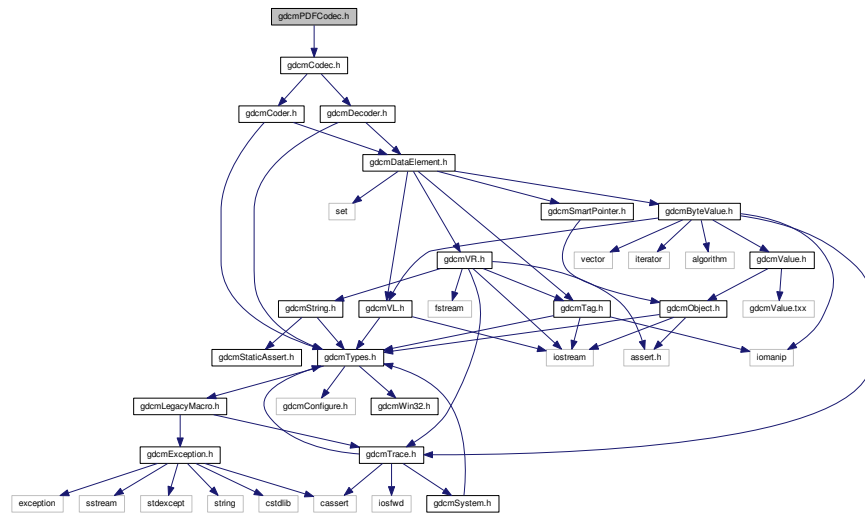
Namespaces

- `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const PDBElement &val)`

Include dependency graph for gdcPDFCodec.h:



Classes

- class `gdcm::PDFCodec`
PDFCodec class.

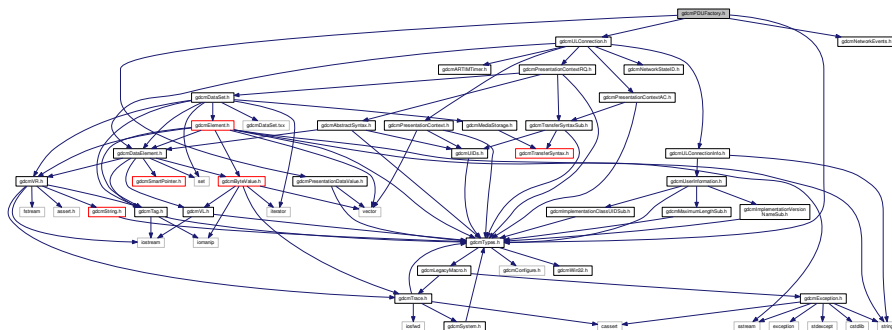
Namespaces

- **gdcm**

28.187 gdcmPDUFactory.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmNetworkEvents.h"
#include "gdcmULConnection.h"
#include "gdcmPresentationDataValue.h"
```

Include dependency graph for gdcmPDUFactory.h:



Classes

- class [gdcm::network::PDUFactory](#)

PDUFactory basically, given an initial byte, construct the appropriate PDU. This way, the event loop doesn't have to know about all the different PDU types.

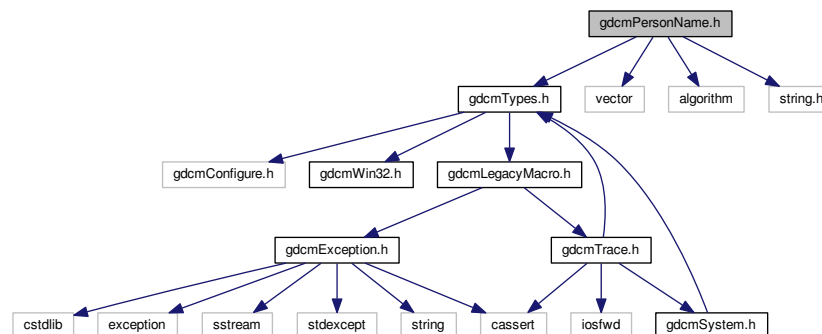
Namespaces

- [gdcm](#)
- [gdcm::network](#)

28.188 gdcmPersonName.h File Reference

```
#include "gdcmTypes.h"
#include <vector>
#include <algorithm>
#include <string.h>
```

Include dependency graph for gdcmPersonName.h:



Classes

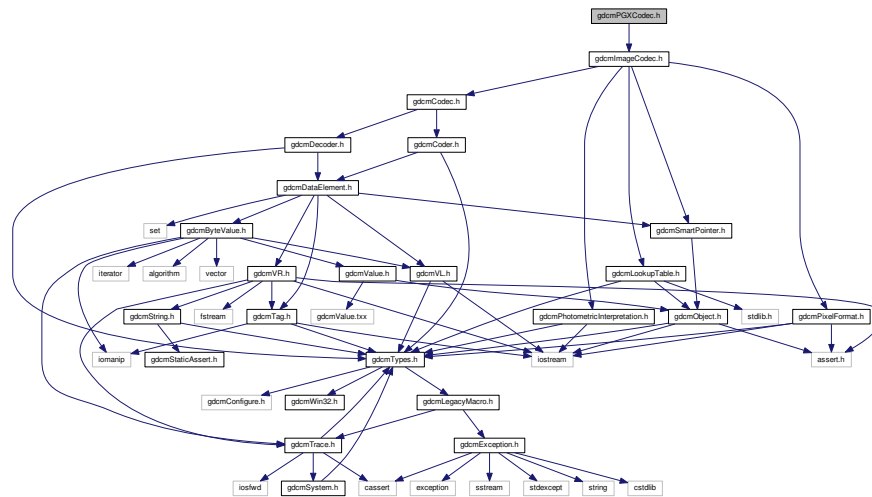
- class [gdcm::PersonName](#)
PersonName class.

Namespaces

- [gdcm](#)

28.189 gdcmPGXCodec.h File Reference

```
#include "gdcmImageCodec.h"
```



Classes

- class `gdcm::PGXCodec`

Class to do PGX See PGX as used in JPEG 2000 implementation and reference images.

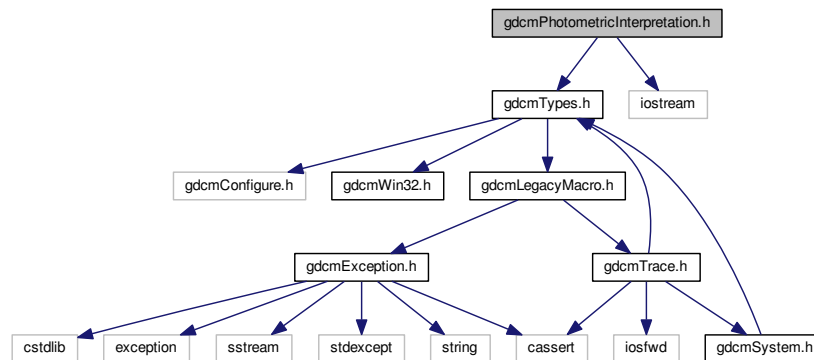
Namespaces

- **gdcm**

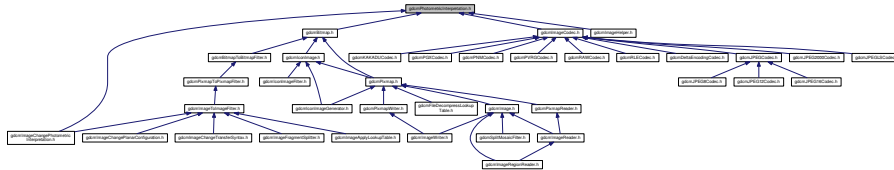
28.190 gdcMPhotometricInterpretation.h File Reference

```
#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](#)

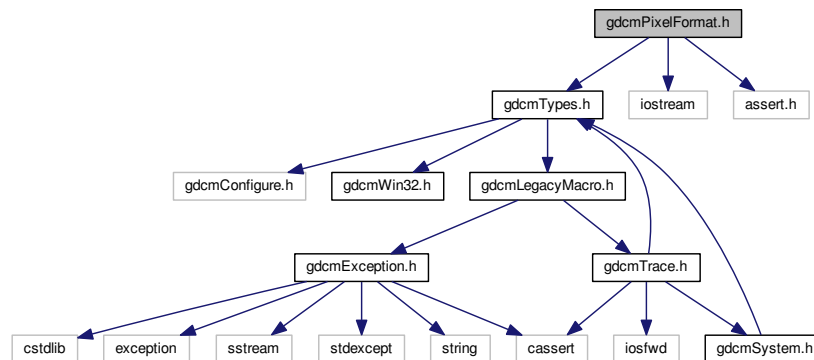
Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const PhotometricInterpretation &val)`

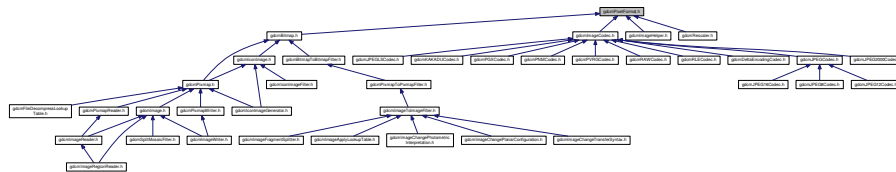
28.191 gdcmPixelFormat.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
#include <assert.h>
```

Include dependency graph for `gdcmPidelFormat.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)`

28.192 gdcmPidap.h File Reference

```

#include "gdcmPidap.h"
#include "gdcmPidap.h"
#include "gdcmPidap.h"
#include "gdcmPidap.h"

```

[illegible]

```

graph TD
    gdcmPixmap_h[gdcmPixmap.h]
    gdcmFileDecompressLookupTable_h[gdcmFileDecompressLookupTable.h]
    gdcmIconImageGenerator_h[gdcmIconImageGenerator.h]
    gdcmPixmapReader_h[gdcmPixmapReader.h]
    gdcmImage_h[gdcmImage.h]
    gdcmPixmapWriter_h[gdcmPixmapWriter.h]
    gdcmImageReader_h[gdcmImageReader.h]
    gdcmImageRegionReader_h[gdcmImageRegionReader.h]
    gdcmSplitMosaicFilter_h[gdcmSplitMosaicFilter.h]
    gdcmImageWriter_h[gdcmImageWriter.h]

    gdcmPixmap_h --> gdcmFileDecompressLookupTable_h
    gdcmPixmap_h --> gdcmIconImageGenerator_h
    gdcmPixmap_h --> gdcmPixmapReader_h
    gdcmPixmap_h --> gdcmImage_h
    gdcmPixmap_h --> gdcmPixmapWriter_h
    gdcmImageReader_h --> gdcmPixmapReader_h
    gdcmImageRegionReader_h --> gdcmImageReader_h
    gdcmImageRegionReader_h --> gdcmImage_h
    gdcmSplitMosaicFilter_h --> gdcmImage_h
    gdcmImageWriter_h --> gdcmImage_h
    gdcmImageWriter_h --> gdcmPixmapWriter_h
  
```

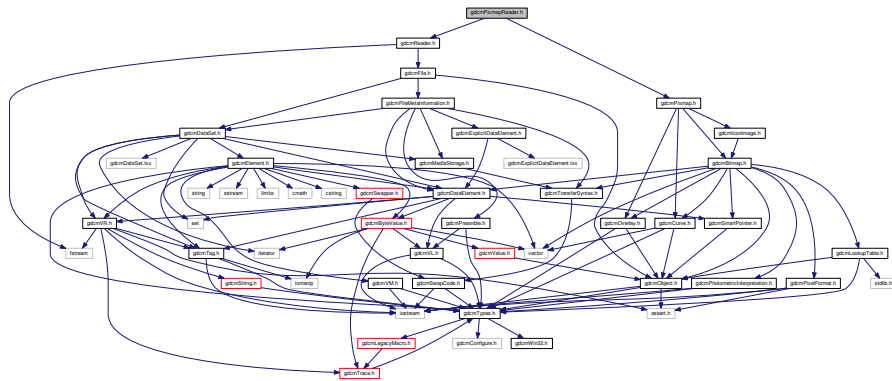
- class `gdcm::Pixmap`

Namespaces

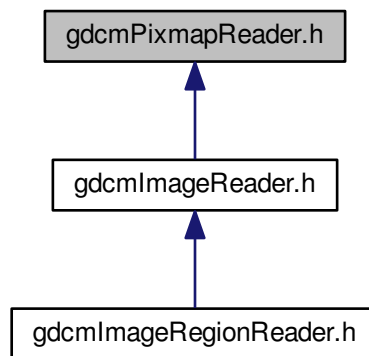
- gdc

```
#include "gdcmReader.h"
#include "gdcmPixmap.h"
```

Include dependency graph for `gdcmPixmapReader.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::PixmapReader`
PixmapReader.

Namespaces

- `gdcm`

28.194 gdcmPixmapToPixmapFilter.h File Reference

```
#include "gdcmBitmapToBitmapFilter.h"
```

[illegible]

```

graph TD
    A[gdcmImageToImageFilter.h] --> B[gdcmImageToImageFilter.h]
    A --> C[gdcmImageChangeTransferSyntax.h]
    A --> D[gdcmImageChangePlanarConfiguration.h]
    A --> E[gdcmImageChangePhotometricInterpretation.h]
    A --> F[gdcmImageApplyLookupTable.h]
    B --> G[gdcmImageToImageFilter.h]
    C --> H[gdcmImageChangeTransferSyntax.h]
    D --> I[gdcmImageChangePlanarConfiguration.h]
    E --> J[gdcmImageChangePhotometricInterpretation.h]
    F --> K[gdcmImageApplyLookupTable.h]
  
```

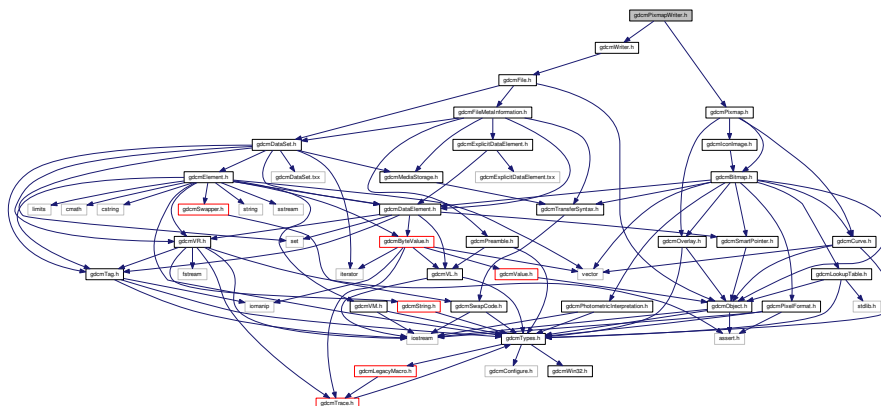
- class `gdcm::PixmapToPixmapFilter`

Namespaces

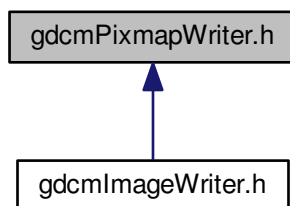
- **gdcm**

```
#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 This class will takes two inputs:

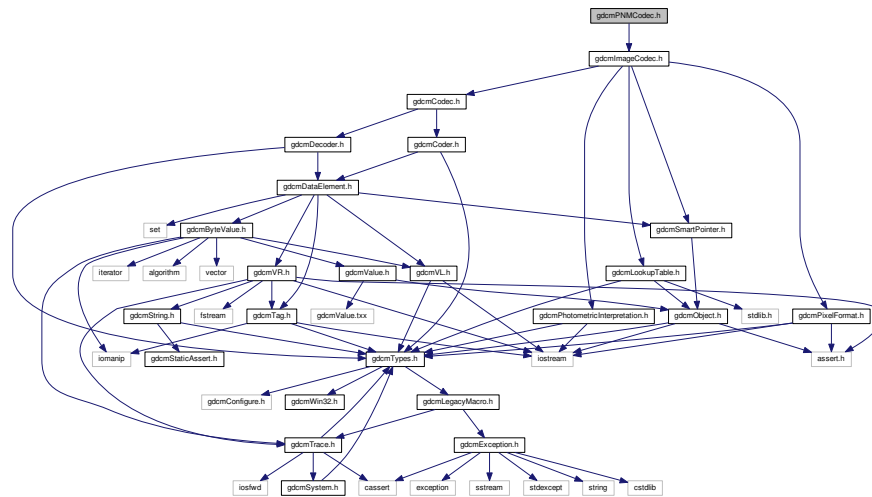
Namespaces

- `gdcm`

28.196 gdcmPNMCodec.h File Reference

```
#include "gdcmImageCodec.h"
```


Include dependency graph for gdcmPNMCodec.h:



Classes

- class [gdcm::PNMCodec](#)

Class to do PNM PNM is the Portable anymap file format. The main web page can be found at: <http://netpbm.sourceforge.net/>.

Namespaces

- [gdcm](#)

28.197 gdcmPreamble.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmVL.h"
```

```
graph TD;
    Preamble[gdcmPreamble.h] --> VL[gdcmVL.h];
    Preamble --> Types[gdcmTypes.h];
    Preamble --> LegacyMacro[gdcmLegacyMacro.h];
    VL --> Types;
    VL --> iostream[iostream];
    Types --> Configure[gdcmConfigure.h];
    Types --> Win32[gdcmWin32.h];
    Types --> LegacyMacro;
    Types --> Exception[gdcmException.h];
    Types --> Trace[gdcmTrace.h];
    LegacyMacro --> Exception;
    LegacyMacro --> Trace;
    Exception --> cstdlib[cstdlib];
    Exception --> exception[exception];
    Exception --> sstream[sstream];
    Exception --> stdexcept[stdexcept];
    Exception --> string[string];
    Exception --> cassert[cassert];
    Exception --> System[gdcmSystem.h];
    Trace --> cassert;
    Trace --> iosfwd[iosfwd];
    Trace --> System;
    System --> Preamble;
```

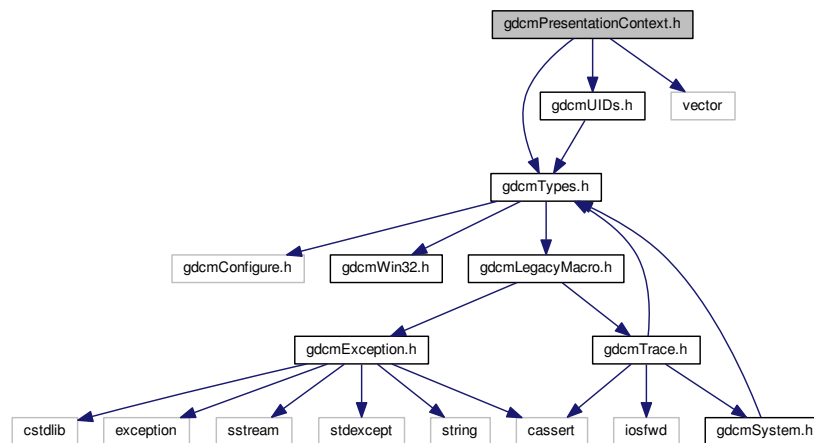
- class `gdcm::Preamble`
DICOM Preamble (Part 10)

- **gdcm**

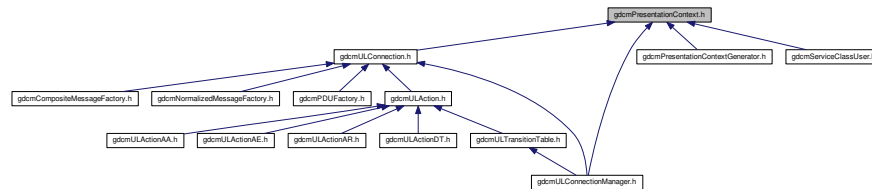
- `std::ostream & gdcmm::operator<< (std::ostream &os, const Preamble &val)`

```
#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.

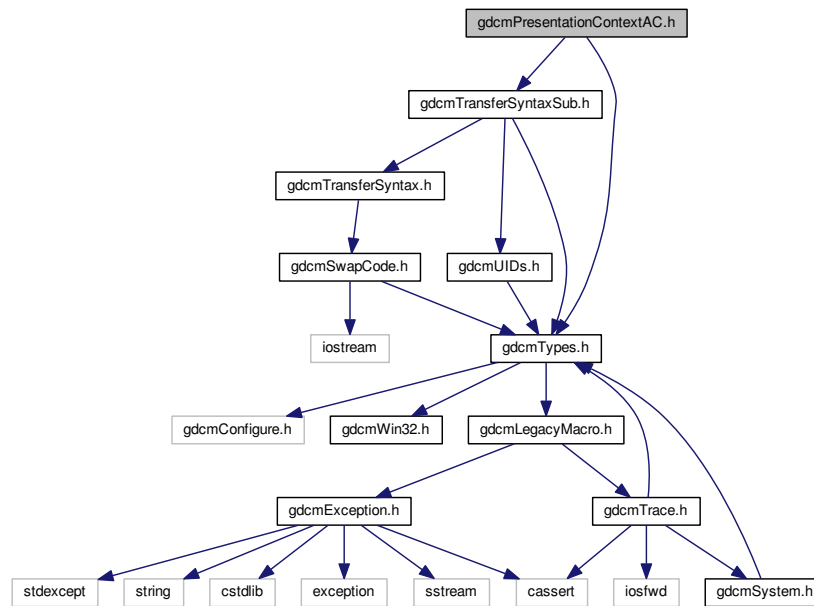
Namespaces

- [gdcm](#)

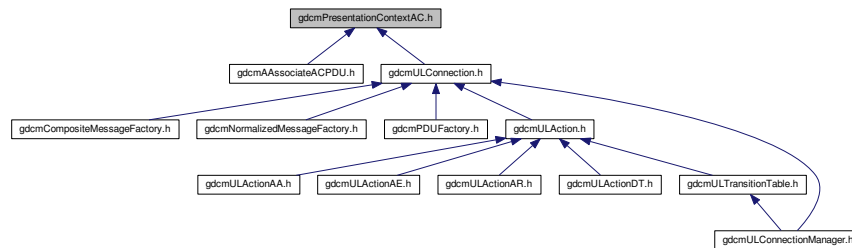
28.199 gdcmPresentationContextAC.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTransferSyntaxSub.h"
```

Include dependency graph for `gdcmPresentationContextAC.h`:



This graph shows which files directly or indirectly include this file:



Classes

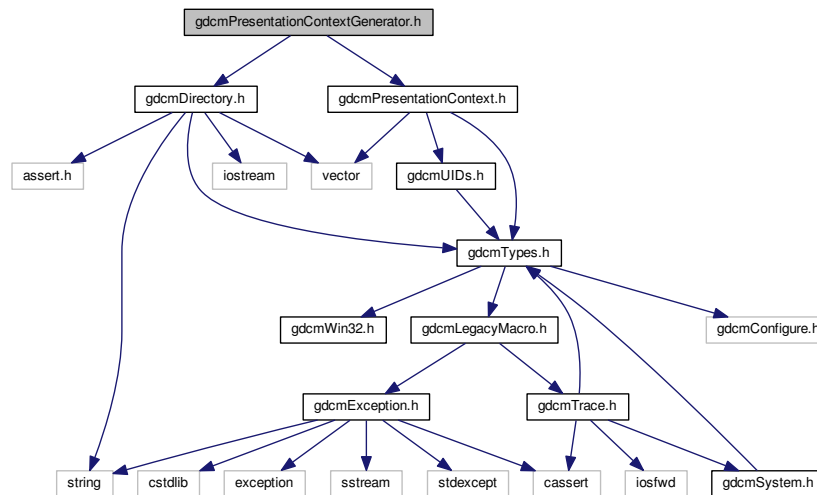
- class `gdcm::network::PresentationContextAC`
PresentationContextAC Table 9-18 PRESENTATION CONTEXT ITEM FIELDS.

Namespaces

- `gdcm`
- `gdcm::network`

28.200 gdcmPresentationContextGenerator.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmPresentationContext.h"
Include dependency graph for gdcmPresentationContextGenerator.h:
```



Classes

- class [gdcm::PresentationContextGenerator](#)

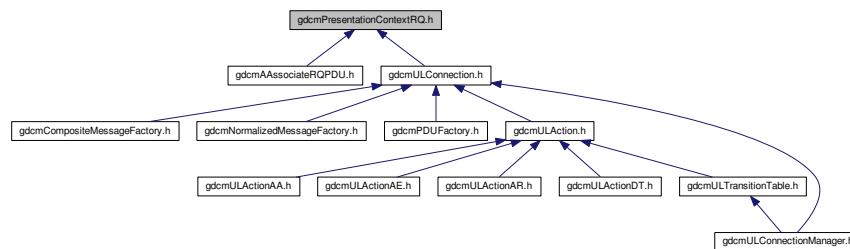
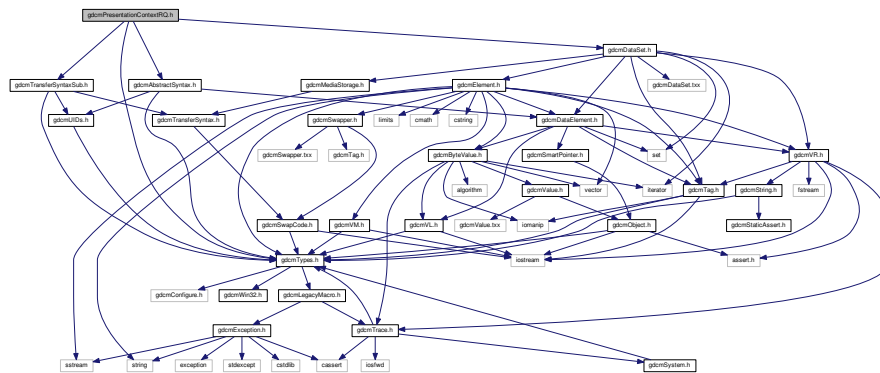
[PresentationContextGenerator](#) This class is responsible for generating the proper [PresentationContext](#) that will be used in subsequent operation during a DICOM Query/Retrieve association. The step of the association is very sensible as special care need to be taken to explicitly define what instance are going to be send and how they are encoded.

Namespaces

- [gdcm](#)

28.201 gdcmPresentationContextRQ.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmAbstractSyntax.h"
#include "gdcmTransferSyntaxSub.h"
#include "gdcmDataSet.h"
```



- DEPARTMENT OF COMMERCE

```

graph TD
    gdcmPresentationDataValue.h --> gdcmTypes.h
    gdcmPresentationDataValue.h --> vector
    gdcmTypes.h --> gdcmConfigure.h
    gdcmTypes.h --> gdcmWin32.h
    gdcmTypes.h --> gdcmLegacyMacro.h
    gdcmTypes.h --> gdcmTrace.h
    gdcmLegacyMacro.h --> gdcmException.h
    gdcmLegacyMacro.h --> gdcmTrace.h
    gdcmTrace.h --> gdcmSystem.h
    gdcmException.h --> cstdlib
    gdcmException.h --> exception
    gdcmException.h --> sstream
    gdcmException.h --> stdexcept
    gdcmException.h --> string
    gdcmException.h --> cassert
    gdcmException.h --> iostream
  
```

- class `gdcm::network::PresentationDataValue`

Namespaces

- `gdcm`
- `gdcm::network`

```
#include "gdcmFile.h"
#include "gdcmDataElement.h"
```

```
graph BT; gdcDictPrinter.h --> gdcPrinter.h; gdcDumper.h --> gdcPrinter.h
```

A diagram illustrating inheritance. At the top is a box labeled `gdcPrinter.h`. Below it are two boxes: `gdcDictPrinter.h` on the left and `gdcDumper.h` on the right. Blue arrows point from each of these bottom boxes up to the `gdcPrinter.h` box, indicating that both `gdcDictPrinter.h` and `gdcDumper.h` inherit from `gdcPrinter.h`.

- class `gdcm::Printer`
Printer class.

- **gdcm**

```
#include "gdcmTag.h"
```

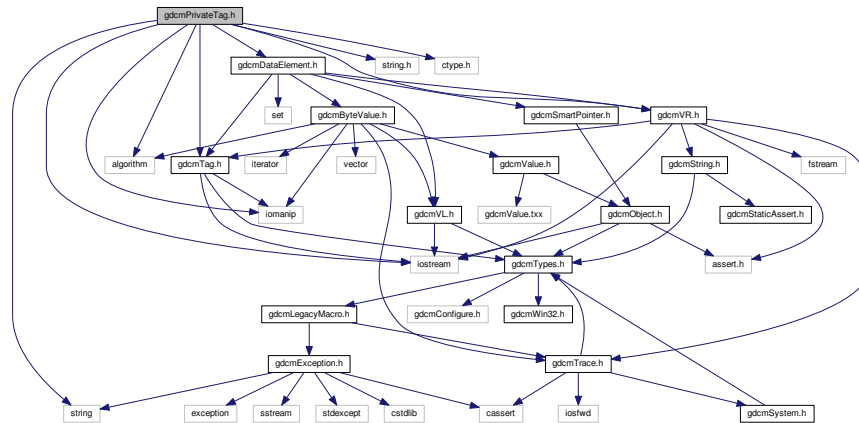


```

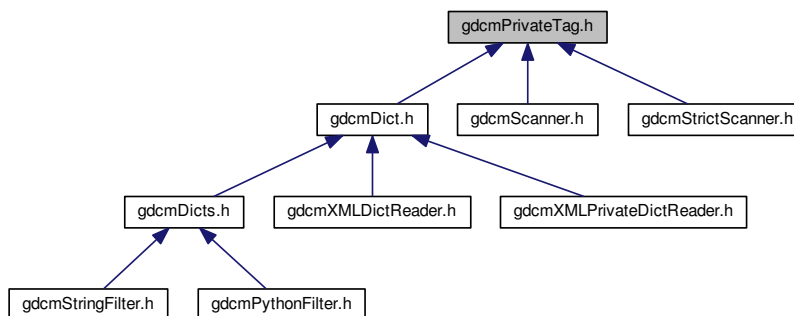
#include "gdcmVR.h"
#include "gdcmDataElement.h"
#include <iostream>
#include <iomanip>
#include <string>
#include <algorithm>
#include <string.h>
#include <ctype.h>

```

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

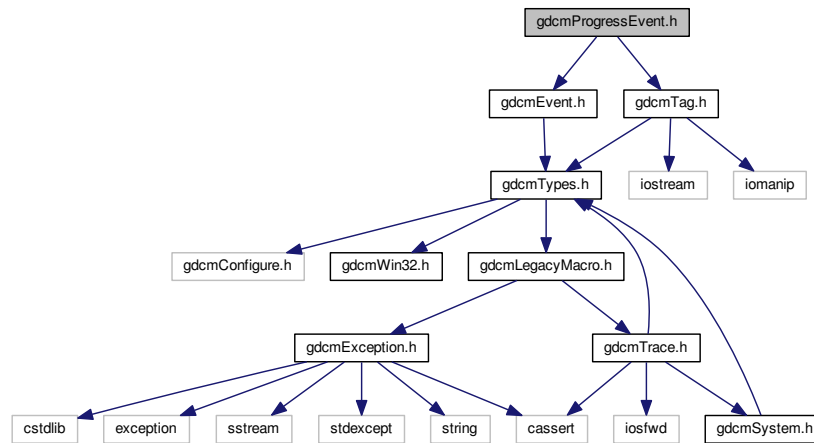
- `std::ostream & gdcm::operator<< (std::ostream &os, const PrivateTag &val)`

28.205 gdcmProgressEvent.h File Reference

```
#include "gdcmEvent.h"
```

```
#include "gdcmTag.h"
```

Include dependency graph for `gdcmProgressEvent.h`:



Classes

- class [gdcm::ProgressEvent](#)
ProgressEvent Special type of event triggered during.

Namespaces

- [gdcm](#)

28.206 gdcmPVRGCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

- class `gdcm::PVRGCodec`
PVRGCodec.

- **gdcm**

```
#include <Python.h>
#include "gdcmDataElement.h"
#include "gdcmDicts.h"
#include "gdcmFile.h"
```

[illegible]

Classes

- class [gdcm::PythonFilter](#)

PythonFilter [PythonFilter](#) is the class that make *gdcm2.x* looks more like *gdcm1* and transform the binary blob contained in a [DataElement](#) into a string, typically this is a nice feature to have for wrapped language.

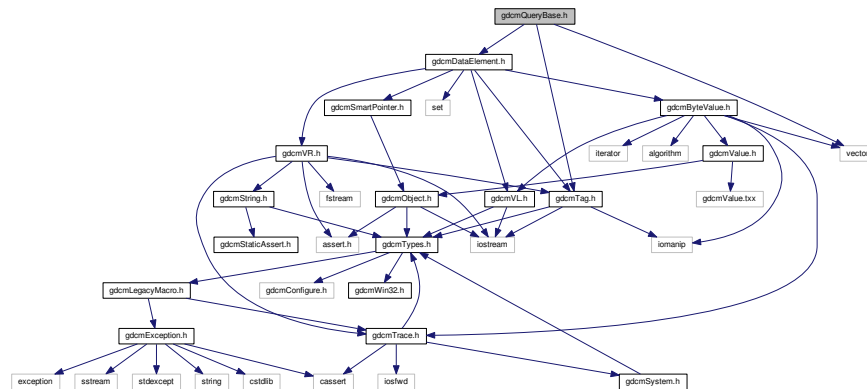
Namespaces

- [gdcm](#)

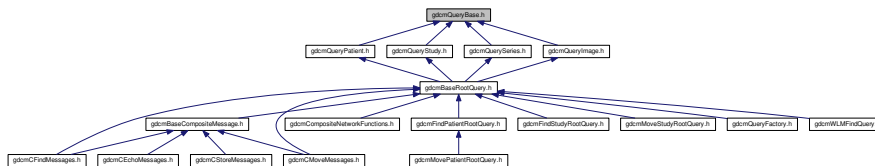
28.208 gdcmQueryBase.h File Reference

```
#include "gdcmTag.h"
#include "gdcmDataElement.h"
#include <vector>
```

Include dependency graph for `gdcmQueryBase.h`:



This graph shows which files directly or indirectly include this file:



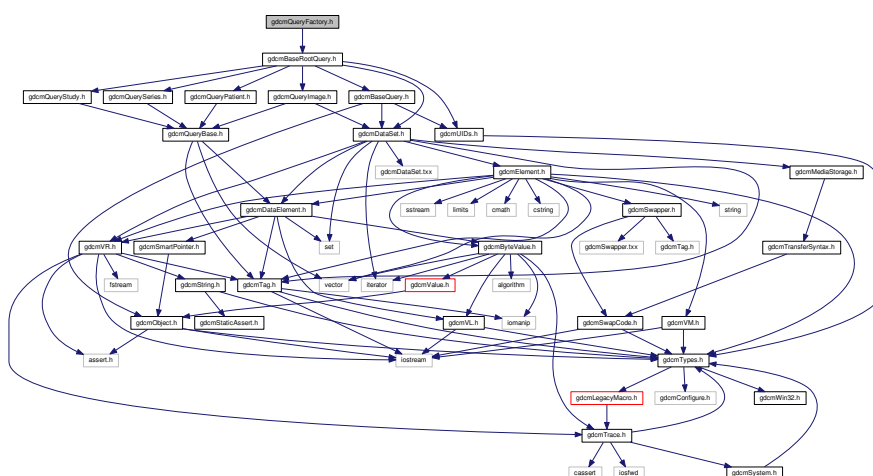
Classes

- class [gdcm::QueryBase](#)

QueryBase contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE.

- **gdcm**

- enum `gdcm::ERootType` {
`gdcm::ePatientRootType`,
`gdcm::eStudyRootType` }



- class `gdcm::QueryFactory`
QueryFactory.h

- **gdcm**

Classes

- class `gdcm::QueryImage`

QueryImage contains: class to construct an image-based query for C-FIND and C-MOVE.

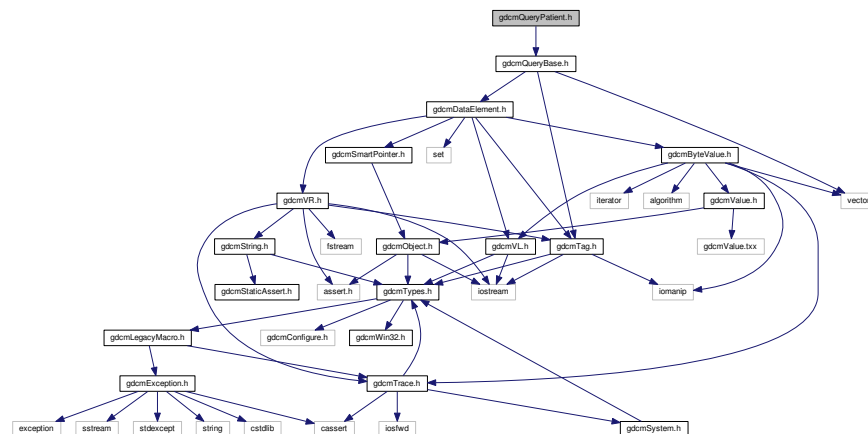
Namespaces

- **gdcm**

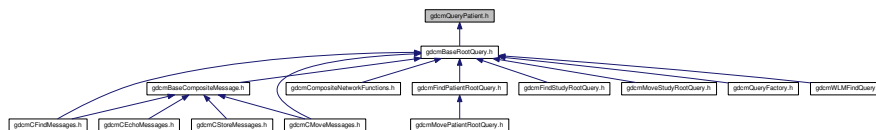
28.211 gdcMQueryPatient.h File Reference

```
#include "gdcmQueryBase.h"
```

Include dependency graph for gdcmQueryPatient.h:



This graph shows which files directly or indirectly include this file:



Classes

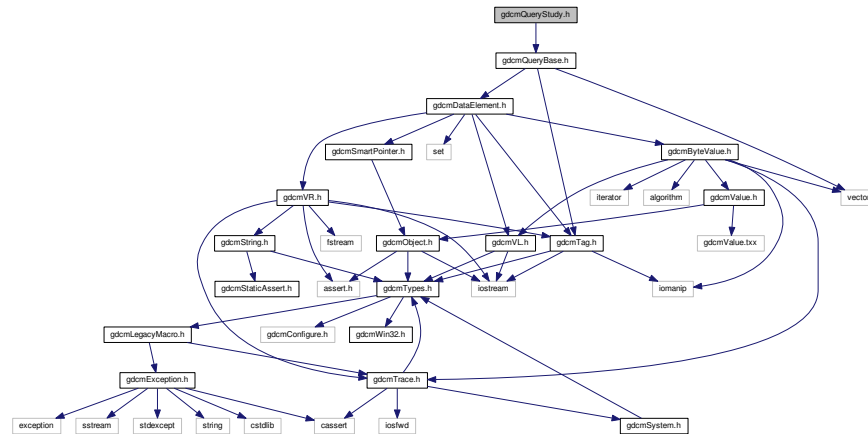
- class `gdcm::QueryPatient`

QueryPatient contains: class to construct a patient-based query for c-find and c-move.

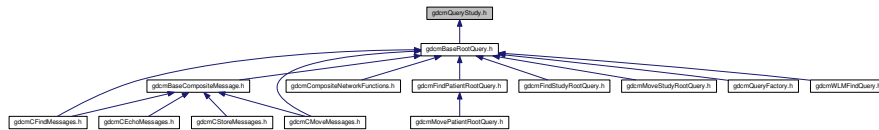
Namespaces

- **gdcm**

Include dependency graph for gdcmmQueryStudy.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcmm::QueryStudy](#)

QueryStudy.h contains: class to construct a study-based query for C-FIND and C-MOVE.

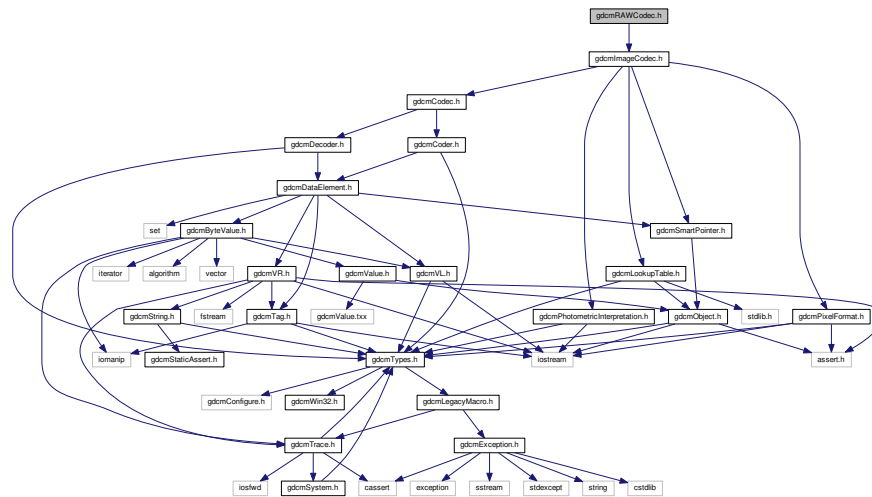
Namespaces

- [gdcmm](#)

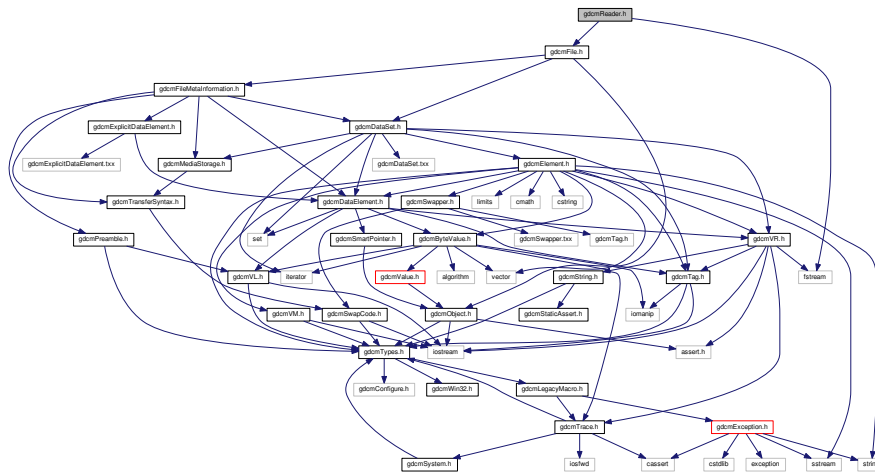
28.214 gdcmmraw.dox File Reference

28.215 gdcmmRAWCodec.h File Reference

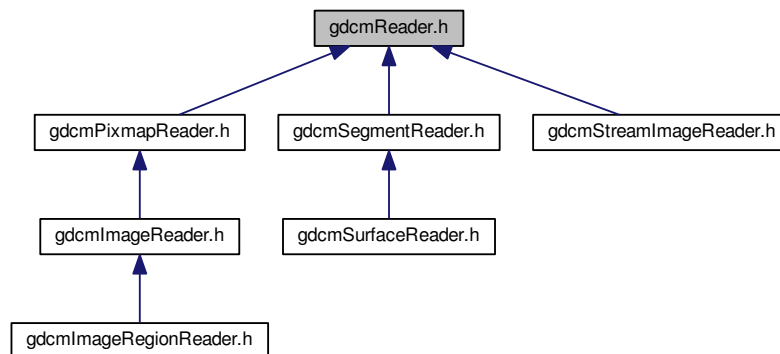
```
#include "gdcmmImageCodec.h"
```



Include dependency graph for gdcmReader.h:



This graph shows which files directly or indirectly include this file:



Classes

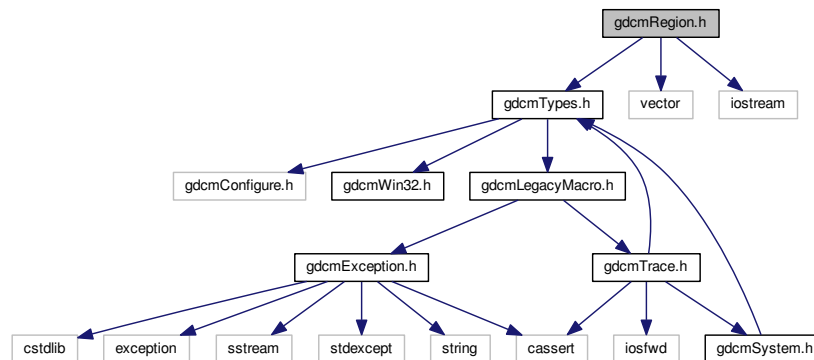
- class [gdcm::Reader](#)
Reader ala DOM (Document *Object* Model)

Namespaces

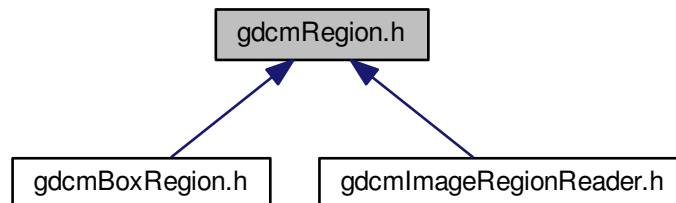
- [gdcm](#)

28.217 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

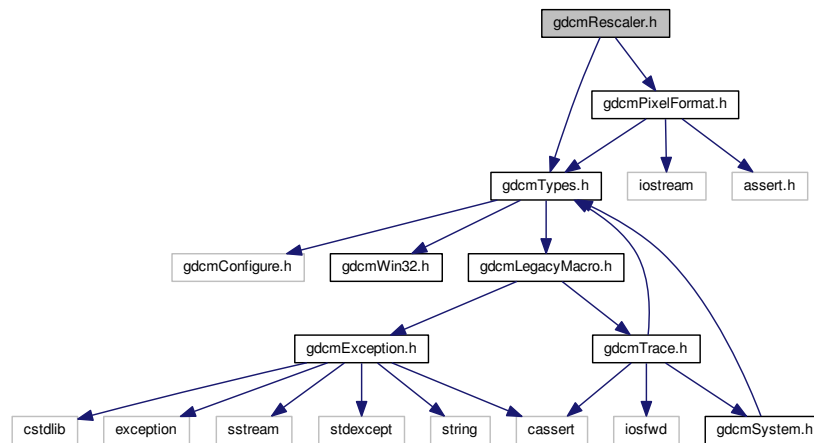
- `std::ostream & gdcm::operator<< (std::ostream &os, const Region &r)`

28.218 gdcmRescaler.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmPixelFormat.h"
```

Include dependency graph for gdcmRescaler.h:



Classes

- class `gdcm::Rescaler`

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.

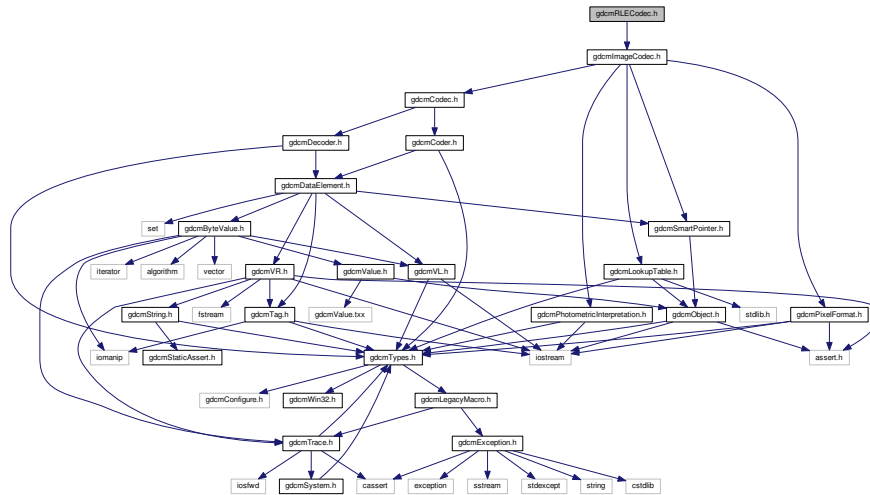
Namespaces

- `gdcm`

28.219 gdcmRLECodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmRLECodec.h:



Classes

- class `gdcm::RLECodec`
Class to do RLE.

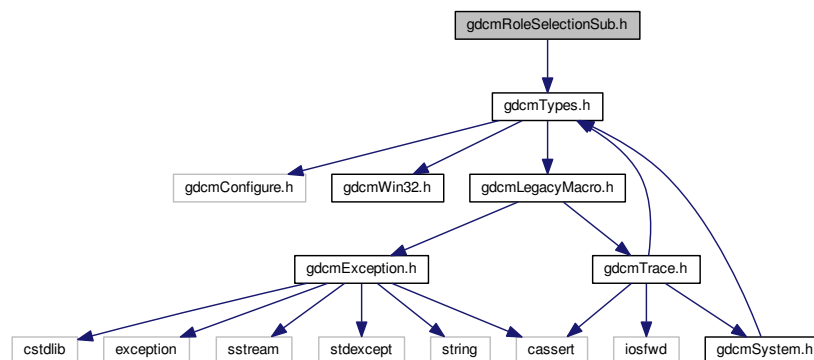
Namespaces

- **gdcm**

28.220 gdcMRoleSelectionSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcMRoleSelectionSub.h:



Namespaces

- [gdcm](#)

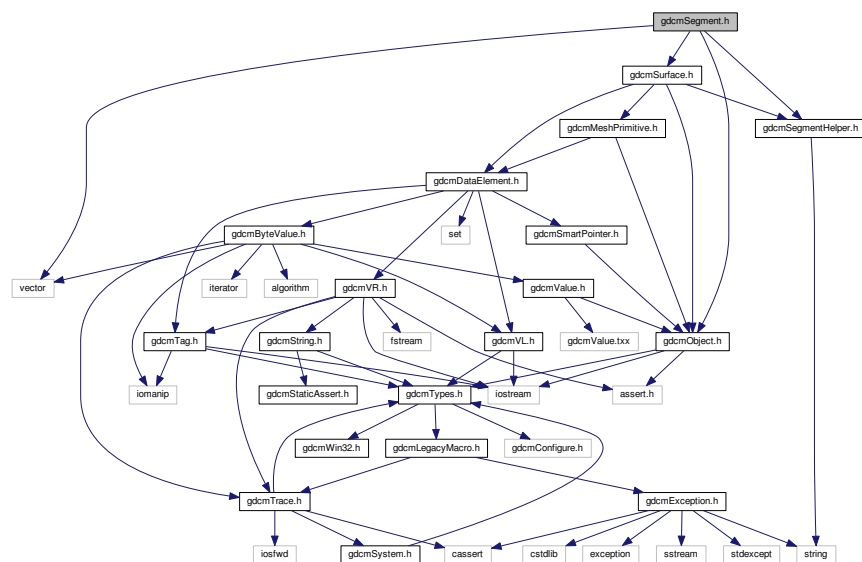
Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Scanner &s)`

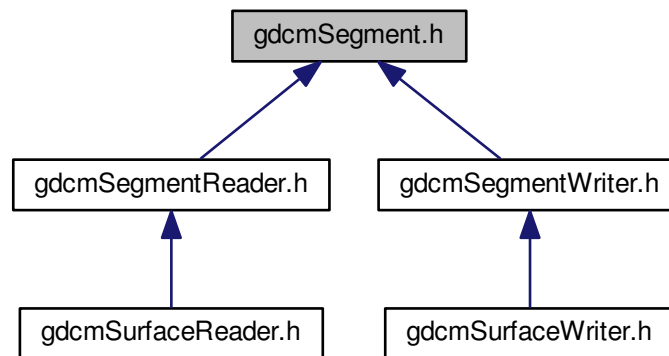
28.223 gdcmscu.dox File Reference

28.224 gdcmSegment.h File Reference

```
#include <vector>
#include <gdcmObject.h>
#include <gdcmSurface.h>
#include "gdcmSegmentHelper.h"
Include dependency graph for gdcmSegment.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::Segment](#)

This class defines a segment. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface.

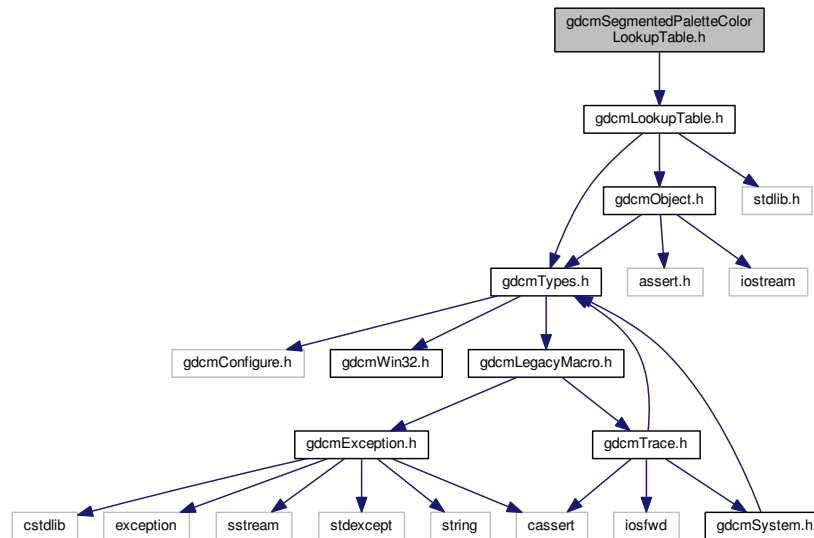
Namespaces

- [gdcm](#)

28.225 gdcmSegmentedPaletteColorLookupTable.h File Reference

```
#include "gdcmLookupTable.h"
```

Include dependency graph for `gdcmSegmentedPaletteColorLookupTable.h`:



Classes

- class `gdcm::SegmentedPaletteColorLookupTable`

SegmentedPaletteColorLookupTable class.

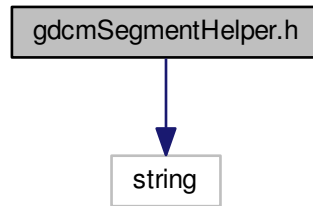
Namespaces

- `gdcm`

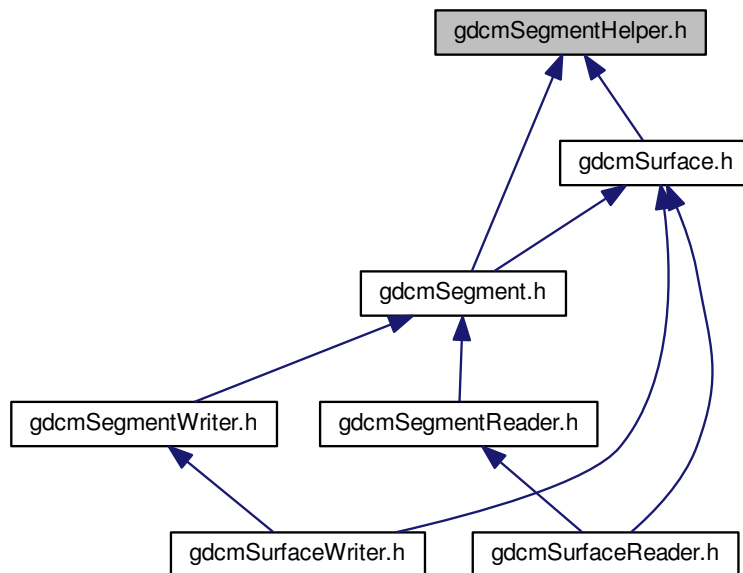
28.226 gdcmSegmentHelper.h File Reference

```
#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](#)

Namespaces

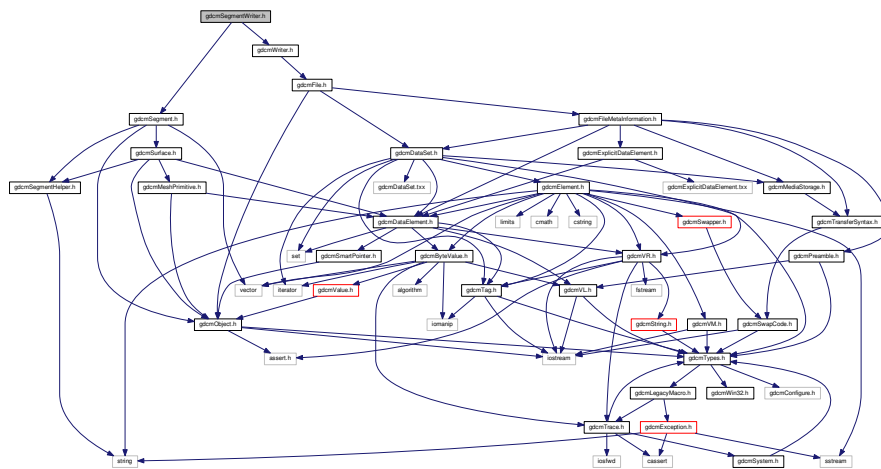
- [gdcm](#)

28.228 gdcmSegmentWriter.h File Reference

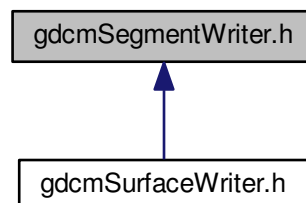
```
#include <gdcmWriter.h>
```

```
#include <gdcmSegment.h>
```

Include dependency graph for gdcmSegmentWriter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::SegmentWriter](#)

This class defines a segment writer. It writes attributes of group 0x0062.

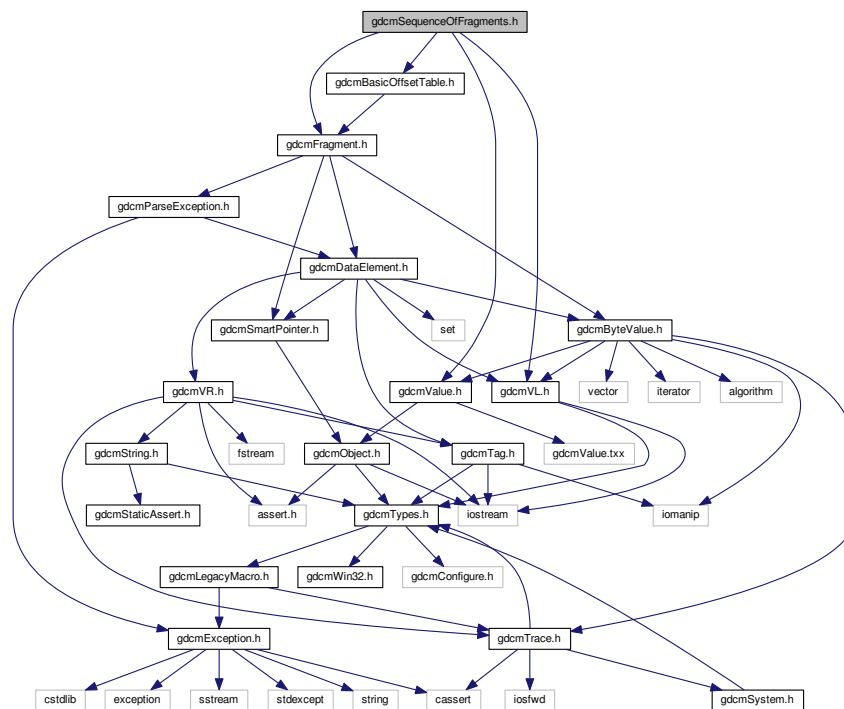
Namespaces

- [gdcm](#)

28.229 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

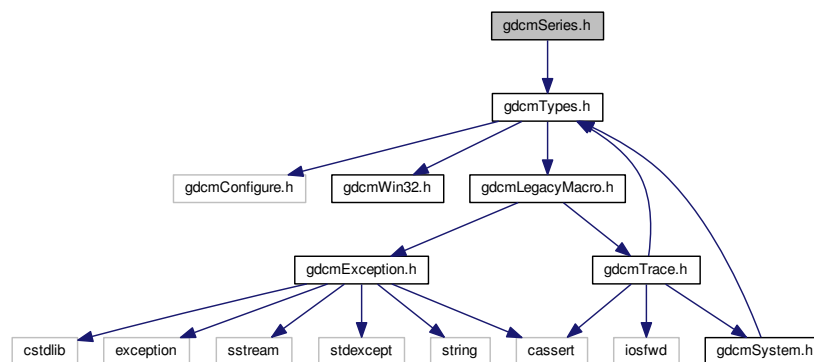
- [gdcm](#)

- enum `gdcm::LodModeType` {
`gdcm::LD_ALL = 0x00000000,`
`gdcm::LD_NOSEQ = 0x00000001,`
`gdcm::LD_NOSHADOW = 0x00000002,`
`gdcm::LD_NOSHADOWSEQ = 0x00000004 }`

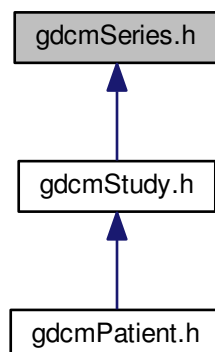
28.232 gdcmSeries.h File Reference

```
#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](#).

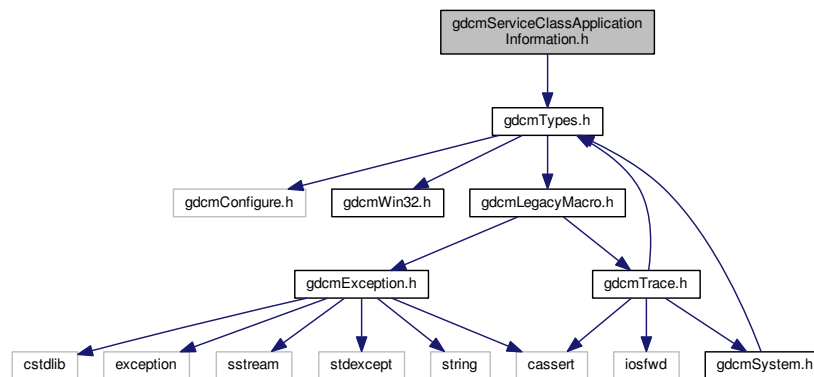
Namespaces

- [gdcm](#)

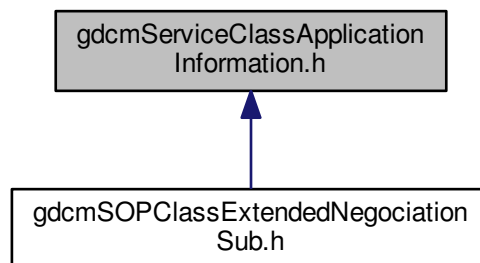
28.233 gdcmServiceClassApplicationInformation.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmServiceClassApplicationInformation.h:



This graph shows which files directly or indirectly include this file:



Classes

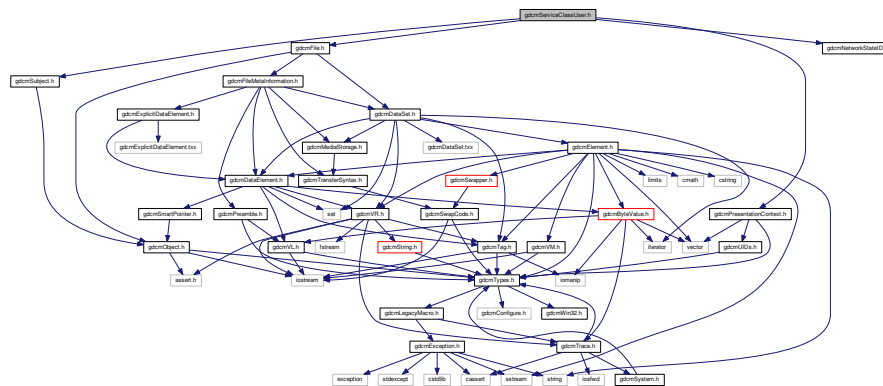
- class [gdcm::network::ServiceClassApplicationInformation](#)

Namespaces

- [gdcm](#)
- [gdcm::network](#)

28.234 gdcmServiceClassUser.h File Reference

```
#include "gdcmSubject.h"
#include "gdcmPresentationContext.h"
#include "gdcmFile.h"
#include "gdcmNetworkStateID.h"
Include dependency graph for gdcmServiceClassUser.h:
```



Classes

- class [gdcm::ServiceClassUser](#)
ServiceClassUser.

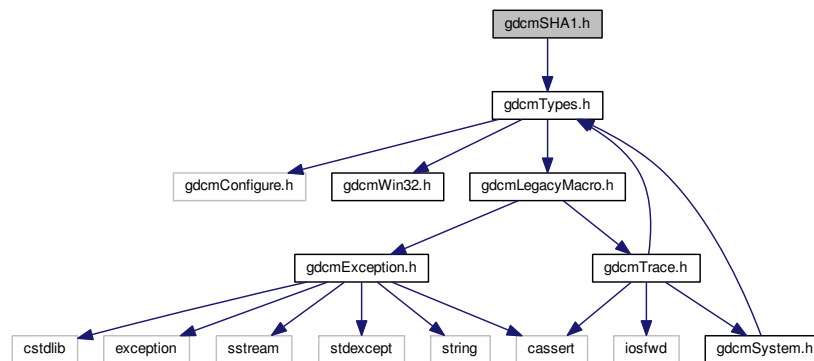
Namespaces

- [gdcm](#)
- [gdcm::network](#)

28.235 gdcmSHA1.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for `gdcmsHA1.h`:



Classes

- class `gdcms::SHA1`

Class for `SHA1`.

Namespaces

- `gdcms`

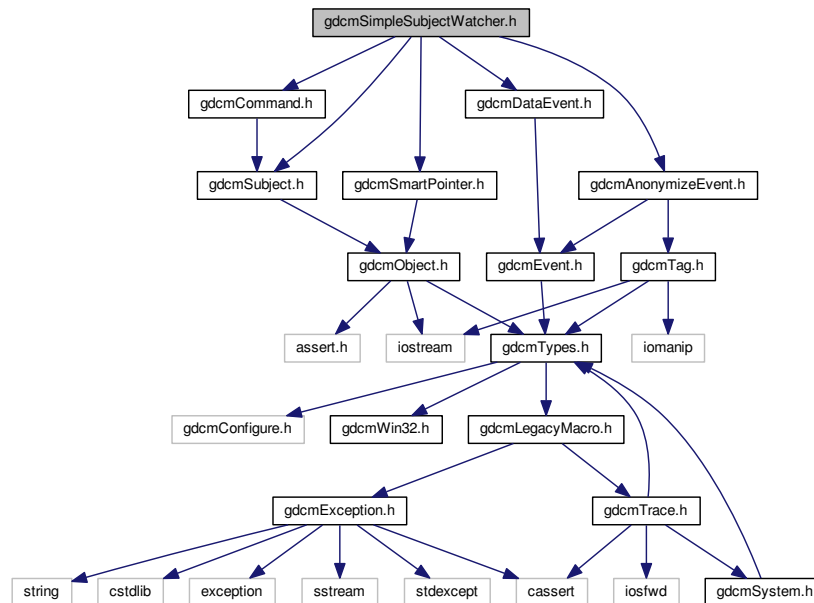
28.236 `gdcmsSimpleSubjectWatcher.h` File Reference

```

#include "gdcmsSubject.h"
#include "gdcmsCommand.h"
#include "gdcmsSmartPointer.h"
#include "gdcmsAnonymizeEvent.h"
#include "gdcmsDataEvent.h"

```

Include dependency graph for gdcmSimpleSubjectWatcher.h:



Classes

- class [gdcm::SimpleSubjectWatcher](#)

SimpleSubjectWatcher This is a typical *Subject* *Watcher* class. It will observe all events.

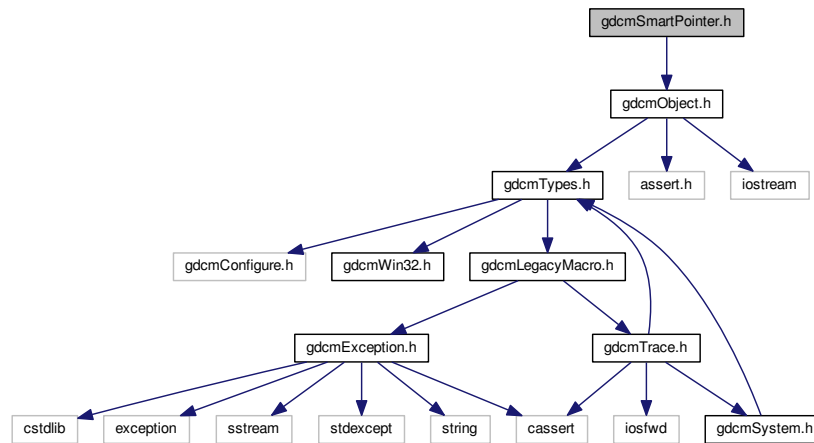
Namespaces

- [gdcm](#)

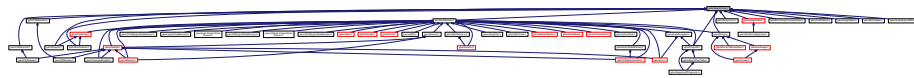
28.237 gdcmSmartPointer.h File Reference

```
#include "gdcmObject.h"
```

Include dependency graph for `gdcSmartPointer.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdc::SmartPointer< ObjectType >`

Class for Smart Pointer.

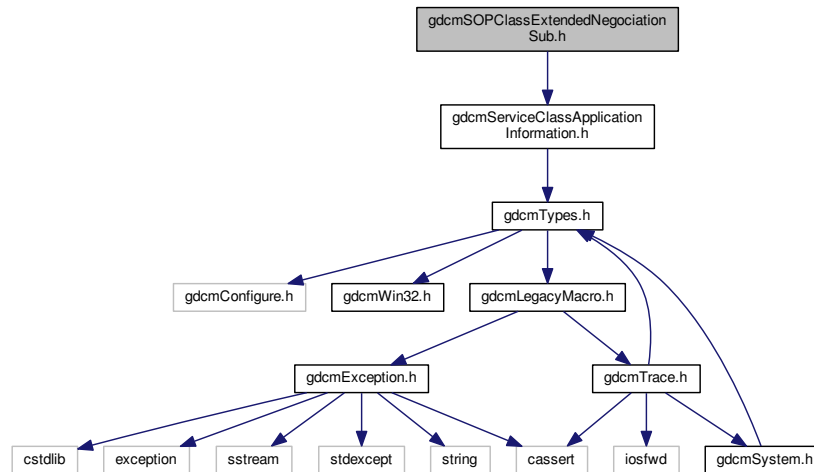
Namespaces

- `gdc`

28.238 gdcSOPClassExtendedNegociationSub.h File Reference

```
#include "gdcServiceClassApplicationInformation.h"
```

Include dependency graph for gdcmSOPClassExtendedNegociationSub.h:



Classes

- class [gdcm::network::SOPClassExtendedNegociationSub](#)

[SOPClassExtendedNegociationSub](#) PS 3.7 Table D.3-11 SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-ASSOCIATE-AC)

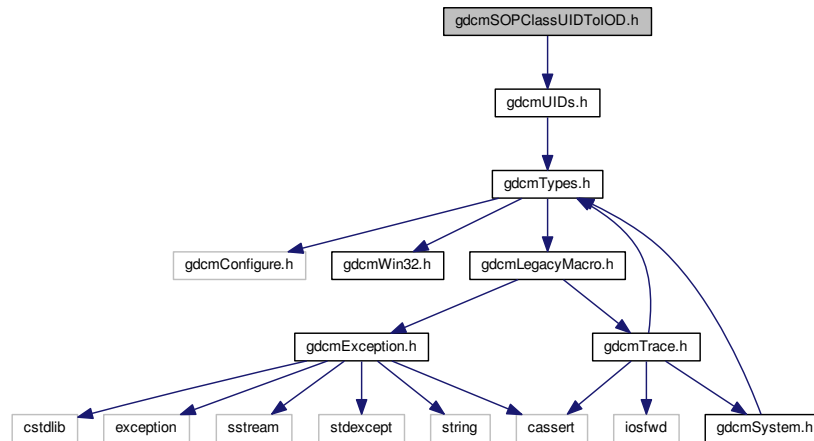
Namespaces

- [gdcm](#)
- [gdcm::network](#)

28.239 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`

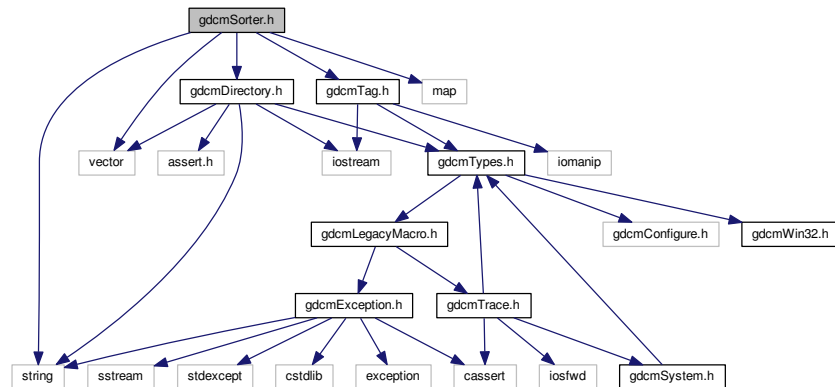
28.240 gdcmSorter.h File Reference

```

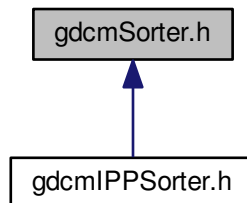
#include "gdcmDirectory.h"
#include "gdcmTag.h"
#include <vector>
#include <string>
#include <map>

```


Include dependency graph for gdcmSorter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::Sorter](#)

[Sorter](#) General class to do sorting using a custom function You simply need to provide a function of type: [Sorter::SortFunction](#).

Namespaces

- [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Sorter &s)`

- class `gdcm::Spectroscopy`
Spectroscopy class.

- **gdcm**

```
#include "gdcmFile.h"
#include "gdcmImage.h"
```

Classes

- class [gdcm::SplitMosaicFilter](#)

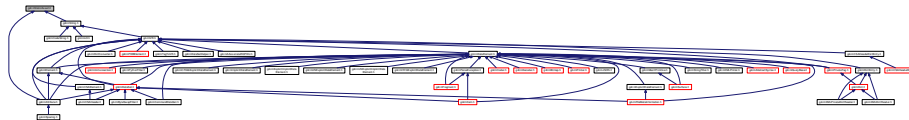
[SplitMosaicFilter](#) class Class to reshuffle bytes for a SIEMENS Mosaic image Siemens CSA [Image](#) Header CSA:= Common Siemens Architecture, sometimes also known as Common syngo Architecture.

Namespaces

- [gdcm](#)

28.244 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< 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.

28.244.1 Macro Definition Documentation

28.244.1.1 #define `GDCM_DO_JOIN(X, Y) GDCM_DO_JOIN2(X,Y)`

28.244.1.2 #define `GDCM_DO_JOIN2(X, Y) X##Y`

28.244.1.3 #define `GDCM_JOIN(X, Y) GDCM_DO_JOIN(X, Y)`

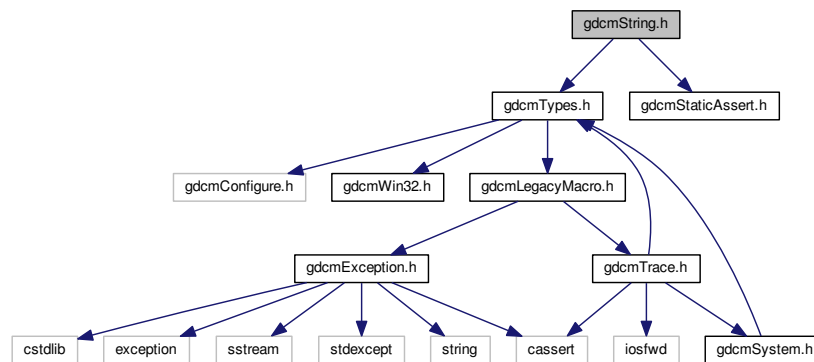
- struct `gdcm::StrictScanner::Itstr`
- class `gdcm::StrictScanner`

- **gdcm**

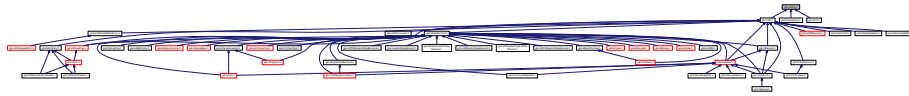
- `std::ostream & gdcmm::operator<< (std::ostream &os, const StrictScanner &s)`

```
#include "gdcmTypes.h"
#include "gdcmStaticAssert.h"
```

Include dependency graph for `gdcmString.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::String< TDelimiter, TMaxLength, TPadChar >`
String.

Namespaces

- `gdcm`

Functions

- `template<char TDelimiter, unsigned int TMaxLength, char TPadChar>`
`std::istream & gdcm::operator>> (std::istream &is, String< TDelimiter, TMaxLength, TPadChar > &ms)`

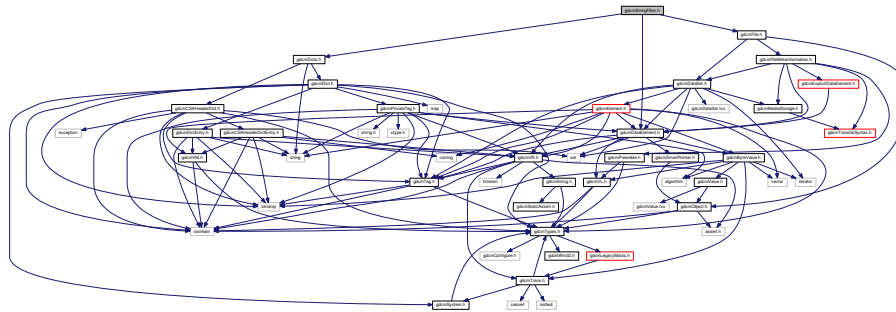
28.249 gdcmStringFilter.h File Reference

```

#include "gdcmDataElement.h"
#include "gdcmDicts.h"
#include "gdcmFile.h"

```


Include dependency graph for gdcmStringFilter.h:



Classes

- class [gdcm::StringFilter](#)

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.

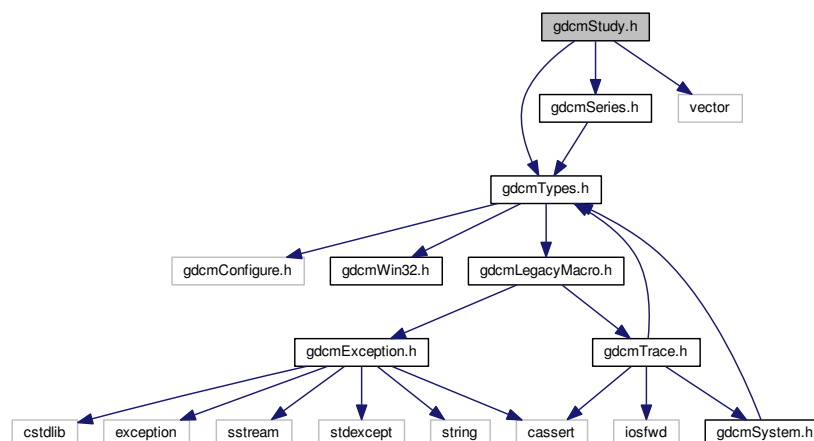
Namespaces

- [gdcm](#)

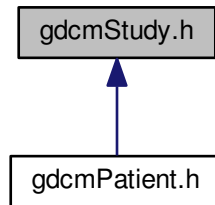
28.250 gdcmStudy.h File Reference

```
#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](#).

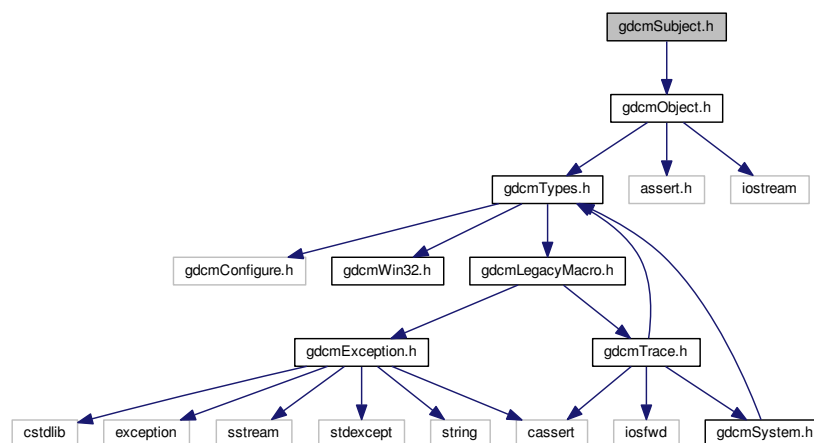
Namespaces

- [gdc](#)

28.251 gdcSubject.h File Reference

```
#include "gdcObject.h"
```

Include dependency graph for gdcSubject.h:



This graph shows which files directly or indirectly include this file:



Classes

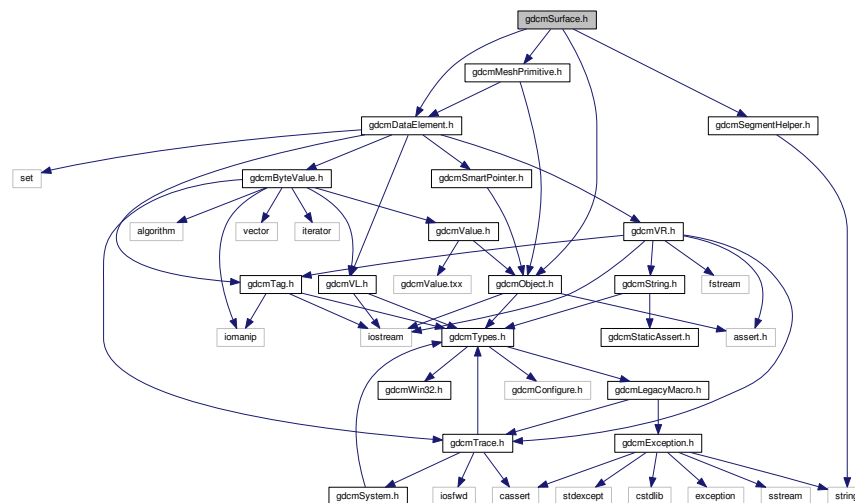
- class [gdcm::Subject](#)
Subject.

Namespaces

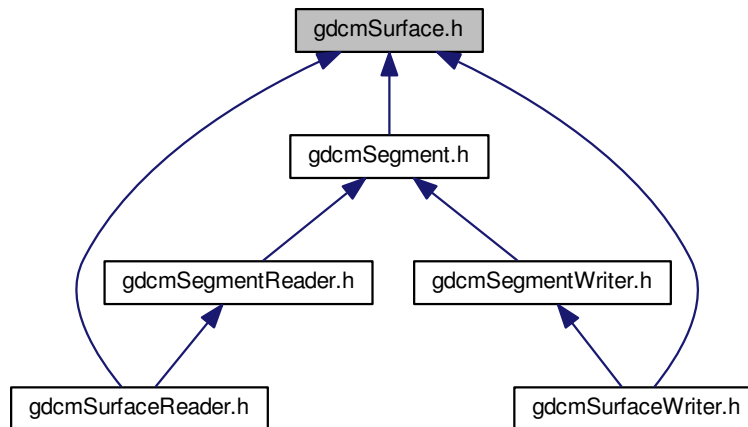
- [gdcm](#)

28.252 gdcmSurface.h File Reference

```
#include <gdcmObject.h>
#include <gdcmDataElement.h>
#include <gdcmMeshPrimitive.h>
#include "gdcmSegmentHelper.h"
Include dependency graph for gdcmSurface.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcms::Surface](#)

This class defines a SURFACE IE. This members are taken from required surface mesh module attributes.

Namespaces

- [gdcms](#)

28.253 gdcmsurfacehelper.h File Reference

```
#include "gdcmtypes.h"
#include <vector>
#include <iostream>
```

```

graph TD
    gdcmSurfaceHelper.h[gdcmSurfaceHelper.h] --> gdcmTypes.h[gdcmTypes.h]
    gdcmSurfaceHelper.h --> vector
    gdcmSurfaceHelper.h --> iostream
    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
    gdcmException.h --> exception
    gdcmException.h --> sstream
    gdcmException.h --> stdexcept
    gdcmException.h --> string
    gdcmException.h --> cassert
    gdcmException.h --> iostwd
    gdcmTrace.h --> iostwd
    gdcmTrace.h --> gdcmSystem.h[gdcmSystem.h]
  
```

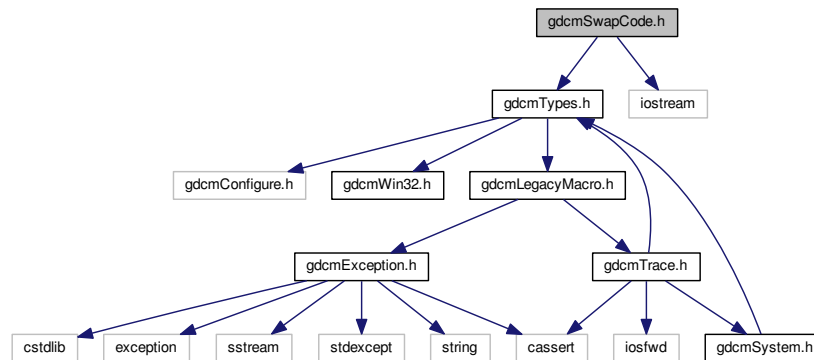
- class `gdcm::SurfaceHelper`
SurfaceHelper Helper class for *Surface* object.

- **gdcm**

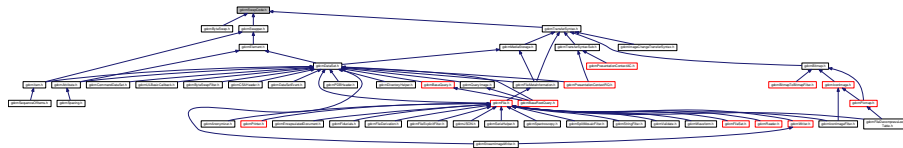
```
#include <gdcmSegmentReader.h>
#include <gdcmSurface.h>
```

[illegible]

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](#)

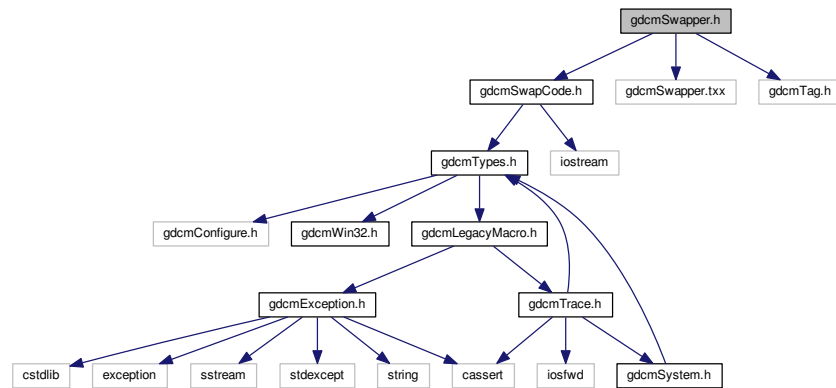
Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const SwapCode &sc)`

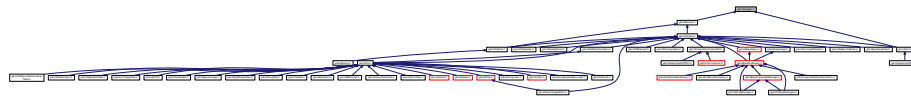
28.257 gdcmSwapper.h File Reference

```
#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`

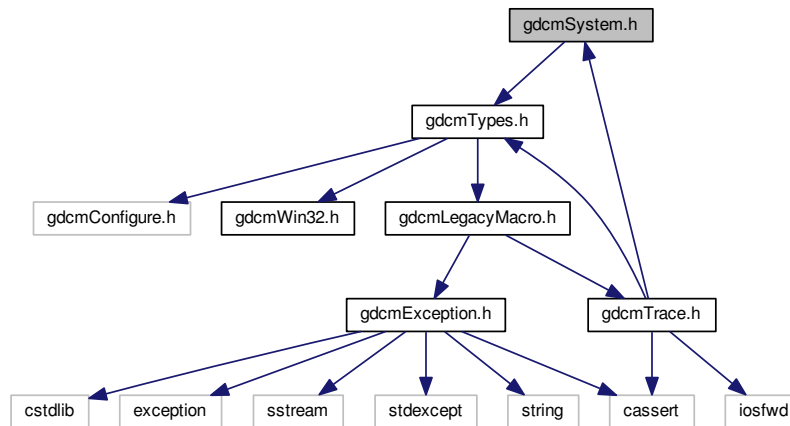
Namespaces

- `gdcm`

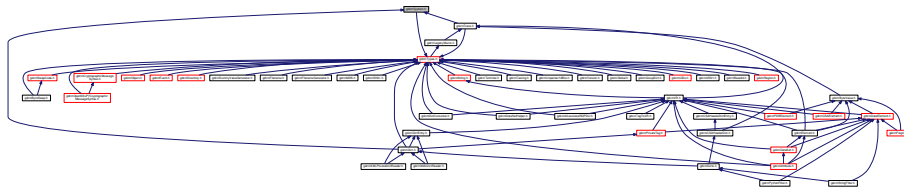
28.258 gdcmSystem.h File Reference

```
#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.

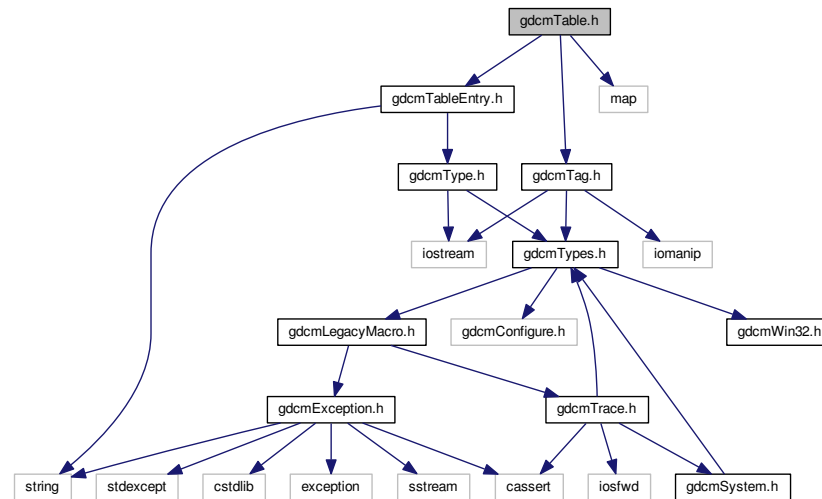
Namespaces

- [gdcm](#)

28.259 gdcmTable.h File Reference

```
#include "gdcmTableEntry.h"
#include "gdcmTag.h"
#include <map>
```

Include dependency graph for `gdcmTable.h`:



Classes

- class [gdcm::Table](#)

Table.

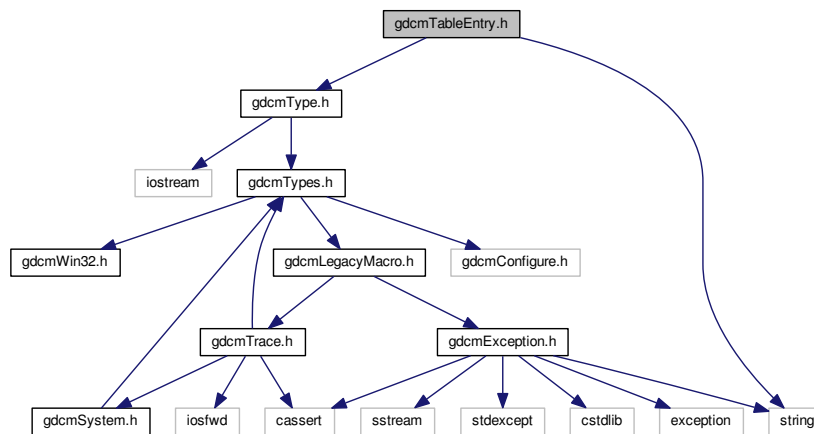
Namespaces

- [gdcm](#)

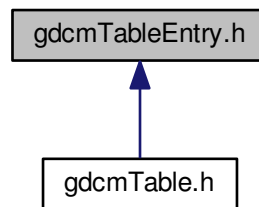
28.260 gdcmTableEntry.h File Reference

```
#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.

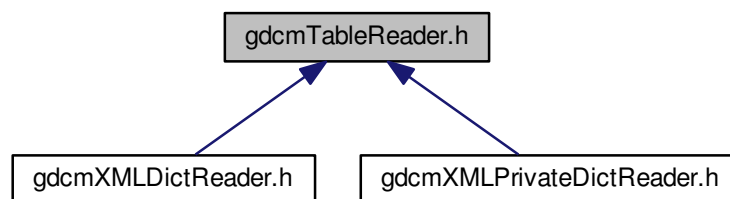
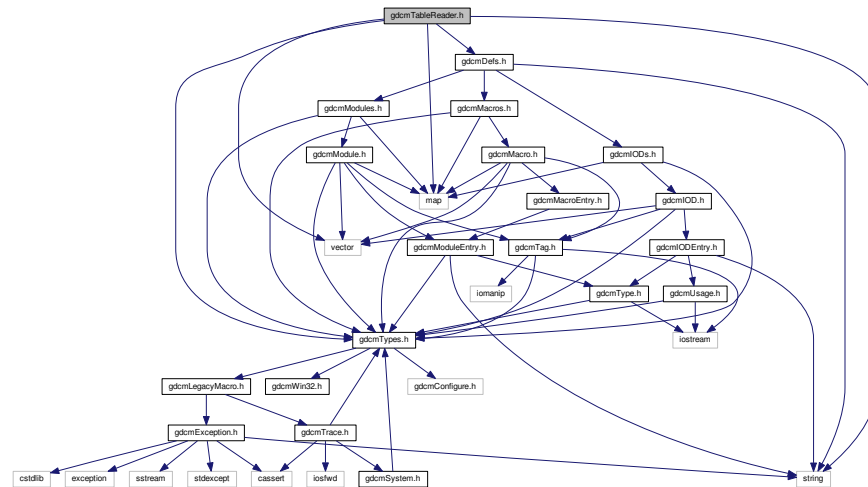
Namespaces

- [gdcm](#)

28.261 gdcmTableReader.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmTableReader.h:



- class `gdcm::TableReader`

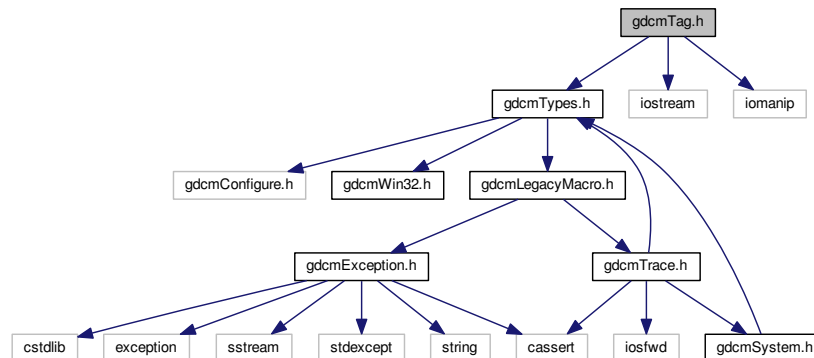
Class for representing a [TableReader](#).

- **gdcm**

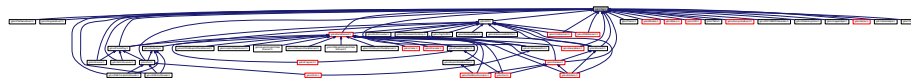
- **gdcm**

28.262 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*). Basically an `uint32_t` which can also be expressed as two `uint16_t` (*group* and *element*)

Namespaces

- [gdcm](#)

Functions

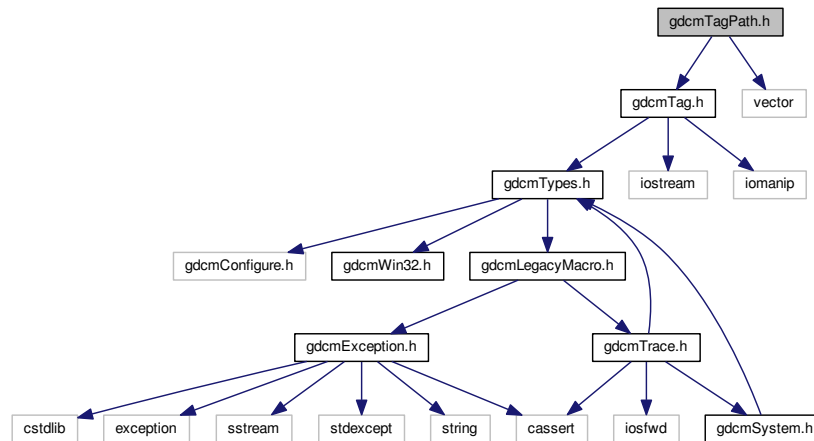
- `std::ostream & gdcm::operator<< (std::ostream &_os, const Tag &_val)`
- `std::istream & gdcm::operator>> (std::istream &_is, Tag &_val)`

28.263 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.

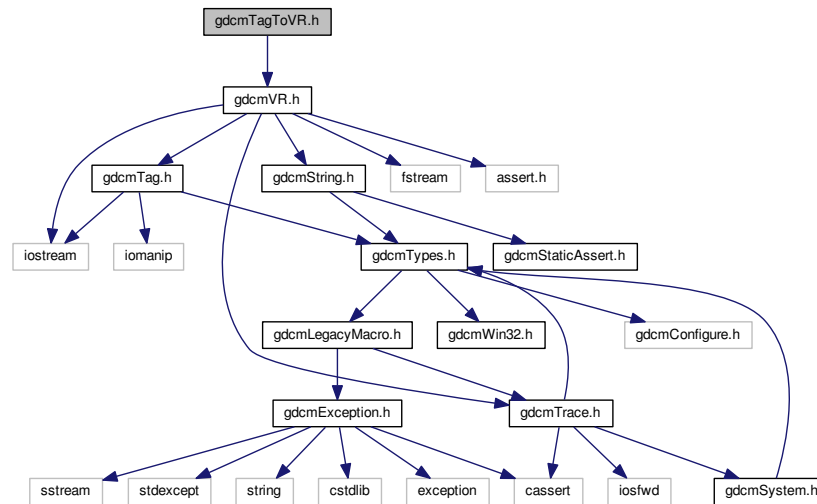
Namespaces

- [gdcm](#)

28.264 gdcmTagToVR.h File Reference

```
#include "gdcmVR.h"
```

Include dependency graph for gdcmTagToVR.h:



Namespaces

- [gdcm](#)

Functions

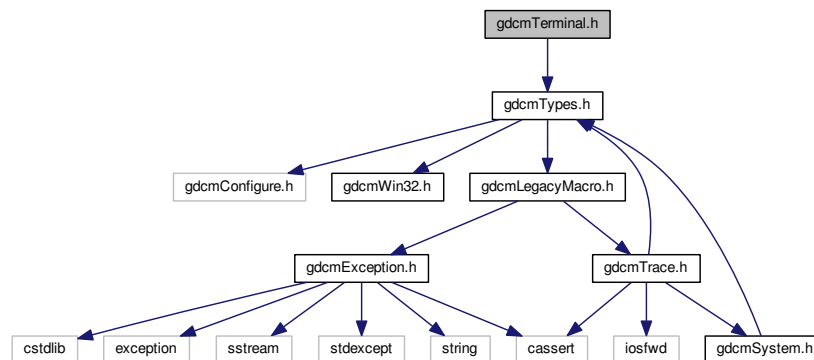
- VR::VRType [gdcm::GetVRFromTag](#) (Tag const &tag)

28.265 gdcmtar.dox File Reference

28.266 gdcmTerminal.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for `gdcTerminal.h`:



Namespaces

- `gdc`
- `gdc::terminal`

Class for Terminal Allow one to print in color in a shell.

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)

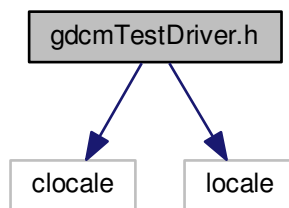
- `GDCM_EXPORT` `std::string` `gdcm::terminal::setbgcolor` (Color c)
- `GDCM_EXPORT` `std::string` `gdcm::terminal::setfgcolor` (Color c)
- `GDCM_EXPORT` `void` `gdcm::terminal::setmode` (Mode m)

28.267 gdcmTestDriver.h File Reference

```
#include <clocale>
```

```
#include <locale>
```

Include dependency graph for `gdcmTestDriver.h`:

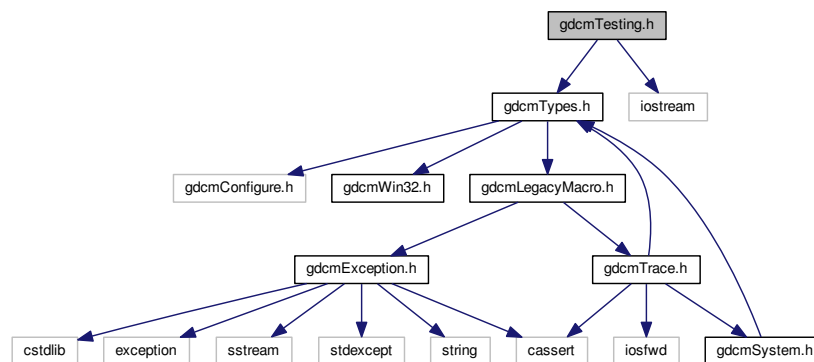


28.268 gdcmTesting.h File Reference

```
#include "gdcmTypes.h"
```

```
#include <iostream>
```

Include dependency graph for `gdcmTesting.h`:



Classes

- class [gdcm::Testing](#)
class for testing

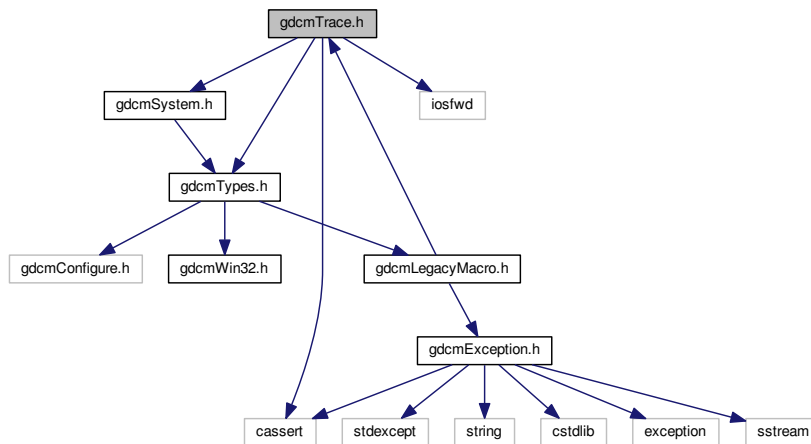
Namespaces

- [gdcm](#)

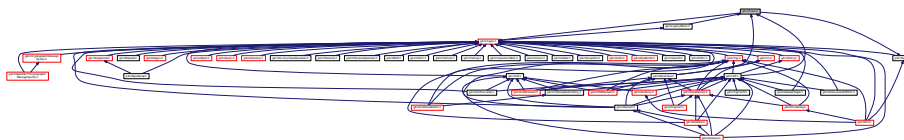
28.269 gdcmTrace.h File Reference

```
#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.

28.269.1 Macro Definition Documentation

28.269.1.1 #define [GDCM_FUNCTION](#) "<unknown>"

28.269.1.2 #define [gdcmAssertAlwaysMacro](#)(arg) [gdcmAssertMacro](#)(arg)

[AssertAlways.](#)

Parameters

<i>arg</i>	argument to test An easy solution to pass also a message is to do: gdcmAssertMacro ("my message" && 2 < 3)
------------	--

Referenced by [gdcm::SequenceOfFragments::ReadValue\(\)](#), and [gdcm::VR::Write\(\)](#).

28.269.1.3 #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" && 2 < 3)</code>
------------	---

Referenced by `gdcm::PixelFormat::SetSamplesPerPixel()`.

28.269.1.4 `#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
------------	--------------

Referenced by `gdcm::ByteValue::ByteValue()`, `gdcm::OpenSSLCryptoFactory::OpenSSLCryptoFactory()`, `gdcm::OpenSSLP7CryptoFactory::OpenSSLP7CryptoFactory()`, `gdcm::BasicOffsetTable::Read()`, `gdcm::Item::Read()`, `gdcm::SequenceOfItems::Read()`, `gdcm::VR::Read()`, `gdcm::SequenceOfFragments::ReadPreValue()`, and `gdcm::SequenceOfFragments::ReadValue()`.

28.269.1.5 `#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
------------	---------------------

Referenced by `gdcm::CommandDataSet::Insert()`, `gdcm::FileMetaInformation::Insert()`, `gdcm::DataSet::Insert()`, `gdcm::Item::Read()`, and `gdcm::Fragment::ReadBacktrack()`.

28.269.1.6 `#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

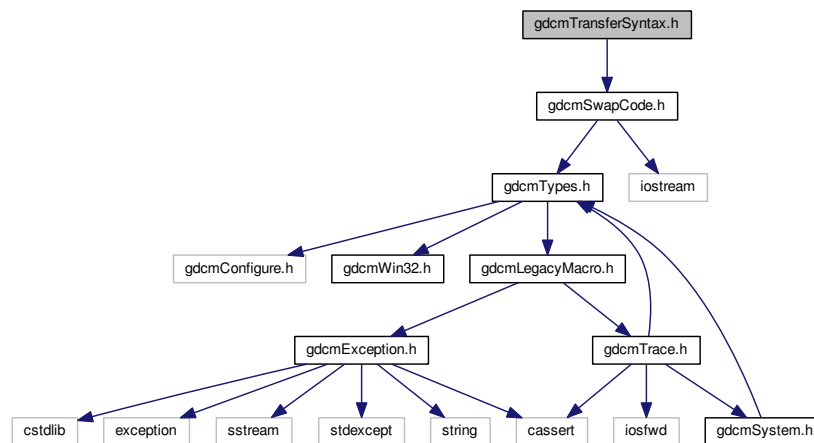
<i>msg</i>	message part
------------	--------------

Referenced by `gdcm::DataSet::InsertDataElement()`, `gdcm::Item::Read()`, `gdcm::SequenceOfItems::Read()`, `gdcm::Fragment::ReadBacktrack()`, `gdcm::Fragment::ReadValue()`, `gdcm::SequenceOfFragments::ReadValue()`, `gdcm::OpenSSL7CryptographicMessageSyntax::SetPassword()`, and `gdcm::Item::Write()`.

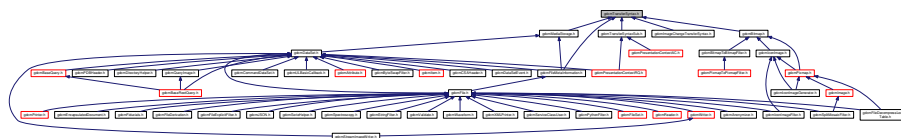
28.270 gdcmTransferSyntax.h File Reference

```
#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

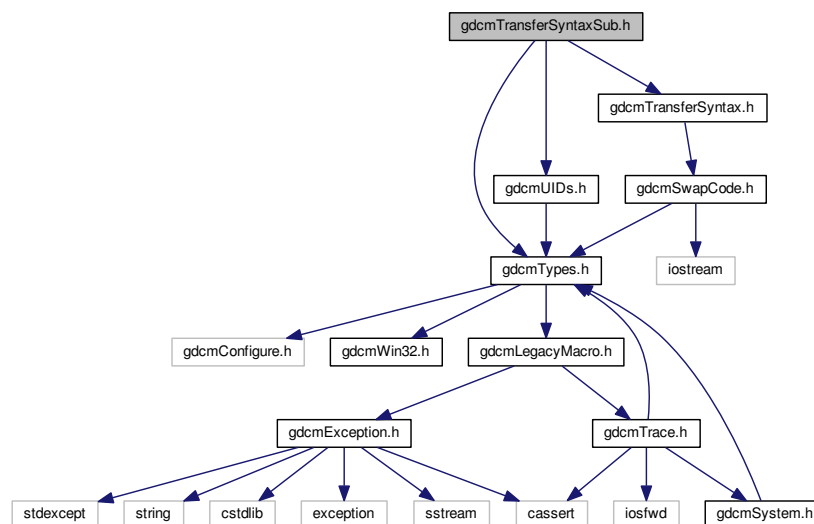
- [gdcm](#)

Functions

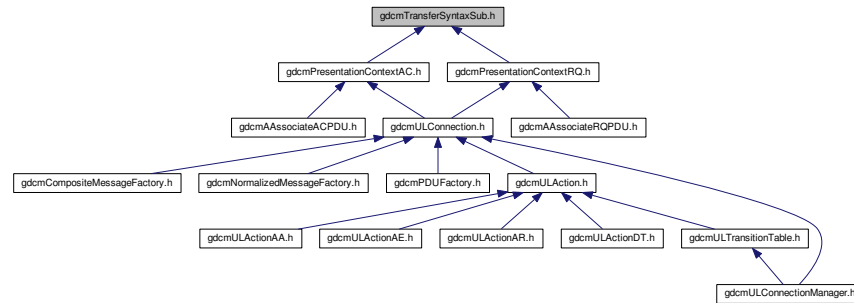
- `std::ostream & gdcm::operator<< (std::ostream &_os, const TransferSyntax &ts)`

28.271 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 Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS.

Namespaces

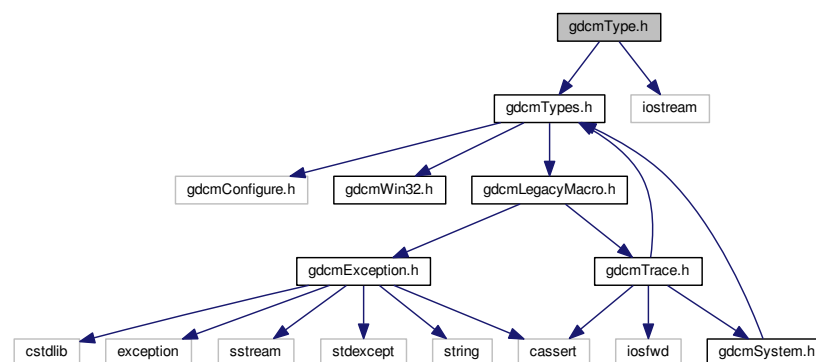
- [gdcm](#)
- [gdcm::network](#)

28.272 gdcType.h File Reference

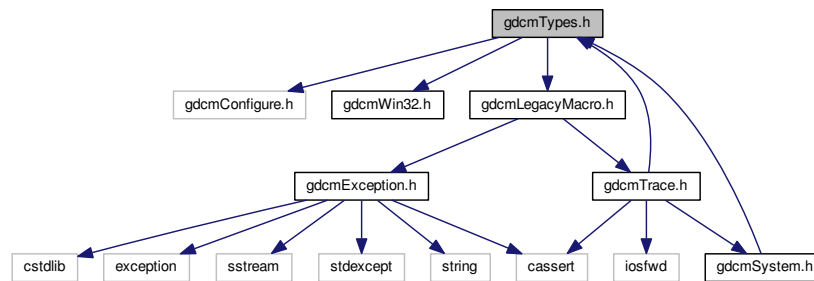
```
#include "gdcmTypes.h"
```

```
#include <iostream>
```

Include dependency graph for gdcType.h:



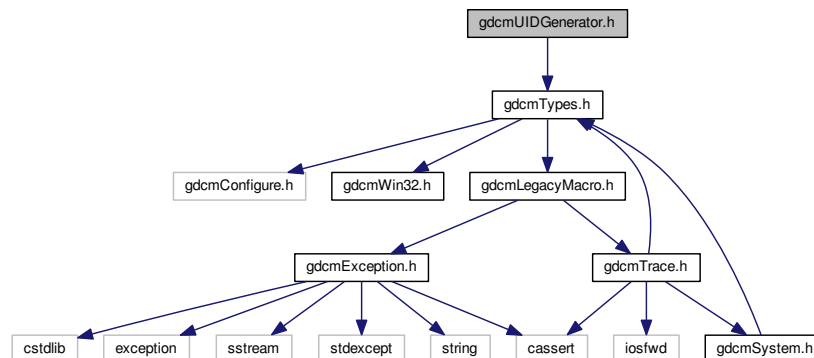
Include dependency graph for gdcmTypes.h:



28.274 gdcmUIDGenerator.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmUIDGenerator.h:



Classes

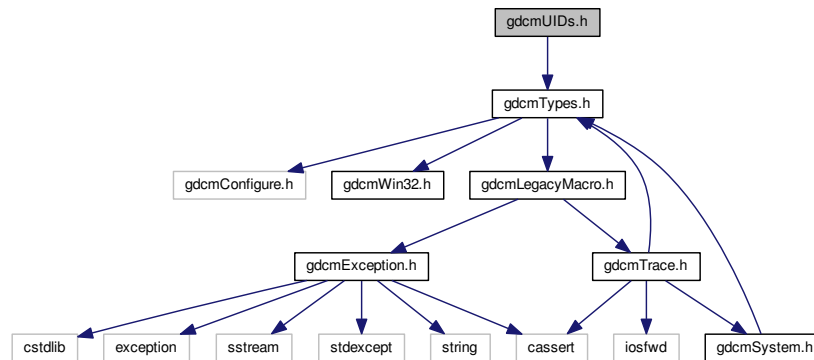
- class [gdcm::UIDGenerator](#)
Class for generating unique UID.

Namespaces

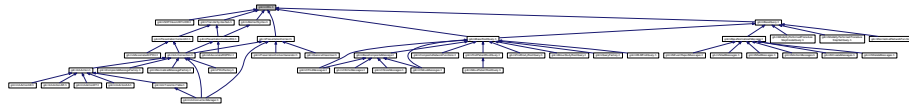
- [gdcm](#)

28.275 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`

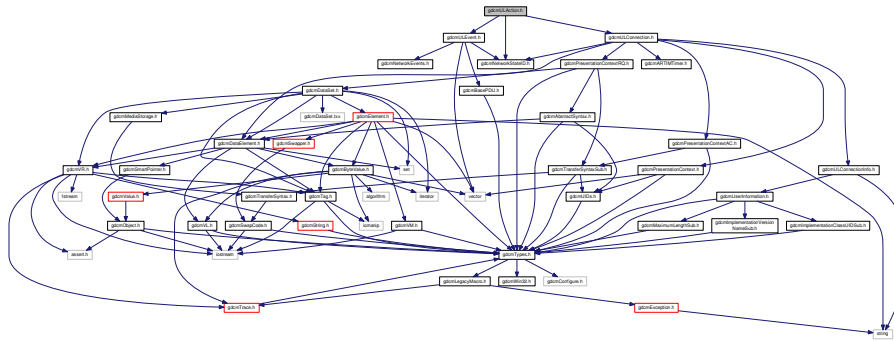
Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const UIDs &uid)`

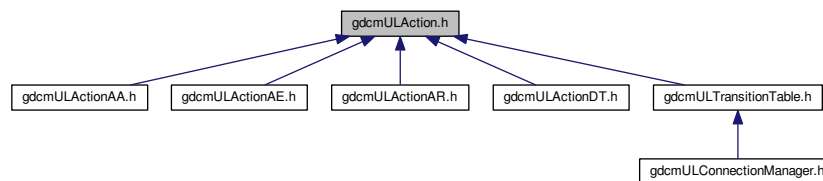
28.276 gdcmULAction.h File Reference

```
#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](#) A [ULConnection](#) in a given [ULState](#) can perform certain [ULActions](#). This base class provides the interface for running those [ULActions](#) on a given [ULConnection](#).

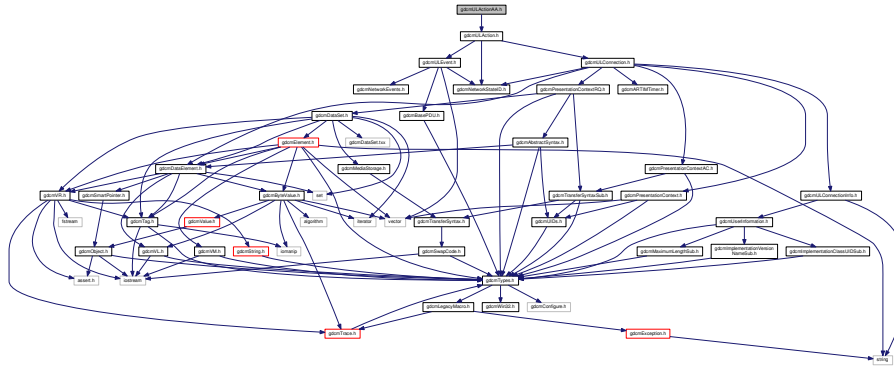
Namespaces

- [gdcm](#)
- [gdcm::network](#)

28.277 gdcmULActionAA.h File Reference

```
#include "gdcmULAction.h"
```

Include dependency graph for `gdcmlActionAA.h`:



Classes

- class `gdcml::network::ULActionAA1`
- class `gdcml::network::ULActionAA2`
- class `gdcml::network::ULActionAA3`
- class `gdcml::network::ULActionAA4`
- class `gdcml::network::ULActionAA5`
- class `gdcml::network::ULActionAA6`
- class `gdcml::network::ULActionAA7`
- class `gdcml::network::ULActionAA8`

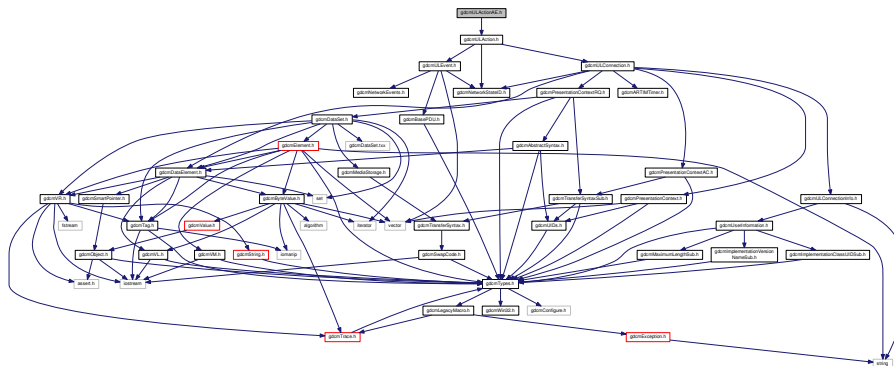
Namespaces

- `gdcml`
- `gdcml::network`

28.278 gdcmlActionAE.h File Reference

```
#include "gdcmlAction.h"
```

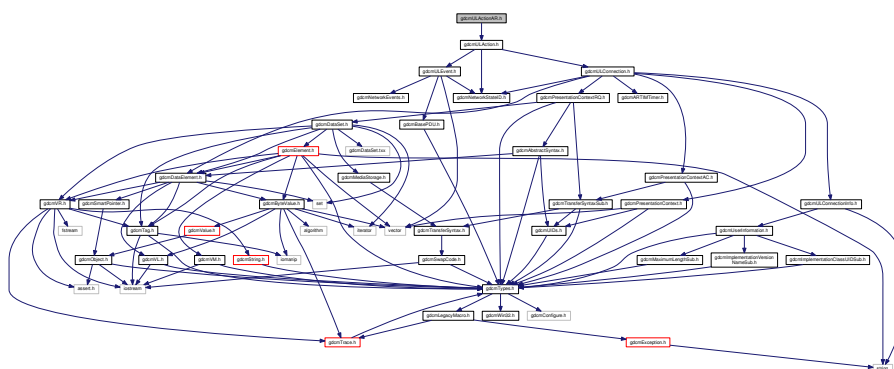
Include dependency graph for `gdcmlActionAE.h`:



- class [gdcn::network::ULActionAE1](#)
- class [gdcn::network::ULActionAE2](#)
- class [gdcn::network::ULActionAE3](#)
- class [gdcn::network::ULActionAE4](#)
- class [gdcn::network::ULActionAE5](#)
- class [gdcn::network::ULActionAE6](#)
- class [gdcn::network::ULActionAE7](#)
- class [gdcn::network::ULActionAE8](#)

- `gdcm`
- `gdcm::network`

Include dependency graph for gdcmlActionAR.h:



- class `gdcmm::network::ULActionAR1`
- class `gdcmm::network::ULActionAR10`
- class `gdcmm::network::ULActionAR2`
- class `gdcmm::network::ULActionAR3`
- class `gdcmm::network::ULActionAR4`
- class `gdcmm::network::ULActionAR5`
- class `gdcmm::network::ULActionAR6`
- class `gdcmm::network::ULActionAR7`
- class `gdcmm::network::ULActionAR8`
- class `gdcmm::network::ULActionAR9`

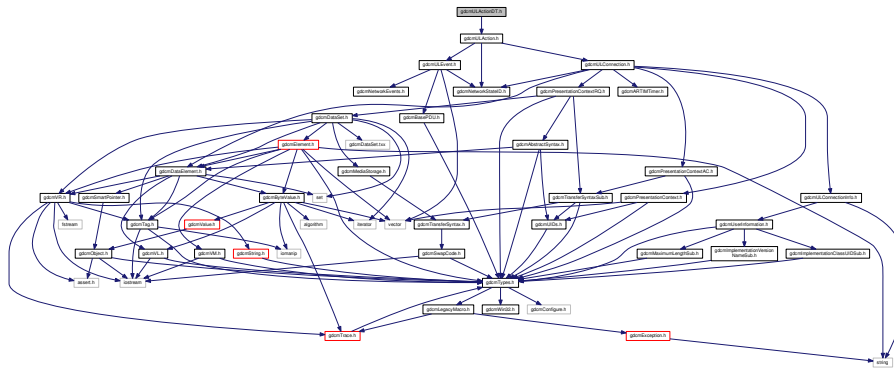
Namespaces

- [gdcm](#)
- [gdcm::network](#)

28.280 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](#)

28.281 gdcmULBasicCallback.h File Reference

```
#include "gdcmULConnectionCallback.h"
```

```
#include "gdcmDataSet.h"
```

```
#include <vector>
```

[illegible]

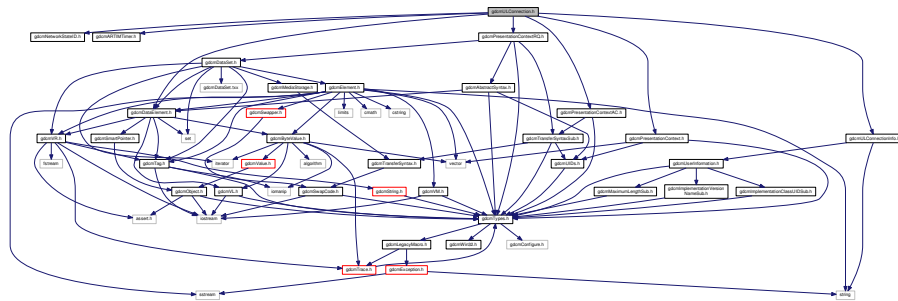
- class `gdcm::network::ULBasicCallback`

Namespaces

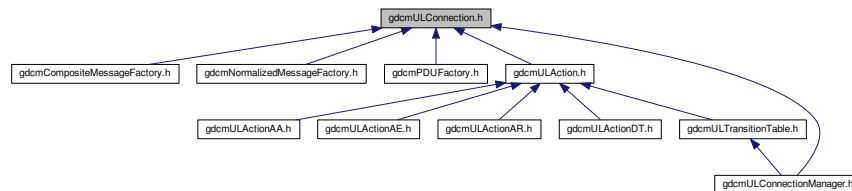
- ## 28.282 gdcmULConnection.h File Reference

Generated on Mon Dec 21 2015 23:27:54 for GDCM by Doxygen

Include dependency graph for `gdcmULConnection.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::network::ULConnection](#)

***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.*

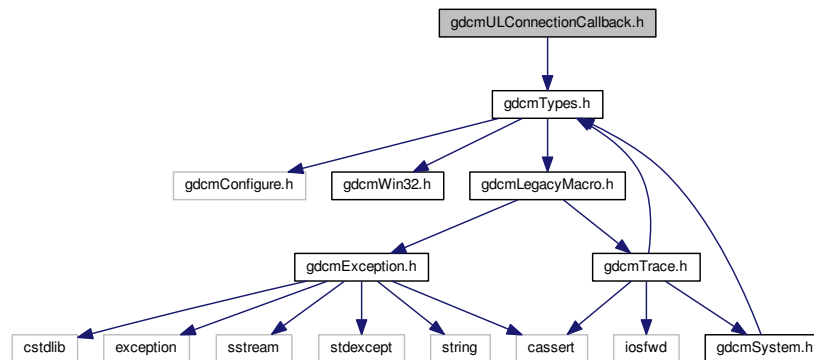
Namespaces

- [gdcm](#)
- [gdcm::network](#)

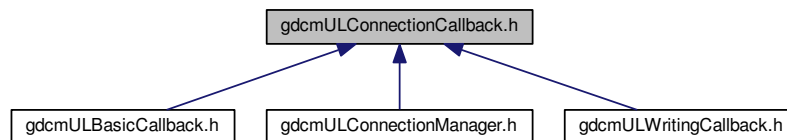
28.283 gdcmULConnectionCallback.h File Reference

```
#include "gdcmTypes.h"
```


Include dependency graph for gdcmULConnectionCallback.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::network::ULConnectionCallback](#)

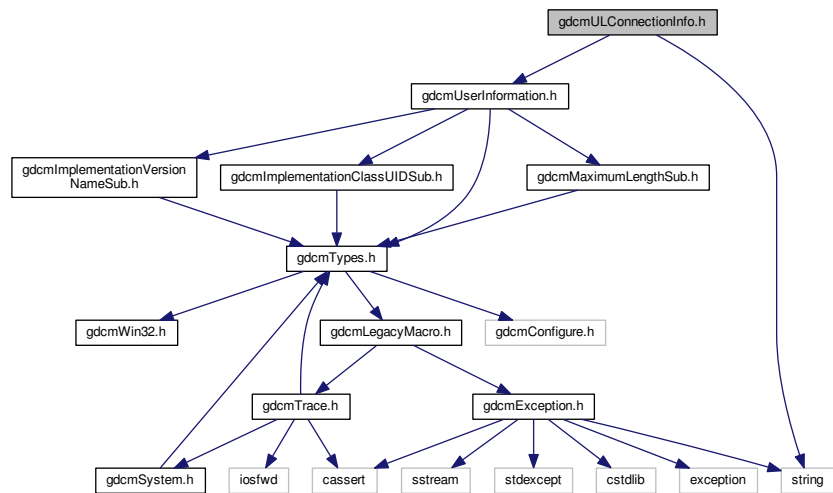
Namespaces

- [gdcm](#)
- [gdcm::network](#)

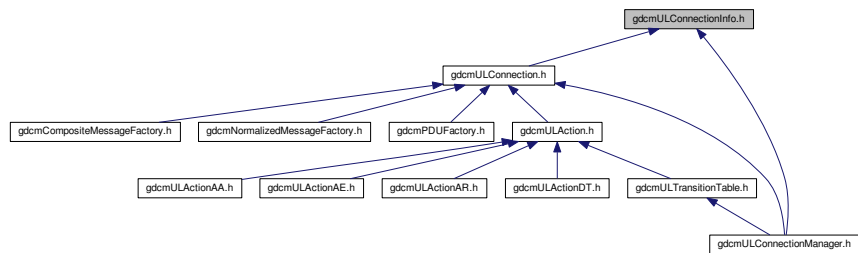
28.284 gdcmULConnectionInfo.h File Reference

```
#include "gdcmUserInformation.h"
#include <string>
```

Include dependency graph for `gdcmULConnectionInfo.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::network::ULConnectionInfo`

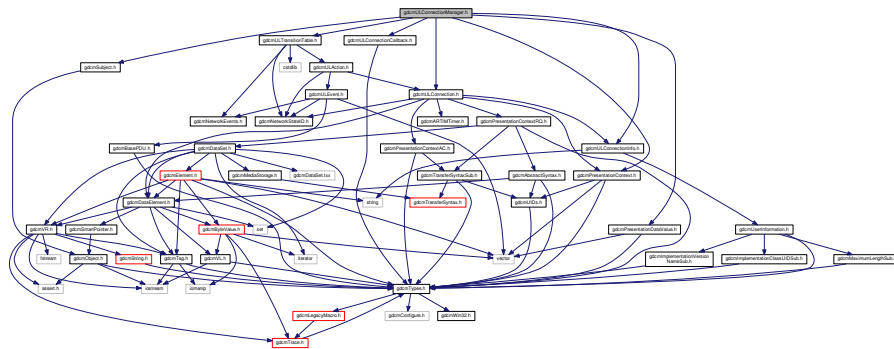
`ULConnectionInfo` this class contains all the information about a particular connection as established by the user. That is, it's: User Information Calling AE Title Called AE Title IP address/computer name IP Port A connection must be established with this information, that's subsequently placed into various primitives for actual communication.

Namespaces

- `gdcm`
- `gdcm::network`

28.285 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](#) 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).

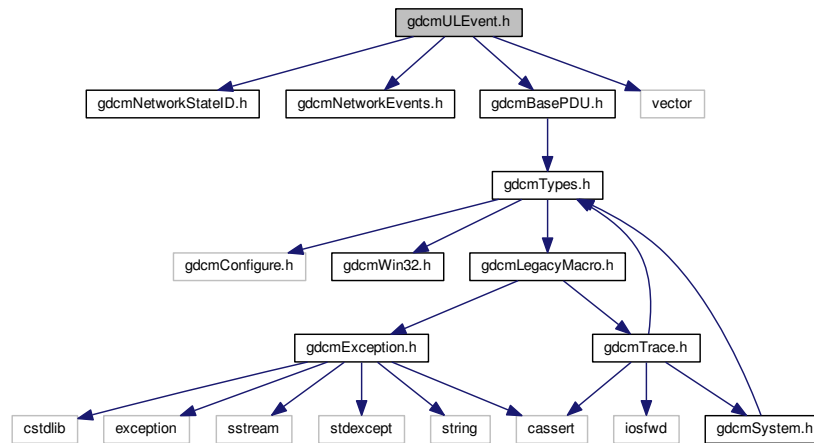
Namespaces

- [gdcm](#)
- [gdcm::network](#)

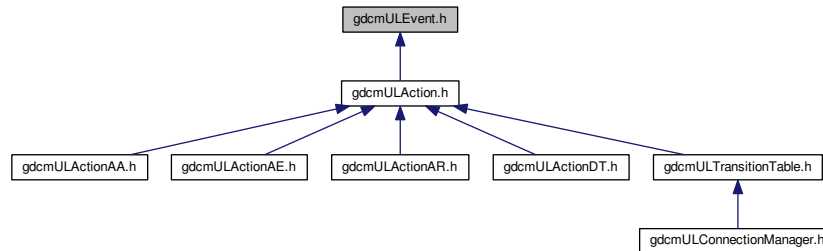
28.286 gdcmULEvent.h File Reference

```
#include "gdcmNetworkStateID.h"
#include "gdcmNetworkEvents.h"
#include "gdcmBasePDU.h"
#include <vector>
```

Include dependency graph for `gdcmlEvent.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcml::network::UEvent`
UEvent base class for network events.

Namespaces

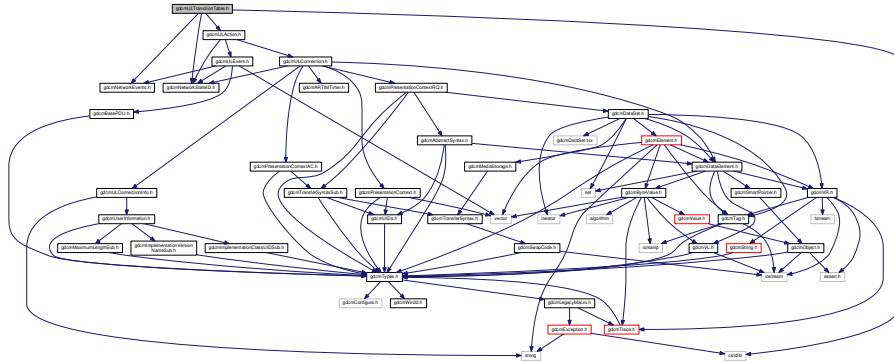
- `gdcml`
- `gdcml::network`

28.287 gdcmlTransitionTable.h File Reference

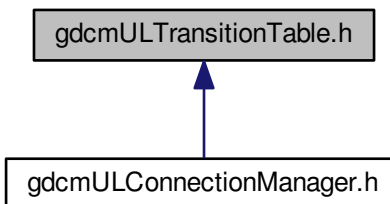
```
#include "gdcmlNetworkStateID.h"
```

```
#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

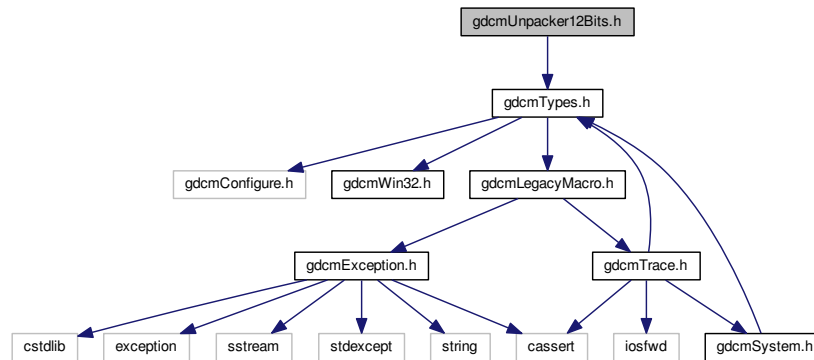
- class [gdcm::network::TableRow](#)
- struct [gdcm::network::Transition](#)
- class [gdcm::network::ULTransitionTable](#)

[ULTransitionTable](#) The transition table of all the ULEvents, new ULActions, and ULStates.

Namespaces

- [gdcm](#)
- [gdcm::network](#)

Include dependency graph for `gdcmUnpacker12Bits.h`:



Classes

- class [gdcm::Unpacker12Bits](#)
Pack/Unpack 12 bits pixel into 16bits.

Namespaces

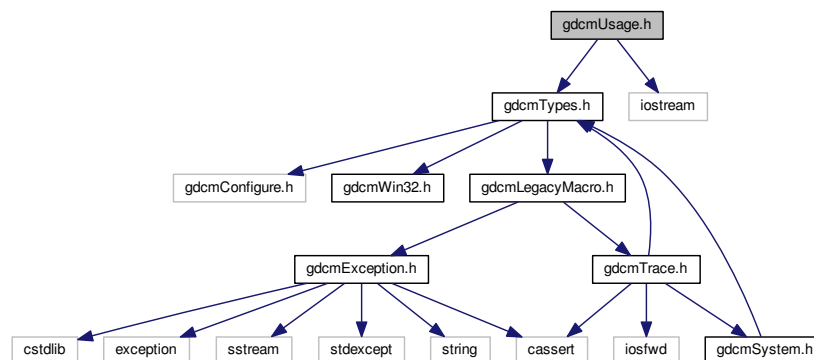
- [gdcm](#)

28.292 gdcmUsage.h File Reference

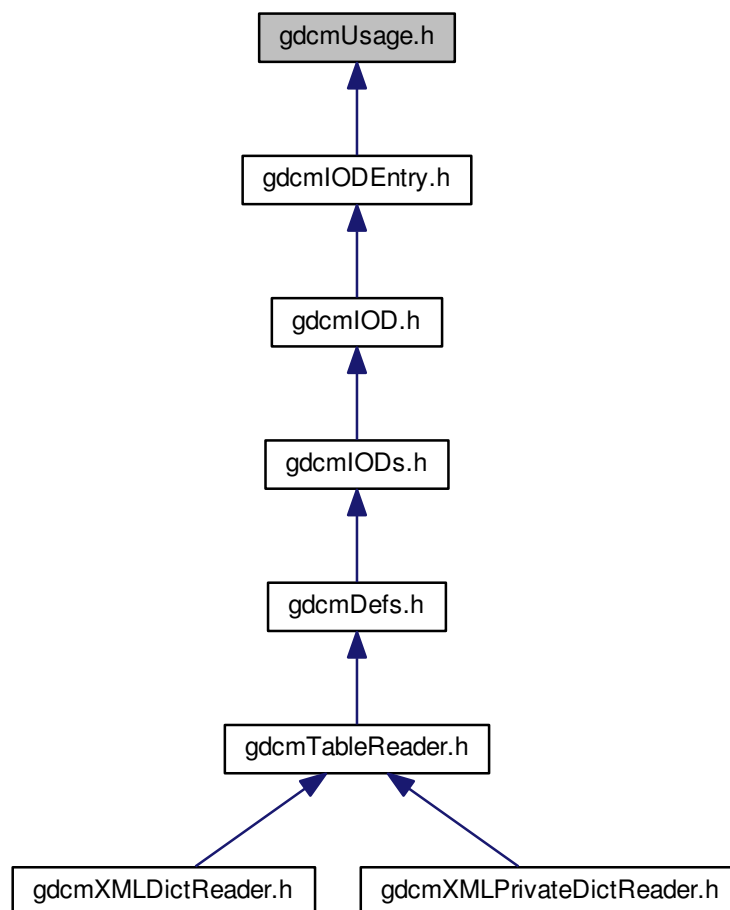
```
#include "gdcmTypes.h"
```

```
#include <iostream>
```

Include dependency graph for `gdcmUsage.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::Usage`
Usage.

Namespaces

- `gdcm`

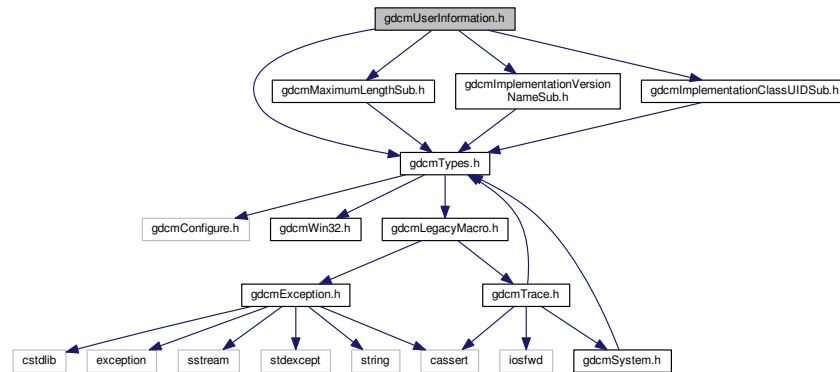
Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Usage &val)`

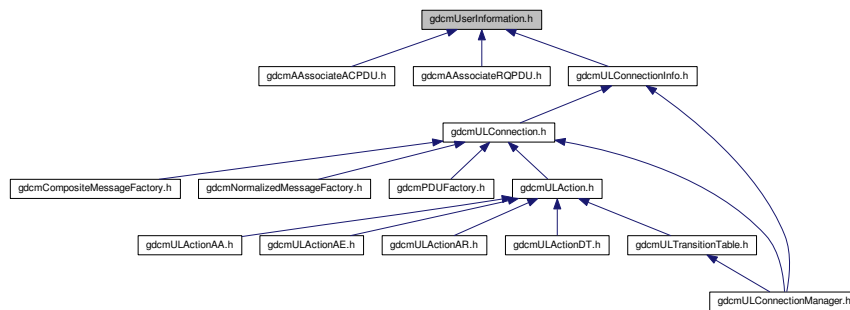
28.293 gdcmUserInformation.h File Reference

```
#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 Table 9-16 USER INFORMATION ITEM FIELDS.

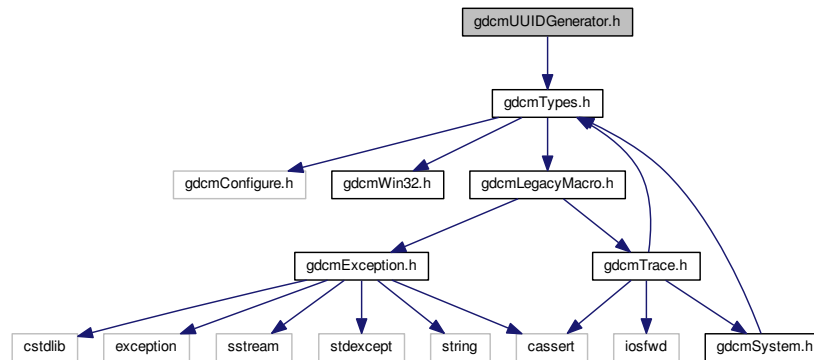
Namespaces

- [gdcm](#)
- [gdcm::network](#)

28.294 gdcmUUIDGenerator.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmUUIDGenerator.h:



Classes

- class [gdcm::UUIDGenerator](#)

Class for generating unique UUID generate DCE 1.1 uid.

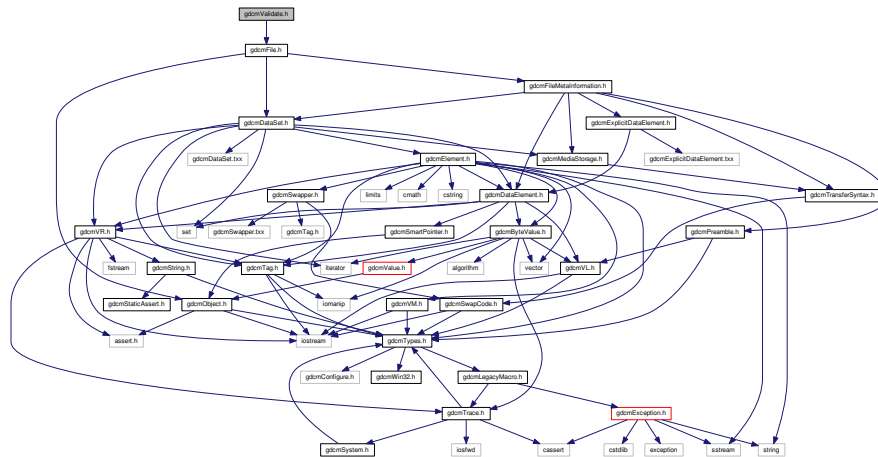
Namespaces

- [gdcm](#)

28.295 gdcmValidate.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for `gdcmValidate.h`:



Classes

- class `gdcm::Validate`

Validate class.

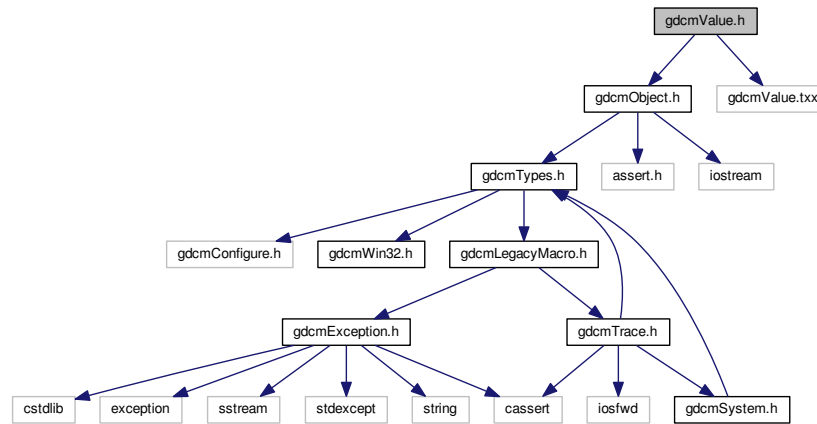
Namespaces

- `gdcm`

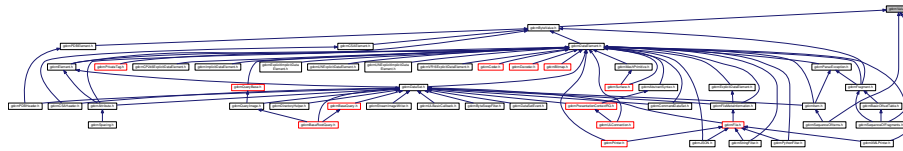
28.296 gdcmValue.h File Reference

```
#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](#).

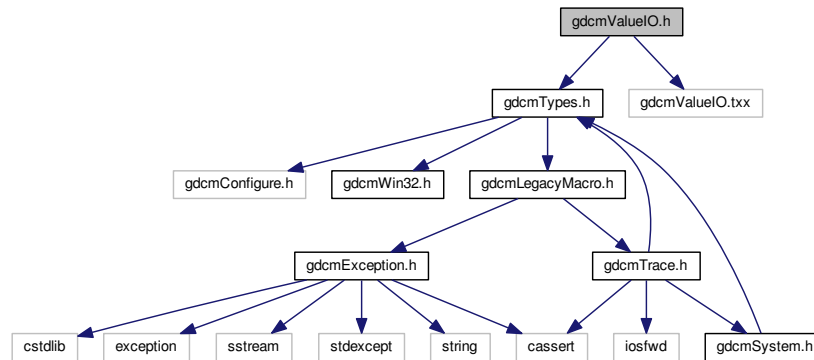
Namespaces

- [gdcm](#)

28.297 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

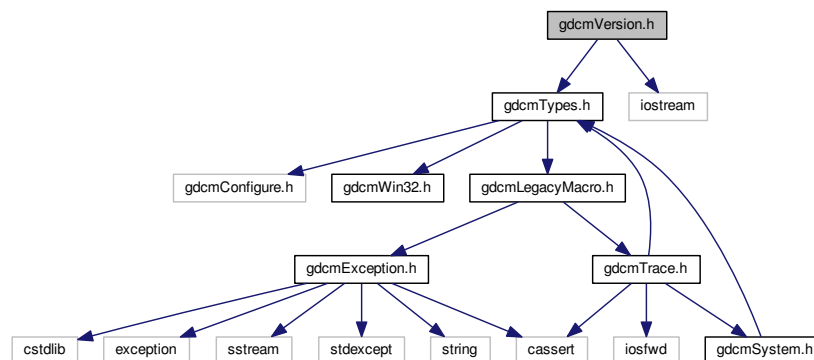
- `gdcm`

28.298 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](#)

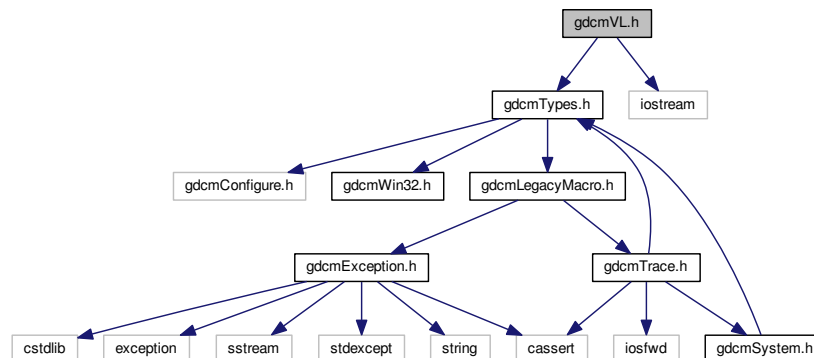
Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Version &v)`

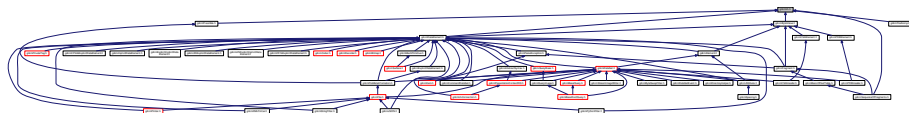
28.299 gdcviewer.dox File Reference

28.300 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](#)

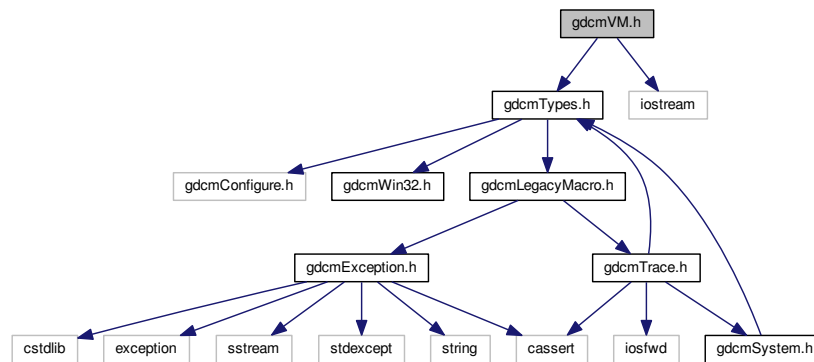
Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const VL &val)`

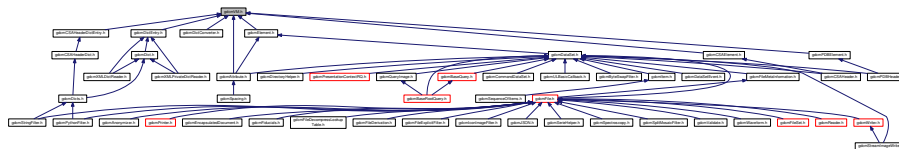
28.301 gdcmVM.h File Reference

```
#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.

- struct [gdcm::VMToLength< T >](#)

Namespaces

- [gdcm](#)

Macros

- `#define TYPETOLENGTH(type, length)`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const VM &_val)`

28.301.1 Macro Definition Documentation

28.301.1.1 `#define TYPETOLENGTH(type, length)`

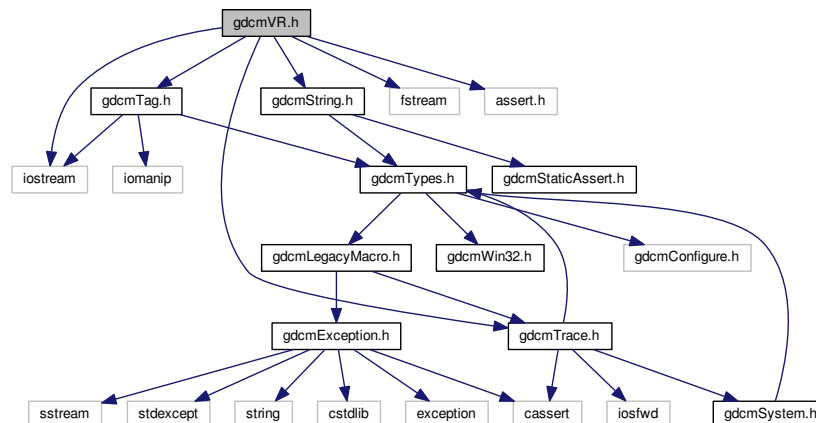
Value:

```
template<> struct VMToLength<VM::type> \
{ enum { Length = length }; };
```

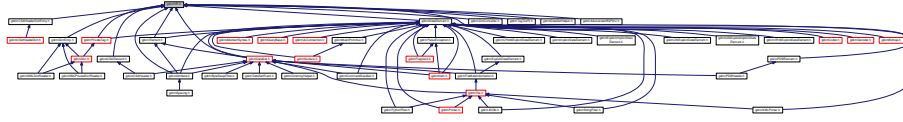
28.302 gdcmVR.h File Reference

```
#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 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.

- struct [gdcm::VRToEncoding< T >](#)
- struct [gdcm::VRToType< T >](#)

Namespaces

- [gdcm](#)

Macros

- #define [TYPETOENCODING](#)(type, rep, rtype)
- #define [VRTypeTemplateCase](#)(type)

Typedefs

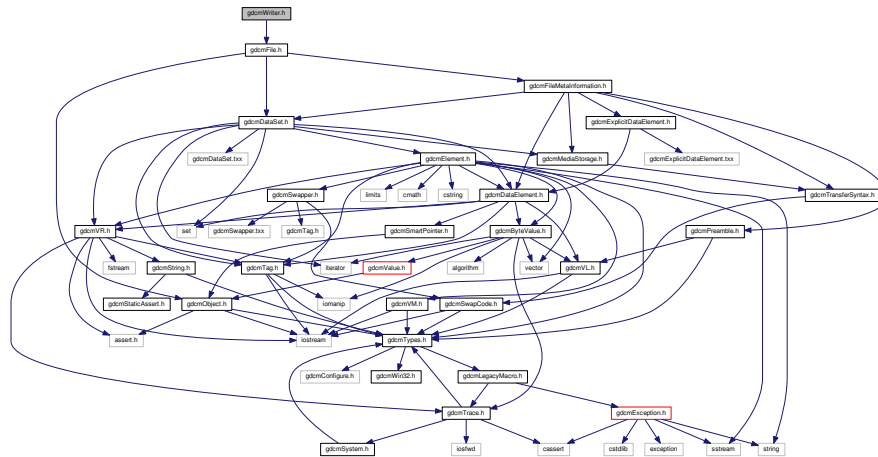
- typedef String<"\\", 16 > [gdcm::AEComp](#)
- typedef String<"\\", 64 > [gdcm::ASComp](#)
- typedef String<"\\", 16 > [gdcm::CSComp](#)
- typedef String<"\\", 64 > [gdcm::DAComp](#)
- typedef String<"\\", 64 > [gdcm::DTComp](#)
- typedef String<"\\", 64 > [gdcm::LOComp](#)
- typedef String<"\\", 64 > [gdcm::LTComp](#)
- typedef String<"\\", 64 > [gdcm::PNComp](#)
- typedef String<"\\", 64 > [gdcm::SHComp](#)
- typedef String<"\\", 64 > [gdcm::STComp](#)
- typedef String<"\\", 16 > [gdcm::TMComp](#)
- typedef String<"\\", 64, 0 > [gdcm::UIComp](#)
- typedef String<"\\", 64 > [gdcm::UTComp](#)

Functions

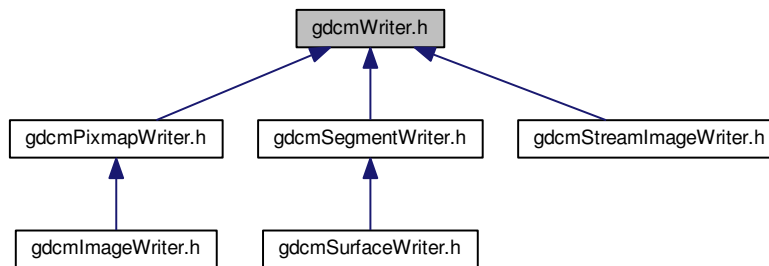
- std::ostream & [gdcm::operator<<](#) (std::ostream &_os, const VR &val)
- std::ostream & [gdcm::operator<<](#) (std::ostream &_os, const UI &_val)
- [gdcm::TYPETOENCODING](#) (SQ, VRBINARY, unsigned char) TYPETOENCODING(UN

- `gdcm::VRBINARY`

Include dependency graph for `gdcmWriter.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::Writer](#)

Writer ala DOM (Document *Object* Model) This class is a non-validating writer, it will only performs well- formedness check only.

Namespaces

- [gdcm](#)

28.308 gdcmxml.dox File Reference

Class for representing a [XMLPrivateDictReader](#).

Namespaces

- [gdcm](#)

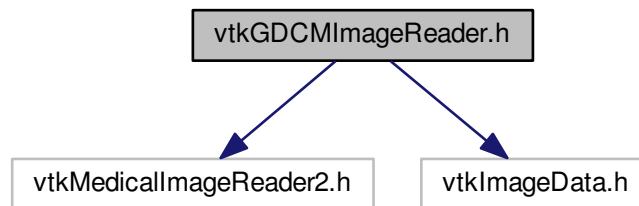
28.312 README.txt File Reference

28.313 TestsList.txt File Reference

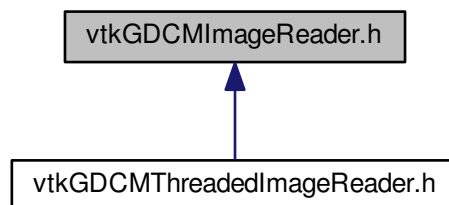
28.314 vtkGDCMImageReader.h File Reference

```
#include "vtkMedicalImageReader2.h"  
#include "vtkImageData.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`

28.314.1 Macro Definition Documentation

28.314.1.1 `#define VTK_CMYK 8`

28.314.1.2 `#define VTK_INVERSE_LUMINANCE 5`

28.314.1.3 `#define VTK_LOOKUP_TABLE 6`

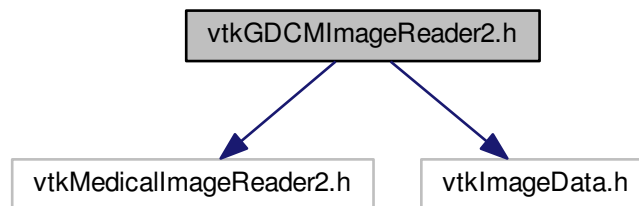
28.314.1.4 `#define VTK_YBR 7`

28.315 [vtkGDCMImageReader2.h](#) File Reference

```
#include "vtkMedicalImageReader2.h"
```

```
#include "vtkImageData.h"
```

Include dependency graph for [vtkGDCMImageReader2.h](#):



Classes

- class [vtkGDCMImageReader2](#)

Namespaces

- [gdcm](#)

Macros

- `#define VTK_CMYK` 8
- `#define VTK_INVERSE_LUMINANCE` 5
- `#define VTK_LOOKUP_TABLE` 6
- `#define VTK_YBR` 7

28.315.1 Macro Definition Documentation

28.315.1.1 `#define VTK_CMYK` 8

28.315.1.2 `#define VTK_INVERSE_LUMINANCE` 5

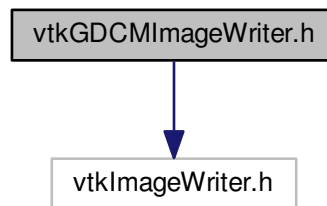
28.315.1.3 `#define VTK_LOOKUP_TABLE` 6

28.315.1.4 `#define VTK_YBR` 7

28.316 vtkGDCMImageWriter.h File Reference

```
#include "vtkImageWriter.h"
```

Include dependency graph for vtkGDCMImageWriter.h:



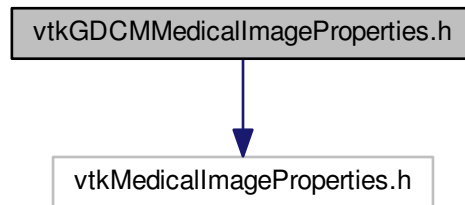
Classes

- class [vtkGDCMImageWriter](#)

28.317 vtkGDCMMedicalImageProperties.h File Reference

```
#include "vtkMedicalImageProperties.h"
```

Include dependency graph for `vtkGDCMMedicalImageProperties.h`:



Classes

- class [vtkGDCMMedicalImageProperties](#)

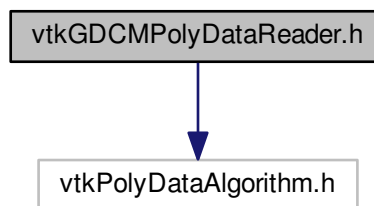
Namespaces

- [gdc](#)

28.318 vtkGDCMPolyDataReader.h File Reference

```
#include "vtkPolyDataAlgorithm.h"
```

Include dependency graph for `vtkGDCMPolyDataReader.h`:



Classes

- class [vtkGDCMPolyDataReader](#)

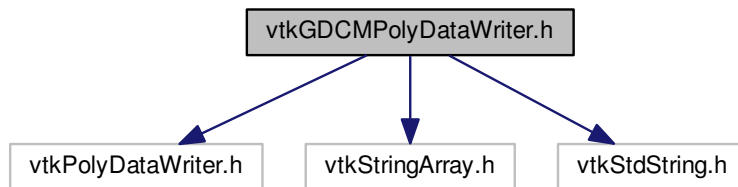
Namespaces

- [gdc](#)

28.319 vtkGDCMPolyDataWriter.h File Reference

```
#include "vtkPolyDataWriter.h"  
#include "vtkStringArray.h"  
#include "vtkStdString.h"
```

Include dependency graph for vtkGDCMPolyDataWriter.h:



Classes

- class [vtkGDCMPolyDataWriter](#)

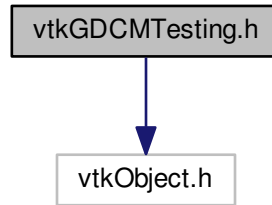
Namespaces

- [gdc](#)

28.320 vtkGDCMTesting.h File Reference

```
#include "vtkObject.h"
```

Include dependency graph for `vtkGDCMTesting.h`:



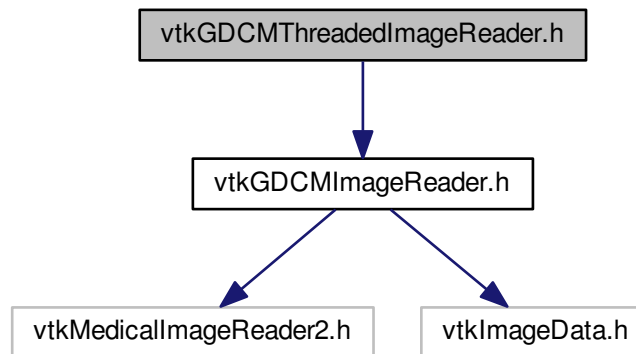
Classes

- class [vtkGDCMTesting](#)

28.321 vtkGDCMThreadedImageReader.h File Reference

```
#include "vtkGDCMImageReader.h"
```

Include dependency graph for `vtkGDCMThreadedImageReader.h`:



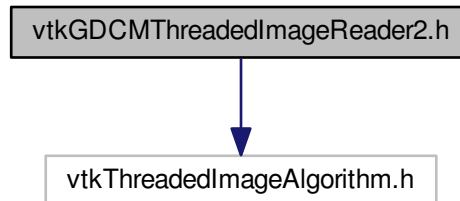
Classes

- class [vtkGDCMThreadedImageReader](#)

28.322 vtkGDCMThreadedImageReader2.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkGDCMThreadedImageReader2.h:



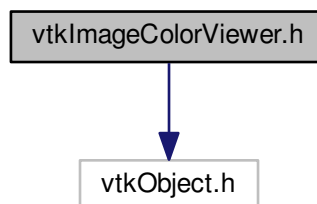
Classes

- class [vtkGDCMThreadedImageReader2](#)

28.323 vtkImageColorViewer.h File Reference

```
#include "vtkObject.h"
```

Include dependency graph for vtkImageColorViewer.h:



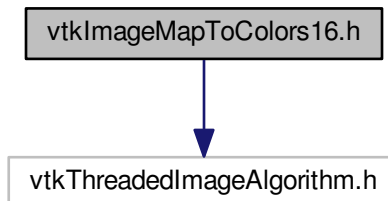
Classes

- class [vtkImageColorViewer](#)

28.324 vtkImageMapToColors16.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageMapToColors16.h:



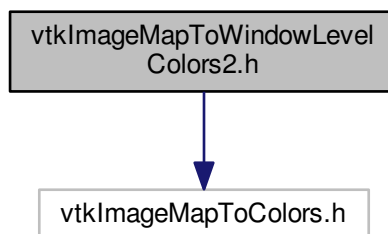
Classes

- class [vtkImageMapToColors16](#)

28.325 vtkImageMapToWindowLevelColors2.h File Reference

```
#include "vtkImageMapToColors.h"
```

Include dependency graph for vtkImageMapToWindowLevelColors2.h:



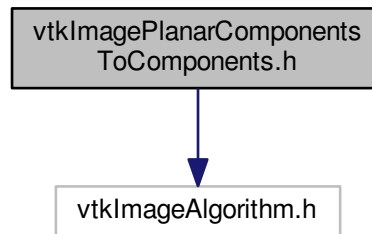
Classes

- class [vtkImageMapToWindowLevelColors2](#)

28.326 vtkImagePlanarComponentsToComponents.h File Reference

```
#include "vtkImageAlgorithm.h"
```

Include dependency graph for vtkImagePlanarComponentsToComponents.h:



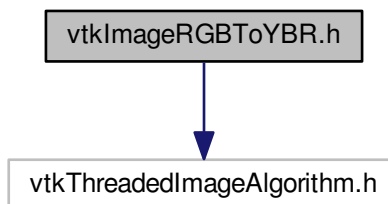
Classes

- class [vtkImagePlanarComponentsToComponents](#)

28.327 vtkImageRGBToYBR.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageRGBToYBR.h:



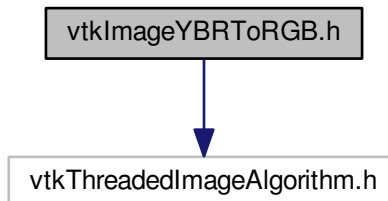
Classes

- class [vtkImageRGBToYBR](#)

28.328 vtkImageYBRToRGB.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageYBRToRGB.h:



Classes

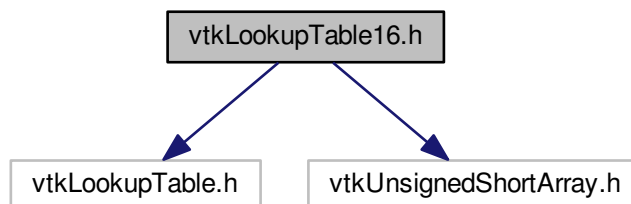
- class [vtkImageYBRToRGB](#)

28.329 vtkLookupTable16.h File Reference

```
#include "vtkLookupTable.h"
```

```
#include "vtkUnsignedShortArray.h"
```

Include dependency graph for vtkLookupTable16.h:



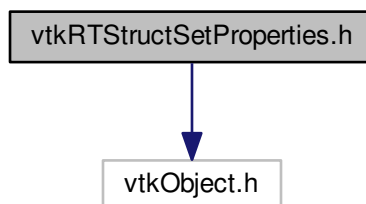
Classes

- class [vtkLookupTable16](#)

28.330 vtkRTStructSetProperties.h File Reference

```
#include "vtkObject.h"
```

Include dependency graph for vtkRTStructSetProperties.h:



Classes

- class [vtkRTStructSetProperties](#)

Chapter 29

Example Documentation

29.1 AWTMedical3.java

```
/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
package examples;

import vtk.*;
//import gdcm.*;

import vtk.util.VtkPanelContainer;
import vtk.util.VtkPanelUtil;
import vtk.util.VtkUtil;

import java.util.ArrayList;

import javax.swing.*;
import java.awt.*;
import java.io.File;

public class AWTMedical3 extends JComponent implements VtkPanelContainer {

    private vtkPanel renWin;

    vtkImageData ReadDataFile(File inSelectedFile){

        vtkImageData outImageData = null;
        Directory theDir = new Directory();

        String theInputDirectory = inSelectedFile.getPath();
        theDir.Load(theInputDirectory);

        Scanner theScanner = new Scanner();
        Tag theStudyTag = new Tag(0x0020,0x000d);
        Tag theSeriesTag = new Tag(0x0020,0x000e);
        theScanner.AddTag(theStudyTag); //get studies,
        theScanner.AddTag(theSeriesTag); //get studies,
        theScanner.Scan(theDir.GetFilenames());

        FilenamesType theStudyValues = theScanner.GetOrderedValues(theStudyTag);
        long theNumStudies = theStudyValues.size();
        //for now, take the first study, and nothing else.
        //and the return is actually not FilenamesType, just a
        //vector of strings
    }
}
```

```

    if (theNumStudies != 1)
        return outImageData;
    String theStudyVal = theStudyValues.get(0);
    //now, get all the values from the scanner that are in that
    //study, then from that get their different series
    FilenamesType theFilenames =
        theScanner.GetAllFilenamesFromTagToValue(theStudyTag, theStudyVal);

    //from that set of filenames, isolate individual series
    //conclude that singleton series = RT struct (can do further
    //checking for things like MIPs and the like)
    //and multiple series entries = volumetric data
    theScanner.Scan(theFilenames);
    FilenamesType theSeriesValues = theScanner.GetOrderedValues(theSeriesTag);
    String studyUID = theScanner.GetValue(theScanner.GetFilenames().get(0), theStudyTag);
    long theNumSeries = theSeriesValues.size();
    for (int i = 0; i < theNumSeries; i++) {
        FilenamesType theSeriesFiles =
            theScanner.GetAllFilenamesFromTagToValue(theSeriesTag, theSeriesValues.get(i));
        long theNumFilesInSeries = theSeriesFiles.size();
        if (theNumFilesInSeries > 1) { //assume it's CT or volumetric data
            //for now, assume a single volume
            //could have multiples, like PET and CT

            IPPSorter sorter = new IPPSorter();
            sorter.SetComputeZSpacing(true);
            sorter.SetZSpacingTolerance(0.001);
            Boolean sorted = sorter.Sort(theSeriesFiles);
            if (!sorted){
                //need some better way to handle failures here
                return outImageData;
            }

            FilenamesType sortedFT = sorter.GetFilenames();
            long theSize = sortedFT.size();
            vtkStringArray sa = new vtkStringArray();
            ArrayList<String> theStrings = new ArrayList<String>();

            vtkGDCMImageReader gdcmReader = new
            vtkGDCMImageReader();
            for (int j = 0; j < theSize; j++) {
                String theFileName = sortedFT.get(j);
                if (gdcmReader.CanReadFile(theFileName) > 0){
                    theStrings.add(theFileName);
                    sa.InsertNextValue(theFileName);
                } else {
                    //this is a busted series
                    //need some more appropriate error here
                    return outImageData;
                }
            }

            gdcmReader.SetFileNames(sa);

            gdcmReader.Update();

            outImageData = gdcmReader.GetOutput(); //the zeroth output should be the image
        }
    }
    String theImageInfo = "";
    if (outImageData != null){
        theImageInfo = outImageData.Print();
    }
    return outImageData;
}

//this function is a rewrite of Medical3 to see if data can
//be loaded via gdcm easily
public AWTMedical3(File inFile) {
    // Create the buttons.
    renWin = new vtkPanel();

    vtkImageData theImageData = ReadDataFile(inFile);

    // An isosurface, or contour value of 500 is known to correspond to the
    // skin of the patient. Once generated, a vtkPolyDataNormals filter is
    // is used to create normals for smooth surface shading during rendering.
    // The triangle stripper is used to create triangle strips from the
    // isosurface these render much faster on some systems.
    vtkContourFilter skinExtractor = new vtkContourFilter();
    skinExtractor.SetInput(theImageData);

```

```

skinExtractor.SetValue(0, 500);
vtkPolyDataNormals skinNormals = new vtkPolyDataNormals();
skinNormals.SetInput(skinExtractor.GetOutput());
skinNormals.SetFeatureAngle(60.0);
//      vtkStripper skinStripper = new vtkStripper();
//      skinStripper.SetInput(skinNormals.GetOutput());
vtkPolyDataMapper skinMapper = new vtkPolyDataMapper();
skinMapper.SetInput(skinNormals.GetOutput());
skinMapper.ScalarVisibilityOff();
vtkActor skin = new vtkActor();
skin.SetMapper(skinMapper);
skin.GetProperty().SetDiffuseColor(1, .49, .25);
skin.GetProperty().SetSpecular(.3);
skin.GetProperty().SetSpecularPower(20);

// An isosurface, or contour value of 1150 is known to correspond to the
// skin of the patient. Once generated, a vtkPolyDataNormals filter is
// is used to create normals for smooth surface shading during rendering.
// The triangle stripper is used to create triangle strips from the
// isosurface these render much faster on some systems.
vtkContourFilter boneExtractor = new vtkContourFilter();
boneExtractor.SetInput(theImageData);
boneExtractor.SetValue(0, 1150);
vtkPolyDataNormals boneNormals = new vtkPolyDataNormals();
boneNormals.SetInput(boneExtractor.GetOutput());
boneNormals.SetFeatureAngle(60.0);
vtkStripper boneStripper = new vtkStripper();
boneStripper.SetInput(boneNormals.GetOutput());
vtkPolyDataMapper boneMapper = new vtkPolyDataMapper();
boneMapper.SetInput(boneStripper.GetOutput());
boneMapper.ScalarVisibilityOff();
vtkActor bone = new vtkActor();
bone.SetMapper(boneMapper);
bone.GetProperty().SetDiffuseColor(1, 1, .9412);

// An outline provides context around the data.
vtkOutlineFilter outlineData = new vtkOutlineFilter();
outlineData.SetInput(theImageData);
vtkPolyDataMapper mapOutline = new vtkPolyDataMapper();
mapOutline.SetInput(outlineData.GetOutput());
vtkActor outline = new vtkActor();
outline.SetMapper(mapOutline);
outline.GetProperty().SetColor(0, 0, 0);

// Now we are creating three orthogonal planes passing through the
// volume. Each plane uses a different texture map and therefore has
// different coloration.

// Start by 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);
    }
}

```

29.2 BasicAnonymizer.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/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;
    }
}

```

29.3 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];

        // instanciate 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;
    }
}

```

29.4 CastConvertPhilips.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #   This software is distributed WITHOUT ANY WARRANTY; without even
10 #   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #   PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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 vtkgdcm
31 import vtk
32 import sys
33 import gdcm
34
35 def ProcessOneFilePublic(filename, outfilename, tmpfile):
36     gdcm.ImageHelper.SetForceRescaleInterceptSlope(True)
37     vtkreader = vtkgdcm.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 = vtkgdcm.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 = gdcm.ImageReader()
60     tmpreader.SetFileName( tmpfile )
61     if not tmpreader.Read():
62         sys.exit(1)
63
64     reader = gdcm.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         gdcm.Tag(0x0028,0x1052),
73         gdcm.Tag(0x0028,0x1053),
74         gdcm.Tag(0x0028,0x1053),
75     ]
76     for tag in tags:
77         ds.Remove( tag )
78

```

```

79 writer = gdcmm.ImageWriter()
80 writer.SetFileName( outfile )
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, outfile, 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( outfile )
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     gdcmm.Trace.DebugOff()
169     gdcmm.Trace.WarningOff()
170     #filename = sys.argv[1]
171     #outfilename = sys.argv[2]
172     tmpfile = "/tmp/philips_rescaled.dcm"
173     #ProcessOneFile( filename, outfile, tmpfile )
174     rescaletype = sys.argv[1]
175     assert rescaletype == "--public" or rescaletype == "--private"
176     dirname = sys.argv[2]
177     d = gdcmm.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"

```

29.5 ChangePrivateTags.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 "gdcmmPrivateTag.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];

    gdcmm::Reader reader;
    reader.SetFileName( filename );
    if (! reader.Read() )
    {
        return 1;
    }

    // (0029,0010) LO [SIEMENS CSA HEADER] # 18,1 Private Creator
    // (0029,0011) LO [SIEMENS MEDCOM HEADER ] # 22,1 Private Creator
    // (0029,0012) LO [SIEMENS MEDCOM HEADER2] # 22,1 Private Creator
    // [...]
    // (0029,1018) CS [MR] # 2,1 CSA Series Header Type
    // (0029,1134) CS [DB TO DICOM ] # 12,1 PMTF Information 4
    // (0029,1260) LO [com ] # 4,1 Series Workflow Status

    gdcmm::File &file = reader.GetFile();
    gdcmm::DataSet &ds = file.GetDataSet();

    // Declare private tag we need to find:
    gdcmm::PrivateTag pt1( 0x29,0x18, "SIEMENS CSA HEADER" );
    gdcmm::PrivateTag pt2( 0x29,0x34, "SIEMENS MEDCOM HEADER" );

```

```

gdcmm::PrivateTag pt3( 0x29,0x60, "SIEMENS MEDCOM HEADER2" );

const char str1[] = "GDCM was here 3!";
if( !ds.FindDataElement( pt1 ) ) return 1;
gdcmm::DataElement de1 = ds.GetDataElement( pt1 ); // Convert Private tag,
    into actual DataElement
std::cout << de1 << std::endl;
de1.SetByteValue( str1, (uint32_t)strlen(str1) );
ds.Replace( de1 );

const char str2[] = "GDCM was here 2!";
if( !ds.FindDataElement( pt2 ) ) return 1;
gdcmm::DataElement de2 = ds.GetDataElement( pt2 );
std::cout << de2 << std::endl;
de2.SetByteValue( str2, (uint32_t)strlen(str2) );
ds.Replace( de2 );

const char str3[] = "GDCM was here 3!";
if( !ds.FindDataElement( pt3 ) ) return 1;
gdcmm::DataElement de3 = ds.GetDataElement( pt3 );
std::cout << de3 << std::endl;
de3.SetByteValue( str3, (uint32_t)strlen(str3) );
ds.Replace( de3 );

gdcmm::Writer writer;
writer.SetFile( file );
writer.SetFileName( outfilename );
if ( !writer.Write() )
{
    return 1;
}

return 0;
}

```

29.6 ChangeSequenceUltrasound.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
#include "gdcmmReader.h"
#include "gdcmmWriter.h"
#include "gdcmmSmartPointer.h"
#include "gdcmmDataSetHelper.h"

/*
./ChangeSequenceUltrasound gdcmmData/D_CLUNIE_CT1_J2KI.dcm myoutput.dcm

This is the exact C++ translation of the original python example: ManipulateSequence.py
*/

int main(int argc, char* argv[] )
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    gdcmm::Reader reader;
    reader.SetFileName( filename );
    if ( ! reader.Read() )
    {
        return 1;
    }
}

```

```

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( outfilename );
if ( !writer.Write() )
{
    return 1;
}

return 0;
}

```

29.7 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;
}

```

29.8 ClinicalTrialAnnotate.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
/*
 * Dummy implementation of C.7.1.3 Clinical Trial Subject Module
 *
 * Usage:
 * ClinicalTrialAnnotate gdcmData/012345.002.050.dcm out.dcm
 */

#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmAnonymizer.h"

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Could not read: " << filename << std::endl;
        return 1;
    }

    // The output of gdcm::Reader is a gdcm::File
    //gdcm::File &file = reader.GetFile();

    // the dataset is the the set of element we are interested in:
    //gdcm::DataSet &ds = file.GetDataSet();

    gdcm::Anonymizer ano;
    ano.SetFile( reader.GetFile() );
    ano.RemoveGroupLength();
    ano.RemovePrivateTags();

    // PS 3.3 - 2008
    // C.7.1.3 Clinical Trial Subject Module
    // <entry group="0012" element="0010" vr="LO" vm="1" name="Clinical Trial Sponsor Name"/>
    ano.Replace( gdcm::Tag(0x12,0x10), "BigCompany name" );
    // <entry group="0012" element="0020" vr="LO" vm="1" name="Clinical Trial Protocol ID"/>
    ano.Replace( gdcm::Tag(0x12,0x20), "My Clinical Trial Protocol ID" );
    // <entry group="0012" element="0021" vr="LO" vm="1" name="Clinical Trial Protocol Name"/>
    ano.Replace( gdcm::Tag(0x12,0x21), "My Clinical Trial Protocol Name" );
    // <entry group="0012" element="0030" vr="LO" vm="1" name="Clinical Trial Site ID"/>
    ano.Replace( gdcm::Tag(0x12,0x30), "My Clinical Trial Site ID" );
    // <entry group="0012" element="0031" vr="LO" vm="1" name="Clinical Trial Site Name"/>
    ano.Replace( gdcm::Tag(0x12,0x31), "My Clinical Trial Site Name" );
    // <entry group="0012" element="0040" vr="LO" vm="1" name="Clinical Trial Subject ID"/>
    ano.Replace( gdcm::Tag(0x12,0x40), "My Clinical Trial Subject ID" );
    // <entry group="0012" element="0042" vr="LO" vm="1" name="Clinical Trial Subject Reading ID"/>

```

```

ano.Replace( gdcM::Tag(0x12,0x42), "My Clinical Trial Subject Reading ID" );

gdcM::Writer writer;
writer.SetFile( reader.GetFile() );
writer.SetFileName( outfilename );
if( !writer.Write() )
{
    return 1;
}

return 0;
}

```

29.9 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 gdcm.Tag(0x0012,0x0010), "MySponsorName");
        ano.Replace( new gdcm.Tag(0x0012,0x0020), "MyProtocolID");
        ano.Replace( new gdcm.Tag(0x0012,0x0021), "MyProtocolName");
        ano.Replace( new gdcm.Tag(0x0012,0x0030), "MySiteId");
        ano.Replace( new gdcm.Tag(0x0012,0x0031), "MySiteName");
        ano.Replace( new gdcm.Tag(0x0012,0x0040), "MySponsorId");
        ano.Replace( new gdcm.Tag(0x0012,0x0050), "MyTPId");
        ano.Replace( new gdcm.Tag(0x0012,0x0051), "MyTPDescription");

        // The following two are not required as they are guaranteed to be filled in by the
        // Basic Application Level Confidentiality Profile. Only override if you understand what
        // you are doing
        //ano.Replace( new gdcm.Tag(0x0012,0x0062), "YES");
        //ano.Replace( new gdcm.Tag(0x0012,0x0063), "My Super Duper Anonymization Overload");

        // We might be generating a subdirectory. Let's make sure the subdir exist:
        gdcm.Filename fn = new gdcm.Filename( outfilename );
        string subdir = fn.GetPath();
        if( !gdcm.PosixEmulation.MakeDirectory( subdir ) )
        {
            return false;
        }

        gdcm.FileMetaInformation fmi = ano.GetFile().GetHeader();
        // The following three lines make sure to regenerate any value:
        fmi.Remove( new gdcm.Tag(0x0002,0x0012) );
        fmi.Remove( new gdcm.Tag(0x0002,0x0013) );
        fmi.Remove( new gdcm.Tag(0x0002,0x0016) );

        Writer writer = new Writer();
        writer.SetFileName( outfilename );
        writer.SetFile( ano.GetFile() );
        ret = writer.Write();
        if( !ret )
        {
            return false;
        }
    }
}

```

```

    return true;
}

public static int Main(string[] args)
{
    gdcmm.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";
    gdcmm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
    System.Console.WriteLine( "Root dir is now: " + gdcmm.UIDGenerator.
        GetRoot() );

    gdcmm.Global global = gdcmm.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( !gdcmm.PosixEmulation.FileIsDirectory(dir1) )
    {
        System.Console.WriteLine( "Input directory: " + dir1 + " does not exist. Sorry" );
        return 1;
    }
    if( !gdcmm.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 = gdcmm.FileName.Join(gdcmm.Testing.
        GetSourceDirectory(), "/Testing/Source/Data/certificate.pem" );
    gdcmm.CryptoFactory fact = gdcmm.CryptoFactory.
        GetFactoryInstance();
    gdcmm.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;
}
}

```

29.10 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( outfilename, std::ios::binary );
    //image.GetBuffer2(out);
    //out.close();
    gdcm::ImageWriter writer;
    writer.SetImage( change.GetOutput() );
    writer.SetFile( reader.GetFile() );
    writer.SetFileName( outfilename );
    if( !writer.Write() )
    {
        return 1;
    }

    return 0;
}

```

29.11 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;
}
}

```

29.12 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"

/*
 * 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] );
}

gdcmm::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 ippszspacing = 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;
}

```

29.13 Convert16BitsTo8Bits.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 "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageCast.h"

#include "gdcmmTesting.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:
// gdcmmData/012345.002.050.dcm

int main(int, char *[])
{
    const char *directory = gdcmm::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;
}

```

29.14 ConvertMPL.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #   This software is distributed WITHOUT ANY WARRANTY; without even
10 #   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #   PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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     ## use float for accurate scaling
63     result = numpy.frombuffer(gdcM_array, dtype=dtype).astype(float)
64     ## optional gamma scaling
65     #maxV = float(result[result.argmax()])
66     #result = result + .5*(maxV-result)
67     #result = numpy.log(result+50) ## apprx background level
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     ## many colormaps are available
82     imshow(numpy_array, interpolation='bilinear', cmap=cm.jet)
83     ## set the plot sizes and placement
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()

```

29.15 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::ostream os;
    os << "singleframe";
    os << "%04d.dcm";
    gdcm::FilenameGenerator fg;
    fg.SetPattern( os.str().c_str() );
    unsigned int nfiles = dims[2];
    fg.SetNumberOfFileNames( nfiles );
    bool b = fg.Generate();
    if( !b )
    {
        std::cerr << "FilenameGenerator::Generate() failed" << std::endl;
        return 1;
    }
    if( !fg.GetNumberOfFileNames() )
    {
        std::cerr << "FilenameGenerator::Generate() failed somehow..." << std::endl;
        return 1;
    }

    // By default write them as Secondary Capture (for portability)
    vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
    vtkStringArray *filenames = vtkStringArray::New();
    for(unsigned int i = 0; i < fg.GetNumberOfFileNames(); ++i)
    {
        filenames->InsertNextValue( fg.GetFilename(i) );
    }
    assert( filenames->GetNumberOfValues() == (int)fg.GetNumberOfFileNames() );
    writer->SetFileNames( filenames );
    filenames->Delete();
    writer->SetFileDimensionality( 2 );
    #if (VTK_MAJOR_VERSION >= 6)
    writer->SetInputConnection( reader->GetOutputPort() );
    #else
    writer->SetInput( reader->GetOutput() );
    #endif
    writer->SetImageFormat( reader->GetImageFormat() );
    writer->Write();

    reader->Delete();
    writer->Delete();

    return 0;
}

```

29.16 ConvertNumpy.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #

```

```

9 # This software is distributed WITHOUT ANY WARRANTY; without even
10 # the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 # PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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_tymap():
29     """Returns the GDCM Pixel Format to numpy array type mapping."""
30     _gdcm_np = {gdcm.PixelFormat.UINT8 :numpy.int8,
31                 gdcm.PixelFormat.INT8 :numpy.uint8,
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_tymap()[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_tymap().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     gdcm_array = image.GetBuffer()
61     result = numpy.frombuffer(gdcm_array, dtype=dtype)
62     result.shape = shape
63     return result
64
65 if __name__ == "__main__":
66     import sys
67     r = gdcm.ImageReader()
68     filename = sys.argv[1]
69     r.SetFileName( filename )
70     if not r.Read():
71         sys.exit(1)
72
73     numpy_array = gdcm_to_numpy( r.GetImage() )
74     print numpy_array

```

29.17 ConvertPIL.py

```

1 #####
2 #
3 # Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 # Copyright (c) 2006-2011 Mathieu Malaterre
6 # All rights reserved.
7 # See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #

```

```

9 # This software is distributed WITHOUT ANY WARRANTY; without even
10 # the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 # PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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 gdcm
30 import numpy
31 from PIL import Image, ImageOps
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     result = numpy.frombuffer(gdcm_array, dtype=dtype)
63     maxV = float(result[result.argmax()])
64     ## linear gamma adjust
65     #result = result + .5*(maxV-result)
66     ## log gamma
67     result = numpy.log(result+50) ## 50 is apprxx background level
68     maxV = float(result[result.argmax()])
69     result = result*(2.**8/maxV) ## histogram stretch
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     ## L is 8 bit grey
81     ## http://www.pythonware.com/library/pil/handbook/concepts.htm
82     pilImage = Image.frombuffer('L',
83                                numpy_array.shape,
84                                numpy_array.astype(numpy.uint8),
85                                'raw','L',0,1)
86     ## cutoff removes background noise and spikes
87     pilImage = ImageOps.autocontrast(pilImage, cutoff=.1)
88     pilImage.save(sys.argv[1]+'.jpg')

```

29.18 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;
}

```

29.19 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;
}

```


29.20 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 gdcmlData/012345.002.050.dcm is [0,192], we could simply do:
        // *pubbuffer++ = *buffer16;
        // *pubbuffer++ = *buffer16;
        // *pubbuffer++ = *buffer16;
        // instead do it right:
        *pubbuffer++ = (unsigned char)std::min(255, (32768 + *buffer16) / 255);
        *pubbuffer++ = (unsigned char)std::min(255, (32768 + *buffer16) / 255);
        *pubbuffer++ = (unsigned char)std::min(255, (32768 + *buffer16) / 255);
        buffer16++;
    }

    QImage *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;
}

```

29.21 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;
}

```

29.22 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( outfile );
    if( !writer.Write() )
    {
        return 1;
    }
    delete[] buf;

    return 0;
}
```

29.23 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;
}

```

29.24 CreateFakeRTDOSE.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageWriter.h"
#include "vtkImageReader.h"
#include "vtkImageCast.h"
#include "vtkImageData.h"
#include "vtkPointData.h"
#include "vtkDataArray.h"
#include "vtkMedicalImageProperties.h"

#include "gdcmlTrace.h"
#include "gdcmlReader.h"
#include "gdcmlWriter.h"
#include "gdcmlAttribute.h"

/*
 * Minimal example to create a fake RTDOSE file. The data contains a sphere
 * just for testing.
 * The vtkMedicalImageProperties is not properly filled, but only contains a
 * single field which is required to set the proper SOP Class
 */
int main(int, char *[])
{
    //gdcml::Trace::DebugOn();

    const vtkIdType xSize = 512;
    const vtkIdType ySize = 512;
    const vtkIdType zSize = 512;

    vtkImageData *image = vtkImageData::New();
    image->SetDimensions(xSize,ySize,zSize);
    image->SetOrigin(-350.684,350.0,890.76);

```

```

    image->SetSpacing(5.4688,-5.4688,-3.27);
    #if VTK_MAJOR_VERSION <= 5
        image->SetNumberOfScalarComponents(1);
        image->SetScalarTypeToDouble();
    #else
        image->AllocateScalars(VTK_DOUBLE,1);
    #endif

    double pt[3];
    for( int z = 0; z < zSize; ++z )
        for( int y = 0; y < ySize; ++y )
            for( int x = 0; x < xSize; ++x )
                {
                    pt[0] = x;
                    pt[1] = y;
                    pt[2] = z;
                    pt[0] -= xSize / 2;
                    pt[1] -= ySize / 2;
                    pt[2] -= zSize / 2;
                    pt[0] /= xSize / 2;
                    pt[1] /= ySize / 2;
                    pt[2] /= zSize / 2;
                    const double unit = pt[0] * pt[0] + pt[1] * pt[1] + pt[2] * pt[2];
                    const double inval = unit <= 1. ? (3 * unit + 7) : 0.; // just for fun => max == 10.
                    double* pixel= static_cast<double*>(image->GetScalarPointer(x,y,z));
                    pixel[0] = inval;
                }

    vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
    writer->SetFileDimensionality( 3 );
    writer->SetFileName( "rtdose.dcm" );
    #if (VTK_MAJOR_VERSION >= 6)
        writer->SetInputData( image );
    #else
        writer->SetInput( image );
    #endif
    writer->GetMedicalImageProperties()->SetSliceThickness("1.5");
    writer->GetMedicalImageProperties()->AddUserDefinedValue( "Dose Units", "GY");
    writer->GetMedicalImageProperties()->AddUserDefinedValue( "Dose Summation Type", "PLAN");
    writer->GetMedicalImageProperties()->AddUserDefinedValue( "Dose Type", "PHYSICAL");
    writer->GetMedicalImageProperties()->AddUserDefinedValue( "Frame of Reference UID", "
        1.3.12.2.1107.5.6.1.68100.30270111041215391275000000001");
    writer->GetMedicalImageProperties()->SetModality( "RTDOSE" );
    //writer->GetMedicalImageProperties()->SetModality( "PT" ); // debug
    writer->SetScale( 0.0042 ); // why not
    writer->Write();

    image->Delete();
    writer->Delete();

    // BEGIN HACK
    // In GDCM version 2.4.3 and before, the following tag was missing which caused issue with some RTDose
    // software:

    // Open the DICOM file that was temporarily created. This will allows me to used
    // GDCM to append specific tags that allows the RTDOSE to be associated with the
    // relevant CT images.
    gdcm::Reader reader2;
    reader2.SetFileName("rtdose.dcm" );
    reader2.Read();
    gdcm::File &file = reader2.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();

    // Required by some software and not 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;
}

```

29.25 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( outfile );

    file.GetHeader().SetDataSetTransferSyntax(
        gdcm::TransferSyntax::JPIPReferenced );

    gdcm::Anonymizer anon;
    anon.SetFile( file );

    gdcm::MediaStorage ms =
        gdcm::MediaStorage::SecondaryCaptureImageStorage;

    gdcm::UIDGenerator gen;
    anon.Replace( gdcm::Tag(0x0008,0x16), ms.GetString() );
    std::cout << ms.GetString() << std::endl;
    anon.Replace( gdcm::Tag(0x0008,0x18), gen.Generate() );
    //
    anon.Replace( gdcm::Tag(0x0010,0x10), "JPIP^EXAMPLE" );
    anon.Replace( gdcm::Tag(0x0010,0x20), "012345" );
    anon.Empty( gdcm::Tag(0x0010,0x30) );
    anon.Empty( gdcm::Tag(0x0010,0x40) );
    anon.Empty( gdcm::Tag(0x0008,0x20) );
    anon.Empty( gdcm::Tag(0x0008,0x30) );
    anon.Empty( gdcm::Tag(0x0008,0x90) );
    anon.Empty( gdcm::Tag(0x0020,0x10) );
    anon.Empty( gdcm::Tag(0x0020,0x11) );
    anon.Empty( gdcm::Tag(0x0008,0x50) );
    anon.Empty( gdcm::Tag(0x0020,0x0013) );
    anon.Replace( gdcm::Tag(0x0020,0xd), gen.Generate() );
    anon.Replace( gdcm::Tag(0x0020,0xe), gen.Generate() );
    anon.Replace( gdcm::Tag(0x0008,0x64), "WSD " );
    anon.Replace( gdcm::Tag(0x0008,0x60), "OT" );

    gdcm::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;
}

```

29.26 CreateRAWStorage.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #       This software is distributed WITHOUT ANY WARRANTY; without even
10 #       the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #       PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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" retired=
17     "false"/>
18 """
19
20 import gdcm
21 import sys,os
22
23 if __name__ == "__main__":
24     r = gdcm.Reader()
25     # Will require Testing...
26     dataroot = gdcm.Testing.GetDataRoot()
27     filename = os.path.join( dataroot, '012345.002.050.dcm' )
28     r.SetFileName( filename )
29     r.Read()
30     f = r.GetFile()
31     ds = f.GetDataSet()
32
33     uid = "1.2.840.10008.5.1.4.1.1.66"
34     # f = gdcm.File()
35     ds = f.GetDataSet()
36     de = gdcm.DataElement( gdcm.Tag(0x0008,0x0016) )
37     de.SetByteValue( uid, gdcm.VL(len(uid)) )
38     vr = gdcm.VR( gdcm.VR.UI )
39     de.SetVR( vr )
40     ds.Replace( de )
41
42     ano = gdcm.Anonymizer()
43     ano.SetFile( r.GetFile() )
44     ano.RemovePrivateTags()
45     ano.RemoveGroupLength()
46     taglist = [
47         gdcm.Tag(0x0008,0x0008),
48         gdcm.Tag(0x0008,0x0022),
49         gdcm.Tag(0x0008,0x0032),
50         gdcm.Tag(0x0008,0x2111),
51         gdcm.Tag(0x0008,0x1150),
52         gdcm.Tag(0x0008,0x1155),
53         gdcm.Tag(0x0008,0x0100),
54         gdcm.Tag(0x0008,0x0102),
55         gdcm.Tag(0x0008,0x0104),
56         gdcm.Tag(0x0040,0xa170),
57         gdcm.Tag(0x0008,0x2112),
58         gdcm.Tag(0x0008,0x0100),
59         gdcm.Tag(0x0008,0x0102),
60         gdcm.Tag(0x0008,0x0104),
61         gdcm.Tag(0x0008,0x9215),
62         gdcm.Tag(0x0018,0x0010),
63         gdcm.Tag(0x0018,0x0022),
64         gdcm.Tag(0x0018,0x0050),
65         gdcm.Tag(0x0018,0x0060),
66         gdcm.Tag(0x0018,0x0088),
67         gdcm.Tag(0x0018,0x0090),
68         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.SetByteValue( uid, gdcM.VL(len(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.SetByteValue( uid, gdcm.VL(len(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()

```

29.27 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 &cсаel = csa.GetCSAElementByName( "Columns" )
        ;
        std::cout << cсаel << std::endl;
        //const gdcm::ByteValue *bv = cсаel.GetByteValue();
        gdcm::Element<gdcm::VR::IS, gdcm::VM::VM1> el;
        el.Set( cсаel.GetValue() );
        dims[0] = el.GetValue();
        std::cout << "Columns:" << el.GetValue() << std::endl;
    }

    if( csa.FindCSAElementByName( "Rows" ) )
    {
        const gdcm::CSAElement &cсаel2 = csa.GetCSAElementByName( "Rows" );
        std::cout << cсаel2 << std::endl;
        gdcm::Element<gdcm::VR::IS, gdcm::VM::VM1> el2;
        el2.Set( cсаel2.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 &cсаel3 = csa.GetCSAElementByName( "
        PixelSpacing" );
        if( !cсаel3.IsEmpty() )
        {
            std::cout << cсаel3 << std::endl;
            gdcm::Element<gdcm::VR::DS, gdcm::VM::VM2> el3;
            el3.Set( cсаel3.GetValue() );
            spacing[0] = el3.GetValue(0);
            spacing[1] = el3.GetValue(1);
            std::cout << "PixelSpacing:" << el3.GetValue() << "," << el3.
            GetValue(1) << std::endl;
            spacingfound = true;
        }
    }

    if( !spacingfound )
    {
        std::cerr << "Problem with PixelSpacing" << std::endl;
        //return 1;
    }

    if( !dims[0] || !dims[1] )
    {
        std::cerr << "Problem with dims" << std::endl;
        return 1;
    }

    gdcm::ImageWriter writer;

    gdcm::Image &image = writer.GetImage();
    image.SetNumberOfDimensions( 2 ); // good default
    image.SetDimension(0, dims[0] );
    image.SetDimension(1, dims[1] );
    image.SetSpacing(0, spacing[0] );
    image.SetSpacing(1, spacing[1] );
    gdcm::PixelFormat pixeltype = gdcm::PixelFormat::INT16; //
        bytepix = 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() );

```

```

gdcmm::PrivateTag csanonimaget(0x7fel,0x10,"SIEMENS CSA NON-IMAGE");
const gdcmm::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;
}

```

29.28 CStoreQtProgress.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 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 gdcmmData/MR_Spectroscopy_SIEMENS_OF.dcm
 *
 */

#include "gdcmmServiceClassUser.h"
#include "gdcmmSimpleSubjectWatcher.h"
#include "gdcmmProgressEvent.h"
#include "gdcmmDirectory.h"
#include "gdcmmPresentationContextGenerator.h"

#include <QApplication>
#include <QProgressDialog>
#include <QVBoxLayout>

namespace gdcmm {
/*
 * This class is a little more complicated than what this example demonstrate
 * This watcher is capable of handling nested progress. Since the Progress
 * grows from [0 to 1] on a per file basis and we only have one instance of a
 * watcher per association, we need some calculation to compute the global
 * (total) progress
 * In fact we simply divide the per-file progress by the number of files.
 *
 * This QtWatcher class will then update the progress bar according to the
 * progress.
 */
class MyQtWatcher : public SimpleSubjectWatcher
{
    size_t nfiles;
    double progress;
    size_t index;
    double refprogress;
    QWidget* win;
    QProgressDialog* qtprogress;
public:
    MyQtWatcher(Subject * s, const char *comment = "", QWidget *w = NULL, QProgressDialog* p = NULL, size_t n
        = 1):

```

```

SimpleSubjectWatcher(s,comment),nfiles(n),progress(0),index(0),refprogress(0),win(w),qtprogress(p){}
void ShowIteration()
{
    index++;
    assert( index <= nfiles );
    // update refprogress (we are moving to the next file)
    refprogress = progress;
}
void ShowProgress(Subject *, const Event &evt)
{
    // Retrieve the ProgressEvent:
    const ProgressEvent &pe = dynamic_cast<const ProgressEvent&>(evt);
    // compute global progress:
    progress = refprogress + (1. / (double)nfiles ) * pe.GetProgress();
    // Print Global and local progress to stdout:
    std::cout << "Global Progress: " << progress << " per file progress " << pe.GetProgress() << std::endl;
    //set progress value in the QtProgress bar
    int i = (int)(progress * 100 + 0.5); // round to next int
    qtprogress->setValue(i);
    win->show();
}
virtual void ShowDataSet(Subject *caller, const Event &evt)
{
    (void)caller;
    (void)evt;
}
};
} // end namespace gdcm

int main(int argc, char *argv[])
{
    if( argc < 4 )
    {
        std::cerr << argv[0] << " remote_server port filename" << std::endl;
        return 1;
    }
    QApplication a(argc, argv);

    std::ostringstream error_log;
    gdcm::Trace::SetErrorStream( error_log );

    const char *remote = argv[1];
    int portno = atoi(argv[2]);
    const char *filename = argv[3];

    QVBoxLayout* layout = new QVBoxLayout;
    QWidget* win = new QWidget;

    QProgressDialog* progress = new QProgressDialog("Sending data...", "Cancel", 0, 100);
    progress->setWindowModality(Qt::WindowModal);

    layout->addWidget( progress,Qt::AlignCenter);
    win->setLayout( layout);

    gdcm::SmartPointer<gdcm::ServiceClassUser> scup = new
        gdcm::ServiceClassUser;
    gdcm::ServiceClassUser &scu = *scup;
    //gdcm::SimpleSubjectWatcher w( &scu, "TestServiceClassUser" );
    // let's use a more complicated progress reported in this example
    gdcm::MyQtWatcher w( &scu, "QtWatcher", win, progress );

    scu.SetHostname( remote );
    scu.SetPort( (uint16_t)portno );
    scu.SetTimeout( 1000 );
    scu.SetCalledAETitle( "GDCM_STORE" );

    if( !scu.InitializeConnection() )
    {
        std::cerr << "Could not InitializeConnection" << std::endl;
        return 1;
    }

    gdcm::Directory::FileNamesType filenames;
    filenames.push_back( filename );

    // setup the PC(s) based on the filenames:
    gdcm::PresentationContextGenerator generator;
    if( !generator.GenerateFromFileNames(filenames) )
    {
        std::cerr << "Could not GenerateFromFileNames" << std::endl;
        return 1;
    }

```

```

    }

    // Setup PresentationContext(s)
    scu.SetPresentationContexts( generator.
        GetPresentationContexts() );

    // Start ASSOCIATION
    if( !scu.StartAssociation() )
    {
        std::cerr << "Could not Start" << std::endl;
        return 1;
    }

    // Send C-STORE
    if( !scu.SendStore( filename ) )
    {
        std::cerr << "Could not Store" << std::endl;
        std::cerr << "Error log is:" << std::endl;
        std::cerr << error_log.str() << std::endl;
        return 1;
    }

    // Stop ASSOCIATION
    if( !scu.StopAssociation() )
    {
        std::cerr << "Could not Stop" << std::endl;
        return 1;
    }

    win->show();

    return a.exec();
}

```

29.29 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;
        }

        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;
}
}

```

29.30 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 );
        }
    }
}

```

29.31 DecompressImage.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #   This software is distributed WITHOUT ANY WARRANTY; without even
10 #   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #   PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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  image = gdcm.Image()
35  ir = r.GetImage()
36
37  image.SetNumberOfDimensions( ir.GetNumberOfDimensions() );
38  dims = ir.GetDimensions();
39  print ir.GetDimension(0);
40  print ir.GetDimension(1);
41  print "Dims:", dims
42
43  # Just for fun:
44  dircos = ir.GetDirectionCosines()
45  t = gdcm.Orientation.GetType(dircos)
46  l = gdcm.Orientation.GetLabel(t)
47  print "Orientation label:", l
48
49  image.SetDimension(0, ir.GetDimension(0) );
50  image.SetDimension(1, ir.GetDimension(1) );
51
52  pixeltype = ir.GetPixelFormat();
53  image.SetPixelFormat( pixeltype );
54
55  pi = ir.GetPhotometricInterpretation();
56  image.SetPhotometricInterpretation( pi );
57
58  pixeldata = gdcm.DataElement( gdcm.Tag(0x7fe0,0x0010) )
59  str1 = ir.GetBuffer()
60  #print ir.GetBufferLength()
61  pixeldata.SetByteValue( str1, gdcm.VL( len(str1) ) )
62  image.SetDataElement( pixeldata )
63
64  w = gdcm.ImageWriter()
65  w.SetFileName( file2 )
66  w.SetFile( r.GetFile() )
67  w.SetImage( image )
68  if not w.Write():
69      sys.exit(1)

```

29.32 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;
}
}

```

29.33 DecompressJPEGFile.cs

This is a C# example on how to use [gdcm::SequenceOfFragments](#)

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/DecompressJPEGFile.exe somejpegfile.jpg
 */
using System;
using gdcm;

public class DecompressJPEGFile
{
    public static int Main(string[] args)
    {
        string file1 = args[0];
        System.IO.FileStream infile =
            new System.IO.FileStream(file1, System.IO.FileMode.Open, System.IO.FileAccess.Read);
        uint fsize = gdcm.PosixEmulation.FileSize(file1);

        byte[] jstream = new byte[fsize];
        infile.Read(jstream, 0, jstream.Length);

        Trace.DebugOn();
        Image image = new Image();
        image.SetNumberOfDimensions( 2 ); // important for now
        DataElement pixeldata = new DataElement( new gdcm.Tag(0x7fe0,0x0010) );

        // DO NOT set a ByteValue here, JPEG is a particular kind of encapsulated syntax
        // in which can one cannot use a simple byte array for storage. Instead, see
        // gdcm.SequenceOfFragments
        //pixeldata.SetByteValue( jstream, new gdcm.VL( (uint)jstream.Length ) );

        // Create a new SequenceOfFragments C++ object, store it as a SmartPointer :
        SmartPtrFrag sq = SequenceOfFragments.New();
        Fragment frag = new Fragment();
        frag.SetByteValue( jstream, new gdcm.VL( (uint)jstream.Length ) );
        // Single file => single fragment
        sq.AddFragment( frag );
        // Pass by reference:
        pixeldata.SetValue( sq.__ref__() );

        // insert:
        image.SetDataElement( pixeldata );

        // JPEG use YBR to achieve better compression ratio by default (not RGB)
    }
}

```

```

// FIXME hardcoded:
PhotometricInterpretation pi = new PhotometricInterpretation( PhotometricInterpretation.PIType.YBR_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;
}
}

```

29.34 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 );
}
}
}

```

29.35 DiffFile.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input1.dcm input2.dcm" << std::endl;
        return 1;
    }
    const char *filename1 = argv[1];
    const char *filename2 = argv[2];

    gdcm::Reader reader1;
    reader1.SetFileName( filename1 );
    if( !reader1.Read() )
    {
        return 1;
    }

    gdcm::Reader reader2;
    reader2.SetFileName( filename2 );
    if( !reader2.Read() )
    {
        return 1;
    }

    const gdcm::File &file1 = reader1.GetFile();
    const gdcm::File &file2 = reader2.GetFile();

    const gdcm::DataSet &ds1 = file1.GetDataSet();
    const gdcm::DataSet &ds2 = file2.GetDataSet();

    gdcm::DataSet::ConstIterator it1 = ds1.Begin();
    gdcm::DataSet::ConstIterator it2 = ds2.Begin();

```

```

const gdcm::DataElement &de1 = *it1;
const gdcm::DataElement &de2 = *it2;
if( de1 == de2 )
{
}
while( it1 != ds1.End() && it2 != ds2.End() && *it1 == *it2 )
{
  ++it1;
  ++it2;
}

if( it1 != ds1.End() || it2 != ds2.End() )
{
  std::cerr << "Problem with:" << std::endl;
  if( it1 != ds1.End() )
  {
    std::cerr << "ds1: " << *it1 << std::endl;
  }
  if( it2 != ds2.End() )
  {
    std::cerr << "ds2: " << *it2 << std::endl;
  }
  return 1;
}

return 0;
}

```

29.36 DiscriminateVolume.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmScanner.h"
#include "gdcmTesting.h"
#include "gdcmIPPSorter.h"
#include "gdcmDirectionCosines.h"
#include <cmath>

/*
 * The following example is a basic sorted which should work in generic cases.
 * It sort files based on:
 * Study Instance UID
 * Series Instance UID
 * Frame of Reference UID
 * Image Orientation (Patient)
 * Image Position (Patient) (Sorting based on IPP + IOP)
 */

namespace gdcm {
  const Tag t1(0x0020,0x000d); // Study Instance UID
  const Tag t2(0x0020,0x000e); // Series Instance UID
  const Tag t3(0x0020,0x0052); // Frame of Reference UID
  const Tag t4(0x0020,0x0037); // Image Orientation (Patient)

  class DiscriminateVolume
  {
  private:
    std::vector< Directory::FilenameType > SortedFiles;
    std::vector< Directory::FilenameType > UnsortedFiles;

    Directory::FilenameType GetAllFileNamesFromTagToValue(
      Scanner const & s, Directory::FilenameType const &filesubset, Tag const &t,
      const char *valueref)
    {
      Directory::FilenameType 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 intertwinet. 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.GetFileNames(), 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::FilenameType >::const_iterator it = UnsortedFiles.begin();
    it != UnsortedFiles.end(); ++it )
{
    os << "Group: " << std::endl;
    for(
        Directory::FilenameType::const_iterator file = it->begin();
        file != it->end(); ++file)
    {
        os << *file << std::endl;
    }
}

std::vector< Directory::FilenameType > const & GetSortedFiles() const { return SortedFiles; }
std::vector< Directory::FilenameType > const & GetUnsortedFiles() const { return UnsortedFiles; }

void ProcessIntoVolume( Scanner const & s )
{
    gdcm::Scanner::ValueType vt1 = s.GetValues( gdcm::t1 );
    for(
        gdcm::Scanner::ValueType::const_iterator it = vt1.begin()
        ; it != vt1.end(); ++it )
    {
        ProcessAStudy( s, it->c_str() );
    }
}

};

} // namespace gdcm

int main(int argc, char *argv[])
{
    std::string dirl;
    if( argc < 2 )
    {
        const char *extradataroot = NULL;
#ifdef GDCM_BUILD_TESTING
        extradataroot = gdcm::Testing::GetDataExtraRoot();
#endif
        if( !extradataroot )
        {
            return 1;
        }
        dirl = extradataroot;
        dirl += "/gdcmSampleData/ForSeriesTesting/VariousIncidences/ST1";
    }
    else
    {
        dirl = argv[1];
    }

    gdcm::Directory d;
    d.Load( dirl.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;
}

```

29.37 DumbAnonymizer.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #   This software is distributed WITHOUT ANY WARRANTY; without even
10 #   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #   PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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","GenerateMSOPId"),
40     (0x0008,0x1155):("Method","GenerateMSOPId"),
41     (0x0008,0x0018):("Method","GenerateMSOPId"),
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 GenerateMSOPId(self):
79     def GenerateMSOPId(self):

```

```

79     generator = gdcmm.UIDGenerator()
80     return generator.Generate()
81 def GetSiteId(self):
82     return "MySiteId"
83 def GetSiteName(self):
84     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     gdcmm.FileMetaInformation.SetSourceApplicationEntityTitle
93     ( "DumbAnonymizer" )
94     gdcmm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT )
95
96     r = gdcmm.Reader()
97     filename = sys.argv[1]
98     r.SetFileName( filename )
99     if not r.Read(): sys.exit(1)
100
101     obj = MyAnon()
102
103     w = gdcmm.Writer()
104     ano = gdcmm.Anonymizer()
105     ano.SetFile( r.GetFile() )
106     ano.RemoveGroupLength()
107     for tag,rule in tag_rules.items():
108         if rule[0] == 'Value':
109             print tag,rule
110             ano.Replace( gdcmm.Tag( tag[0], tag[1] ), rule[1] )
111         elif rule[0] == 'Method':
112             print tag,rule
113             # result = locals()[rule[1]]()
114             methodname = rule[1]
115             if hasattr(obj, methodname):
116                 _member = getattr(obj, methodname)
117                 result = _member()
118                 ano.Replace( gdcmm.Tag( tag[0], tag[1] ), result )
119             else:
120                 print "Problem with: ", methodname
121
122     outfilename = sys.argv[2]
123     w.SetFileName( outfilename )
124     w.SetFile( ano.GetFile() )
125     if not w.Write(): sys.exit(1)

```

29.38 DumpADAC.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.

=====*/
/*
 * 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 "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 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, NULL }
};

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 << "\n";
    }
    else
    {
        (void)len;
        os << " " << buffer << "\n";
    }
}

bool DumpADAC( std::istream & is )
{
    std::ostream &os = std::cout;

    char magic[6 + 1];
    magic[6] = 0;
    is.read( magic, 6);
    // std::cout << magic << " ";
    assert( strcmp( magic, "adac01" ) == 0 );
    int c = is.get();
    assert( c == 0 ); (void)c;
    c = is.get();
    assert( c == 'X' );

    uint16_t v;
    v = readint16(is);
    // std::cout << v << std::endl;
    assert( v == 512 ); (void)v; // ??

    int nel = 87;
    for( int i = 0; i <= nel; ++i )
    {
        readelement( is );
    }

    char buffer[512];
    for( int i = 0; i <= nel; ++i )
    {
        const el &e = Vel[i];

```

```

int diff;
if( i == nel )
{
    diff = 2048 - e.v3;
    if( diff > 512 ) diff = 512;
}
else
{
    const el &enext = Vel[i+1];
    diff = enext.v3 - e.v3;
}
is.seekg( e.v3, std::ios::beg );
//std::cout << "(" << std::hex << std::setw( 2 ) << std::setfill( '0' ) << e.v1 << ") " << std::hex <<
    std::setw( 3 ) << std::setfill( '0' ) << e.v2 << " ";
printname( diff, 0, e.v1 );
int mult = 1;
if( e.v2 == 0 )
{
    is.read( buffer, diff);
    buffer[ diff ] = 0;
    printascii( e.v1, buffer, diff);
}
else if( e.v2 == 0x100 )
{
    mult = diff / 2;
    assert( diff == 2 * mult );
    for ( int ii = 0; ii < mult; ++ii )
    {
        if ( ii ) os << "\\ ";
        uint16_t val = readint16(is);
        os << " " << std::dec << val << " ";
    }
}
else if( e.v2 == 0x200 )
{
    assert( diff == 4 );
    uint32_t val = readint32(is);
    os << " " << std::dec << val << " ";
}
else if( e.v2 == 0x300 )
{
    assert( diff == 4 );
    float val = readfloat32(is);
    os << " " << std::dec << val << " ";
}
else
{
    assert( 0 );
}
os << std::endl;
}
return true;
}

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }
    const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

    // (0019,1061) UN (OB) 61\64\61\63\30 # 2048,1 Ver200 ADAC Pegasys Headers
    const gdcm::PrivateTag tver200adacpegasysheaders(0x0019,0x61,"ADAC_IMG");
    if( !ds.FindDataElement( tver200adacpegasysheaders ) ) return 1;
    const gdcm::DataElement& ver200adacpegasysheaders = ds.
        GetDataElement( tver200adacpegasysheaders );
    if ( ver200adacpegasysheaders.IsEmpty() ) return 1;
    const gdcm::ByteValue *bv = ver200adacpegasysheaders.
        GetByteValue();

    // (0019,1021) US 1 # 2,1 Ver200 Number of ADAC Headers
    // TODO

    // (0019,1041) IS [2048\221184 ] # 12,1-n Ver200 ADAC Header/Image Size
    if( bv->GetLength() != 2048 ) return 1;

```



```

gdcmm::Element<gdcmm::VR::IS,gdcmm::VM::VM2> el;
const gdcmm::PrivateTag tver200adacheaderimagesize(0x0019,0x41,"ADAC_IMG");
if( !ds.FindDataElement( tver200adacheaderimagesize ) ) return 1;
const gdcmm::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;
}

```

29.39 DumpExamCard.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.

=====*/
/*

Try to extract contents of Philips RAW storage class:

(0002,0002) UI [1.2.840.10008.5.1.4.1.1.66] # 26,1 Media Storage SOP Class UID
(0002,0003) UI [1.3.46.670589.11.17240.5.23.4.1.3012.2010032409482568018] # 56,1 Media Storage SOP
Instance UID
(0002,0010) UI [1.2.840.10008.1.2.1] # 20,1 Transfer Syntax UID
(0002,0012) UI [1.3.46.670589.11.0.0.51.4.4.1] # 30,1 Implementation Class UID
(0002,0013) SH [MR DICOM 4.1] # 12,1 Implementation Version Name

* Everything done in this code is for the sole purpose of writing interoperable
* software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
* If you believe anything in this code violates any law or any of your rights,
* please contact us (gdcmm-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 "gdcmmReader.h"
#include "gdcmmDataSet.h"
#include "gdcmmPrivateTag.h"
#include "gdcmmBase64.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
    "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_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,
    param_string,
    param_3, // ??
    param_enum,
} param_type;

static const char *gettypenamefromtype( int i )
{
    const char *ret = NULL;
    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));
        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];
    int8_t boolean;
    int32_t type;
    uint32_t dim;
    uint32_t v4;
    /*int32_t*/ std::streamoff offset;
    param_type gettype() const { return (param_type)type; }
    uint32_t getdim() const { return dim; }
    void read( std::istream & is )
    {
        is.read( name, 32 + 1 );
        //assert( name[32] == 0 ); // fails sometimes...
        // 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 ); // sigh
        is.read( (char*)&boolean, 1 );
        assert( boolean == 0 || boolean == 1 ); // some kind of bool...
        is.read( (char*)&type, sizeof( type ) );
        assert( gettypenamefromtype( type ) );
        is.read( (char*)&dim, sizeof( dim ) );
        is.read( (char*)&v4, sizeof( v4 ) );
        //assert( v4 == 0 ); // always 0 ? sometimes not...
        const std::streamoff cur = is.tellg();
        is.read( (char*)&offset, sizeof( offset ) );
        offset += cur;
    }

    void print( std::ostream & os ) const
    {
        os << name << ", ";
        os << (int)boolean << ", ";
        os << type << ", ";
        os << dim << ", ";
        os << v4 << ", ";
        os << offset << std::endl;
    }

    void printvalue( std::ostream & os, std::istream & is ) const
    {
        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:
            {
                for( uint32_t idx = 0; idx < dim; ++idx )
                {
                    if( idx ) os << ", ";
                    int32_t v;
                    is.read( (char*)&v, sizeof(v) );
                    os << v;
                }
            }
            break;
        }
    }
};

```

```

    case param_string:
    {
        std::string v;
        v.resize( dim );
        is.read( &v[0], dim );
        os << v;
    }
    break;
case param_enum:
{
    for( uint32_t idx = 0; idx < dim; ++idx )
    {
        if( idx ) os << ", ";
        int32_t v;
        is.read( (char*)&v, sizeof(v) );
        os << v;
    }
}
break;
}

}

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( gdcmm::DataSet & ds )
{
    /*
    TODO:
    Looks like the real length of the blob is stored here:
(2005,1132) SQ # u/1,1 ?
    (fffe,e000) na (Item with undefined length)
    (2005,0011) LO [Philips MR Imaging DD 002 ] # 26,1 Private Creator
    (2005,1143) SL 3103 # 4,1 ?

Wotsit ?
(2005,1132) SQ # u/1,1 ?
    (fffe,e000) na (Item with undefined length)
    (2005,0011) LO [Philips MR Imaging DD 002 ] # 26,1 Private Creator
    (2005,1147) CS [Y ] # 2,1 ?
    */
    bool ret = false;

    // (2005,1137) PN (LO) [PDF_CONTROL_GEN_PARS] # 20,1 ?
    const gdcmm::PrivateTag pt0(0x2005,0x37,"Philips MR Imaging DD 002");
    if( !ds.FindDataElement( pt0 ) ) return false;
    const gdcmm::DataElement &de0 = ds.GetDataElement( pt0 );
    if( de0.IsEmpty() ) return false;
    const gdcmm::ByteValue * bv0 = de0.GetByteValue();
    std::string s0( bv0->GetPointer() , bv0->GetLength() );

    // (2005,1139) LO [IEEE_PDF] # 8,1 ?
    const gdcmm::PrivateTag pt1(0x2005,0x39,"Philips MR Imaging DD 002");
    if( !ds.FindDataElement( pt1 ) ) return false;
    const gdcmm::DataElement &de1 = ds.GetDataElement( pt1 );

    const gdcmm::PrivateTag pt(0x2005,0x44,"Philips MR Imaging DD 002");
    if( !ds.FindDataElement( pt ) ) return false;
    const gdcmm::DataElement &de = ds.GetDataElement( pt );

```

```

if( de.IsEmpty() ) return false;
const gdcm::ByteValue * bv = de.GetByteValue();

if( s0 == "ExamCardBlob" )
{
    assert( del.IsEmpty() );

    std::string fn = gdcm::LOComp::Trim( s0.c_str() ); // remove trailing space
    fn += ".xml";
    std::ofstream out( fn.c_str() );

    // remove trailing \0
    size_t len = strlen( bv->GetPointer() );
    out.write( bv->GetPointer() , len );
    out.close();

    // Extract binary64 thingy (this is a ugly hack, better use an XML parser)
    std::string dup( bv->GetPointer(), len );
    std::string::size_type pos1 = dup.find( "<ExamCardBlob" );
    std::string::size_type pos2 = dup.find( "</ExamCardBlob" );

    std::string b64( bv->GetPointer() + pos1 + 14, pos2 - (pos1 + 14) );

    // ugly hack to remove \r\n from input base64:
    std::string::iterator r_pos = std::remove(b64.begin(), b64.end(), '\r');
    b64.erase(r_pos, b64.end());
    std::string::iterator n_pos = std::remove(b64.begin(), b64.end(), '\n');
    b64.erase(n_pos, b64.end());
#ifdef 0
    std::ofstream out2( "debug" );
    out2.write( b64.c_str(), b64.size() );
    out2.close();
#endif

    const size_t dlen = gdcm::Base64::GetDecodeLength( b64.c_str(), b64.size() );

    std::string decoded;
    decoded.resize( dlen );
    gdcm::Base64::Decode( &decoded[0], decoded.size(), b64.c_str(), b64.size() );

    std::ofstream f64( "soap.xml" );
    f64.write( decoded.c_str(), decoded.size() );
    f64.close();

    ret = true;
}
else
{
    if( del.IsEmpty() ) return false;
    const gdcm::ByteValue * bvl = del.GetByteValue();
    std::string s1( bvl->GetPointer() , bvl->GetLength() );

    if( s1 == "IEEE_PDF" )
    {
        // std::cout << "Len= " << bv->GetLength() << std::endl;
#ifdef 0
        std::string fn = gdcm::LOComp::Trim( s.c_str() ); // remove trailing space
        std::ofstream out( fn.c_str() );
        out.write( bv->GetPointer(), bv->GetLength() );
        out.close();
#endif
    }
    else
    {
        std::istringstream is;
        std::string dup( bv->GetPointer(), bv->GetLength() );
        is.str( dup );

        header h;
        h.read( is );
#ifdef 0
        std::cout << s0.c_str() << std::endl;
        h.print( std::cout );
#endif
    }

    assert( is.tellg() == std::streampos(0x20) );
    is.seekg( 0x20 );

    std::vector< param > params;
    param p;
    for( uint32_t i = 0; i < h.getnparams(); ++i )
    {

```

```

        p.read( is );
        //p.print( std::cout );
        params.push_back( p );
    }

    std::string fn = gdc::LOComp::Trim( s0.c_str() ); // remove trailing space
    assert( !isvalidpdfstring( fn.c_str() ) );
    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:
                ;
        }
    }

#ifdef 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 " )
{
#ifdef 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();

#ifdef 0
    int array[ 128 ];
    memcpy( array, bv->GetPointer(), 512 );
    for( int i = 0; i < 14; ++i )
    {
        std::cout << array[i] << std::endl;
    }
#endif

    ret = true;
}
// else -> ret == false
assert( ret );

return ret;
}

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }
    const gdcm::DataSet& ds = reader.GetFile().GetDataSet();
    /*
(2005,1132) SQ                                     # u/1,1 ?
(fffe,e000) na (Item with undefined length)
(2005,0011) LO [Philips MR Imaging DD 002 ]         # 26,1 Private Creator
(2005,1137) PN (LO) [PDF_CONTROL_GEN_PARS]          # 20,1 ?
(2005,1138) PN (LO) (no value)                     # 0,1 ?
(2005,1139) PN (LO) [IEEE_PDF]                    # 8,1 ?
(2005,1140) PN (LO) (no value)                     # 0,1 ?
(2005,1141) PN (LO) (no value)                     # 0,1 ?
(2005,1143) SL 3103                                # 4,1 ?
(2005,1144) OW
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 ) ) return 1;
}

return 0;
}

```

29.40 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 VRTToType<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> e11;
    e11.SetFromDataElement( index2 );

    UL copy = (UL)e11.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> e12;
        e12.SetFromDataElement( value );
        std::cout << e12.GetValue() << std::endl;
    }
    else if( ds.FindDataElement( tvaluefloat1 ) )
    {
        const DataElement & value = ds.GetDataElement( tvaluefloat1 );
        gdcm::Element<VR::FL,VM::VM1> e12;
        e12.SetFromDataElement( value );
        std::cout << e12.GetValue() << std::endl;
    }
    else if( ds.FindDataElement( tvaluefloat ) )
    {
        const DataElement & value = ds.GetDataElement( tvaluefloat );
        gdcm::Element<VR::FD,VM::VM1> e12;
        e12.SetFromDataElement( value );
        std::cout << e12.GetValue() << std::endl;
    }
    else if( ds.FindDataElement( tvaluesl ) )
    {
        const DataElement & value = ds.GetDataElement( tvaluesl );
        gdcm::Element<VR::SL,VM::VM1> e12;
        e12.SetFromDataElement( value );
        std::cout << e12.GetValue() << std::endl;
    }
    else if( ds.FindDataElement( tvalueul ) )
    {
        const DataElement & value = ds.GetDataElement( tvalueul );
        gdcm::Element<VR::UL,VM::VM1_n> e12;
        e12.SetFromDataElement( value );
        assert( e12.GetLength() == 1 );
        std::cout << e12.GetValue() << std::endl;
    }
    else if( ds.FindDataElement( tvalueob ) )
    {
        const DataElement & value = ds.GetDataElement( tvalueob );
        // gdcm::Element<VR::SL,VM::VM1> e12;

```

```

//      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 1;
    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 1;
    }
    const gdcm::DataElement& seq_values10 = subds.GetDataElement( privtag0 );
    gdcm::SmartPointer<gdcm::SequenceOfItems> sqi_values10 =
        seq_values10.GetValueAssQ();

    size_t nil = sqi_values10->GetNumberOfItems();
    // assert( nil == 1 );
    for( size_t il = 1; il <= nil; ++il )
    {
        gdcm::Item &item_10 = sqi_values10->GetItem(il);
        gdcm::DataSet &ds10 = item_10.GetNestedDataSet();
        assert( ds10.Size() == 2 + 3 );
        // (7fe1,0010)
        // (7fe1,1012)
        // (7fe1,1018)
        // (7fe1,1020)
        // (7fe1,1083)

        PrintNameValueMapping3( privtag1, privtag2, ds10, sqi_dict, "  " );
        std::cout << std::endl;

        const gdcm::PrivateTag tseq_values20(0x7fe1,0x20,"GEMS_Ultrasound_MovieGroup_001");
        if( !ds10.FindDataElement( tseq_values20 ) )
        {
            assert( 0 );
            return 1;
        }
        const gdcm::DataElement& seq_values20 = ds10.GetDataElement(
            tseq_values20 );
        gdcm::SmartPointer<gdcm::SequenceOfItems> sqi_values20 =
            seq_values20.GetValueAssQ();
    }
}

```

```

size_t ni2 = sqi_values20->GetNumberOfItems();
//assert( ni == 1 );
for( size_t i2 = 1; i2 <= ni2; ++i2 )
{
    gdcm::Item &item_20 = sqi_values20->GetItem(i2);
    gdcm::DataSet &ds20 = item_20.GetNestedDataSet();
    size_t count = ds20.Size(); (void)count;
    assert( ds20.Size() == 2 + 3 || ds20.Size() == 2 + 2 );
    // (7fe1,0010)
    // (7fe1,1024)
    // (7fe1,1026)
    // (7fe1,1036)
    // (7fe1,103a)
    // (7fe1,1083) (*)

    const gdcm::PrivateTag tseq_values20name(0x7fe1,0x24,"GEMS_Ultrasound_MovieGroup_001"
);
    const gdcm::PrivateTag tseq_values26(0x7fe1,0x26,"GEMS_Ultrasound_MovieGroup_001");
    PrintNameValueMapping3( tseq_values20name, tseq_values26, ds20, sqi_dict, "  ");
    std::cout << std::endl;

    print36(ds20, sqi_dict, "  ");
    print83(ds20, sqi_dict, "  ");
}

print83(ds10, sqi_dict, "  ");
}
return true;
}

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    using namespace gdcm;
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() ) return 1;

    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();
    const PrivateTag tseq(0x7fe1,0x1,"GEMS_Ultrasound_MovieGroup_001");

    if( !ds.FindDataElement( tseq ) ) return 1;
    const DataElement& seq = ds.GetDataElement( tseq );

    SmartPointer<SequenceOfItems> sqi = seq.
        GetValueAsSQ();
    assert( sqi->GetNumberOfItems() == 1 );

    Item &item = sqi->GetItem(1);
    DataSet &subds = item.GetNestedDataSet();

    const PrivateTag tseq_dict(0x7fe1,0x70,"GEMS_Ultrasound_MovieGroup_001");
    if( !subds.FindDataElement( tseq_dict ) ) return 1;
    const DataElement& seq_dict = subds.GetDataElement( tseq_dict );
    SmartPointer<SequenceOfItems> sqi_dict = seq_dict.
        GetValueAsSQ();

    const PrivateTag tseq_values8(0x7fe1,0x8,"GEMS_Ultrasound_MovieGroup_001");
    if( !subds.FindDataElement( tseq_values8 ) ) return 1;
    const DataElement& seq_values8 = subds.GetDataElement( tseq_values8 );
    SmartPointer<SequenceOfItems> sqi_values8 = seq_values8.
        GetValueAsSQ();

    const PrivateTag tseq_values8name(0x7fe1,0x2,"GEMS_Ultrasound_MovieGroup_001");
    if( !subds.FindDataElement( tseq_values8name ) ) return 1;
    const DataElement& values8name = subds.GetDataElement( tseq_values8name );
    {
        Element<VR::LO,VM::VM1> el;
        el.SetFromDataElement( values8name );
        std::cout << el.GetValue() << std::endl;
    }
    size_t count = subds.Size(); (void)count;
    assert( subds.Size() == 3 + 2 + 1 || subds.Size() == 3 + 2 + 2 );

    // (7fe1,0010) # 30,1 Private Creator
    // (7fe1,1002) # 8,1 US MovieGroup Value 0008 Name
    // (7fe1,1003) # 4,1 ?
    // (7fe1,1008) # 8140,1 US MovieGroup Value 0008 Sequence
    // (7fe1,1010) # 1372196,1 ?

```

```
// (7fe1,1070) # 33684,1 US MovieGroup Dict
// (7fe1,1073) (*)
PrintNameValueMapping( sqi_values8, sqi_dict, " ");

const PrivateTag tseq_values10(0x7fe1,0x10,"GEMS_Ultrasound_MovieGroup_001");
const PrivateTag tseq_values10name(0x7fe1,0x12,"GEMS_Ultrasound_MovieGroup_001");
const PrivateTag tseq_values18(0x7fe1,0x18,"GEMS_Ultrasound_MovieGroup_001");
PrintNameValueMapping4( tseq_values10, subds, tseq_values10name, tseq_values18, sqi_dict, " ");

print73( subds, sqi_dict, " ");

#if 0
gdcm::DataSet::ConstIterator it = subds.Begin();
for( ; it != subds.End(); ++it )
{
    const gdcm::DataElement &de = *it;
    std::cout << de.GetTag() << std::endl;
}
#endif

return 0;
}
```

29.41 DumpImageHeaderInfo.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.

=====*/
/*
 * Dump TOSHIBA MDW HEADER / Image Header Info
 */
#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 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 );

    uint32_t l;
    is.read( (char*)&l, sizeof(l) );
}
```

Generated on Mon Dec 21 2015 23:27:54 for GDCM by Doxygen

```

}

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::stringstream 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;
}

```

29.42 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
* $ gdcmmimg --sop-class-uid 1.2.840.10008.5.1.4.1.1.3.1 --size 608,427,88 raw_us_img.raw raw_us_img.dcm
*/

// header:
struct hframe
{
    uint32_t val0; // 800 increment ?
    uint16_t val1[2];
    uint16_t val2[2];
    uint32_t imgsize;

    bool operator==(const hframe &h) const
    {
        return val0 == h.val0 &&
            val1[0] == h.val1[0] &&
            val1[1] == h.val1[1] &&
            val2[0] == h.val2[0] &&
            val2[1] == h.val2[1] &&
            imgsize == h.imgsize;
    }
};

static bool ProcessDeflate( const char *outfilename, const int nslices, const
    int buf_size, const char *buf, const std::streampos len,
    const char *crdbuf, const size_t crclen )
{
    std::vector< hframe > crchheaders;
    crchheaders.reserve( nslices );
    {
        std::istream is;
        is.str( std::string( crdbuf, crclen ) );
        hframe header;
        for( int r = 0; r < nslices; ++r )
        {
            is.read( (char*)&header, sizeof( header ) );
        }
    }
    #if 0
        std::cout << header.val0
            << " " << header.val1[0]
            << " " << header.val1[1]
            << " " << header.val2[0]
            << " " << header.val2[1]
            << " " << header.imgsize << std::endl;
    #endif
    crchheaders.push_back( header );
    }

    std::istream is;
    is.str( std::string( buf, 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 << crchheaders[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 == crchheaders[r] );
    assert( header.val1[0] == 2000 );
    assert( header.val1[1] == 3 );
    assert( header.val2[0] == 1 );
    assert( header.val2[1] == 1280 );

    uLongf destLen = buf_size; // >= 608,427
    Bytef *dest = (Bytef*)&outbuf[0];
    assert( is.tellg() == offsets[r] + 16 );
    const Bytef *source = (Bytef*)buf + offsets[r] + 16;
    uLong sourceLen;
    if( r + 1 == nframes )
        sourceLen = totalsize - offsets[r] - 16;
    else
        sourceLen = offsets[r+1] - 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 > crchheaders;
    crchheaders.reserve( nslices );
    {
        std::istringstream is;
        is.str( std::string( crdbuf, crclen ) );
        hframe header;
        for( int r = 0; r < nslices; ++r )
        {
            is.read( (char*)&header, sizeof( header ) );
        }
#ifdef 0
        std::cout << header.val0
            << " " << header.val1[0]
            << " " << header.val1[1]
            << " " << header.val2[0]
            << " " << header.val2[1]
            << " " << header.imgsize << std::endl;
#endif
        crchheaders.push_back( header );
    }
}

std::istringstream is;
is.str( std::string( buf, len ) );

std::streampos totalsize;
is.read( (char*)&totalsize, sizeof( totalsize ) );
assert( totalsize == len );

```

```

uint32_t nframes;
is.read( (char*)&nframes, sizeof( nframes ));
assert( nframes == (uint32_t)nslices );

std::vector< uint32_t > offsets;
offsets.reserve( nframes );
for( uint32_t frame = 0; frame < nframes ; ++frame )
{
    uint32_t offset;
    is.read( (char*)&offset, sizeof( offset ));
    offsets.push_back( offset );
    //std::cout << offset << std::endl;
}

std::vector<char> outbuf;
// No idea how to present the data, I'll just append everything, and present it as 2D
std::stringstream ss;
ss << outfilename;
ss << '_';
ss << 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 == crcheaders[r] );

    is.read( buffer, buf_size - 16 );
    os.write( buffer, header.imgsize );
}
assert( is.tellg() == totalsize );
os.close();

return true;
}

#ifdef NDEBUG
static const char * const UDM_USD_DATATYPE_STRINGS[] = {
    "UDM_USD_DATATYPE_DIN_2D_ECHO",
    "UDM_USD_DATATYPE_DIN_2D_ECHO_CONTRAST",
    "UDM_USD_DATATYPE_DIN_DOPPLER_CW",
    "UDM_USD_DATATYPE_DIN_DOPPLER_PW",
    "UDM_USD_DATATYPE_DIN_DOPPLER_PW_TDI",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_FLOW",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_PMI",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_CPA",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_TDI",
    "UDM_USD_DATATYPE_DIN_MMODE_ECHO",
    "UDM_USD_DATATYPE_DIN_MMODE_COLOR",
    "UDM_USD_DATATYPE_DIN_MMODE_COLOR_TDI",
    "UDM_USD_DATATYPE_DIN_PARAM_BLOCK",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_VELOCITY",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_POWER",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_VARIANCE",
    "UDM_USD_DATATYPE_DIN_DOPPLER_AUDIO",
    "UDM_USD_DATATYPE_DIN_DOPPLER_HIGHQ",
    "UDM_USD_DATATYPE_DIN_PHYSIO",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_STRAIN",
    "UDM_USD_DATATYPE_DIN_COMPOSITE_RGB",
    "UDM_USD_DATATYPE_DIN_XFOV_REALTIME_GRAPHICS",
    "UDM_USD_DATATYPE_DIN_XFOV_MOSAIC",
    "UDM_USD_DATATYPE_DIN_COMPOSITE_R",
    "UDM_USD_DATATYPE_DIN_COMPOSITE_G",
    "UDM_USD_DATATYPE_DIN_COMPOSITE_B",
    "UDM_USD_DATATYPE_DIN_MMODE_COLOR_VELOCITY",
    "UDM_USD_DATATYPE_DIN_MMODE_COLOR_POWER",

```

```

    "UDM_USD_DATATYPE_DIN_MMODE_COLOR_VARIANCE",
    "UDM_USD_DATATYPE_DIN_2D_ELASTO",
};

static inline bool is_valid( const char * datatype_str )
{
    static const int n = sizeof( UDM_USD_DATATYPE_STRINGS ) / sizeof( *UDM_USD_DATATYPE_STRINGS );
    bool found = false;
    if( datatype_str )
    {
        for( int i = 0; !found && i < n; ++i )
        {
            found = strcmp( datatype_str, UDM_USD_DATATYPE_STRINGS[i] ) == 0;
        }
    }
    return found;
}
#endif

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    using namespace gdcm;
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() ) return 1;

    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds1 = file.GetDataSet();

    const PrivateTag tseq1(0x200d,0x3cf8,"Philips US Imaging DD 045");
    if( !ds1.FindDataElement( tseq1 ) ) return 1;
    const DataElement& seq1 = ds1.GetDataElement( tseq1 );

    SmartPointer<SequenceOfItems> sq1 = seq1.
        GetValueAsSQ();
    assert( sq1->GetNumberOfItems() >= 1 );

    const size_t nitems = sq1->GetNumberOfItems();
    for( size_t item = 1; item < nitems; ++item )
    {
        Item &item1 = sq1->GetItem(item);
        DataSet &ds2 = item1.GetNestedDataSet();

        // (200d,300d) LO 28 UDM_USD_DATATYPE_DIN_2D_ECHO
        const PrivateTag tdatatype(0x200d,0x300d,"Philips US Imaging DD 033");
        if( !ds2.FindDataElement( tdatatype ) ) return 1;
        const DataElement& datatype = ds2.GetDataElement( tdatatype );
        const ByteValue *bvdatatype = datatype.GetByteValue();
        if( !bvdatatype ) return 1;

        const PrivateTag tseq2(0x200d,0x3cf1,"Philips US Imaging DD 045");
        if( !ds2.FindDataElement( tseq2 ) ) return 1;
        const DataElement& seq2 = ds2.GetDataElement( tseq2 );

        SmartPointer<SequenceOfItems> sq2 = seq2.
            GetValueAsSQ();
        assert( sq2->GetNumberOfItems() >= 1 );

        // FIXME: what if not in first Item ?
        assert( sq2->GetNumberOfItems() == 1 );
        Item &item2 = sq2->GetItem(1);
        DataSet &ds3 = item2.GetNestedDataSet();

        const PrivateTag tzlib(0x200d,0x3cfa,"Philips US Imaging DD 045");
        if( !ds3.FindDataElement( tzlib ) ) return 1;
        const DataElement& zlib = ds3.GetDataElement( tzlib );

        const ByteValue *bv = zlib.GetByteValue();
        if( !bv ) return 1;
        if( bv->GetLength() != 4 ) return 1;

        // (200d,3010) IS 2 88
        const PrivateTag tnslices(0x200d,0x3010,"Philips US Imaging DD 033");
        if( !ds3.FindDataElement( tnslices ) ) return 1;
        const DataElement& nslices = ds3.GetDataElement( tnslices );
        Element<VR::IS,VM::VM1> elnslices;
        elnslices.SetFromDataElement( nslices );
        const int nslicesref = elnslices.GetValue();
        assert( nslicesref >= 0 );
    }
}

```

```

// (200d,3011) IS 6 259648
const PrivateTag tzalloc(0x200d,0x3011,"Philips US Imaging DD 033");
if( !ds3.FindDataElement( tzalloc ) ) return 1;
const DataElement& zalloc = ds3.GetDataElement( tzalloc );
Element<VR::IS,VM::VM1> elzalloc;
elzalloc.SetFromDataElement( zalloc );
const int zallocref = elzalloc.GetValue();
assert( zallocref >= 0 );
// (200d,3021) IS 2 0
const PrivateTag tzero(0x200d,0x3021,"Philips US Imaging DD 033");
if( !ds3.FindDataElement( tzero ) ) return 1;
const DataElement& zero = ds3.GetDataElement( tzero );
Element<VR::IS,VM::VM1> elzero;
elzero.SetFromDataElement( zero );
const int zerocref = elzero.GetValue();
assert( zerocref == 0 ); (void)zerocref;

// (200d,3cf3) OB
const PrivateTag tdeflate(0x200d,0x3cf3,"Philips US Imaging DD 045");
if( !ds3.FindDataElement( tdeflate ) ) return 1;
const DataElement& deflate = ds3.GetDataElement( tdeflate );
const ByteValue *bv2 = deflate.GetByteValue();

// (200d,3cfb) OB
const PrivateTag tcrc(0x200d,0x3cfb,"Philips US Imaging DD 045");
if( !ds3.FindDataElement( tcrc ) ) return 1;
const DataElement& crc = ds3.GetDataElement( tcrc );
const ByteValue *bv3 = crc.GetByteValue();

std::string outfile = std::string( bvdatatype->GetPointer(), bvdatatype->
    GetLength() );
outfile = LOComp::Trim( outfile.c_str() );
const char *outfilename = outfile.c_str();
assert( is_valid(outfilename) );
if( bv2 )
{
    assert( bv3 );
    assert( zallocref > 0 );
    assert( nslicesref > 0 );
    std::cout << ds2 << std::endl;

    if( strcmp(bv->GetPointer(), "ZLib", 4) == 0 )
    {
        if( !ProcessDeflate( outfile, nslicesref, zallocref, bv2->GetPointer(),
            std::streampos(bv2->GetLength()), bv3->GetPointer(), bv3->
            GetLength() ) )
        {
            return 1;
        }
    }
    else if( strcmp(bv->GetPointer(), "None", 4) == 0 )
    {
        if( !ProcessNone( outfile, nslicesref, zallocref, bv2->GetPointer(),
            std::streampos(bv2->GetLength()), bv3->GetPointer(), bv3->
            GetLength() ) )
        {
            return 1;
        }
    }
    else
    {
        std::string str( bv->GetPointer(), bv->GetLength() );
        std::cerr << "Unhandled: " << str << std::endl;
        return 1;
    }
}
}

return 0;
}

```

29.43 DumpToshibaDTI.cxx

```

/*=====

```

```

Program: GDCM (Grassroots DICOM). A DICOM library

```

Copyright (c) 2006-2011 Mathieu Malaterre
 All rights reserved.
 See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even
 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
 PURPOSE. See the above copyright notice for more information.

```

=====*/
/*
 * https://groups.google.com/d/msg/comp.protocols.dicom/7IaIkT0ZG5U/k7LPu81VvAMJ
 */
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmPrinter.h"
#include "gdcmDictPrinter.h"

#include <iostream>
#include <fstream>
#include <vector>

#include <assert.h>

bool DumpToshibaDTI( const char * input, size_t len )
{
    if( len % 2 ) return false;

    std::vector<char> copy( input, input + len );
    std::reverse( copy.begin(), copy.end() );

    std::istringstream is;
    std::string dup( &copy[0], copy.size() );
    is.str( dup );

    gdcm::Reader reader;
    reader.SetStream( is );
    if( !reader.Read() )
        return false;

    //std::cout << reader.GetFile().GetDataSet() << std::endl;
    //gdcm::DictPrinter p;
    gdcm::Printer p;
    p.SetFile( reader.GetFile() );
    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/l,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;
}

```

29.44 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(0);

    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.GetFileNames() );
    if( !b0 ) return 1;
    time_t time_scanner = time(0);

    std::cout << "Finished loading data from : " << nfiles << " files" << std::endl;

    // MappingType const &mappings = s.GetMappings();

```

```

sqlite3* db;
sqlite3_open("./dicom.db", &db);

if(db == 0)
{
    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, 0, 0, &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,
    0 // 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(0);

```



```

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;
}

```

29.45 DuplicatePCDE.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
/*
Usage:
DuplicatePCDE gdcmData/D_CLUNIE_CT1_J2KI.dcm out.dcm

aka:
medical.nema.org/medical/dicom/DataSets/WG04/IMAGES/J2KI/CT1_J2KI

See:
gdcmConformanceTests/CT1_J2KI_DuplicatePCDE.dcm

Original thread can be found at:

http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/82f28c4db28963af

Question:
1.
There is no restriction for a specific Private Creator Data Element
(PCDE) to be unique within the same group, right ?
Decoders of Private Data would have to handle the case where a PCDE
would be repeated and should NOT stop on the first instance of a
particular PCDE, right ?

Eg. when searching for the tag associated with
(0x0029,0x0010,"SIEMENS CSA HEADER") in the following (pseudo)
dataset:

(0029,0010) LO [SIEMENS CSA HEADER] # 18, 1
PrivateCreator
(0029,0011) LO [SIEMENS MEDCOM HEADER] # 22, 1
PrivateCreator
(0029,0012) LO [SIEMENS MEDCOM HEADER2] # 22, 1
PrivateCreator
(0029,0013) LO [SIEMENS CSA HEADER] # 18, 1
PrivateCreator
(0029,1008) CS [IMAGE NUM 4] # 12, 1
CSAImageHeaderType
(0029,1009) LO [20050723] # 8, 1
CSAImageHeaderVersion
(0029,1010) OB 53\56\31\30\04\03\02\01\38\00\00\00\4d
\00\00\00\45\63\68\6f\4c\69... # 6788, 1 CSAImageHeaderInfo
(0029,1018) CS [MR] # 2, 1
CSASeriesHeaderType
(0029,1019) LO [20050723] # 8, 1
CSASeriesHeaderVersion
(0029,1020) OB 53\56\31\30\04\03\02\01\2c\00\00\00\4d
\00\00\00\55\73\65\64\50\61... # 51520, 1 CSASeriesHeaderInfo
(0029,1131) LO [4.0.163088300] # 14, 1
PMTFInformation1

```

```

(0029,1132) UL 32768 # 4, 1
PMTFInformation2
(0029,1133) UL 0 # 4, 1
PMTFInformation3
(0029,1134) CS [DB TO DICOM] # 12, 1
PMTFInformation4
(0029,1260) ?? 63\6f\6d\20 # 4, 1
Unknown Tag & Data
(0029,1310) OB 53\56\31\30\04\03\02\01\38\00\00\00\4d
\00\00\00\45\63\68\6f\4c\69... # 6788, 1 CSImageHeaderInfo

```

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( outfilename );
if ( !w.Write() )
{
    return 1;
}

return 0;
}

```

29.46 ELSCINT1WaveToText.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"

/*
 * This example shows how to read a Wave Information tag from ELSCINT1
 * The wave information is stored in Tag (01e1,18,ELSCINT1) hidden in a
 * Secondary Capture Image Storage (usually a 'N' Symbol is shown)
 *
 * Everything done in this code is for the sole purpose of writing interoperable
 * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
 * If you believe anything in this code violates any law or any of your rights,
 * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
 * find a solution.
 */

```

```

*
* Everything you do with this code is at your own risk, since decompression
* algorithm was not written from specification documents.
*
* Special thanks to:
* Gauthier Bouilhol
*/

template <typename T>
bool dumpargs(std::ostream & os, T c1, T c2, T c3, T c4, T c5, T c6, T c7, T c8)
{
    static const char sep = '\t';
    os << c1 << sep << c2 << sep << c3 << sep << c4 << sep << c5 << sep << c6 << sep << c7 << sep << c8;
    os << std::endl;
    return true;
}

bool wave2stream( std::ostream &text_file, const char *in, size_t len )
{
    short * buffer = (short*)in;
    size_t length = len / sizeof( short );
    text_file << "COMPLETE_WAVE" << '\t' << "MASK" << '\t' << "AQUISITION_PROFIL" << '\t' << "
    END-INHALE" << '\t' << "END-EXHALE" << '\t' << "AQUISITION_WAVE" << '\t' << "WAVE_STATISTICS" << '\t' << "MASK"
    << std::endl;
    for (size_t i=0;i<length-76;i+=2)
    {
        if ( i < 74 )
        {
            if (buffer[i+75] == 0)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0 << '\t' << " "
                << '\t' << " " << '\t' << " " << '\t' << buffer[i] << '\t' << buffer
                [i+1] << std::endl;
            if (buffer[i+75] == 16384)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0 << '\t' <<
                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+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' << " " << '\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 << '\t' << " "
                << '\t' << buffer[i+74] << '\t' << buffer[i+74] << '\t' << buffer[i] << '\t' << buffer
                [i+1] << std::endl;
        }
        else
        {
            if (buffer[i+75] == 0)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0 << '\t' << " "
                << '\t' << " " << '\t' << " " << '\t' << " "
                << std::endl;
            if (buffer[i+75] == 16384)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0 << '\t' <<
                buffer[i+74] << '\t' << " " << '\t' << " " << '\t' << " " << '\t' << " "
                << std::endl;
            if (buffer[i+75] == 256)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0 << '\t' << " "
                << '\t' << buffer[i+74] << '\t' << " " << '\t' << " " << '\t' << " "
                << std::endl;
            if (buffer[i+75] == -32768)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1 << '\t' << " "
                << '\t' << " " << '\t' << buffer[i+74] << '\t' << " " << '\t' << " "
                << std::endl;
            if (buffer[i+75] == -16384)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1 << '\t' <<
                buffer[i+74] << '\t' << " " << '\t' << buffer[i+74] << '\t' << " " << '\t' << " "
                << std::endl;
            if (buffer[i+75] == -32512)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1 << '\t' << " "
                << '\t' << buffer[i+74] << '\t' << buffer[i+74] << '\t' << " " << '\t' << " "
                << std::endl;
        }
    }
}

```

```

    return true;
}

int main(int argc, char *argv [])
{
    if( argc < 3 ) return 1;
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }
    const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

    const gdcm::PrivateTag twave(0x01e1,0x18,"ELSCINT1");
    if( !ds.FindDataElement( twave ) ) return 1;
    const gdcm::DataElement& wave = ds.GetDataElement( twave );
    if ( wave.IsEmpty() ) return 1;
    const gdcm::ByteValue *bv = wave.GetByteValue();
    assert( bv );

    std::ofstream os( outfile, std::ios::binary );
    // Dump that to a CSV file:
    wave2stream( os, bv->GetPointer(), bv->GetLength() );
    os.close();

    return 0;
}

```

29.47 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;
}

```

29.48 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\2e\3\cf\d3\20\0d\31\30\20\30\20... # 40718, 1
 * EncapsulatedDocument
 * (0042,0012) LO [application/pdf] # 16, 1 MIMETypeOfEncapsulatedDocument
 * ...
 *
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcM/debug-gcc/bin
 * $ mono bin/ExtractEncapsulatedFile.exe some_pdf_encapsulated.dcm
 */
using System;

```

```

using gdcm;

public class ExtractEncapsulatedFile
{
    public static int Main(string[] args)
    {
        string file = args[0];
        Reader reader = new Reader();
        reader.SetFileName( file );
        bool ret = reader.Read();
        if( !ret )
        {
            return 1;
        }

        File f = reader.GetFile();
        DataSet ds = f.GetDataSet();
        Tag tencapsulated_stream = new Tag(0x0042,0x0011); // Encapsulated Document
        if( !ds.FindDataElement( tencapsulated_stream ) )
        {
            return 1;
        }
        // else
        DataElement de = ds.GetDataElement( tencapsulated_stream );
        ByteValue bv = de.GetByteValue();
        uint len = bv.GetLength();
        byte[] encapsulated_stream = new byte[len];
        bv.GetBuffer( encapsulated_stream, len );

        // Write out the decompressed bytes
        //System.Console.WriteLine(image.toString());
        using (System.IO.Stream stream =
            System.IO.File.Open(@"tmp/dd.pdf",
                System.IO.FileMode.Create))
        {
            System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
            writer.Write( encapsulated_stream );
        }

        return 0;
    }
}

```

29.49 ExtractEncryptedContent.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
#include "gdcmReader.h"

#include <fstream>

/*

openssl smime -encrypt -binary -aes256 -in outputfile.dcm -inform DER -out outputfile.der -outform DER ../
trunk/Testing/Source/Data/certificate.pem

openssl smime -decrypt -binary -in out.der -inform DER -out outputfile.dcm -outform DER -inkey ../trunk/
Testing/Source/Data/privatekey.pem ../trunk/Testing/Source/Data/certificate.pem

*/

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {

```

```

    std::cerr << argv[0] << " input.dcm output.der" << std::endl;
    return 1;
}
const char *filename = argv[1];
const char *outfilename = argv[2];

gdcm::Reader reader;
reader.SetFileName( filename );
if( !reader.Read() )
{
    return 1;
}

gdcm::File &file = reader.GetFile();
gdcm::DataSet &ds = file.GetDataSet();

const gdcm::DataElement &EncryptedAttributesSequence = ds.
    GetDataElement( gdcm::Tag( 0x0400,0x0500 ) );

gdcm::SequenceOfItems *sqi = EncryptedAttributesSequence.
    GetValueAsSQ();

if ( !sqi || sqi->GetNumberOfItems() != 1 ) return 1;

gdcm::Item &item = sqi->GetItem(1);

gdcm::DataSet &nestedds = item.GetNestedDataSet();

if( ! nestedds.FindDataElement( gdcm::Tag( 0x0400,0x0520) ) ) return 1;

const gdcm::DataElement &EncryptedContent = nestedds.
    GetDataElement( gdcm::Tag( 0x0400,0x0520) );

const gdcm::ByteValue *bv = EncryptedContent.GetByteValue();

std::ofstream of( outfile, std::ios::binary );
of.write( bv->GetPointer(), bv->GetLength() );
of.close();

return 0;
}

```

29.50 ExtractIconFromFile.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * This example shows how to either retrieve an Icon if present somewhere
 * in the file, or else generate one.
 */
#include "gdcmImageReader.h"
#include "gdcmPNMCodec.h"
#include "gdcmIconImageFilter.h"
#include "gdcmIconImageGenerator.h"

bool WriteIconAsPNM(const char* filename, const gdcm::IconImage& icon)
{
    gdcm::PNMCodec pnm;
    pnm.SetDimensions( icon.GetDimensions() );
    pnm.SetPixelFormat( icon.GetPixelFormat() );
    pnm.SetPhotometricInterpretation( icon.
        GetPhotometricInterpretation() );
    pnm.SetLUT( icon.GetLUT() );
    const gdcm::DataElement& in = icon.GetDataElement();
    bool b = pnm.Write( filename, in );
    assert( b );
}

```



```

    return b;
}

int main(int argc, char *argv [])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcm::ImageReader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read (or not image): " << filename << std::endl;
        return 1;
    }

    gdcm::IconImageFilter iif;
    iif.SetFile( reader.GetFile() );
    bool b = iif.Extract();

    if( b )
    {
        const gdcm::IconImage &icon = iif.GetIconImage(0);
        icon.Print( std::cout );

        if( !icon.GetTransferSyntax().IsEncapsulated() )
        {
            // Let's write out this icon as PNM file
            WriteIconAsPNM("icon.ppm", icon);
        }
        else if( icon.GetTransferSyntax() ==
            gdcm::TransferSyntax::JPEGBaselineProcess1
            || icon.GetTransferSyntax() ==
            gdcm::TransferSyntax::JPEGExtendedProcess2_4
        )
        {
            const gdcm::DataElement& in = icon.GetDataElement();
            const gdcm::ByteValue *bv = in.GetByteValue();
            assert( bv );
            std::ofstream out( "icon.jpg", std::ios::binary );
            out.write( bv->GetPointer(), bv->GetLength() );
            out.close();
        }
    }
    else
    {
        assert( iif.GetNumberOfIconImages() == 0 );
        std::cerr << "No Icon Found anywhere in file" << std::endl;

        const gdcm::Image &img = reader.GetImage();
        gdcm::IconImageGenerator iig;
        iig.AutoPixelMinMax(true);
        iig.SetPixmap( img );
        const unsigned int idims[2] = { 64, 64 };
        iig.SetOutputDimensions( idims );
        //iig.SetPixelMinMax(60, 868);
        if( !iig.Generate() ) return 1;
        const gdcm::IconImage & icon = iig.GetIconImage();
        WriteIconAsPNM("icon.ppm", icon);
    }

    return 0;
}

```

29.51 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;
}
}

```

29.52 ExtractImageRegion.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.

=====*/

/*
 * This small code shows how to use the gdcml.ImageRegionReader API
 * In this example we are taking each frame by frame and dump them to
 * /tmp/frame.raw.
 *
 * Usage:
 * $ LD_LIBRARY_PATH=. CLASSPATH=gdcml.jar:. java ExtractImageRegion input.dcm
 */
import gdcml.*;
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");
        }
    }
}

```

29.53 ExtractImageRegionWithLUT.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/

/*
 * This small code shows how to use the gdcm.ImageRegionReader API
 * In this example we are taking each frame by frame and dump them to
 * /tmp/frame.raw.
 * Furthermore we are applying the LUT on this image.
 * Special care should be taken in case the image is not PALETTE COLOR
 *
 * Usage:
 * $ bin/ExtractImageRegionWithLUT.exe input.dcm
 *
 * Example:
 * $ bin/ExtractImageRegionWithLUT.exe gdcmData/rle16l00.dcm
 * $ md5sum /tmp/frame_rgb.raw
 * 73bf61325fdb6e2830244a2b7b0c4ae2 /tmp/frame_rgb.raw
 * $ gdcming --depth 16 --spp 3 --size 600,430 /tmp/frame_rgb.raw rgb.dcm
 * $ gdcmvviewer rgb.dcm
 */
using System;
using gdcm;

public class ExtractImageRegion
{
    public static int Main(string[] args)
    {
        string filename = args[0];

        // instantiate the reader:
        gdcm.ImageRegionReader reader = new gdcm.
            ImageRegionReader();
        reader.SetFileName( filename );

        // pull DICOM info:
        if (!reader.ReadInformation()) return 1;
        // Get file infos
        gdcm.File f = reader.GetFile();

        gdcm.LookupTable lut = reader.GetImage().GetLUT();

        // get some info about image
        UIntArrayType dims = ImageHelper.GetDimensionsValue(f);
        PixelFormat pf = ImageHelper.GetPixelFormatValue (f);
        int pixelsize = pf.GetPixelSize();

        // buffer to get the pixels
        byte[] buffer = new byte[ dims[0] * dims[1] * pixelsize ];
    }
}

```

```

// output buffer for the RGB decoded image:
byte[] buffer2 = new byte[ dims[0] * dims[1] * pixelsize * 3 ];

// define a simple box region.
BoxRegion box = new BoxRegion();
for (uint z = 0; z < dims[2]; z++)
{
    // Define that I want the image 0, full size (dimx x dimy pixels)
    // and do that for each z:
    box.SetDomain(0, dims[0] - 1, 0, dims[1] - 1, z, z);
    //System.Console.WriteLine( box.ToString() );
    reader.SetRegion( box );

    // reader will try to load the uncompressed image region into buffer.
    // the call returns an error when buffer.Length is too small. For instance
    // one can call:
    // uint buf_len = reader.ComputeBufferLength(); // take into account pixel size
    // to get the exact size of minimum buffer
    if (reader.ReadIntoBuffer(buffer, (uint)buffer.Length))
    {
        if( !lut.Decode( buffer2, (uint)buffer2.Length, buffer, (uint)buffer.Length ) )
        {
            throw new Exception("can't decode");
        }

        using (System.IO.Stream stream =
            System.IO.File.Open(@"tmp/frame_rgb.raw",
                System.IO.FileMode.Create))
        {
            System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
            writer.Write(buffer2);
        }
    }
    else
    {
        throw new Exception("can't read pixels error");
    }
}

return 0;
}
}

```

29.54 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 <openjpeg.h>
#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 "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/MAROTECH_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);
        //gdcmmErrorMacro( "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;
        }
    }

    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 del( gdcmm::Tag(0x8,0x16) );
    del.SetVR( gdcmm::VR::UI );
    gdcmm::MediaStorage ms( gdcmm::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;
        el1.SetValue(col,0);
        el1.SetValue(row,1);
        gdcm::DataElement brr = el1.GetAsDataElement();
        brr.SetTag( gdcm::Tag(0x0048,0x0202) ); //brr --> bottom right col/row
        gdcm::Item it;
        gdcm::DataSet &nds = it.GetNestedDataSet();
        nds.Insert( rfn );
        nds.Insert( ulr );
        nds.Insert( brr );

        sq->AddItem(it);
    }

    gdcm::Writer w1;
    gdcm::File &file1 = w1.GetFile();
    gdcm::DataSet &ds1 = file1.GetDataSet();
    file1.GetHeader().SetDataSetTransferSyntax(
        gdcm::TransferSyntax::ExplicitVRLittleEndian );

```



```

gdcmm::UIDGenerator uid1;
gdcmm::DataElement dea( gdcmm::Tag(0x8,0x18) ); // SOP Instance UID
dea.SetVR( gdcmm::VR::UI );
const char *ul = uid1.Generate();
dea.SetByteValue( ul, strlen(ul) );
dsl.Insert( dea );

gdcmm::DataElement deb( gdcmm::Tag(0x8,0x16) );
deb.SetVR( gdcmm::VR::UI );
gdcmm::MediaStorage msl(
    gdcmm::MediaStorage::VLWholeSlideMicroscopyImageStorage
);
deb.SetByteValue( msl.GetString(), strlen( msl.GetString() ) );
dsl.Insert( deb );

const char mystr1[] = "MONOCHROME2 ";
gdcmm::DataElement dec( gdcmm::Tag(0x28,0x04) );
//de.SetTag( gdcmm::Tag(0x28,0x04) );
dec.SetVR( gdcmm::VR::CS );
dec.SetByteValue( mystr, strlen( mystr ) );
dsl.Insert( dec );

gdcmm::Attribute<0x0028,0x0010> row1 = {image->y1};
//row.SetValue(512);
dsl.Insert( row1.GetAsDataElement() );
// w.SetCheckFileMetaInformation( true );
gdcmm::Attribute<0x0028,0x0011> col1 = {image->x1};
dsl.Insert( col1.GetAsDataElement() );
gdcmm::Attribute<0x0028,0x0008> Number_Of_Frames1 = {tccp->numresolutions};
dsl.Insert( Number_Of_Frames1.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0100> ata = {8};
dsl.Insert( ata.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0002> atb = {image->numcomps};
dsl.Insert( atb.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0101> atc = {8};
dsl.Insert( atc.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0102> atd = {7};
dsl.Insert( atd.GetAsDataElement() );

theStreamWriter.SetFile(file1);

gdcmm::DataElement des( gdcmm::Tag(0x0048,0x0200) );
des.SetVR( gdcmm::VR::SQ );
//des.SetVR( gdcmm::VR::VM1 );
des.SetValue(*sq);
des.SetVLTToUndefined();

dsl.Insert( des );

if (!theStreamWriter.WriteImageInformation()){
    std::cerr << "unable to write image information" << std::endl;
    return 1; //the CanWrite function should prevent getting here, else,
    //that's a test failure
}

theStreamWriter.SetFile(file);

if (!theStreamWriter.CanWriteFile()){
    delete [] raw;
    std::cout << "Not able to write";
    return 0; //this means that the file was unwritable, period.
    //very similar to a ReadImageInformation failure
}
else
    std::cout<<"\nable to read";

// Important to write here
std::vector<unsigned int> extent = gdcmm::ImageHelper::GetDimensionsValue
    (file);

unsigned short xmax = extent[0];
unsigned short ymax = extent[1];
unsigned short theChunkSize = 4;

```

```

unsigned short ychunk = extent[1]/theChunkSize; //go in chunk sizes of theChunkSize
unsigned short zmax = extent[2];
std::cout << "\n"<<xmax << "\n" << ymax<<"\n"<<zmax<<"\n" << image->numcomps<<"\n";

if (xmax == 0 || ymax == 0)
{
    std::cerr << "Image has no size, unable to write zero-sized image." << std::endl;
    return 0;
}

int z, y, nexty;
unsigned long prevLen = 0; //when going through the char buffer, make sure to grab
//the bytes sequentially. So, store how far you got in the buffer with each iteration.
for (z = 0; z < zmax; ++z){
    for (y = 0; y < ymax; y += ychunk){
        nexty = y + ychunk;
        if (nexty > ymax) nexty = ymax;
        theStreamWriter.DefinePixelExtent(0, xmax, y, nexty, z, z+1);
        unsigned long len = theStreamWriter.DefineProperBufferLength();
        std::cout << "\n" <<len;
        char* finalBuffer = new char[len];
        memcpy(finalBuffer, &(raw[prevLen]), len);
        std::cout << "\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 = 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;
}

```

29.55 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;
}
}

```

29.56 Fake_Image_Using_Stream_Image_Writer.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcM.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
// This work was realised during the GSOC 2011 by Manoj Alwani

#include "gdcMReader.h"
#include "gdcMMediaStorage.h"
#include "gdcMWriter.h"
#include "gdcMItem.h"
#include "gdcMImageReader.h"
#include "gdcMAttribute.h"
#include "gdcMFile.h"
#include "gdcMTag.h"
#include "gdcMTransferSyntax.h"
#include "gdcMUIDGenerator.h"
#include "gdcMAnonymizer.h"
#include "gdcMStreamImageWriter.h"

```

```

#include "gdcmImageHelper.h"
#include "gdcmTrace.h"

int main(int, char *[])
{

    char * buffer = new char[ 256 * 256 *3 ];
    // *p = (uint8_t*)buffer;
    char * p = buffer;

    gdcm::Trace::DebugOn();
    gdcm::Trace::WarningOn();

    for(int row = 0; row < 256; ++row)
    {
        for(int col = 0; col < 256; ++col)
            //for(int b = 0; b < 256; ++b)
            {
                *p++ = 255;
                *p++ = 0;
                *p++ = 0;
            }
    }

    gdcm::Writer w;
    gdcm::File &file = w.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();

    file.GetHeader().SetDataSetTransferSyntax(
        gdcm::TransferSyntax::ExplicitVRLittleEndian );

    gdcm::UIDGenerator uid;
    gdcm::DataElement de( gdcm::Tag(0x8,0x18) ); // SOP Instance UID
    de.SetVR( gdcm::VR::UI );
    const char *u = uid.Generate();
    de.SetByteValue( u, strlen(u) );
    ds.Insert( de );

    gdcm::DataElement del( gdcm::Tag(0x8,0x16) );
    del.SetVR( gdcm::VR::UI );
    gdcm::MediaStorage ms(
        gdcm::MediaStorage::VLWholeSlideMicroscopyImageStorage
    );
    del.SetByteValue( ms.GetString(), strlen(ms.GetString()) );
    ds.Insert( del );

    const char mystr[] = "RGB";
    gdcm::DataElement de2( gdcm::Tag(0x28,0x04) );
    //de.SetTag(gdcm::Tag(0x28,0x04));
    de2.SetVR( gdcm::VR::CS );
    de2.SetByteValue(mystr, strlen(mystr));
    ds.Insert( de2 );

    gdcm::Attribute<0x0028,0x0010> row = {256};
    //row.SetValue(512);
    ds.Insert( row.GetAsDataElement() );
    // w.SetCheckFileMetaInformation( true );
    gdcm::Attribute<0x0028,0x0011> col = {256};
    ds.Insert( col.GetAsDataElement() );

    gdcm::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
    ds.Insert( Number_Of_Frames.GetAsDataElement() );

    gdcm::Attribute<0x0028,0x0100> at = {8};
    ds.Insert( at.GetAsDataElement() );

    gdcm::Attribute<0x0028,0x0002> at1 = {3}; //bits per pixel
    ds.Insert( at1.GetAsDataElement() );

    gdcm::Attribute<0x0028,0x0101> at2 = {8};
    ds.Insert( at2.GetAsDataElement() );

    gdcm::Attribute<0x0028,0x0102> at3 = {7};
    ds.Insert( at3.GetAsDataElement() );

    gdcm::Attribute<0x0028,0x0006> at4 = {0};
    ds.Insert( at4.GetAsDataElement() );

    gdcm::Attribute<0x0028,0x0103> at5 = {0};
    ds.Insert( at5.GetAsDataElement() );

```

```

//de.SetTag(gdcm::Tag(0x7fe0,0x0010));
//ds.Insert(de);

gdcm::StreamImageWriter theStreamWriter;
gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new
    gdcm::SequenceOfItems();
sq->SetLengthToUndefined();

uint16_t row1 = 256;
uint16_t col1 = 256;
//std::cout << row;

gdcm::Element<gdcm::VR::IS, gdcm::VM::VM1> el2;
el2.SetValue(1);
gdcm::DataElement rfn = el2.GetAsDataElement(); //rfn --->
    reference frame number
rfn.SetTag( gdcm::Tag(0x0008,0x1160) );

gdcm::Element<gdcm::VR::US, gdcm::VM::VM2> el;
el.SetValue(1,0);
el.SetValue(1,1);
gdcm::DataElement ulr = el.GetAsDataElement(); //ulr --> upper
    left col/row
ulr.SetTag( gdcm::Tag(0x0048,0x0201) );

gdcm::Element<gdcm::VR::US, gdcm::VM::VM2> ell;
ell.SetValue(col1,0);
ell.SetValue(row1,1);
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::DataElement des( gdcm::Tag(0x0048,0x0200) );
des.SetVR(gdcm::VR::SQ);
des.SetValue(*sq);
des.SetVLToUndefined();

ds.Insert(des);

theStreamWriter.SetFile(file);

std::ofstream of;
of.open( "output.dcm", std::ios::out | std::ios::binary );
theStreamWriter.SetStream(of);

if (!theStreamWriter.CanWriteFile()){
    delete [] buffer;
    std::cout << "Not able to write";
    return 0; //this means that the file was unwritable, period.
    //very similar to a ReadImageInformation failure
}
else
    std::cout<<"\nable to read";

if (!theStreamWriter.WriteImageInformation()){
    std::cerr << "unable to write image information" << std::endl;
    delete [] buffer;
    return 1; //the CanWrite function should prevent getting here, else,
    //that's a test failure
}

std::vector<unsigned int> extent =
    gdcm::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 = 0xffff;
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;
}

```

29.57 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;
    }
}

```

29.58 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" );
}
}

```

29.59 FileChangeTS.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Simple C# example
 *
 * Shows multiple steps:
 * Steps 1.
 * Create a fake (dummy) DICOM file, with size 512 x 512 x 2 We use a small
 * image to be able to create the volume in memory Of course you can use any
 * existing DICOM instead
 *
 * Step 2.
 * Hack the DICOM file to pretend the number of frames is 1000 (instead of 2)
 * At this point in time this makes the DICOM file invalid (truncated). But the
 * next step will fix this.
 *
 * Step 3.
 * Use C# to create a binary data which will represent our source object for
 * image.
 *
 * Step 4.
 * We use gdcm.FileStreamer to merge the template DICOM file from Step 2, with
 * the binary data from Step 3. We decide to read a scanline at a time, but
 * this can be read with any number of bytes. AppendToDataElement() will always
 * do the proper computation.
 *
 * Step 5.
 * We compress this gigantic file, into [JPEG Lossless, Non-Hierarchical,
 * First-Order Prediction (Process 14 [Selection Value 1])]
 *
 * Usage:
 * $ mono bin/FileChangeTS.exe small.dcm big.dcm raw.data merge.dcm jpeg.dcm
 */
using System;
using System.IO;
using gdcm;

public class FileChangeTS
{
    public static byte[] StrToByteArray(string str)
    {
        System.Text.Encoding encoding=new System.Text.Encoding();
        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 AssembledDICOMAndRaw(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;
}
}

```

29.60 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 AssembledDICOMAndRaw(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;
}
}

```

29.61 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;
}

```

29.62 FindAllPatientName.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #       This software is distributed WITHOUT ANY WARRANTY; without even
10 #       the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #       PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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.SetByteValue('F*',gdcm.VL(2))
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]

```

29.63 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
//
// -> BuggyJ2Kvvvua-fixed2-j2k.dcm

/*
 * This program attempts to fix a broken J2K/DICOM:
 * It contains 2 bugs:
 * 1. The first 8 bytes seems to be random bytes: remove them
 * 2. YCC is set to 1, while image is grayscale need to set it back to 0
 *
 * Ref:
 * It's a software from http://rentgenprom.ru/ , shipped with universal digital radiographic units
 * "ProScan-2000". The Ukrainian manufacturer developed own digital radiographic unit and it is
 * compatible with software from "ProScan-2000".
 * Information found in DICOM file is:
 *
 * (0008,0070) LO [ZAO "Renthenprom" (JSC Rentgenprom) ]           # 36,1 Manufacturer
 * (0018,1020) LO [2.13.1.7]                                       # 8,1-n Software Version(s)
 *
 */
int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcm::File &file = reader.GetFile();
    const gdcm::DataElement &pixeldata0 = file.GetDataSet().
        GetDataElement( gdcm::Tag(0x7fe0,0x0010) );
    const gdcm::SequenceOfFragments *sqf = pixeldata0.
        GetSequenceOfFragments();
    if( !sqf )
    {
        return 1;
    }
    const gdcm::Fragment &frag0 = sqf->GetFragment(0);

    const gdcm::ByteValue *bv = frag0.GetByteValue();
    const char *ptr = bv->GetPointer();
    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;
    const char *cod_marker = ptr + 0x35; /* 0x2d + 0x8 */ // FIXME
    if( cod_marker[0] == (char)0xff && cod_marker[1] == 0x52 )
    {
        // found start of COD
        if( cod_marker[6+2] == 1 )
        {
            // Change in place:
            *((char*)cod_marker + 6+2) = 0;
            // Prepare a new DataElement:
            gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
            pixeldata.SetVR( gdcm::VR::OB );
            gdcm::SmartPointer<gdcm::SequenceOfFragments> sq = new
            gdcm::SequenceOfFragments;

            gdcm::Fragment frag;

```



```

        // remove 8 first bytes:
        frag.SetByteValue( ptr + 8, (uint32_t)(len - 8) );
        sq->AddFragment( frag );
        pixeldata.SetValue( *sq );
        file.GetDataSet().Replace( pixeldata );
    }
    else
    {
        return 1;
    }
}
else
{
    std::cerr << "COD not found" << (int)cod_marker[0] << std::endl;
    return 1;
}

gdcm::Writer writer;
writer.SetFile( reader.GetFile() );
writer.SetFileName( outfilename );
writer.CheckFileMetaInformationOff();
if( !writer.Write() )
{
    std::cerr << "Could not write" << std::endl;
}

// paranoid check:
gdcm::ImageReader ireader;
ireader.SetFileName( outfilename );
if( !ireader.Read() )
{
    std::cerr << "file written is still not valid, please report" << std::endl;
    return 1;
}

return 0;
}

```

29.64 FixCommaBug.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #       This software is distributed WITHOUT ANY WARRANTY; without even
10 #       the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #       PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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

```

29.65 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 --jpegls' 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<BYTE> rgbbyteOutall;
    for(unsigned int i = 0; i < sf->GetNumberOfFragments(); ++i)
    {
        const gdcm::Fragment &frag = sf->GetFragment(i);
        if( frag.IsEmpty() ) return 1;
        const gdcm::ByteValue *bv = frag.GetByteValue();
        if( !bv ) return 1;
        unsigned long totalLen = bv->GetLength();

        std::vector<char> vbuffer;
        vbuffer.resize( totalLen );
        char *buffer = &vbuffer[0];
        bv->GetBuffer(buffer, totalLen);
        const BYTE* pbyteCompressed0 = (const BYTE*)buffer;
        while( totalLen > 0 && pbyteCompressed0[totalLen-1] != 0xd9 )
        {
            totalLen--;
        }

        JlsParameters metadata;
        if (JpegLsReadHeader(buffer, totalLen, &metadata) != OK)
        {
            std::cerr << "Cant parse jpegls" << std::endl;
            return false;
        }

        std::cout << metadata.width << std::endl;
        std::cout << metadata.height << std::endl;
        std::cout << metadata.bitspersample << std::endl;

        gdcm::PixelFormat const &pf = image.GetPixelFormat();
        std::cout << pf << std::endl;
    }
}

```

```

// http://charls.codeplex.com/discussions/230307?ProjectName=charls
unsigned char marker_lse_13[] = {
    0xFF, 0xF8, 0x00, 0x0D,
    0x01,
    0x1F, 0xFF,
    0x00, 0x22, // T1 = 34
    0x00, 0x83, // T2 = 131
    0x02, 0x24, // T3 = 548
    0x00, 0x40
};

unsigned char marker_lse_14[] = {
    0xFF, 0xF8, 0x00, 0x0D,
    0x01,
    0x3F, 0xFF,
    0x00, 0x42, // T1 = 66
    0x01, 0x03, // T2 = 259
    0x04, 0x44, // T3 = 1092
    0x00, 0x40
};

unsigned char marker_lse_15[] = {
    0xFF, 0xF8, 0x00, 0x0D,
    0x01,
    0x7F, 0xFF,
    0x00, 0x82, // T1 = 130
    0x02, 0x03, // T2 = 515
    0x08, 0x84, // T3 = 2180
    0x00, 0x40
};

unsigned char marker_lse_16[] = {
    0xFF, 0xF8, 0x00, 0x0D,
    0x01,
    0xFF, 0xFF,
    0x01, 0x02, // T1 = 258
    0x04, 0x03, // T2 = 1027
    0x11, 0x04, // T3 = 4356
    0x00, 0x40
};

const unsigned char *marker_lse = NULL;
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);

std::vector<BYTE> rgbyteOut;
//rgbyteOut.resize( image.GetBufferLength() );

```

```

    rgbyteOut.resize(params.height * params.width * ((params.bitspersample + 7)
        / 8) * params.components);

    JLS_ERROR result =
        JpegLsDecode(&rgbyteOut[0], rgbyteOut.size(), pbyteCompressed, cbyteCompressed, &params );
    if (result != OK)
    {
        std::cerr << "Could not patch JAI-JPEGLS" << std::endl;
        return 1;
    }
    rgbyteOutall.insert( rgbyteOutall.end(), rgbyteOut.begin(), rgbyteOut.end() );
}

gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
pixeldata.SetVR( gdcm::VR::OW );
pixeldata.SetByteValue( (char*)&rgbyteOutall[0], (uint32_t)rgbyteOutall.size() );

// Add the pixel data element
reader.GetFile().GetDataSet().Replace( pixeldata );
reader.GetFile().GetHeader().SetDataSetTransferSyntax(
    gdcm::TransferSyntax::ExplicitVRLittleEndian);

gdcm::Writer writer;
writer.SetFileName( outfilename );
writer.SetFile( reader.GetFile() );
writer.Write();

std::cout << "Success !" << std::endl;

return 0;
}

```

29.66 FixOrientation.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmFile.h"
#include "gdcmOrientation.h"
#include "gdcmAttribute.h"

// Very simple orientation changer, fix invalid dataset
int main(int argc, char* argv[] )
{
    // assume AXIAL input for now
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    gdcm::Reader reader;
    reader.SetFileName( filename );
    if (! reader.Read() )
    {
        return 1;
    }

    const double axial[] = { 1,0,0, 0,1,0 };
    (void)axial;
    const double coronal[] = { 0,0,1, 1,0,0 };
    (void)coronal;

```

```

const double sagittal[] = { 0,1,0, 0,0,1 };
(void)sagittal;
gdcm::Attribute<0x0020,0x0032> at1; // IPP
(void)at1;
gdcm::Attribute<0x0020,0x0037> at2; // IOP
(void)at2;

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( outfilename );
if ( !writer.Write() )
{
    return 1;
}

return 0;
}

```

29.67 gdcmorthoplanes.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

#include "vtkActor.h"
#include "vtkCamera.h"
#include "vtkMatrix4x4.h"
#include "vtkTransform.h"
#include "vtkAssembly.h"
#include "vtkCellPicker.h"
#include "vtkCommand.h"
#include "vtkImageActor.h"
#include "vtkImageMapToColors.h"
#include "vtkImageOrthoPlanes.h"
#include "vtkImagePlaneWidget.h"
#include "vtkImageReader.h"
#include "vtkInteractorEventRecorder.h"
#include "vtkLookupTable.h"
#include "vtkOutlineFilter.h"
#include "vtkPolyDataMapper.h"

```

```

#include "vtkProperty.h"
#include "vtkRenderWindow.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkRenderer.h"
#include "vtkVolume16Reader.h"
#include "vtkImageData.h"
#include "vtkImageChangeInformation.h"
#include "vtkOrientationMarkerWidget.h"
#include "vtkAnnotatedCubeActor.h"
#include "vtkAxesActor.h"
#include "vtkCaptionActor2D.h"
#include "vtkTextProperty.h"
#include "vtkPropAssembly.h"

#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkStringArray.h"

#include "gdcmsystem.h"
#include "gdcmdir.h"
#include "gdcmppsorter.h"

#ifdef vtkFloatingPointType
#define vtkFloatingPointType float
#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 && gdcmm::System::FileIsDirectory( filename ) )
    {
        std::cout << "Loading directory: " << filename << std::endl;
        bool recursive = false;
        gdcmm::Directory d;
        d.Load(filename, recursive);
        gdcmm::Directory::FileNamesType const &files = d.
        GetFileNames();
        for( gdcmm::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( gdcmm::System::FileExists( filename ) )
            {
                if( gdcmm::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 )
{
    //gdcmm::Trace::DebugOn();
    //gdcmm::Trace::WarningOn();
    gdcmm::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.GetFileNames();
    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();
#endif

    vtkOutlineFilter* outline = vtkOutlineFilter::New();
    outline->SetInputConnection(v16->GetOutputPort());

    vtkPolyDataMapper* outlineMapper = vtkPolyDataMapper::New();
    outlineMapper->SetInputConnection(outline->GetOutputPort());

    vtkActor* outlineActor = vtkActor::New();
    outlineActor->SetMapper( outlineMapper);

    vtkRenderer* ren1 = vtkRenderer::New();
    vtkRenderer* ren2 = vtkRenderer::New();

    vtkRenderWindow* renWin = vtkRenderWindow::New();
    renWin->AddRenderer(ren2);
    renWin->AddRenderer(ren1);

    vtkRenderWindowInteractor* iren = vtkRenderWindowInteractor::New();
    iren->SetRenderWindow(renWin);

    vtkCellPicker* picker = vtkCellPicker::New();
    picker->SetTolerance(0.005);

    vtkProperty* ipwProp = vtkProperty::New();
    //assign default props to the ipw's texture plane actor

    vtkImagePlaneWidget* planeWidgetX = vtkImagePlaneWidget::New();
    planeWidgetX->SetInteractor( iren);
    planeWidgetX->SetKeyPressActivationValue('x');
    planeWidgetX->SetPicker(picker);
    planeWidgetX->RestrictPlaneToVolumeOn();
    planeWidgetX->GetPlaneProperty()->SetColor(1,0,0);
    planeWidgetX->SetTexturePlaneProperty(ipwProp);
    planeWidgetX->TextureInterpolateOff();
    planeWidgetX->SetResliceInterpolateToNearestNeighbour();
#if (VTK_MAJOR_VERSION >= 6)
    planeWidgetX->SetInputConnection(v16->GetOutputPort());
#else
    planeWidgetX->SetInput(v16->GetOutput());
#endif
    planeWidgetX->SetPlaneOrientationToXAxes();
    //planeWidgetX->SetSliceIndex(32);
    planeWidgetX->DisplayTextOn();

```

```

planeWidgetX->On();
planeWidgetX->InteractionOff();
planeWidgetX->InteractionOn();

vtkImagePlaneWidget* planeWidgetY = vtkImagePlaneWidget::New();
planeWidgetY->SetInteractor(iren);
planeWidgetY->SetKeyPressActivationValue('y');
planeWidgetY->SetPicker(picker);
planeWidgetY->GetPlaneProperty()->SetColor(1,1,0);
planeWidgetY->SetTexturePlaneProperty(ipwProp);
planeWidgetY->TextureInterpolateOn();
planeWidgetY->SetResliceInterpolateToLinear();
#if (VTK_MAJOR_VERSION >= 6)
planeWidgetY->SetInputConnection(vl6->GetOutputPort());
#else
planeWidgetY->SetInput(vl6->GetOutput());
#endif
planeWidgetY->SetPlaneOrientationToYAxes();
//planeWidgetY->SetSlicePosition(102.4);
planeWidgetY->SetLookupTable(planeWidgetX->GetLookupTable());
planeWidgetY->DisplayTextOn();
planeWidgetY->UpdatePlacement();
planeWidgetY->On();

vtkImagePlaneWidget* planeWidgetZ = vtkImagePlaneWidget::New();
planeWidgetZ->SetInteractor(iren);
planeWidgetZ->SetKeyPressActivationValue('z');
planeWidgetZ->SetPicker(picker);
planeWidgetZ->GetPlaneProperty()->SetColor(0,0,1);
planeWidgetZ->SetTexturePlaneProperty(ipwProp);
planeWidgetZ->TextureInterpolateOn();
planeWidgetZ->SetResliceInterpolateToCubic();
#if (VTK_MAJOR_VERSION >= 6)
planeWidgetZ->SetInputConnection(vl6->GetOutputPort());
#else
planeWidgetZ->SetInput(vl6->GetOutput());
#endif
planeWidgetZ->SetPlaneOrientationToZAxes();
//planeWidgetZ->SetSliceIndex(25);
planeWidgetZ->SetLookupTable(planeWidgetX->GetLookupTable());
planeWidgetZ->DisplayTextOn();
planeWidgetZ->On();

vtkImageOrthoPlanes* orthoPlanes = vtkImageOrthoPlanes::New();
orthoPlanes->SetPlane(0, planeWidgetX);
orthoPlanes->SetPlane(1, planeWidgetY);
orthoPlanes->SetPlane(2, planeWidgetZ);
orthoPlanes->ResetPlanes();

vtkOrthoPlanesCallback* cbk = vtkOrthoPlanesCallback::New();
cbk->WidgetX = planeWidgetX;
cbk->WidgetY = planeWidgetY;
cbk->WidgetZ = planeWidgetZ;
planeWidgetX->AddObserver(vtkCommand::EndWindowLevelEvent, cbk);
planeWidgetY->AddObserver(vtkCommand::EndWindowLevelEvent, cbk);
planeWidgetZ->AddObserver(vtkCommand::EndWindowLevelEvent, cbk);
cbk->Delete();

double wl[2];
planeWidgetZ->GetWindowLevel(wl);

// Add a 2D image to test the GetReslice method
//
vtkImageMapToColors* colorMap = vtkImageMapToColors::New();
colorMap->PassAlphaToOutputOff();
colorMap->SetActiveComponent(0);
colorMap->SetOutputFormatToLuminance();
#if (VTK_MAJOR_VERSION >= 6)
colorMap->SetInputData(planeWidgetZ->GetResliceOutput());
#else
colorMap->SetInput(planeWidgetZ->GetResliceOutput());
#endif
colorMap->SetLookupTable(planeWidgetX->GetLookupTable());

vtkImageActor* imageActor = vtkImageActor::New();
imageActor->PickableOff();
#if (VTK_MAJOR_VERSION >= 6)
imageActor->SetInputData(colorMap->GetOutput());
#else
imageActor->SetInput(colorMap->GetOutput());
#endif

```

```

// Add the actors
//
ren1->AddActor( outlineActor);
ren2->AddActor( imageActor);

ren1->SetBackground( 0.1, 0.1, 0.2);
ren2->SetBackground( 0.2, 0.1, 0.2);

renWin->SetSize( 600, 350);

ren1->SetViewport(0,0,0.58333,1);
ren2->SetViewport(0.58333,0,1,1);

// Set the actors' postions
//
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;
}

```

29.68 gdcmmreslice.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
#include "vtkGDCMImageReader.h"

#include "vtkRenderer.h"
#include "vtkAssembly.h"
#include "vtkImageFlip.h"
#include "vtkImageReslice.h"
#include "vtkRenderWindow.h"

```

```

#include "vtkAnnotatedCubeActor.h"
#include "vtkTransform.h"
#include "vtkAxesActor.h"
#include "vtkTextProperty.h"
#include "vtkCaptionActor2D.h"
#include "vtkPropAssembly.h"
#include "vtkOrientationMarkerWidget.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkImageData.h"
#include "vtkLookupTable.h"
#include "vtkTexture.h"
#include "vtkPlaneSource.h"

int main( int argc, char *argv[] )
{
    if( argc < 2 ) return 1;
    vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
    reader->SetFileName( argv[1] );
    //reader->FileLowerLeftOn();
    reader->Update();

    vtkImageFlip *flip = vtkImageFlip::New();
    #if (VTK_MAJOR_VERSION >= 6)
        flip->SetInputConnection( reader->GetOutputPort() );
    #else
        flip->SetInput( reader->GetOutput() );
    #endif
    flip->SetFilteredAxis(0);
    flip->Update();

    vtkImageReslice *reslice = vtkImageReslice::New();
    //reslice->SetInput( reader->GetOutput() );
    #if (VTK_MAJOR_VERSION >= 6)
        reslice->SetInputConnection( flip->GetOutputPort() );
    #else
        reslice->SetInput( flip->GetOutput() );
    #endif
    //reslice->SetResliceAxesDirectionCosines()
    reader->GetDirectionCosines()->Print(std::cout);
    vtkMatrix4x4 *invert = vtkMatrix4x4::New();
    invert->DeepCopy( reader->GetDirectionCosines() );
    invert->Invert();

    //reslice->SetResliceAxes( reader->GetDirectionCosines() );
    reslice->SetResliceAxes( invert );
    reslice->Update();
    vtkImageData* ima = reslice->GetOutput();

    vtkLookupTable* table = vtkLookupTable::New();
    table->SetNumberOfColors(1000);
    table->SetTableRange(0,1000);
    table->SetSaturationRange(0,0);
    table->SetHueRange(0,1);
    table->SetValueRange(0,1);
    table->SetAlphaRange(1,1);
    table->Build();

    // Texture
    vtkTexture* texture = vtkTexture::New();
    #if (VTK_MAJOR_VERSION >= 6)
        texture->SetInputData(ima);
    #else
        texture->SetInput(ima);
    #endif
    texture->InterpolateOn();
    texture->SetLookupTable(table);

    // PlaneSource
    vtkPlaneSource* plane = vtkPlaneSource::New();

    // PolyDataMapper
    vtkPolyDataMapper *planeMapper = vtkPolyDataMapper::New();
    #if (VTK_MAJOR_VERSION >= 6)
        planeMapper->SetInputConnection(plane->GetOutputPort());
    #else
        planeMapper->SetInput(plane->GetOutput());
    #endif

    // Actor

```

```

vtkActor* planeActor = vtkActor::New();
planeActor->SetTexture(texture);
planeActor->SetMapper(planeMapper);
planeActor->PickableOn();

// Final rendering with simple interactor:
vtkRenderer *ren = vtkRenderer::New();
vtkRenderWindow *renwin = vtkRenderWindow::New();
renwin->AddRenderer(ren);
vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
iren->SetRenderWindow(renwin);
ren->AddActor(planeActor);
ren->SetBackground(0,0,0.5);

// DICOM is RAH:
vtkAnnotatedCubeActor* cube = vtkAnnotatedCubeActor::New();
cube->SetXPlusFaceText ( "R" );
cube->SetXMinusFaceText ( "L" );
cube->SetYPlusFaceText ( "A" );
cube->SetYMinusFaceText ( "P" );
cube->SetZPlusFaceText ( "H" );
cube->SetZMinusFaceText ( "F" );

vtkAxesActor* axes2 = vtkAxesActor::New();

vtkTransform *transform = vtkTransform::New();
transform->Identity();
//reader->GetDirectionCosines()->Print(std::cout);
transform->Concatenate(invert);
//axes2->SetShaftTypeToCylinder();
axes2->SetUserTransform( transform );
cube->GetAssembly()->SetUserTransform( transform ); // 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;
}

```

29.69 gdcmrtnonplan.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 <vtkXMLPolyDataWriter.h>
#include <vtkRenderWindowInteractor.h>
#include <vtkImageColorViewer.h>

#include "gdcmsReader.h"
#include "gdcmsAttribute.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];

    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,03a2) SQ # u/l,1 Ion Beam Sequence
(ffff,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/l,1 Ion Range 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\35.43\35.43\34.54\34.54\34.71\36.10\38.62\44.88\44.88\45.00\45.00\45.66\45.66\46.42\39.77\39.77\39.77\39.77\
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 gdcm::DataSet& ds = reader.GetFile().GetDataSet();
gdcm::Tag tbeamsq(0x300a,0x03a2);
if( !ds.FindDataElement( tbeamsq ) )
{
    return 1;
}
const gdcm::DataElement &tbeamsq = 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(1); // Item start at #1
//     const gdcm::DataSet& nestedds = item.GetNestedDataSet();
//     //std::cout << nestedds << std::endl;
//     gdcm::Tag tcompensatorsq(0x300a,0x02ea);
//     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);

#if (VTK_MAJOR_VERSION >= 6)
#else
    img->Update();
#endif
img->Print(std::cout);

    vtkXMLImageDataWriter *writeb= vtkXMLImageDataWriter::New();
#if (VTK_MAJOR_VERSION >= 6)
    writeb->SetInputData( img );
#else
    writeb->SetInput( img );
#endif
writeb->SetFileName( outfilename );
writeb->Write();

/*
    (300a,03a6) SQ                                     # u/l,1 Ion Block Sequence
    (fffe,e000) na (Item with undefined length)
    (300a,00e1) SH [brass ]                             # 6,1 Material ID
    (300a,00f7) FL 95.03                                # 4,1 Isocenter to Block Tray Distance
    (300a,00f8) CS [APERTURE]                           # 8,1 Block Type
    (300a,00fa) CS [ABSENT]                             # 6,1 Block Divergence
    (300a,00fb) CS [SOURCE_SIDE ]                      # 12,1 Block Mounting Position
    (300a,00fc) IS [1 ]                                # 2,1 Block Number
    (300a,0100) DS [50.00 ]                             # 6,1 Block Thickness
    (300a,0104) IS [179 ]                              # 4,1 Block Number of Points
    (300a,0106) DS
    [1.7\50.0\14.3\50.0\16.7\49.4\18.7\48.2\19.4\47.7\20.1\47.1\21.0\47.0\22.3\47.0\23.7\
    46.8\25.7\46.2\27.0\45.6\27.2\45.4\28.2\44.6\28.9\44.2\29.7\43.9\31.5\43.5\33.0\42.8\33.7\42.4\35.2\41.3\38.2\40.4\39.6\39.7\
    2\37.4\43.0\37.1\44.7\36] # 1934,2-2n Block Data
    (fffe,e00d)
    (fffe,e0dd)

*/
gdcml::Tag tblocksq(0x300a,0x03a6);
if( !nestedds.FindDataElement( tblocksq ) )
{
    return 1;
}
const gdcml::DataElement &tblocksq = nestedds.GetDataElement( tblocksq );
//std::cout << tblocksq << std::endl;
gdcml::SmartPointer<gdcml::SequenceOfItems> sssqi = tblocksq.
    GetValueAsSQ();
const gdcml::Item &item3 = sssqi->GetItem(1); // Item start at #1
const gdcml::DataSet &nestedds3 = item3.GetNestedDataSet();

gdcml::Tag tblockdata(0x300a,0x0106);
if( !nestedds3.FindDataElement( tblockdata ) )
{
    return 1;
}
const gdcml::DataElement &tblockdata = nestedds3.
    GetDataElement( tblockdata );
// std::cout << tblockdata << std::endl;
gdcml::Attribute<0x300a,0x0106> at_;
at_.SetFromDataElement( tblockdata );

vtkDoubleArray *scalars = vtkDoubleArray::New();
scalars->SetNumberOfComponents(3);

gdcml::Attribute<0x300a,0x0104> bnpts; // IS [179 ]
# 4,1 Block Number of Points
if( !nestedds3.FindDataElement( bnpts.GetTag() ) )
{
    return 1;
}
const gdcml::DataElement &tblocknpts = nestedds3.
    GetDataElement( bnpts.GetTag() );

```

```

bnpts.SetFromDataElement( blocknpts );
//std::cout << bnpts.GetValue() << std::endl;

vtkPolyData *output = vtkPolyData::New();
vtkPoints *newPts = vtkPoints::New();
vtkCellArray *polys = vtkCellArray::New();
const double *ptr = at_.GetValues();
//unsigned int npts = bnpts.GetNumberOfValues() / 2;
unsigned int npts = bnpts.GetValue();
vtkIdType *ptIds = new vtkIdType[npts];
for(unsigned int i = 0; i < npts; ++i)
{
    float x[3] = {};
    x[0] = (float)ptr[2*i+0];
    x[1] = (float)ptr[2*i+1];
    //x[2] = ptr[i+2];
    vtkIdType ptId = newPts->InsertNextPoint( x );
    //std::cout << x[0] << ", " << x[1] << ", " << x[2] << std::endl;
    ptIds[i] = ptId;
}
vtkIdType cellId = polys->InsertNextCell(npts, ptIds);
(void)cellId;
delete[] ptIds;

output->SetPoints(newPts);
newPts->Delete();
output->SetPolys(polys);
polys->Delete();
//output->GetCellData()->SetScalars(scalars);
//scalars->Delete();
#if (VTK_MAJOR_VERSION >= 6)
#else
    output->Update();
#endif
output->Print( std::cout );

// }

vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();

vtkImageColorViewer *viewer = vtkImageColorViewer::New();
#if (VTK_MAJOR_VERSION >= 6)
viewer->SetInputData(img);
#else
viewer->SetInput(img);
#endif
viewer->SetupInteractor(iren);
viewer->SetSize(600, 600);
viewer->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;
}

```

29.70 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
  (fffe,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
(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\35.43\35.43\34.54\34.54\34.71\36.10\38.62\44.88\44.88\44.88\45.00\45.00\45.00\45.66\45.66\46.42\39.77\39.77\39.77\39.77]
Data
(300a,02e0) CS [ABSENT] # 6,1 Compensator Divergence
(300a,02e1) CS [SOURCE_SIDE ] # 12,1 Compensator Mounting Position
(fffe,e00d)
(fffe,e000) na (Item with undefined length)
(fffe,e00d)
(fffe,e0dd)
*/
const gdcmm::DataSet& ds = reader.GetFile().GetDataSet();
gdcmm::Tag tbeamsq(0x300a,0x00b0);
if( !ds.FindElement( tbeamsq ) )
{
    return 1;
}
const gdcmm::DataElement &beamsq = ds.GetDataElement( tbeamsq );
//std::cout << beamsq << std::endl;
gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqi = beamsq.
    GetValueAsSQ();
if( !sqi || !sqi->GetNumberOfItems() )
{
    return 1;
}

//for(unsigned int pd = 0; pd < sqi->GetNumberOfItems(); ++pd)
// {
//const gdcmm::Item & item = sqi->GetItem(1); // Item start at #1
const gdcmm::Item & item = sqi->GetItem(2); // Item start at #1
const gdcmm::DataSet& nestedds = item.GetNestedDataSet();
//std::cout << nestedds << std::endl;
gdcmm::Tag tcompensatorsq(0x300a,0x00e3);
if( !nestedds.FindElement( tcompensatorsq ) )
{
    return 1;
}
const gdcmm::DataElement &compensatorsq = nestedds.
    GetDataElement( tcompensatorsq );
//std::cout << compensatorsq << std::endl;
gdcmm::SmartPointer<gdcmm::SequenceOfItems> ssqi = compensatorsq.
    GetValueAsSQ();
const gdcmm::Item & item2 = ssqi->GetItem(1); // Item start at #1
const gdcmm::DataSet& nestedds2 = item2.GetNestedDataSet();
//std::cout << nestedds2 << std::endl;
gdcmm::Tag tcompensatorthicknessdata(0x300a,0x00ec);
if( !nestedds2.FindElement( tcompensatorthicknessdata ) )
{
    return 1;
}
const gdcmm::DataElement &compensatorthicknessdata = nestedds2.
    GetDataElement( tcompensatorthicknessdata );
// std::cout << compensatorthicknessdata << std::endl;
gdcmm::Attribute<0x300a,0x00ec> at;
at.SetFromDataElement( compensatorthicknessdata );
const double* pts = at.GetValues();
// (300a,00e7) IS [35] # 2,1 Compensator Rows
gdcmm::Attribute<0x300a,0x00e7> at1;
const gdcmm::DataElement &compensatorrows = nestedds2.
    GetDataElement( at1.GetTag() );
at1.SetFromDataElement( compensatorrows );
std::cout << at1.GetValue() << std::endl;
// (300a,00e8) IS [37] # 2,1 Compensator Columns
gdcmm::Attribute<0x300a,0x00e8> at2;
const gdcmm::DataElement &compensatorcols = nestedds2.
    GetDataElement( at2.GetTag() );
at2.SetFromDataElement( compensatorcols );
std::cout << at2.GetValue() << std::endl;

// (300a,00e9) DS [3.679991\4.249288 ] # 18,2 Compensator Pixel Spacing
gdcmm::Attribute<0x300a,0x00e9> at3;
const gdcmm::DataElement &compensatorpixelspacing = nestedds2.

```

```

        GetDataElement( at3.GetTag() );
at3.SetFromDataElement( compensatorpixelspacing );
std::cout << at3.GetValue(0) << std::endl;
        // (300a,00ea) DS [-76.00\62.50] # 12,2 Compensator Position
gdcml::Attribute<0x300a,0x00ea> at4;
const gdcml::DataElement &compensatorposition = nesteddds2.
        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 );
//img->SetExtent(1, xdim, 1, ydim, 1, zdim);
#if (VTK_MAJOR_VERSION >= 6)
    assert(0);
#else
    img->SetScalarTypeToDouble();
#endif
    img->SetSpacing( at3.GetValue(1), at3.GetValue(0), 1); // FIXME image is upside down
    img->SetOrigin( at4.GetValue(0), at4.GetValue(1), 1);
#if (VTK_MAJOR_VERSION >= 6)
    assert(0);
#else
    img->SetNumberOfScalarComponents(1);
#endif
    img->GetPointData()->SetScalars(d);

    vtkXMLImageDataWriter *writeb= vtkXMLImageDataWriter::New();
#if (VTK_MAJOR_VERSION >= 6)
    writeb->SetInputData( img );
#else
    writeb->SetInput( img );
#endif
    writeb->SetFileName( outfilename );
    writeb->Write();
/*
(300a,00f4) SQ # u/1,1 Block Sequence
    (fffe,e000) na (Item with undefined length)
        (300a,00e1) SH [brass ] # 6,1 Material ID
        (300a,00f8) CS [APERTURE] # 8,1 Block Type
        (300a,00fa) CS [ABSENT] # 6,1 Block Divergence
        (300a,00fb) CS [SOURCE_SIDE ] # 12,1 Block Mounting Position
        (300a,00fc) IS [1] # 2,1 Block Number
        (300a,0100) DS [50.00 ] # 6,1 Block Thickness
        (300a,0104) IS [179 ] # 4,1 Block Number of Points
        (300a,0106) DS
        [1.7\50.0\14.3\50.0\16.7\49.4\18.7\48.2\19.4\47.7\20.1\47.1\21.0\47.0\22.3\47.0\23.7\
        46.8\25.7\46.2\27.0\45.6\27.2\45.4\28.2\44.6\28.9\44.2\29.7\43.9\31.5\43.5\33.0\42.8\33.7\42.4\35.2\41.3\38.2\40.4\39.6\39.7\
        (fffe,e00d)
        (fffe,e000) na (Item with undefined length)
        (fffe,e00d)
    (fffe,e0dd)
*/
gdcml::Tag tblocksq(0x300a,0x00f4);
if( !nesteddds.FindDataElement( tblocksq ) )
{
    return 1;
}
const gdcml::DataElement &blocksq = nesteddds.GetDataElement( tblocksq );
//std::cout << blocksq << std::endl;
gdcml::SmartPointer<gdcml::SequenceOfItems> sssqi = blocksq.
    GetValueAsSQ();
const gdcml::Item & item3 = sssqi->GetItem(1); // Item start at #1
const gdcml::DataSet& nesteddds3 = item3.GetNestedDataSet();

gdcml::Tag tblockdata(0x300a,0x0106);
if( !nesteddds3.FindDataElement( tblockdata ) )
{
    return 1;
}
const gdcml::DataElement &blockdata = nesteddds3.
    GetDataElement( tblockdata );
// std::cout << blockdata << std::endl;
gdcml::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
    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;
}

```

29.71 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;
}

```

29.72 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;
}

```

29.73 gdcmvolume.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"
#include "vtkPiecewiseFunction.h"
#include "vtkColorTransferFunction.h"
#include "vtkVolume.h"
#include "vtkVolumeProperty.h"
#include "vtkVolumeTextureMapper3D.h"
#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;
}

```

29.74 GenAIIVR.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();

```

```

gdcmm::FileExplicitFilter fef;
//fef.SetChangePrivateTags( true );
fef.SetFile( w.GetFile() );
if( !fef.Change() )
{
    std::cerr << "Failed to change" << std::endl;
    return 1;
}

gdcmm::SmartPointer<gdcmm::SequenceOfItems> sq = new
    gdcmm::SequenceOfItems();
sq->SetLengthToUndefined();

// gdcmm::DummyValueGenerator dv;

const std::size_t len = 10;
char ss[len+1];
ss[len] = '\0';

const char owner_str[] = "GDCM CONFORMANCE TESTS";
gdcmm::DataElement owner( gdcmm::Tag(0x4d4d, 0x10) );
owner.SetByteValue(owner_str, (uint32_t)strlen(owner_str));
owner.SetVR( gdcmm::VR::LO );

// Create an item
gdcmm::Item it;
it.SetVLToUndefined();
gdcmm::DataSet &nds = it.GetNestedDataSet();
// nds.Insert(owner);
// nds.Insert(de);

// Insert sequence into data set
gdcmm::DataElement des( gdcmm::Tag(0x4d4d, 0x1001) );
des.SetVR(gdcmm::VR::SQ);
des.SetValue(*sq);
des.SetVLToUndefined();

ds.Insert(owner);
ds.Insert(des);

// avoid INVALID = 0
for(int i = 1; i < 27; ++i)
{
    VR vr = (VR::VRType)(1 << i);
    Tag t = FindTagFromVR( pubdict, vr );
    if( vr != VR::UN && vr != VR::SQ )
    {
        assert( t != Tag(0xffff, 0xffff) );
        gdcmm::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
gdcmm::UIDGenerator uid;
gdcmm::DataElement de( Tag(0x8, 0x18) ); // SOP Instance UID
de.SetVR( VR::UI );
const char *u = uid.Generate();
de.SetByteValue( u, (uint32_t)strlen(u) );
ds.Insert( de );

de.SetTag( Tag(0x8, 0x16) ); // SOP Class UID
de.SetVR( VR::UI );
gdcmm::MediaStorage ms( gdcmm::MediaStorage::RawDataStorage
    );
de.SetByteValue( ms.GetString(), (uint32_t)strlen(ms.GetString()));
ds.Insert( de );

gdcmm::FileMetaInformation &fmi = f.GetHeader();
//fmi.SetDataSetTransferSyntax( gdcmm::TransferSyntax::ImplicitVRLittleEndian );
fmi.SetDataSetTransferSyntax(
    gdcmm::TransferSyntax::ExplicitVRLittleEndian );

w.SetCheckFileMetaInformation( true );
w.SetFileName( outfilename );
if (!w.Write() )
{

```

```

        return 1;
    }

    return 0;
}

```

29.75 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://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 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.GetFilenames();

        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;
    }
}

```

29.76 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);

    gdc::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";
        gdc::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;
}

```

29.77 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;
}

```

29.78 GenFakelIdentifyFile.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmGlobal.h"
#include "gdcmDummyValueGenerator.h"
#include "gdcmMediaStorage.h"

```

```

#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmAttribute.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmDict.h"
#include "gdcmDictEntry.h"
#include "gdcmDicts.h"
#include "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
    {
        gdcmm::Attribute<0x0008,0x0000> at1 = { 0 }; // This element has no
            reason to be 'anonymized'...
        nds.Insert( at1.GetAsDataElement() );
        gdcmm::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 gdcmm::Tag;
    using gdcmm::VR;
    const char *outfilename = argv[1];

    std::vector<gdcmm::Tag> balcptags =
        gdcmm::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes
            ();

    gdcmm::Writer w;
    gdcmm::File &f = w.GetFile();
    gdcmm::DataSet &ds = f.GetDataSet();

    // Add attribute that need to be anonymized:
    std::vector<gdcmm::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 gdcmm::Global &g = gdcmm::Global::GetInstance();
    static const gdcmm::Dicts &dicts = g.GetDicts();
    static const gdcmm::Dict &pubdict = dicts.GetPublicDict();

    using gdcmm::Dict;
    Dict::ConstIterator dictit = pubdict.Begin();
    for(; dictit != pubdict.End(); ++dictit)
    {
        const gdcmm::Tag &dicttag = dictit->first;
        if( dicttag == Tag(0x6e65,0x6146) ) break;
        //const gdcmm::DictEntry &dictentry = dictit->second;
        ds.Insert( CreateFakeElement( dicttag, false ) );
    }
    ds.Remove( gdcmm::Tag(0x400,0x500) );
    ds.Remove( gdcmm::Tag(0x12,0x62) );
    ds.Remove( gdcmm::Tag(0x12,0x63) );

    // Make sure to override any UID stuff
    gdcmm::UIDGenerator uid;
    gdcmm::DataElement de( Tag(0x8,0x18) ); // SOP Instance UID
    de.SetVR( VR::UI );
    const char *u = uid.Generate();
    de.SetByteValue( u, (uint32_t)strlen(u) );
    //ds.Insert( de );
    ds.Replace( de );

    de.SetTag( Tag(0x8,0x16) ); // SOP Class UID
    de.SetVR( VR::UI );
    gdcmm::MediaStorage ms( gdcmm::MediaStorage::RawDataStorage
        );
    de.SetByteValue( ms.GetString(), (uint32_t)strlen(ms.GetString()) );
    ds.Replace( de ); // replace !

    gdcmm::FileMetaInformation &fmi = f.GetHeader();
    //fmi.SetDataSetTransferSyntax( gdcmm::TransferSyntax::ImplicitVRLittleEndian );
    fmi.SetDataSetTransferSyntax(
        gdcmm::TransferSyntax::ExplicitVRLittleEndian );

```

```

w.SetCheckFileMetaInformation( true );
w.SetFileName( outfilename );
if ( !w.Write() )
{
    return 1;
}

return 0;
}

```

29.79 GenFakelImage.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 "gdcmlImage.h"
#include "gdcmlImageWriter.h"
#include "gdcmlFileDerivation.h"
#include "gdcmlUIDGenerator.h"
// #include "gdcmlImageChangePhotometricInterpretation.h"

/*
 * This example shows two things:
 * 1. How to create an image ex-nihilo
 * 2. How to use the gdcmlFileDerivation 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
    gdcml::SmartPointer<gdcml::Image> im = new
        gdcml::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[2] = b;
                //ybr[0] = r;
                //ybr[1] = g;
                //ybr[2] = b;

                ybr2[0] = r;
                ybr2[1] = g;
                ybr2[2] = b;
                //gdcml::ImageChangePhotometricInterpretation::YBR2RGB(rgb, ybr);
                //gdcml::ImageChangePhotometricInterpretation::RGB2YBR(ybr2, rgb);
                *p++ = (char)ybr2[0];
                *p++ = (char)ybr2[1];
                *p++ = (char)ybr2[2];
            }
}

```

```

im->SetNumberOfDimensions( 2 );
im->SetDimension(0, 256 );
im->SetDimension(1, 256 );

im->GetPixelFormat().SetSamplesPerPixel(3);
//im->SetPhotometricInterpretation( gdcm::PhotometricInterpretation::RGB );
im->SetPhotometricInterpretation(
    gdcm::PhotometricInterpretation::YBR_FULL );

unsigned long l = im->GetBufferLength();
if( l != 256 * 256 * 3 )
{
    return 1;
}
gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
pixeldata.SetByteValue( buffer, (uint32_t)l );
delete[] buffer;
im->SetDataElement( pixeldata );

gdcm::UIDGenerator uid; // helper for uid generation

gdcm::SmartPointer<gdcm::File> file = new
    gdcm::File; // empty file

// Step 2: DERIVED object
gdcm::FileDerivation fd;
// For the purpose of this exercise we will pretend that this image is referencing
// two source image (we need to generate fake UID for that).
const char ReferencedSOPClassUID[] = "1.2.840.10008.5.1.4.1.1.7"; // Secondary Capture
fd.AddReference( ReferencedSOPClassUID, uid.Generate() );
fd.AddReference( ReferencedSOPClassUID, uid.Generate() );

// Again for the purpose of the exercise we will pretend that the image is a
// multiplanar reformat (MPR):
// CID 7202 Source Image Purposes of Reference
// { "DCM",121322,"Source image for image processing operation"},
fd.SetPurposeOfReferenceCodeSequenceCodeValue( 121322 );
// CID 7203 Image Derivation
// { "DCM",113072,"Multiplanar reformatting" },
fd.SetDerivationCodeSequenceCodeValue( 113072 );
fd.SetFile( *file );
// If all Code Value are ok the filter will execute properly
if( !fd.Derive() )
{
    std::cerr << "Sorry could not derive using input info" << std::endl;
    return 1;
}

// We pass both :
// 1. the fake generated image
// 2. the 'DERIVED' dataset object
// to the writer.
gdcm::ImageWriter w;
w.SetImage( *im );
w.SetFile( fd.GetFile() );

// Set the filename:
w.SetFileName( "ybr2.dcm" );
if( !w.Write() )
{
    return 1;
}

return 0;
}

```

29.80 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 nitens = 1000;
    nitens += std::numeric_limits<uint32_t>::max();
    for(unsigned int idx = 0; idx < nitens; ++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);
}

```

```

gdcmm::Writer w;
w.SetFile( file );
//w.SetCheckFileMetaInformation( true );
w.SetFileName( outfilename );
if ( !w.Write() )
{
    return 1;
}

return 0;
}

```

29.81 GenSeqs.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 "gdcmmItem.h"
#include "gdcmmImageReader.h"
#include "gdcmmSequenceOfItems.h"
#include "gdcmmFile.h"
#include "gdcmmTag.h"

/*
 * This example is used to generate the file:
 *
 * gdcmmConformanceTests/SequenceWithUndefinedLengthNotConvertibleToDefinedLength.dcm
 *
 * There is a flaw in the DICOM design where it is assumed that Sequence can be
 * either represented as undefined length or defined length. This should work
 * in most case, but the undefined length is a little more general and can
 * store sequence of items that a defined length cannot.
 * Deflated syntax was used in this case since this synthetic example can be
 * nicely compressed using this transfer syntax.
 *
 * Warning: do not try to compute the group length elements !
 * Warning: You may need a 64bits machine for this example to work.
 */
int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    gdcmm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcmm::File &file = reader.GetFile();
    gdcmm::DataSet &ds = file.GetDataSet();

    //const unsigned int nitems = 1000;
    const unsigned int ptr_len = 42; /*94967296 / nitems; */
    //assert( ptr_len == 42949672 );
    char *ptr = new char[ptr_len];
    memset(ptr,0,ptr_len);

    // Create a Sequence

```



```

gdcmm::SmartPointer<gdcmm::SequenceOfItems> sq = new
    gdcmm::SequenceOfItems();
sq->SetLengthToUndefined();

const char owner_str[] = "GDCM CONFORMANCE TESTS";
gdcmm::DataElement owner( gdcmm::Tag(0x4d4d, 0x10) );
owner.SetByteValue( owner_str, (uint32_t)strlen( owner_str ) );
owner.SetVR( gdcmm::VR::LO );

for( unsigned int idx = 0; idx < 10/* nitems*/; ++idx)
{
    // Create a dataelement
    gdcmm::DataElement de( gdcmm::Tag(0x4d4d, 0x1002) );
    de.SetByteValue( ptr, ptr_len );
    de.SetVR( gdcmm::VR::OB );

    // Create an item
    gdcmm::Item it;
    it.SetVLToUndefined();
    gdcmm::DataSet &nds = it.GetNestedDataSet();
    nds.Insert( owner );
    nds.Insert( de );

    sq->AddItem( it );
}

// Insert sequence into data set
gdcmm::DataElement des( gdcmm::Tag(0x4d4d, 0x1001) );
des.SetVR( gdcmm::VR::SQ );
des.SetValue( *sq );
des.SetVLToUndefined();

ds.Insert( owner );
ds.Insert( des );

gdcmm::Writer w;
w.SetFile( file );
//w.SetCheckFileMetaInformation( true );
w.SetFileName( outfilename );
if ( !w.Write() )
{
    return 1;
}

return 0;
}

```

29.82 GetArray.cs

```

/*=====
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/GetArray.exe gdcmmData/012345.002.050.dcm
 */
using System;
using gdcmm;

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;
}
}

```

29.83 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;
}

```

29.84 GetPortionCSAHeader.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #   This software is distributed WITHOUT ANY WARRANTY; without even
10 #   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #   PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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  !
56  print diffgraddir
57
58  # repeat for t2 if you like it:
59  if ds.FindDataElement( t2 ):
60      csa_t2.LoadFromDataElement( ds.GetDataElement( t2 ) )
61      # print csa_t2
62
63  gdt = csa_t2.GetCSAElementByName( "GradientDelayTime" )
64  print gdt
65
66  bv = gdt.GetByteValue();
67  #print bv
68  str = bv.GetPointer()
69  print str.split("\\")

```

29.85 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;
}

```

29.86 GetSubSequenceData.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 "gdcmlImage.h"
#include "gdcmlImageWriter.h"
#include "gdcmlDataElement.h"
#include "gdcmlPrivateTag.h"
#include "gdcmlUIDGenerator.h"

#include <iostream>
#include <string>

#include <map>

/*
 * This example will extract the Movie from the private group of
 * GEMS_Ultrasound_MovieGroup_001 See Attribute
 * (7fe1,60,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 gdcml;
    const char *filename = argv[1];
    gdcml::Reader reader;
    reader.SetFileName( filename );
    reader.Read();

    gdcml::File &file = reader.GetFile();
    gdcml::DataSet &ds = file.GetDataSet();
    const PrivateTag tseq(0x7fe1,0x1,"GEMS_Ultrasound_MovieGroup_001");

    if( !ds.FindDataElement( tseq ) ) return 1;
    const DataElement& seq = ds.GetDataElement( tseq );

    SmartPointer<SequenceOfItems> sqi = seq.
        GetValueAsSQ();
    assert( sqi->GetNumberOfItems() == 1 );
    Item &item = sqi->GetItem(1);
    DataSet &subds = item.GetNestedDataSet();

    const PrivateTag tseq1(0x7fe1,0x10,"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(0x7fe1,0x20,"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 1;
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 << "E1= " << 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 1;
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 1;
    const DataElement& de8 = subds4.GetDataElement( tseq8 );
    Element<VR::UL,VM::VM1> ldimz;
    ldimz.SetFromDataElement( de8 );
    dimz += ldimz.GetValue();
    if( !subds4.FindDataElement( tseq4 ) ) return 1;
    const DataElement& seq4 = subds4.GetDataElement( tseq4 );
    if( !subds4.FindDataElement( tseq5 ) ) return 1;
    const DataElement& seq5 = subds4.GetDataElement( tseq5 );

    //      std::cout << seq4 << std::endl;
    //      std::cout << seq5 << std::endl;

    const ByteValue *bv4 = seq4.GetByteValue();
    (void)bv4;
#ifdef 0
    {
        std::ofstream out( "/tmp/mo4", std::ios::binary );
        out.write( bv4->GetPointer(), bv4->GetLength() );
        out.close();
    }
#endif
    const ByteValue *bv5 = seq5.GetByteValue();
#ifdef 0
    {

```



```

        std::ofstream out( "/tmp/mo5", std::ios::binary );
        out.write( bv5->GetPointer(), bv5->GetLength());
        out.close();
    }
#endif

    std::cout << bv5->GetLength() << std::endl;
    imbuffer.insert( imbuffer.begin(), bv5->GetPointer(), bv5->
        GetPointer() + bv5->GetLength() );
    }
    DataElement fakedata;
    fakedata.SetByteValue( &imbuffer[0], (uint32_t)imbuffer.size() );

    gdcmm::SmartPointer<gdcmm::Image> im = new
        gdcmm::Image;
    im->SetNumberOfDimensions( 3 );

    im->SetDimension(0, dimx );
    im->SetDimension(1, dimy );
    im->SetDimension(2, dimz );
    size_t l1 = imbuffer.size();
    (void)l1;
    size_t l2 = im->GetBufferLength();
    (void)l2;
    assert( im->GetBufferLength() == imbuffer.size() );
    im->SetPhotometricInterpretation(
        gdcmm::PhotometricInterpretation::MONOCHROME2 );

    im->SetDataElement( fakedata );

    gdcmm::ImageWriter w;
    w.SetImage( *im );
    DataSet &dataset = w.GetFile().GetDataSet();

    gdcmm::UIDGenerator uid;
    gdcmm::DataElement de( Tag(0x8,0x18) ); // SOP Instance UID
    de.SetVR( VR::UI );
    const char *u = uid.Generate();
    de.SetByteValue( u, (uint32_t)strlen(u) );
    //ds.Insert( de );
    dataset.Replace( de );

    de.SetTag( Tag(0x8,0x16) ); // SOP Class UID
    de.SetVR( VR::UI );
    gdcmm::MediaStorage ms(
        gdcmm::MediaStorage::MultiframeGrayscaleByteSecondaryCaptureImageStorage
    );
    de.SetByteValue( ms.GetString(), (uint32_t)strlen(ms.
        GetString()) );
    dataset.Replace( de ); // replace !

    w.SetFileName( "outvid.dcm" );
    if( !w.Write() )
    {
        return 1;
    }

    return 0;
}

```

29.87 headsq2dcm.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.
8 #
9 #   This software is distributed WITHOUT ANY WARRANTY; without even
10 #   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #   PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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()

```

29.88 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(vtkgdc.vtkImageData imgin)
{
    HandleRef rawCppThis = imgin.GetCppThis();
    Kitware.VTK.vtkImageData imgout = new Kitware.VTK.vtkImageData( rawCppThis.Handle, false, false);
    return imgout;
}

static vtkgdc.vtkImageData ConnectActivizToSWIG(Kitware.VTK.vtkImageData imgin)
{
    HandleRef rawCppThis = imgin.GetCppThis();
    vtkgdc.vtkImageData imgout = new vtkgdc.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

    vtkgdc.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;
}
}

```

29.89 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() == vtkgdcm.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;
}

```

29.90 HelloActiviz3.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
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.WriteLine(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;
    }
}

```

29.91 HelloActiviz4.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/

```

```

=====*/
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;
    }
}

```

29.92 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;
}

```

29.93 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=gdcmm.jar javac ../../gdcmm/Examples/Java/HelloSimple.java -d .
 *
 * Usage:
 * $ LD_LIBRARY_PATH=. CLASSPATH=gdcmm.jar:. java HelloSimple gdcmmData/012345.002.050.dcm
 */
import gdcmm.*;

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 );
    }
}

```

29.94 HelloVizWorld.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
/*
 * Basic example for dealing with a DICOM file that contains an Image
 * (read: Pixel Data element)
 */

#include "gdcmmImageReader.h"
#include "gdcmmImageWriter.h"
#include "gdcmmImage.h"
#include "gdcmmPhotometricInterpretation.h"

#include <iostream>

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    // Instantiate the image reader:
    gdcmm::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( outfilename );
//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;
}

```

29.95 HelloVTKWorld.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
using vtkgdcm;

/*
 * This test only test the SWIG/VTK part, you do not need Activiz
 */
public class HelloVTKWorld
{
    public static int Main(string[] args)
    {
        {
            string filename = args[0];
            vtkGDCMImageReader reader = vtkGDCMImageReader.
                New();
            reader.SetFileName( filename );
            reader.Update();

            vtkMedicalImageProperties prop = reader.GetMedicalImageProperties();
            System.Console.WriteLine( prop.GetPatientName() ); //

            if( reader.GetImageFormat() == vtkgdcm.vtkgdcm.VTK_LUMINANCE ) // MONOCHROME2

```

```

    {
        System.Console.WriteLine( "Image is MONOCHROME2" ); //
    }

    // Just for fun, invert the direction cosines, output should reflect that:
    vtkMatrix4x4 dircos = reader.GetDirectionCosines();
    dircos.Invert();

    string outfilename = args[1];
    vtkGDCMImageWriter writer = vtkGDCMImageWriter.
        New();
    writer.SetMedicalImageProperties( reader.GetMedicalImageProperties() );
    writer.SetDirectionCosines( dircos );
    writer.SetShift( reader.GetShift() );
    writer.SetScale( reader.GetScale() );
    writer.SetImageFormat( reader.GetImageFormat() );
    writer.SetFileName( outfilename );
    //writer.SetInputConnection( reader.GetOutputPort() ); // new
    writer.SetInput( reader.GetOutput() ); // old
    writer.Write();

    return 0;
}
}

```

29.96 HelloVTKWorld.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
// We are required to call the package 'vtk' eventhough I (MM) would have preferred
// an import statement along the line of:
// import vtkgdc.*;
import vtk.*;

/*
 * Compilation:
 * CLASSPATH=vtkgdc.jar:/usr/share/java/vtk.jar javac HelloVTKWorld.java
 *
 * Usage:
 * LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/jni:. CLASSPATH=/usr/share/java/
 *   vtk.jar:vtkgdc.jar:gdcm.jar:. java HelloVTKWorld gdcmData/012345.002.050.dcm bla.dcm
 */
public class HelloVTKWorld
{
    static {
        System.loadLibrary("vtkCommonJava");
        System.loadLibrary("vtkFilteringJava");
        System.loadLibrary("vtkIOJava");
        System.loadLibrary("vtkImagingJava");
        System.loadLibrary("vtkGraphicsJava");
        System.loadLibrary("vtkgdcJava");
        try {
            System.loadLibrary("vtkRenderingJava");
        } catch (Throwable e) {
            System.out.println("cannot load vtkHybrid, skipping...");
        }
        try {
            System.loadLibrary("vtkHybridJava");
        } catch (Throwable e) {
            System.out.println("cannot load vtkHybrid, skipping...");
        }
        try {
            System.loadLibrary("vtkVolumeRenderingJava");
        } catch (Throwable e) {
            System.out.println("cannot load vtkVolumeRendering, skipping...");
        }
    }
}

```

```

    }
}

public static void main(String[] args)
{
    String filename = args[0];
    vtkGDCMImageReader reader = new vtkGDCMImageReader();
    reader.SetFileName( filename );
    reader.Update();

    vtkMedicalImageProperties prop = reader.GetMedicalImageProperties();
    System.out.println( prop.GetPatientName() ); //

//    if( reader.GetImageFormat() == vtkgdcm.vtkgdcm.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 );
}
}

```

29.97 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 vtkgdcm;

/*
 * 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();

        vtkVoxel16Reader reader = vtkVoxel16Reader.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.SetInput( reader.GetOutput() );
cast.SetOutputScalarTypeToUnsignedChar();

// By default this is creating a Multiframe Grayscale Word Secondary Capture Image Storage
vtkGDCMImageWriter writer = vtkGDCMImageWriter.
    New();
writer.SetFileName( "headsq.dcm" );
writer.SetInput( reader.GetOutput() );
// cast -> Multiframe Grayscale Byte Secondary Capture Image Storage
// writer.SetInput( cast.GetOutput() );
writer.SetFileDimensionality( 3 );
writer.Write();

return 0;
}
}

```

29.98 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.
 */

```

29.99 HelloWorld.py

```

1 #####
2 #
3 # Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 # Copyright (c) 2006-2011 Mathieu Malaterre
6 # All rights reserved.
7 # See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 # This software is distributed WITHOUT ANY WARRANTY; without even
10 # the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 # PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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     # Instanciate 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))

```

29.100 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;
}

```

29.101 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 midpoinxy = (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/l, 1 ROIContourSequence
    gdcm::Tag tag(0x3006,0x0039);

    const gdcm::DataElement &roicsq = ds.GetDataElement( tag );
    gdcm::SmartPointer<gdcm::SequenceOfItems> sqi = roicsq.
        GetValueAsSQ();
    //sqi->SetNumberOfItems( 1 );
    const gdcm::Item &item = sqi->GetItem(1); // Item start at #1
    const gdcm::DataSet& nestedds = item.GetNestedDataSet();

    gdcm::Tag tcsq(0x3006,0x0040);
    if( !nestedds.FindDataElement( tcsq ) )
    {
        return 0;
    }
    const gdcm::DataElement& csq = nestedds.GetDataElement( tcsq );
    gdcm::SmartPointer<gdcm::SequenceOfItems> sqi2 = csq.
        GetValueAsSQ();
    if( !sqi2 || !sqi2->GetNumberOfItems() )
    {
        return 0;
    }
    //unsigned int nitems = sqi2->GetNumberOfItems();
    gdcm::Item &item2 = sqi2->GetItem(1); // Item start at #1

    gdcm::DataSet& nestedds2 = item2.GetNestedDataSet();
    //item2.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;
}

```

29.102 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;
}

```

29.103 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;
}

```

29.104 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;
    }
}

```

29.105 ManipulateFile.py

```

1 #####
2 #
3 # Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 # Copyright (c) 2006-2011 Mathieu Malaterre
6 # All rights reserved.
7 # See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 # This software is distributed WITHOUT ANY WARRANTY; without even
10 # the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 # PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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)

```

29.106 ManipulateSequence.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #       This software is distributed WITHOUT ANY WARRANTY; without even
10 #       the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #       PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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.SetByteValue( mystr, gdcm.VL( len(mystr) ) )
68
69     w = gdcm.Writer()
70     w.SetFile( f )
71     w.SetFileName( file2 )

```

```

72 if not w.Write():
73     sys.exit(1)

```

29.107 MergeFile.py

```

1 #####
2 #
3 # Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 # Copyright (c) 2006-2011 Mathieu Malaterre
6 # All rights reserved.
7 # See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 # This software is distributed WITHOUT ANY WARRANTY; without even
10 # the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 # PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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)

```

29.108 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;
}

```


29.109 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;
}
}

```

29.110 MIPViewer.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
import vtk.*;
import gdcm.*;
import java.io.File;
import java.awt.Canvas;

/*
 * Compilation:
 * CLASSPATH=vtkgdcm.jar:/usr/share/java/vtk.jar javac MIPViewer.java
 *
 * Usage:
 * LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/jni:. CLASSPATH=/usr/share/java/
   vtk.jar:vtkgdcm.jar:gdcm.jar:. java MIPViewer BRAINX
 */
public class MIPViewer extends Canvas
{
    static {
        // VTK
        System.loadLibrary("vtkCommonJava");
        System.loadLibrary("vtkFilteringJava");
        System.loadLibrary("vtkIOJava");
        System.loadLibrary("vtkImagingJava");
        System.loadLibrary("vtkGraphicsJava");
        System.loadLibrary("vtkRenderingJava");
        System.loadLibrary("vtkVolumeRenderingJava"); // vtkSmartVolumeMapper
    }
}

```

```

    System.loadLibrary("vtkWidgetsJava"); // vtkBoxWidget
    // VTK-GDCM
    System.loadLibrary("vtkgdcmJava");
}

static FilenamesType fns = new FilenamesType();

protected native int Lock();

protected native int UnLock();

public static void process(String path)
{
    fns.add( path );
}

// Process only files under dir
public static void visitAllFiles(File dir)
{
    if (dir.isDirectory())
    {
        String[] children = dir.list();
        for (int i=0; i<children.length; i++)
        {
            visitAllFiles(new File(dir, children[i]));
        }
    }
    else
    {
        process(dir.getPath());
    }
}

public static void main(String[] args) throws Exception
{
    String dirname = args[0];
    if( !PosixEmulation.FileIsDirectory( dirname ) )
    {
        return;
    }

    File dir = new File(dirname);
    visitAllFiles(dir);

    IPPSorter ipp = new IPPSorter();
    ipp.SetComputeZSpacing( true );
    ipp.SetZSpacingTolerance( 1e-3 );
    boolean b = ipp.Sort( fns );
    if(!b)
    {
        throw new Exception("Could not scan");
    }
    double ippzspacing = ipp.GetZSpacing();

    FilenamesType sorted = ipp.GetFilenames();
    vtkStringArray files = new vtkStringArray();
    long nfiles = sorted.size();
    //for( String f : sorted )
    for (int i = 0; i < nfiles; i++) {
        String f = sorted.get(i);
        files.InsertNextValue( f );
    }
    vtkGDCMImageReader reader = new vtkGDCMImageReader();
    reader.SetFileNames( files );
    reader.Update(); // get spacing value

    double[] spacing = reader.GetOutput().GetSpacing();

    vtkImageChangeInformation change = new vtkImageChangeInformation();
    change.SetInputConnection( reader.GetOutputPort() );
    change.SetOutputSpacing( spacing[0], spacing[1], ippzspacing );

    // Create our volume and mapper
    vtkVolume volume = new vtkVolume();
    vtkSmartVolumeMapper mapper = new vtkSmartVolumeMapper();

    vtkRenderWindowInteractor iren = new vtkRenderWindowInteractor();

    // Add a box widget if the clip option was selected
    vtkBoxWidget box = new vtkBoxWidget();
    box.SetInteractor(iren);

```

```

box.SetPlaceFactor(1.01);
box.SetInput(change.GetOutput());

//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();
}
}

```

29.111 MpegVideoInfo.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

```

Copyright (c) 2006-2011 Mathieu Malaterre
 All rights reserved.
 See Copyright.txt or <http://gdc.sourceforge.net/Copyright.html> for details.

This software is distributed 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.informpeg?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 bytesRead = 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 & 0xFFFF0000) >> 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;
}
}

```

29.112 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=vtkgdcm.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:vtkgdcm.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("vtkgdcmJava");
    }

    static FilenamesType fns = new FilenamesType();

    public static void process(String path)
    {
        fns.add( path );
    }

    // Process only files under dir
    public static void visitAllFiles(File dir)
    {
        if (dir.isDirectory())

```

```

    {
        String[] children = dir.list();
        for (int i=0; i<children.length; i++)
        {
            visitAllFiles(new File(dir, children[i]));
        }
    }
    else
    {
        process(dir.getPath());
    }
}

public static void main(String[] args) throws Exception
{
    String dirname = args[0];
    if( !PosixEmulation.FileIsDirectory( dirname ) )
    {
        return;
    }

    File dir = new File(dirname);
    visitAllFiles(dir);

    IPPSorter ipp = new IPPSorter();
    ipp.SetComputeZSpacing( true );
    ipp.SetZSpacingTolerance( 1e-3 );
    boolean b = ipp.Sort( fns );
    if(!b)
    {
        throw new Exception("Could not scan");
    }
    double ippzspacing = ipp.GetZSpacing();

    FilenamesType sorted = ipp.GetFilenames();
    vtkStringArray files = new vtkStringArray();
    long nfiles = sorted.size();
    //for( String f : sorted )
    for (int i = 0; i < nfiles; i++) {
        String f = sorted.get(i);
        files.InsertNextValue( f );
    }
    vtkGDCMImageReader reader = new vtkGDCMImageReader();
    reader.SetFileNames( files );
    reader.Update(); // get spacing value

    double[] spacing = reader.GetOutput().GetSpacing();

    vtkImageChangeInformation change = new vtkImageChangeInformation();
    change.SetInputConnection( reader.GetOutputPort() );
    change.SetOutputSpacing( spacing[0], spacing[1], ippzspacing );

    // A simple vtkInteractorStyleImage example for
    // 3D image viewing with the vtkImageResliceMapper.
    //
    // Drag Left mouse button to window/level
    // Shift-Left drag to rotate (oblique slice)
    // Shift-Middle drag to slice through image
    // OR Ctrl-Right drag to slice through image

    // Create the RenderWindow, Renderer
    vtkRenderer ren1 = new vtkRenderer();
    vtkRenderWindow renWin = new vtkRenderWindow();
    renWin.AddRenderer(ren1);

    vtkImageResliceMapper im = new vtkImageResliceMapper();
    im.SetInputConnection(change.GetOutputPort());
    im.SliceFacesCameraOn();
    im.SliceAtFocalPointOn();
    im.BorderOff();

    vtkImageProperty ip = new vtkImageProperty();
    ip.SetColorWindow(2000);
    ip.SetColorLevel(1000);
    ip.SetAmbient(0.0);
    ip.SetDiffuse(1.0);
    ip.SetOpacity(1.0);
    ip.SetInterpolationTypeToLinear();

    vtkImageSlice ia = new vtkImageSlice();
    ia.SetMapper(im);

```

```

        ia.SetProperty(ip);

        ren1.AddViewProp(ia);
        ren1.SetBackground(0.1,0.2,0.4);
        renWin.SetSize(300,300);

        vtkRenderWindowInteractor iren = new vtkRenderWindowInteractor();
        vtkInteractorStyleImage style = new vtkInteractorStyleImage();
        style.SetInteractionModeToImage3D();
        iren.SetInteractorStyle(style);
        renWin.SetInteractor(iren);

        // render the image
        renWin.Render();
        vtkCamera cam1 = ren1.GetActiveCamera();
        cam1.ParallelProjectionOn();
        ren1.ResetCameraClippingRange();
        renWin.Render();

        iren.Start();
    }
}

```

29.113 MPRViewer2.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
import vtk.*;
import gdcml.*;
import java.io.File;

/*
 * Compilation:
 * CLASSPATH=vtkgdcml.jar:/usr/share/java/vtk.jar javac 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:vtkgdcml.jar:gdcml.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("vtkgdcmlJava");
    }

    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 wextent[] = image.GetWholeExtent();
    int xmin = wextent[0];
    int xmax = wextent[1];
    int ymin = wextent[2];
    int ymax = wextent[3];
    int zmin = wextent[4];
    int zmax = wextent[5];
}

```

```

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], ippszspacing );
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.SetInput( change.GetOutput() );
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.SetInput( change.GetOutput() );
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.SetInput( change.GetOutput() );
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 );
}

```

29.114 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.ulEnableRFSpoiling      = 0x1
sPhysioImaging.lSignal1              = 2
sPhysioImaging.lMethod1              = 2
sPhysioImaging.lSignal2              = 1
sPhysioImaging.lMethod2              = 1
sPhysioImaging.lPhases               = 21
sPhysioImaging.lRetroGatedImages     = 16
sPhysioImaging.sPhysioECG.lScanWindow = 805
sPhysioImaging.sPhysioECG.lTriggerPulses = 1
sPhysioImaging.sPhysioECG.lTriggerWindow = 5
sPhysioImaging.sPhysioECG.lArrhythmiaDetection = 1
sPhysioImaging.sPhysioECG.lCardiacGateOnThreshold = 100000
sPhysioImaging.sPhysioECG.lCardiacGateOffThreshold = 700000
sPhysioImaging.sPhysioPulse.lTriggerPulses = 1
sPhysioImaging.sPhysioPulse.lTriggerWindow = 5
sPhysioImaging.sPhysioPulse.lCardiacGateOnThreshold = 100000
sPhysioImaging.sPhysioPulse.lCardiacGateOffThreshold = 700000
sPhysioImaging.sPhysioExt.lTriggerPulses = 1
sPhysioImaging.sPhysioExt.lTriggerWindow = 5
sPhysioImaging.sPhysioExt.lCardiacGateOnThreshold = 100000
sPhysioImaging.sPhysioExt.lCardiacGateOffThreshold = 700000
sPhysioImaging.sPhysioResp.lRespGateThreshold = 20
sPhysioImaging.sPhysioResp.lRespGatePhase = 2
sPhysioImaging.sPhysioResp.dGatingRatio = 0.3
sSpecPara.lPhaseCyclingType          = 1
sSpecPara.lPhaseEncodingType         = 1
sSpecPara.lRFExcitationBandwidth     = 1
sSpecPara.ucRemoveOversampling       = 0x1
sSpecPara.lDecouplingType            = 1
sSpecPara.lNOEType                   = 1
sSpecPara.lExcitationType            = 1
sSpecPara.lSpectralSuppression        = 1
sDiffusion.ulMode                    = 0x1
sAngio.sFlowArray.asElm[0].nVelocity = 100
sAngio.sFlowArray.asElm[0].nDir      = 0x4
sAngio.sFlowArray.lSize              = 1
sAngio.ucPCFlowMode                  = 0x2
sAngio.ucTOFInflow                   = 0x4
sAngio.ucRephasedImage               = 0x1
sAngio.ucPhaseImage                  = 0x1
sEllipticalFilter.ucMode              = 0x1

```

```
sPat.lAccelFactPE = 1
sPat.lAccelFact3D = 1
sPat.ucPATMode = 0x1
sPat.ucRefScanMode = 0x1
ucAutoMovie = 0x1
ucDisableChangeStoreImages = 0x1
ucReconstructionMode = 0x1
ucPHAPSMode = 0x1
ucDixon = 0x1
lAverages = 2
adFlipAngleDegree[0] = 30
lScanTimeSec = 103
lTotalScanTimeSec = 112
dRefSNR = 165404.1473
dRefSNR_VOI = 165404.1473
tdefaultEVAProt = "%SiemensEvaDefProt%\Inline\Inline.evp"
tcurrentEVAProt = "%CURRENTEVAPROT%\EVA2A5.tmp"
sCOIL_SELECT_MEAS.asList[0].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[0].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[0].sCoilElementID.tElement = "PP6"
sCOIL_SELECT_MEAS.asList[0].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[0].lRxChannelConnected = 1
sCOIL_SELECT_MEAS.asList[1].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[1].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[1].sCoilElementID.tElement = "PP5"
sCOIL_SELECT_MEAS.asList[1].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[1].lRxChannelConnected = 1
sCOIL_SELECT_MEAS.asList[2].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[2].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[2].sCoilElementID.tElement = "PP3"
sCOIL_SELECT_MEAS.asList[2].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[2].lRxChannelConnected = 2
sCOIL_SELECT_MEAS.asList[3].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[3].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[3].sCoilElementID.tElement = "PP4"
sCOIL_SELECT_MEAS.asList[3].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[3].lRxChannelConnected = 3
sCOIL_SELECT_MEAS.asList[4].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[4].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[4].sCoilElementID.tElement = "PP2"
sCOIL_SELECT_MEAS.asList[4].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[4].lRxChannelConnected = 4
sCOIL_SELECT_MEAS.asList[5].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[5].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[5].sCoilElementID.tElement = "PP1"
sCOIL_SELECT_MEAS.asList[5].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[5].lRxChannelConnected = 4
sCOIL_SELECT_MEAS.asList[6].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[6].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[6].sCoilElementID.tElement = "PA6"
sCOIL_SELECT_MEAS.asList[6].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[6].lRxChannelConnected = 5
sCOIL_SELECT_MEAS.asList[7].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[7].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[7].sCoilElementID.tElement = "PA5"
sCOIL_SELECT_MEAS.asList[7].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[7].lRxChannelConnected = 5
sCOIL_SELECT_MEAS.asList[8].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[8].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[8].sCoilElementID.tElement = "PA3"
sCOIL_SELECT_MEAS.asList[8].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[8].lRxChannelConnected = 6
sCOIL_SELECT_MEAS.asList[9].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[9].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[9].sCoilElementID.tElement = "PA4"
sCOIL_SELECT_MEAS.asList[9].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[9].lRxChannelConnected = 7
sCOIL_SELECT_MEAS.asList[10].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[10].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[10].sCoilElementID.tElement = "PA2"
sCOIL_SELECT_MEAS.asList[10].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[10].lRxChannelConnected = 8
sCOIL_SELECT_MEAS.asList[11].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[11].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[11].sCoilElementID.tElement = "PA1"
sCOIL_SELECT_MEAS.asList[11].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[11].lRxChannelConnected = 8
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[0] = 0xff
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[1] = 0x76
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[2] = 0x78
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[3] = 0x87
```

```

sCOIL_SELECT_MEAS.sCOILPLUGS.auiPlugId[4] = 0x67
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[0] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[1] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[2] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[3] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[4] = 0x2
sEFISPEC.bEFIDataValid = 1
### ASCCONV END ###
',
*/

/*
 * Table of equivalence:
 *
ulVersion = 0xbee332
<=>
27 - 'MrProtocolVersion' VM 1, VR IS, SyngoDT 6, NoOfItems 6, Data '12510002'
*/

#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmCSAHeader.h"
#include "gdcmAttribute.h"
#include "gdcmGlobal.h"
#include "gdcmDicts.h"

#include <map>

#include <math.h>

int main(int argc, char *argv [])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcm::ImageReader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }

    gdcm::CSAHeader csa;
    const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

    //const gdcm::PrivateTag &t1 = csa.GetCSAImageHeaderInfoTag();
    const gdcm::PrivateTag &t2 = csa.GetCSASeriesHeaderInfoTag();

    if( ds.FindDataElement( t2 ) )
    {
        csa.LoadFromDataElement( ds.GetDataElement( t2 ) );
        //csa.Print( std::cout );
    }

    if( !csa.FindCSAElementByName( "MrProtocol" ) )
    {
        return 1;
    }
    const gdcm::CSAElement &csael = csa.GetCSAElementByName( "MrProtocol" );
    //std::cout << csael << std::endl;

    const gdcm::ByteValue *bv = csael.GetByteValue();
    if( !bv )
    {
        return 1;
    }
    std::string str(bv->GetPointer(), bv->GetLength());
    std::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 gdcmm::CSAHeaderDict &csadict =
    gdcmm::Global::GetInstance().GetDicts().
    GetCSAHeaderDict();
const gdcmm::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:

...
    <ParamFunctor."<TlmapFunctor">
    {
        <Class> "<TlmapFunctor@IceImagePostProcFunctors">

        <ParamBool."<EXECUTE"> { }
        <ParamDouble."<Flip_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 gdcmm::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;
}

return 0;
}

```

29.115 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;
}

```

29.116 NewSequence.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcM.sourceforge.net/Copyright.html for details.
8 #
9 #   This software is distributed WITHOUT ANY WARRANTY; without even
10 #   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #   PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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  #tsis = gdcmm.Tag(0x0008,0x2112) # SourceImageSequence
40
41  # Create a dataelement
42  de = gdcmm.DataElement(gdcmm.Tag(0x0010, 0x2180))
43  de.SetByteValue("Occupation", gdcmm.VL(len("Occupation")))
44  de.SetVR(gdcmm.VR(gdcmm.VR.SH))
45
46  # Create an item
47  it=gdcmm.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=gdcmm.SequenceOfItems().New()
55  sq.SetLengthToUndefined()
56  sq.AddItem(it)
57
58  # Insert sequence into data set
59  des=gdcmm.DataElement(gdcmm.Tag(0x0400,0x0550))
60  des.SetVR(gdcmm.VR(gdcmm.VR.SQ))
61  des.SetValue(sq.__ref__())
62  des.SetVLToUndefined()
63
64  ds.Insert(des)
65
66  w = gdcmm.Writer()
67  w.SetFile( f )
68  w.SetFileName( file2 )
69  if not w.Write():
70      sys.exit(1)

```

29.117 offscreenimage.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 "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;
}

```

29.118 PatchFile.cxx

This is a C++ example on how to use [gdcm::Attribute](#)

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * The image was a broken file where the Pixel Data element was 8 times too big
 * Apparently multiplying the BitsAllocated to 4 and multiplying the number of
 * frames by 2 would solve the problem
 */

```

```

* This C++ code can be used to patch the header.
*/

#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmWriter.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        return 1;
    }
    const char *f = argv[1];
    const char *out = argv[2];
    gdcm::Reader r;
    r.SetFileName( f );
    if( !r.Read() )
    {
        return 1;
    }

    gdcm::File &file = r.GetFile();
    gdcm::DataSet& ds = file.GetDataSet();
    // (0028,0100) US 16          # 2, 1 BitsAllocated
    // (0028,0101) US 16          # 2, 1 BitsStored
    // (0028,0102) US 15          # 2, 1 HighBit
    //
    {
        gdcm::Attribute<0x28,0x100> at;
        at.SetFromDataElement( ds.GetDataElement( at.
            GetTag() ) );
        if( at.GetValue() != 8 )
        {
            return 1;
        }
        at.SetValue( 32 );
        ds.Replace( at.GetAsDataElement() );
    }
    {
        gdcm::Attribute<0x28,0x101> at;
        at.SetFromDataElement( ds.GetDataElement( at.
            GetTag() ) );
        if( at.GetValue() != 8 )
        {
            return 1;
        }
        at.SetValue( 32 );
        ds.Replace( at.GetAsDataElement() );
    }
    {
        gdcm::Attribute<0x28,0x102> at;
        at.SetFromDataElement( ds.GetDataElement( at.
            GetTag() ) );
        if( at.GetValue() != 7 )
        {
            return 1;
        }
        at.SetValue( 31 );
        ds.Replace( at.GetAsDataElement() );
    }
    // (0028,0008) IS [56]          # 2, 1 NumberOfFrames

    {
        gdcm::Attribute<0x28,0x8> at;
        at.SetFromDataElement( ds.GetDataElement( at.
            GetTag() ) );
        at.SetValue( at.GetValue() * 2 );
        ds.Replace( at.GetAsDataElement() );
    }

    gdcm::Writer w;
    w.SetFile( file );
    w.SetCheckFileMetaInformation( false );
    w.SetFileName( out );
    if( !w.Write() )
    {
        return 1;
    }
}

```

```

// Now let's see if we can read it as an image:
gdcm::ImageReader ir;
ir.SetFileName( out );
if(!ir.Read())
{
    return 1;
}
gdcm::Image &image = ir.GetImage();
unsigned long len = image.GetBufferLength();
const gdcm::ByteValue *bv = ir.GetFile().GetDataSet().
    GetDataElement( gdcm::Tag(0x7fe0,0x0010) ).GetByteValue();
if( !bv || len != bv->GetLength() )
{
    return 1;
}
std::cout << bv->GetLength() << " " << len << std::endl;

std::cout << "Sucess to rewrite image !" << std::endl;
image.Print( std::cout );
return 0;
}

```

29.119 PhilipsPrivateRescaleInterceptSlope.py

```

1 #####
2 #
3 # Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 # Copyright (c) 2006-2011 Mathieu Malaterre
6 # All rights reserved.
7 # See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 # This software is distributed WITHOUT ANY WARRANTY; without even
10 # the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 # PURPOSE. See the above copyright notice for more information.
12 #
13 #####
14
15 """
16 Usage:
17
18 python
19 """
20
21 import gdcm
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 = gdcm.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 = gdcm.PrivateTag(0x2005,0x09,"Philips MR Imaging DD 005")
42 tag2 = gdcm.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 gdcm gives us a reference
48 e11 = gdcm.DataElement( ds.GetDataElement( tag1 ) )
49 print e11
50 e12 = gdcm.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 el1.SetTag( gdcm.Tag(0x0028,0x1052) )
57 el2.SetTag( gdcm.Tag(0x0028,0x1053) )
58
59 ds.Insert( el1 )
60 ds.Insert( el2 )
61
62 w = gdcm.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"

```

29.120 PlaySound.py

```

1 #####
2 #
3 # Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 # Copyright (c) 2006-2011 Mathieu Malaterre
6 # All rights reserved.
7 # See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 # This software is distributed WITHOUT ANY WARRANTY; without even
10 # the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 # PURPOSE. See the above copyright notice for more information.
12 #
13 #####
14
15 """
16 Usage:
17
18 python PlaySound.py input.dcm
19 """
20
21 import gdcm
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()

```

29.121 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
};

byte* src = (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( strncmp( bv->GetPointer(), comprle.c_str(), comprle.size() ) == 0 )
    {
        isrle = true;
        return 1;
    }
    if( strncmp( 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;
}

```

29.122 PrivateDict.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #   This software is distributed WITHOUT ANY WARRANTY; without even
10 #   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #   PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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") )

```

29.123 PublicDict.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcM.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * Dummy example to show GDCM Dict(s) API (Part 6) + Collected Private Attributes:
 */

#include "gdcMGlobal.h"
#include "gdcMDicts.h"
#include "gdcMDict.h"
#include "gdcMCSAHeader.h"
#include "gdcMPrivateTag.h"

int main(int , char *[])
{
    const gdcM::Global& g = gdcM::Global::GetInstance(); // sum of all
        knowledge !
    const gdcM::Dicts &dicts = g.GetDicts();
    const gdcM::Dict &pub = dicts.GetPublicDict(); // Part 6

    //std::cout << pub << std::endl;

    // 3 different ways to access the same information

    // 1. From the public dict only:
    gdcM::Tag patient_name(0x10,0x10);
    const gdcM::DictEntry &entry1 = pub.GetDictEntry(patient_name);
    std::cout << entry1 << std::endl;

    // 2. From all dicts:
    const gdcM::DictEntry &entry2 = dicts.GetDictEntry(patient_name);
    std::cout << entry2 << std::endl;

    // 3. This solution is the most flexible solution as you can request using the same
    // API either a public tag or a private tag
    const char *strowner = 0;
    const gdcM::DictEntry &entry3 = dicts.GetDictEntry(patient_name,strowner);
    std::cout << entry3 << std::endl;

    // Private attributes:

    // try with a private tag now:
    const gdcM::PrivateTag &private_tag =
        gdcM::CSAHeader::GetCSAImageHeaderInfoTag();
    //std::cout << private_tag << std::endl;

```

```

const gdcm::DictEntry &entry4 = dicts.GetDictEntry(private_tag,private_tag.
    GetOwner());
std::cout << entry4 << std::endl;

// Let's pretend that private lookup is on 0x10xx elements:
gdcm::PrivateTag dummy = private_tag;
dummy.SetElement( (uint16_t)(0x1000 + dummy.GetElement()) );
const gdcm::DictEntry &entry5 = dicts.GetDictEntry(dummy,dummy.
    GetOwner());
std::cout << entry5 << std::endl;

return 0;
}

```

29.124 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;
}

```

29.125 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(
gdcmm::Tag (0x0020, 0x000e)))
            sqi->GetItem(itemused).GetDataElement (
gdcmm::Tag (0x0020, 0x000e)).GetValue().Print(strm);
        std::cout << "            SERIE UID" << strm.str() << std::endl;

        //SERIE MODALITY
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(
gdcmm::Tag (0x0008, 0x0060)))
            sqi->GetItem(itemused).GetDataElement (
gdcmm::Tag (0x0008, 0x0060)).GetValue().Print(strm);
        std::cout << "            SERIE MODALITY" << strm.str() << std::endl;

        //SERIE DESCRIPTION
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(
gdcmm::Tag (0x0008, 0x103e)))
            sqi->GetItem(itemused).GetDataElement (
gdcmm::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(
gdcmm::Tag (0x0004, 0x1430)))
            sqi->GetItem(itemused).GetDataElement (
gdcmm::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(
gdcmm::Tag (0x0004, 0x1511)))
                    sqi->GetItem(itemused).GetDataElement (
gdcmm::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(
gdcmm::Tag (0x0004, 0x1500)))
                    sqi->GetItem(itemused).GetDataElement (
gdcmm::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(
gdcmm::Tag (0x0004, 0x1430)))
                    sqi->GetItem(itemused).GetDataElement (
gdcmm::Tag (0x0004, 0x1430)).GetValue().Print(strm);

            }
        }
    }
    itemused++;
}
}
return 0;

```

```
}

```

29.126 ReadAndDumpDICOMDIR.py

```
1 #####
2 #
3 # Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 # Copyright (c) 2006-2011 Mathieu Malaterre
6 # All rights reserved.
7 # See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 # This software is distributed WITHOUT ANY WARRANTY; without even
10 # the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 # PURPOSE. See the above copyright notice for more information.
12 #
13 # File: ReadAndDumpDICOMDIR.py
14 #
15 # Author: Lukas Batteau (lbatteau gmail)
16 #
17 # This example shows how to read and dump a DICOMDIR File.
18 # Based on Tom Marynowski's (lordglub gmail) example.
19 #
20 # Usage:
21 # python ReadAndDumpDICOMDIR.py [DICOMDIR file]
22 #####
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)) !=
60         gdcm.MediaStorage.MediaStorageDirectoryStorage):
61         # File is not a DICOMDIR
62         print "This file is not a DICOMDIR (Media storage type: %s)" % (str(mediaStorage))
63         quit()
64
65     # Check Media Storage SOP Class
66     if (fileMetaInformation.FindDataElement(gdcm.Tag(0x0002, 0x0002))):
67         sopClassUid = str(fileMetaInformation.GetDataElement(gdcm.Tag(0x0002, 0x0002)).GetValue())
68         # Check SOP UID
69         if (sopClassUid != "1.2.840.10008.1.3.10"):
70             # File is not a DICOMDIR
71             print "This file is not a DICOMDIR"
72         else:
73             # 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)).
139                                GetValue())
140
141                            # SERIES
142                            while (value.strip() == "SERIES"):
143                                print value.strip()
144
145                                # Print series UID
146                                if (item.FindDataElement(gdcm.Tag(0x0020, 0x000e))):
147                                    value = str(item.GetDataElement(gdcm.Tag(0x0020, 0x000e)).
148                                    GetValue())
149                                    print value
150

```

```

149             # Print series modality
150             if (item.FindDataElement(gdcm.Tag(0x0008, 0x0060))):
151                 value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x0060))).
152             GetValue()
153             print "Modality"
154             print value
155             # Print series description
156             if (item.FindDataElement(gdcm.Tag(0x0008, 0x103e))):
157                 value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x103e))).
158             GetValue()
159             print "Description"
160             print value
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))).
166             GetValue()
167             # IMAGE
168             while (value.strip() == "IMAGE"):
169                 print value.strip()
170             # Print image UID
171             if (item.FindDataElement(gdcm.Tag(0x0004, 0x1511))):
172                 value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1511))).
173             GetValue()
174             print value
175             # Next
176             if (itemNr < sequence.GetNumberOfItems()):
177                 itemNr = itemNr + 1
178             else:
179                 break
180             item = sequence.GetItem(itemNr)
181             if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
182                 value = str(item.GetDataElement(
183                     gdcm.Tag(0x0004, 0x1430)).GetValue())
184             # Next
185             itemNr = itemNr + 1

```

29.127 ReadAndPrintAttributes.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
/*
 * This small example will show how one can read and print
 * a DICOM Attribute using different technique (by tag or by name)
 */

#include "gdcmReader.h"
#include "gdcmGlobal.h"
#include "gdcmDicts.h"
#include "gdcmDict.h"
#include "gdcmAttribute.h"
#include "gdcmStringFilter.h"

#include <iostream>

int main(int argc, char *argv[])
{
    if( argc < 2 )

```



```

    {
        std::cerr << argv[0] << " input.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];

    // Instantiate the reader:
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Could not read: " << filename << std::endl;
        return 1;
    }

    // The output of gdcm::Reader is a gdcm::File
    gdcm::File &file = reader.GetFile();

    // the dataset is the the set of element we are interested in:
    gdcm::DataSet &ds = file.GetDataSet();

    const gdcm::Global& g = gdcm::Global::GetInstance();
    const gdcm::Dicts &dicts = g.GetDicts();
    const gdcm::Dict &pubdict = dicts.GetPublicDict();

    using namespace gdcm;

    // In this example we will show why using name to lookup attribute can be
    // dangerous.
    Tag tPatientName(0x00,0x00);
    //const DictEntry &de1 =
    pubdict.GetDictEntryByName("Patient Name", tPatientName);

    std::cout << "Found: " << tPatientName << std::endl;

    // Indeed the attribute could not be found. Since DICOM 2003, Patient Name
    // has become Patient's Name.

    Tag tPatientsName;
    //const DictEntry &de2 =
    pubdict.GetDictEntryByName("Patient's Name", tPatientsName);

    std::cout << "Found: " << tPatientsName << std::endl;

    // Let's try to read an arbitrary DICOM Attribute:
    Tag tDoseGridScaling;
    //const DictEntry &de3 =
    pubdict.GetDictEntryByName("Dose Grid Scaling", tDoseGridScaling);

    std::cout << "Found: " << tDoseGridScaling << std::endl;

    if( ds.FindDataElement( tDoseGridScaling ) )
    {
        gdcm::StringFilter sf;
        sf.SetFile(file);
        std::cout << "Attribute Value as String: " << sf.ToString( tDoseGridScaling ) << std::endl;

        // Let's check the name again:
        std::pair<std::string, std::string> pss
            = sf.ToStringPair( tDoseGridScaling );
        std::cout << "Attribute Name Checked: " << pss.first << std::endl;
        std::cout << "Attribute Value (string): " << pss.second << std::endl;

        //const DataElement &dgs = ds.GetDataElement( tDoseGridScaling );

        // Let's assume for a moment we knew the tag number:
        Attribute<0x3004,0x000e> at;
        assert( at.GetTag() == tDoseGridScaling );
        at.SetFromDataSet( ds );
        // For the sake of long term maintenance, we will not write
        // that this particular attribute is stored as a double. What if
        // a user made a mistake. It is much safer to rely on GDCM internal
        // mechanism to deduce the VR::DS type (represented as a iieee double)
        Attribute<0x3004,0x000e>::ArrayType v = at.
            GetValue();
        std::cout << "DoseGridScaling=" << v << std::endl;
    }

    return 0;
}

```

29.128 ReadExplicitLengthSQIVR.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmlReader.h"
#include "gdcmlImplicitDataElement.h"
#include "gdcmlDataSet.h"
#include "gdcmlPrivateTag.h"
#include "gdcmlPrivateTag.h"
#include "gdcmlByteValue.h"
#include "gdcmlSequenceOfItems.h"

using namespace gdcml;

int main(int argc, char *argv[])
{
    if ( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcml::Reader r;
    r.SetFileName( filename );
    r.Read();

    //gdcml::PrivateTag pt(0x01,0x42,"ELSCINT1");
    //gdcml::Tag pt(0x88,0x200);
    gdcml::Tag pt(0x8,0x1140);
    DataSet &ds = r.GetFile().GetDataSet();
    const DataElement &de = ds.GetDataElement( pt );

    std::cout << de << std::endl;
    const ByteValue *bv = de.GetByteValue();
    SmartPointer<SequenceOfItems> sqi = new
        SequenceOfItems;
    sqi->SetLength( bv->GetLength() );
    std::stringstream ss;
    ss.str( std::string( bv->GetPointer(), bv->GetLength() ) );
    sqi->Read<ImplicitDataElement,SwapperNoOp>( ss );

    std::cout << *sqi << std::endl;

    return 0;
}

```

29.129 ReadFiles.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
import gdcml.*;
import java.io.File;

public class ReadFiles
{
    static int i = 0;
    public static void process(String path)

```

```

{
    //String path = file.getPath();
    assert PosixEmulation.FileExists(path) : "Problem converting to 8bits";

    System.out.println("Reading: " + path );
    System.out.println("File: " + i++);
    Reader r = new Reader();
    try
    {
        r.SetFileName( path );
        TagSetType skip = new TagSetType();
        skip.insert( new Tag(0x7fe0,0x10) );
        boolean b = r.ReadUpToTag( new Tag(0x88,0x200), skip );
        //System.out.println("DS:\n" + r.GetFile().GetDataSet().toString() );
    }
    finally
    {
        r.delete(); // will properly call C++ destructor and close file descriptor
    }
}

// Process only files under dir
public static void visitAllFiles(File dir)
{
    if (dir.isDirectory())
    {
        String[] children = dir.list();
        for (int i=0; i<children.length; i++)
        {
            visitAllFiles(new File(dir, children[i]));
        }
    }
    else
    {
        process(dir.getPath());
    }
}

public static void waiting (int n)
{
    long t0, t1;
    t0 = System.currentTimeMillis();
    do
    {
        t1 = System.currentTimeMillis();
    }
    while ((t1 - t0) < (n * 1000));
}

public static void main(String[] args) throws Exception
{
    String directory = args[0];

    Directory gdir = new Directory();
    long n = gdir.Load( directory, true );
    System.out.println( gdir.toString() );
    FilenamesType files = gdir.GetFilenames();
    for( long i = 0; i < n; ++i )
    {
        String path = files.get( (int)i );
        process( path );
    }

    System.out.println( "Java API" );

    //waiting( 10 );
    for( int i = 0; i < 2; ++i )
    {
        File dir = new File(directory);
        visitAllFiles(dir);
    }
}
}

```

29.130 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;
}

```

29.131 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;
}

```

29.132 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.SetInput( reader.GetOutput() );
    writer.SetFileName( "ReadSeriesIntoVTK.mhd" );
    writer.Write();

    System.out.println("Success writing: " + writer.GetFileName() );
}
}

```

29.133 ReadUTF8QtDir.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.

=====*/
/*
 * 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 "gdcmsReader.h"
#include "gdcmsDirectory.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;
    }
    gdcms::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.
        GetFileNames();

    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;
}

```

29.134 RefCounting.cs

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====

```

```

=====*/
using Kitware.VTK;
using Kitware.VTK.GDCM;

/*
 * this is not so much an example but simply a test to make sure ctor / dctor 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;
        }
    }
}

```

29.135 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://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Simple C# example
 *
 * Usage:
 * $ 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 fmi;
        SetSourceApplicationEntityTitle( "My Reformat App" );

        // http://www.oid-info.com/get/1.3.6.1.4.17434
        string THERALYS_ORG_ROOT = "1.3.6.1.4.17434";
        gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
        System.Console.WriteLine( "Root dir is now: " + gdcm.UIDGenerator.
            GetRoot() );

        string filename = args[0];
        string outfilename = args[1];

        Reader reader = new Reader();
        reader.SetFileName( filename );
        if( !reader.Read() )
        {
            System.Console.WriteLine( "Could not read: " + filename );
            return 1;
        }

        UIDGenerator uid = new UIDGenerator(); // helper for uid generation
        FileDerivation fd = new FileDerivation();
        // For the purpose of this exercise we will pretend that this image is referencing
        // two source image (we need to generate fake UID for that).
        string ReferencedSOPClassUID = "1.2.840.10008.5.1.4.1.1.7"; // Secondary Capture
        fd.AddReference( ReferencedSOPClassUID, uid.Generate() );
        fd.AddReference( ReferencedSOPClassUID, uid.Generate() );

        // Again for the purpose of the exercise we will pretend that the image is a
        // multiplanar reformat (MPR):
        // CID 7202 Source Image Purposes of Reference
        // { "DCM",121322,"Source image for image processing operation"},
        fd.SetPurposeOfReferenceCodeSequenceCodeValue( 121322 );
        // CID 7203 Image Derivation
        // { "DCM",113072,"Multiplanar reformatting" },
        fd.SetDerivationCodeSequenceCodeValue( 113072 );
        fd.SetFile( reader.GetFile() );
        // If all Code Value are ok the filter will execute properly
        if( !fd.Derive() )
        {
            return 1;
        }

        gdcm.FileMetaInformation fmi = reader.GetFile().GetHeader();
        // The following three lines make sure to regenerate any value:
        fmi.Remove( new gdcm.Tag(0x0002,0x0012) );
        fmi.Remove( new gdcm.Tag(0x0002,0x0013) );
        fmi.Remove( new gdcm.Tag(0x0002,0x0016) );

        Writer writer = new Writer();
        writer.SetFileName( outfilename );
        writer.SetFile( fd.GetFile() );
        if( !writer.Write() )
        {
            System.Console.WriteLine( "Could not write: " + outfilename );
            return 1;
        }

        return 0;
    }
}

```

29.136 RemovePrivateTags.py

```

1 #####
2 #
3 # Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 # Copyright (c) 2006-2011 Mathieu Malaterre
6 # All rights reserved.
7 # See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 # This software is distributed WITHOUT ANY WARRANTY; without even

```

```

10 #         the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #         PURPOSE.  See the above copyright notice for more information.
12 #
13 #####
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)

```

29.137 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.ComputeInterceptSlopePixelFormat() );

    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;
}
}

```

29.138 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();
    _imageViewer->Delete();

    _sphere->Delete();
    _sphereMapper->Delete();
    _sphereActor->Delete();

    _plane->Delete();
    _cutter->Delete();
    _polyTransform->Delete();
    _ROIMapper->Delete();
    _ROIActor->Delete();

    _annotation->Delete();
}

void CreatePipeline(const char* fileName)
{
    vtkProperty2D* props;

    //_reader=vtkXMLImageDataReader::New();
    //_reader->SetFileName(fileName);
    //_reader->Update();

    //_reader=qzDICOMImageReader::New();
    _reader=vtkGDCMImageReader::New();

    //vtkDirectory *d = vtkDirectory::New();
    //d->Open(fileName);
    //d->Print( std::cout );
    gdcmm::Directory d;
    d.Load(fileName);
    gdcmm::Directory::FileNamesType const &files = d.
    GetFileNames();

    gdcmm::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.GetFileNames();
    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.
    _imageView = vtkImageViewer2::New();
    _imageView->SetupInteractor(_interactor);
    _imageView->SetSize(400, 400);
    _imageView->SetColorWindow(1024);
    _imageView->SetColorLevel(800);
    _imageView->SetInputConnection(_reslice->GetOutputPort());
    _imageView->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());
#if (VTK_MAJOR_VERSION >= 6)
    assert(0);
#else
    imageData->UpdateInformation();
#endif

// Let vtkImageViewer2 handle the slice limits.
_imageViewer->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());

_imageViewer->Render();
}

int GetSlice()
{
    return _imageViewer->GetSlice();
}

// Set the orientation of the view.
void SetOrientation(ResliceRender::ORIENTATION orientation)
{
    vtkCamera* camera=_imageViewer->GetRenderer()->GetActiveCamera();

    double spacing[3];
    double origin[3];
    double point[4];
    double newPoint[4];
    double initialPosition;
    double xDirCosine[3];
    double yDirCosine[3];
    double zDirCosine[3];
    double normal[3];

    vtkImageData* imageData;

```

```

    vtkSmartPointer<vtkMatrix4x4> matrix =
        vtkSmartPointer<vtkMatrix4x4>::New();

    _orientation=orientation;

    // Reset ViewUp
    camera->SetViewUp(0.0, 1.0, 0.0);

    // Compute the cut plane position to the input coordinate system.
    imageData=vtkImageData::SafeDownCast(_reslice->GetInput());
    #if (VTK_MAJOR_VERSION >= 6)
        assert(0);
    #else
        imageData->UpdateInformation();
    #endif
    imageData->GetSpacing(spacing);
    imageData->GetOrigin(origin);

    point[0]=origin[0];
    point[1]=origin[1];
    point[2]=origin[2];
    point[3]=1.0;

    switch (_orientation)
    {
    case AXIAL:
        matrix->DeepCopy(AxialMatrix);
        initialPosition=sphereCenter[2];
        break;

    case CORONAL:
        matrix->DeepCopy(CoronalMatrix);
        initialPosition=sphereCenter[1];
        break;

    case SAGITTAL:
        matrix->DeepCopy(SagittalMatrix);
        initialPosition=sphereCenter[0];
        break;

    case OBLIQUE:
        matrix->DeepCopy(ObliqueMatrix);
        initialPosition=sphereCenter[2];
        break;
    }

    // Move the origin from the original image coordinate system to the
    // resliced image coordinate system.
    matrix->MultiplyPoint(point, newPoint);
    matrix->SetElement(0, 3, newPoint[0]);
    matrix->SetElement(1, 3, newPoint[1]);
    matrix->SetElement(2, 3, newPoint[2]);

    ResetOrientation();
    SetOrientation(matrix);

    // Compute the cutting plane normal and set it.
    // PROBLEM: If the transformation is connected rather than
    // using SetResliceAxes, the Direction Cosines do not reflect
    // the orientation of the vtkImageReslice.
    _reslice->GetResliceAxesDirectionCosines(xDirCosine, yDirCosine,
                                              zDirCosine);
    vtkMath::Cross(xDirCosine, yDirCosine, normal);
    _plane->SetNormal(normal);

    // Set the extents and spacing of the reslice to account for
    // all of the data.
    _reslice->SetOutputExtentToDefault();
    _reslice->SetOutputSpacing(spacing[0], spacing[0], spacing[0]);

    // Force the vtkImageViewer2 to update.
    // PROBLEM: The whole extent does not seem to be set in time
    // for the first render. This results in an error because the
    // slice is positioned outside the old bounds.
    #if (VTK_MAJOR_VERSION >= 6)
        _imageView->SetInputData(NULL);
    #else
        _imageView->SetInput(NULL);
    #endif
    _imageView->SetInputConnection(_reslice->GetOutputPort());

```

```

        _imageView->GetRenderer()->ResetCameraClippingRange();
        _imageView->GetRenderer()->ResetCamera();

        // Set the initial slice to be at the center of the sphere.
        // Divide by the spacing because this will be undone in SetSlice.
        SetSlice( (int)(initialPosition / spacing[0]));
    }

    vtkRenderWindowInteractor* GetInteractor()
    {
        return _interactor;
    }

protected:
    ORIENTATION                _orientation;

    //qzDICOMImageReader*      _reader;
    vtkGDCMImageReader*        _reader;
    vtkImageThreshold*         _threshold;
    vtkImageShiftScale*        _shift;
    vtkImageReslice*           _reslice;
    vtkRenderWindowInteractor* _interactor;
    vtkImageViewer2*           _imageView;

    vtkSphereSource*           _sphere;
    vtkPolyDataMapper*          _sphereMapper;
    vtkActor*                   _sphereActor;

    vtkPlane*                   _plane;
    vtkCutter*                   _cutter;
    vtkTransform*               _transform;
    vtkTransformPolyDataFilter* _polyTransform;
    vtkPolyDataMapper2D*         _ROIMapper;
    vtkActor2D*                 _ROIActor;

    vtkTextActor*               _annotation;
};

// Catch KeyPress events.
// Up Arrow - increases the slice
// Down Arrow - decreases the slice
// 'A' - sets the view to Axial
// 'S' - sets the view to Sagittal
// 'C' - sets the view to Coronal
// 'O' - set the view to Oblique

void KeyCallback::Execute(vtkObject* caller, unsigned long eventId, void *calldata)
{
    (void)caller;
    (void)eventId;
    (void)calldata;
    std::string sym=_reslice->GetInteractor()->GetKeySym();

    if (!sym.compare("Up"))
    {
        _reslice->SetSlice(_reslice->GetSlice() + 1);
    }
    else if (!sym.compare("Down"))
    {
        _reslice->SetSlice(_reslice->GetSlice() - 1);
    }
    else if ((!sym.compare("A")) || (!sym.compare("a")))
    {
        _reslice->SetOrientation(ResliceRender::AXIAL);
    }
    else if ((!sym.compare("C")) || (!sym.compare("c")))
    {
        _reslice->SetOrientation(ResliceRender::CORONAL);
    }
    else if ((!sym.compare("S")) || (!sym.compare("s")))
    {
        _reslice->SetOrientation(ResliceRender::SAGITTAL);
    }
    else if ((!sym.compare("O")) || (!sym.compare("o")))
    {
        _reslice->SetOrientation(ResliceRender::OBLIQUE);
    }
}

void KeyCallback::SetCallbackData(ResliceRender* reslice)

```

```

{
    _reslice=reslice;
}

// Usage: ResliceSphere [fileName]
int main(int argc, char *argv[])
{
    ResliceRender render;

    if (argc == 1)
    {
        const char *root = gdcm::Testing::GetDataExtraRoot();
        std::string dir3 = root;
        dir3 += "/gdcmSampleData/ForSeriesTesting/Dentist/images/";
        render.CreatePipeline(dir3.c_str());
    }
    else
    {
        render.CreatePipeline(argv[1]);
    }

    render.SetOrientation(ResliceRender::AXIAL);
    render.Start();

    return EXIT_SUCCESS;
}

```

29.139 ReWriteSCAsMR.py

```

1 #####
2 #
3 # Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 # Copyright (c) 2006-2011 Mathieu Malaterre
6 # All rights reserved.
7 # See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 # This software is distributed WITHOUT ANY WARRANTY; without even
10 # the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 # PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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

```

```

47  # in all other case simply return the currentspacing:
48  return False
49
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
          Image Storage'
61          # while we would rather have 'MR Image Storage'
62          gdcm.ImageHelper.SetForcePixelSpacing( True )
63          mrspacing = gdcm.ImageHelper.GetSpacingValue( r.GetFile() )
64          # TODO: I cannot do simply the following:
65          #image.SetSpacing( mrspacing )
66          image.SetSpacing(0, mrspacing[0] )
67          image.SetSpacing(1, mrspacing[1] )
68          image.SetSpacing(2, mrspacing[2] )
69          gdcm.ImageHelper.SetForceRescaleInterceptSlope( True )
70          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)

```

29.140 rle2img.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_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 (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:
 * 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 "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmAttribute.h"
#include "gdcmImageWriter.h"

/* FIXME: Why is PhilipsLosslessRice.dcm a 512x512 image ... */
void delta_decode(const char *inbuffer, size_t length, std::vector<unsigned short> &output)
{
    // RLE pass
    std::vector<char> temp;
    for(size_t i = 0; i < length; ++i)
    {
        if( inbuffer[i] == (char)0xa5 )
        {
            //unsigned char repeat = (unsigned char)inbuffer[i+1] + 1;
            //assert( (unsigned char)inbuffer[i+1] != 255 );
            int repeat = (unsigned char)inbuffer[i+1] + 1;
            char value = inbuffer[i+2];
            while(repeat)
            {
                temp.push_back( value );
                --repeat;
            }
            i+=2;
        }
        else
        {
            temp.push_back( inbuffer[i] );
        }
    }

    // Delta encoding pass
    unsigned short delta = 0;
    for(size_t i = 0; i < temp.size(); ++i)
    {
        if( temp[i] == 0x5a )
        {
            unsigned char v1 = (unsigned char)temp[i+1];
            unsigned char v2 = (unsigned char)temp[i+2];
            unsigned short value = (unsigned short)(v2 * 256 + v1);
            output.push_back( value );
            delta = value;
            i+=2;
        }
        else
        {
            unsigned short value = (unsigned short)(temp[i] + delta);
            output.push_back( value );
            delta = value;
        }
        //assert( output[output.size()-1] == ref[output.size()-1] );
    }

    if ( output.size() % 2 )
    {
        output.resize( output.size() - 1 );
    }
    std::cout << length << " -> " << output.size() * 2 << std::endl;
}

int main(int argc, char *argv [])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << "input.dcm [output.dcm]" << std::endl;
        std::cerr << "will default to 'outrle.dcm' unless output.dcm is specified."
        << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
    }
}

```

```

    return 1;
}
const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

// (07a1,1011) CS [PMSCT_RLE1] # 10,1 Tamar Compression Type
const gdcm::PrivateTag tcompressiontype(0x07a1,0x0011,"ELSCINT1");
if( !ds.FindDataElement( tcompressiontype ) ) return 1;
const gdcm::DataElement& compressiontype = ds.GetDataElement(
    tcompressiontype );
if ( compressiontype.IsEmpty() ) return 1;
const gdcm::ByteValue * bv = compressiontype.GetByteValue();
std::string comprle = "PMSCT_RLE1";
std::string comprgb = "PMSCT_RGB1";
bool isrle = false;
bool isrgb = false;
if( strcmp( bv->GetPointer(), comprle.c_str(), comprle.size() ) == 0 )
{
    isrle = true;
}
if( strcmp( 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;

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<0x0028,0x0010> at1;
at1.SetFromDataSet( ds );
gdcm::Attribute<0x0028,0x0011> at2;
at2.SetFromDataSet( ds );

gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
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
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 = "out.rle.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;
}

```

29.141 rtstructapp.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 "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/gdcmlNonImageData/exRT_Structure_Set_Storage.dcm
// gdcmDataExtra/gdcmlNonImageData/RTSTRUCT_1.3.6.1.4.1.22213.1.1396.2.dcm
// gdcmDataExtra/gdcmlNonImageData/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)
{
#ifdef 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();
#ifdef 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;
}

```

29.142 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;
    }
}

```

29.143 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");
    }

    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" );
}

```

29.144 ScanDirectory.py

```

1 #####
2 #
3 # Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 # Copyright (c) 2006-2011 Mathieu Malaterre
6 # All rights reserved.
7 # See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 # This software is distributed WITHOUT ANY WARRANTY; without even
10 # the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 # PURPOSE. See the above copyright notice for more information.
12 #
13 #####
14
15 import gdcm
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  # instanciate 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)

```

29.145 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;
    }
}

```


29.146 SimplePrint.cs

This is a C# example on how to use `gdcm::SWIGDataSet`

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
  Convertor convertor = new Convertor();
  int a = convertor.Convert<int>( some_int_blob );
  double b = convertor.Convert<double>( some_double_blob );
*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/SimplePrint.exe gdcmData/012345.002.050.dcm
 */
using System;
using gdcm;

public class SimplePrint
{
    public static void RecurseDataSet(File f, DataSet ds, string indent)
    {
        CSharpDataSet cds = new CSharpDataSet(ds);
        while(!cds.IsAtEnd())
        {
            DataElement de = cds.GetCurrent();
            // Compute VR from the toplevel file, and the currently processed dataset:
            VR vr = DataSetHelper.ComputeVR(f, ds, de.GetTag() );

            if( vr.Compatible( new VR(VR.VRType.SQ) ) )
            {
                uint uvl = (uint)de.GetVL(); // Test cast is ok
                System.Console.WriteLine( indent + de.GetTag().toString() + ":" + uvl ); // why not ?
                //SequenceOfItems sq = de.GetSequenceOfItems();
                // GetValueAsSQ handle more cases than GetSequenceOfItems
                SmartPtrSQ sq = de.GetValueAsSQ();
                uint n = sq.GetNumberOfItems();
                for( uint i = 1; i <= n; i++) // item starts at 1, not 0
                {
                    Item item = sq.GetItem( i );
                    DataSet nested = item.GetNestedDataSet();
                    RecurseDataSet( f, nested, indent + "  " );
                }
            }
            else
            {
                System.Console.WriteLine( indent + de.toString() );
            }
            cds.Next();
        }
    }

    public static int Main(string[] args)
    {
        string filename = args[0];
        Reader reader = new Reader();
        reader.SetFileName( filename );
        bool ret = reader.Read();
        if( !ret )
        {
            return 1;
        }
        File f = reader.GetFile();
        DataSet ds = f.GetDataSet();

        RecurseDataSet( f, ds, "" );
    }
}

```

```

    return 0;
}
}

```

29.147 SimplePrintPatientName.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/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;
        }
    }
}

```

29.148 SimpleScanner.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

```

Copyright (c) 2006-2011 Mathieu Malaterre
 All rights reserved.
 See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even
 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
 PURPOSE. See the above copyright notice for more information.

```

=====*/
/*
 * Simple example to show how to use Scanner API.
 * It exposes the three different cases:
 * - DICOM Attribute is present and has a value
 * - DICOM Attribute is present and has no value
 * - DICOM Attribute is not present at all
 * It also shows the purpose of the function 'IsKey' to detect whether or
 * not the file has been read by the gdcm::Scanner. Technically most of the time
 * if a file is not a 'Key' this is because it is not a DICOM file. You need to use
 * gdcm::System::FileExists to decide whether or not the file actually exist on the disk.
 *
 * It was tested on this particular image:
 * ./SimpleScanner gdcmData/012345.002.050.dcm
 */

#include "gdcmStrictScanner.h"
#include "gdcmSimpleSubjectWatcher.h"
#include "gdcmFileNameEvent.h"

class MyFileWatcher : public gdcm::SimpleSubjectWatcher
{
public:
  MyFileWatcher(gdcm::Subject * s, const char *comment = ""):
    gdcm::SimpleSubjectWatcher(s,comment){}
  void ShowFileName(gdcm::Subject *, const gdcm::Event &evt)
  {
    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;
}

```

29.149 SortImage.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
*/
#include "gdcmSorter.h"
#include "gdcmScanner.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"

bool mysort(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
    //gdcm::Attribute<0x0020,0x0013> at1; // Instance Number

```

```

gdcmm::Attribute<0x0018,0x1060> at1; // Trigger Time
gdcmm::Attribute<0x0020,0x0032> at11; // Image Position (Patient)
at1.Set( ds1 );
at11.Set( ds1 );
//gdcmm::Attribute<0x0020,0x0013> at2;
gdcmm::Attribute<0x0018,0x1060> at2;
gdcmm::Attribute<0x0020,0x0032> at22;
at2.Set( ds2 );
at22.Set( ds2 );
if( at11 == at22 )
{
    return at1 < at2;
}
return at11 < at22;
}

bool mysort_part1(gdcmm::DataSet const & ds1, gdcmm::DataSet const & ds2 )
{
    gdcmm::Attribute<0x0018,0x1060> at1;
    at1.Set( ds1 );
    gdcmm::Attribute<0x0018,0x1060> at2;
    at2.Set( ds2 );
    return at1 < at2;
}

bool mysort_part2(gdcmm::DataSet const & ds1, gdcmm::DataSet const & ds2 )
{
    gdcmm::Attribute<0x0020,0x0032> at1;
    at1.Set( ds1 );
    gdcmm::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(gdcmm::DataSet const & ds1, gdcmm::DataSet const & ds2 )
{
    gdcmm::Attribute<0x0020,0x0052> at1; // FrameOfReferenceUID
    at1.Set( ds1 );
    gdcmm::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];
    gdcmm::Directory dir;
    unsigned int nfiles = dir.Load( dirname );

    dir.Print( std::cout );

    gdcmm::Sorter sorter;
    sorter.SetSortFunction( mysort );
    sorter.Sort( dir.GetFilesNames() );

    std::cout << "Sorter:" << std::endl;
    sorter.Print( std::cout );

    gdcmm::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 );

    gdcmm::Scanner s;
    s.AddTag( gdcmm::Tag(0x20,0x32) ); // Image Position (Patient)
    //s.AddTag( gdcmm::Tag(0x20,0x37) ); // Image Orientation (Patient)
    s.Scan( dir.GetFilesNames() );

    //s.Print( std::cout );

    // Count how many different IPP there are:
    const gdcmm::Scanner::ValueType &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;
}

```

29.150 SortImage.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #       This software is distributed WITHOUT ANY WARRANTY; without even
10 #       the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #       PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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(dsl, ds2):
29     # compare dsl
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

```

29.151 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;
    }
}

```

29.152 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;
}
}

```


29.153 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 "gdcmJPEGLSCodec.h"
#include "gdcmUIDGenerator.h"
#include "gdcmStreamImageWriter.h"
#include "gdcmAttribute.h"
#include "gdcmFile.h"
#include "gdcmTag.h"

bool StreamImageRead(gdcm::StreamImageWriter & theStreamWriter,
    const char* filename, const char* outfilename, int resolution)
{
    gdcm::StreamImageReader reader;

    reader.SetFileName( filename );

    if (!reader.ReadImageInformation())
    {
        std::cerr << "unable to read image information" << std::endl;
        return 1; //unable to read tags as expected.
    }
    //let's be tricky; each image will be read in portions, first the top half, then the bottom
    //that way, we can test how the stream handles fragmentation of the data
    //we could also loop this to get various different size combinations, but I'm not sure
    //that's useful, yet.
    std::vector<unsigned int> extent =
        gdcm::ImageHelper::GetDimensionsValue(reader.
            GetFile());
    // std::cout << extent[0];
    //at this point, these values aren't used, but may be in the future
    //unsigned short xmin = 0;
    //unsigned short xmax = extent[0];
    //unsigned short ymin = 0;
    //unsigned short ymax = extent[1];
    //unsigned short zmin = 0;
    //unsigned short zmax = extent[2];

    std::cout<< "\n Row: "<<extent[0] <<"\n Col :"<< extent[1]<< "\n Resolution :"<< extent[2] << std::endl;

    int a =1;
    for (int i=1; i<=(extent[2]-resolution);++i)
        a = a*2;

    reader.DefinePixelExtent(0, extent[0]/a, 0, extent[1]/a, resolution-1, resolution);

    unsigned long len = reader.DefineProperBufferLength();
    char* finalBuffer = new char[len];
    memset(finalBuffer, 0, sizeof(char)*len);

    if (reader.CanReadImage())
    {
        bool result = reader.Read(finalBuffer, len);
        if( !result )

```

```

        {
            std::cout << "res2 failure:" << filename << std::endl;
            delete [] finalBuffer;
            return 1;
        }
        else
        {
            std::cout<< "Able to read";
        }
    }
    else
    {
        std::cerr<< "Not able to put in buffer"<< std::endl;
    }
}
/*
//now, read in smaller buffer extents
reader.DefinePixelExtent(xmin, xmax, ymin, ymax);
len = reader.DefineProperBufferLength();

char* buffer = new char[len];
bool res2 = reader.Read(buffer, len);
if( !res2 ){
    std::cerr << "res2 failure:" << filename << std::endl;
    return 1;
}
//copy the result into finalBuffer
memcpy(finalBuffer, buffer, len);

//now read the next half of the image
ymin = ymax;
ymax = extent[1];

reader.DefinePixelExtent(xmin, xmax, ymin, ymax);

//std::cerr << "Success to read image from file: " << filename << std::endl;
unsigned long len2 = reader.DefineProperBufferLength();

char* buffer2 = new char[len2];
bool res3 = reader.Read(buffer2, len2);
if( !res3 ){
    std::cerr << "res3 failure:" << filename << std::endl;
    return 1;
}
//copy the result into finalBuffer
memcpy(&(finalBuffer[len]), buffer2, len2);

delete [] buffer;
delete [] buffer2;
*/

gdcm::Writer w;
gdcm::File &file = w.GetFile();
gdcm::DataSet &ds = file.GetDataSet();

file.GetHeader().SetDataSetTransferSyntax(
    gdcm::TransferSyntax::ExplicitVRLittleEndian );

gdcm::UIDGenerator uid;
gdcm::DataElement de( gdcm::Tag(0x8,0x18) ); // SOP Instance UID
de.SetVR( gdcm::VR::UI );
const char *u = uid.Generate();
de.SetByteValue( u, strlen(u) );
ds.Insert( de );

gdcm::DataElement del( gdcm::Tag(0x8,0x16) );
del.SetVR( gdcm::VR::UI );
gdcm::MediaStorage ms(
    gdcm::MediaStorage::VLWholeSlideMicroscopyImageStorage
);
del.SetByteValue( ms.GetString(), strlen(ms.GetString()) );
ds.Insert( del );

const char mystr[] = "MONOCHROME2 ";
gdcm::DataElement de2( gdcm::Tag(0x28,0x04) );
//de.SetTag(gdcm::Tag(0x28,0x04));
de2.SetVR( gdcm::VR::CS );
de2.SetByteValue(mystr, strlen(mystr));
ds.Insert( de2 );

gdcm::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
ds.Insert( Number_Of_Frames.GetAsDataElement() );

```

```

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;
}

```

29.154 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;
gdcmm::ByteSwap<uint32_t>::SwapRangeFromSwapCodeIntoSystem
    ((uint32_t*)(&vl_str), gdcmm::SwapCode::BigEndian, 1);
memcpy(&vl, vl_str, 4);
if( vl != 0x00000004 )
{
    std::cerr << std::hex << "vl: " << vl << std::endl;
    return 1;
}

gdcmm::ByteSwap<uint32_t>::SwapFromSwapCodeIntoSystem(
    vl, gdcmm::SwapCode::LittleEndian);
if( vl != 0x00000004 )
{
    std::cerr << std::hex << "vl: " << vl << std::endl;
    return 1;
}

gdcmm::ByteSwap<uint32_t>::SwapFromSwapCodeIntoSystem(
    vl, gdcmm::SwapCode::BigEndian);
if( vl != 0x40000000 )
{
    std::cerr << std::hex << "vl: " << vl << std::endl;
    return 1;
}

return 0;
}

int TestByteSwap(int , char *[])
{
    gdcmm::SwapCode sc = gdcmm::SwapCode::Unknown;
    if ( gdcmm::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 == gdcm::SwapCode::LittleEndian )
{
    if( tn != 0x1234 )
    {
        std::cerr << std::hex << "tn: " << tn << std::endl;
        return 1;
    }
}
gdcm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem
(&ul6.tn, gdcm::SwapCode::BigEndian, 1);
tn = ul6.tn;
if( sc == gdcm::SwapCode::LittleEndian )
{
    if( tn != 0x3412 )
    {
        std::cerr << std::hex << "tn: " << tn << std::endl;
        return 1;
    }
}
else if ( sc == gdcm::SwapCode::BigEndian )
{
    if( tn != 0x1234 )
    {
        std::cerr << std::hex << "tn: " << tn << std::endl;
        return 1;
    }
}

if( myfunc() )
{
    return 1;
}

uint16_t array[] = { 0x1234 };
gdcm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem
(array,
gdcm::SwapCode::BigEndian,2);
if ( array[0] != 0x3412 )
{
    std::cerr << std::hex << "array: " << array[0] << std::endl;
    return 1;
}

return 0;
}

```

29.155 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;
}

```

```

gdcmm::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 gdcmm::FileMetaInformation &h = reader.GetFile().GetHeader();
//std::cout << h << std::endl;

//const gdcmm::DataSet &ds = reader.GetFile().GetDataSet();
//std::cout << ds << std::endl;

const char *ref = gdcmm::Testing::GetMediaStorageFromFile(filename);
gdcmm::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 != gdcmm::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
    gdcmm::Trace::DebugOff();
    gdcmm::Trace::WarningOff();
    int r = 0, i = 0;
    const char *filename;
    const char * const *filenames = gdcmm::Testing::GetFileNames();
    while( (filename = filenames[i]) )
    {
        r += TestRead( filename );
        ++i;
    }

    return r;
}

```

29.156 TestReader.py

This is a C++ example on how to use [gdcmm::Reader](#)

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre

```

```

6 # All rights reserved.
7 # See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 # This software is distributed WITHOUT ANY WARRANTY; without even
10 # the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 # PURPOSE. See the above copyright notice for more information.
12 #
13 #####
14
15 import gdcm
16 import os,sys
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         gdcm.Trace.DebugOff()
34         gdcm.Trace.WarningOff()
35         t = gdcm.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)

```

29.157 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 *scalarpointer;
// TODO I should also pass in the dim of the reference image just in case
};

void *ReadFilesThread(void *voidparams)
{
    const threadparams *params = static_cast<const threadparams *>(voidparams);

    const size_t nfiles = params->nfiles;

```



```

for(unsigned int file = 0; file < nfiles; ++file)
{
    /*
    // TODO: update progress
    pthread_mutex_lock(&params->lock);
    //section critique
    ReadingProgress+=params->stepProgress;
    pthread_mutex_unlock(&params->lock);
    */
    const char *filename = params->filenames[file];
    //std::cerr << filename << std::endl;

    gdcm::ImageReader reader;
    reader.SetFileName( filename );
    try
    {
        if( !reader.Read() )
        {
            std::cerr << "Failed to read: " << filename << std::endl;
            break;
        }
    }
    catch( ... )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        break;
    }

    const gdcm::Image &image = reader.GetImage();
    unsigned long len = image.GetBufferLength();
    char * pointer = params->scalarpointer;
    #if 0
    char *tempimage = new char[len];
    image.GetBuffer(tempimage);

    memcpy(pointer + file*len, tempimage, len);
    delete[] tempimage;
    #else
    char *tempimage = pointer + file * len;
    image.GetBuffer(tempimage);
    #endif
    }

    return voidparams;
}

void ShowFilenames(const threadparams &params)
{
    std::cout << "start" << std::endl;
    for(unsigned int i = 0; i < params.nfiles; ++i)
    {
        const char *filename = params.filenames[i];
        std::cout << filename << std::endl;
    }
    std::cout << "end" << std::endl;
}

void ReadFiles(size_t nfiles, const char *filenames[])
{
    // \precondition: nfiles > 0
    assert( nfiles > 0 );
    const char *reference= filenames[0]; // take the first image as reference

    gdcm::ImageReader reader;
    reader.SetFileName( reference );
    if( !reader.Read() )
    {
        // That would be very bad...
        assert(0);
    }

    const gdcm::Image &image = reader.GetImage();
    gdcm::PixelFormat pixeltype = image.GetPixelFormat();
    unsigned long len = image.GetBufferLength();
    const unsigned int *dims = image.GetDimensions();
    unsigned short pixelsize = pixeltype.GetPixelSize();
    (void)pixelsize;
    assert( image.GetNumberOfDimensions() == 2 );

    vtkImageData *output = vtkImageData::New();
    output->SetDimensions(dims[0], dims[1], (int)nfiles);

```

```

#if (VTK_MAJOR_VERSION >= 6)
    int numscal = pixeltype.GetSamplesPerPixel();
    switch( pixeltype )
    {
        case gdcm::PixelFormat::INT8:
            output->AllocateScalars( VTK_SIGNED_CHAR, numscal );
            break;
        case gdcm::PixelFormat::UINT8:
            output->AllocateScalars( VTK_UNSIGNED_CHAR, numscal );
            break;
        case gdcm::PixelFormat::INT16:
            output->AllocateScalars( VTK_SHORT, numscal );
            break;
        case gdcm::PixelFormat::UINT16:
            output->AllocateScalars( VTK_UNSIGNED_SHORT, numscal );
            break;
        case gdcm::PixelFormat::INT32:
            output->AllocateScalars( VTK_INT, numscal );
            break;
        case gdcm::PixelFormat::UINT32:
            output->AllocateScalars( VTK_UNSIGNED_INT, numscal );
            break;
        default:
            assert(0);
    }
#else
    switch( pixeltype )
    {
        case gdcm::PixelFormat::INT8:
            #if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
                output->SetScalarType ( VTK_SIGNED_CHAR );
            #else
                output->SetScalarType ( VTK_CHAR );
            #endif
            break;
        case gdcm::PixelFormat::UINT8:
            output->SetScalarType ( VTK_UNSIGNED_CHAR );
            break;
        case gdcm::PixelFormat::INT16:
            output->SetScalarType ( VTK_SHORT );
            break;
        case gdcm::PixelFormat::UINT16:
            output->SetScalarType ( VTK_UNSIGNED_SHORT );
            break;
        case gdcm::PixelFormat::INT32:
            output->SetScalarType ( VTK_INT );
            break;
        case gdcm::PixelFormat::UINT32:
            output->SetScalarType ( VTK_UNSIGNED_INT );
            break;
        default:
            assert(0);
    }
    output->SetNumberOfScalarComponents ( pixeltype.GetSamplesPerPixel() );
    output->AllocateScalars();
#endif
    char * scalarpointer = static_cast<char*>(output->GetScalarPointer());

    const unsigned int nthreads = 4;
    threadparams params[nthreads];

    //pthread_mutex_t lock;
    //pthread_mutex_init(&lock, NULL);

    pthread_t *pthread = new pthread_t[nthreads];

    // There is nfiles, and nThreads
    assert( nfiles > nthreads );
    const size_t partition = nfiles / nthreads;
    for (unsigned int thread=0; thread < nthreads; ++thread)
    {
        params[thread].filenames = filenames + thread * partition;
        params[thread].nfiles = partition;
        if( thread == nthreads - 1 )
        {
            // There is slightly more files to process in this thread:
            params[thread].nfiles += nfiles % nthreads;
        }
        assert( thread * partition < nfiles );
        params[thread].scalarpointer = scalarpointer + thread * partition * len;
    }

```

```

    //assert( params[thread].scalarpointer < scalarpointer + 2 * dims[0] * dims[1] * dims[2] );
    // start thread:
    int res = pthread_create( &pthread[thread], NULL, ReadFilesThread, &params[thread]);
    if( res )
    {
        std::cerr << "Unable to start a new thread, pthread returned: " << res << std::endl;
        assert(0);
    }
    //ShowFileNames(params[thread]);
}
// DEBUG
size_t total = 0;
for (unsigned int thread=0; thread < nthreads; ++thread)
{
    total += params[thread].nfiles;
}
assert( total == nfiles );
// END DEBUG

for (unsigned int thread=0;thread<nthreads;thread++)
{
    pthread_join( pthread[thread], NULL);
}
delete[] pthread;

//pthread_mutex_destroy(&lock);

// For some reason writing down the file is painfully slow...
vtkStructuredPointsWriter *writer = vtkStructuredPointsWriter::New();
#if (VTK_MAJOR_VERSION >= 6)
writer->SetInputData( output );
#else
writer->SetInput( output );
#endif
writer->SetFileName( "/tmp/threadgdcmm.vtk" );
writer->SetFileTypeToBinary();
//writer->Write();
writer->Delete();

//output->Print( std::cout );
output->Delete();
}

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " [directory|list of filenames]\n";
        return 1;
    }

    // Check if user pass in a single directory
    if( argc == 2 && gdcmm::System::FileIsDirectory( argv[1] ) )
    {
        gdcmm::Directory d;
        d.Load( argv[1] );
        gdcmm::Directory::FileNamesType l = d.
            GetFileNames();
        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;
}

```

29.158 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;
}

```

29.159 uid_unique.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 "gdcmlUIDGenerator.h"

#include <iostream>
#include <string>
#include <set>

int main()
{
    gdcml::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(1)
    {
        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 );
    }
}

```

29.160 VolumeSorter.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 "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];
}

gdcm::Directory d;
d.Load( dir1.c_str(), true ); // recursive !
const gdcm::Directory::FileNamesType &l1 = d.
    GetFileNames();
const size_t nfiles = l1.size();
std::cout << nfiles << std::endl;

//if( nfiles != 280 )
// {
//     return 1;
// }

//d.Print( std::cout );

gdcm::Scanner s0;
const gdcm::Tag t1(0x0020,0x000d); // Study Instance UID
const gdcm::Tag t2(0x0020,0x000e); // Series Instance UID
//const gdcm::Tag t3(0x0010,0x0010); // Patient's Name
s0.AddTag( t1 );
s0.AddTag( t2 );
//s0.AddTag( t3 );
//s0.AddTag( t4 );
//s0.AddTag( t5 );
//s0.AddTag( t6 );
bool b = s0.Scan( d.GetFileNames() );
if( !b )
{
    std::cerr << "Scanner failed" << std::endl;
    return 1;
}

//s0.Print( std::cout );

// Only get the DICOM files:
gdcm::Directory::FileNamesType l2 = s0.GetKeys();
const size_t nfiles2 = l2.size();
std::cout << nfiles2 << std::endl;

if ( nfiles2 > nfiles )
{
    return 1;
}

gdcm::Sorter sorter;
sorter.SetSortFunction( mysort1 );
sorter.StableSort( l2 );

sorter.SetSortFunction( mysort2 );
sorter.StableSort( sorter.GetFileNames() );

sorter.SetSortFunction( mysort3 );
sorter.StableSort( sorter.GetFileNames() );

sorter.SetSortFunction( mysort4 );
sorter.StableSort( sorter.GetFileNames() );

//sorter.Print( std::cout );

// Let's try to check our result:
// assume that IPP is precise enough so that we can test floating point equality:
size_t nvalues = 0;
{
    gdcm::Scanner s;
    s.AddTag( gdcm::Tag(0x20,0x32) ); // Image Position (Patient)
    //s.AddTag( gdcm::Tag(0x20,0x37) ); // Image Orientation (Patient)
    s.Scan( d.GetFileNames() );

    //s.Print( std::cout );

    const gdcm::Scanner::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;
}

gdcm::Directory::FileNamesType sorted_files = sorter.
    GetFileNames();

// Which means we can take nvalues files at a time and execute gdcm::IPPSorter on it:
gdcm::IPPSorter ippsorter;
gdcm::Directory::FileNamesType sub( sorted_files.begin(), sorted_files.
    begin() + nvalues);
std::cout << sub.size() << std::endl;
std::cout << sub[0] << std::endl;
std::cout << sub[nvalues-1] << std::endl;
ippsorter.SetComputeZSpacing( false );
if( !ippsorter.Sort( sub ) )
{
    std::cerr << "Could not sort" << std::endl;
    return 1;
}

std::cout << "IPPSorter:" << std::endl;
ippsorter.Print( std::cout );

return 0;
}

```

29.161 WriteBuffer.py

```

1 #####
2 #
3 # Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 # Copyright (c) 2006-2011 Mathieu Malaterre
6 # All rights reserved.
7 # See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 # This software is distributed WITHOUT ANY WARRANTY; without even
10 # the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 # PURPOSE. See the above copyright notice for more information.
12 #
13 #####
14
15 """
16 Usage:
17
18 http://chuckhahm.com/Ischem/Zurich/XX_0134
19
20 (2005,1132) SQ (Sequence with undefined length #=8) # u/1, 1 Unknown Tag & Data
21 (fffe,e000) na (Item with undefined length #=9) # u/1, 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/1, 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() )
```


Index

- ~ASN1
 - gdcmm::ASN1, [174](#)
- ~AnonymizeEvent
 - gdcmm::AnonymizeEvent, [159](#)
- ~Anonymizer
 - gdcmm::Anonymizer, [162](#)
- ~Attribute
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, [188](#)
- ~AudioCodec
 - gdcmm::AudioCodec, [197](#)
- ~BaseCompositeMessage
 - gdcmm::network::BaseCompositeMessage, [201](#)
- ~BaseNormalizedMessage
 - gdcmm::network::BaseNormalizedMessage, [203](#)
- ~BasePDU
 - gdcmm::network::BasePDU, [205](#)
- ~BaseQuery
 - gdcmm::BaseQuery, [207](#)
- ~BaseRootQuery
 - gdcmm::BaseRootQuery, [210](#)
- ~Bitmap
 - gdcmm::Bitmap, [219](#)
- ~BitmapToBitmapFilter
 - gdcmm::BitmapToBitmapFilter, [225](#)
- ~BoxRegion
 - gdcmm::BoxRegion, [227](#)
- ~ByteSwapFilter
 - gdcmm::ByteSwapFilter, [231](#)
- ~ByteValue
 - gdcmm::ByteValue, [234](#)
- ~CAPICryptographicMessageSyntax
 - gdcmm::CAPICryptographicMessageSyntax, [239](#)
- ~CSAHeader
 - gdcmm::CSAHeader, [277](#)
- ~Coder
 - gdcmm::Coder, [252](#)
- ~Command
 - gdcmm::Command, [257](#)
- ~CommandDataSet
 - gdcmm::CommandDataSet, [259](#)
- ~CryptoFactory
 - gdcmm::CryptoFactory, [268](#)
- ~CryptographicMessageSyntax
 - gdcmm::CryptographicMessageSyntax, [270](#)
- ~Curve
 - gdcmm::Curve, [288](#)
- ~DICOMDIRGenerator
 - gdcmm::DICOMDIRGenerator, [319](#)
- ~DataEvent
 - gdcmm::DataEvent, [301](#)
- ~DataSetEvent
 - gdcmm::DataSetEvent, [310](#)
- ~Decoder
 - gdcmm::Decoder, [312](#)
- ~Defs
 - gdcmm::Defs, [314](#)
- ~DeltaEncodingCodec
 - gdcmm::DeltaEncodingCodec, [317](#)
- ~DictConverter
 - gdcmm::DictConverter, [324](#)
- ~DictPrinter
 - gdcmm::DictPrinter, [329](#)
- ~Dicts
 - gdcmm::Dicts, [330](#)
- ~DirectionCosines
 - gdcmm::DirectionCosines, [334](#)
- ~Directory
 - gdcmm::Directory, [336](#)
- ~Dumper
 - gdcmm::Dumper, [341](#)
- ~Element
 - gdcmm::Element< TVR, VM::VM1_n >, [347](#)
- ~Event
 - gdcmm::Event, [366](#)
- ~Exception
 - gdcmm::Exception, [368](#)
- ~File
 - gdcmm::File, [376](#)
- ~FileAnonymizer
 - gdcmm::FileAnonymizer, [379](#)
- ~FileChangeTransferSyntax
 - gdcmm::FileChangeTransferSyntax, [382](#)
- ~FileDecompressLookupTable
 - gdcmm::FileDecompressLookupTable, [384](#)
- ~FileDerivation
 - gdcmm::FileDerivation, [386](#)
- ~FileExplicitFilter
 - gdcmm::FileExplicitFilter, [388](#)
- ~FileMetaInformation

- gdcmm::FileMetaInformation, 392
- ~FileNameEvent
 - gdcmm::FileNameEvent, 399
- ~FileStreamer
 - gdcmm::FileStreamer, 405
- ~FilenameGenerator
 - gdcmm::FilenameGenerator, 401
- ~Global
 - gdcmm::Global, 416
- ~GroupDict
 - gdcmm::GroupDict, 419
- ~IconImageFilter
 - gdcmm::IconImageFilter, 421
- ~IconImageGenerator
 - gdcmm::IconImageGenerator, 423
- ~Image
 - gdcmm::Image, 427
- ~ImageApplyLookupTable
 - gdcmm::ImageApplyLookupTable, 431
- ~ImageChangePhotometricInterpretation
 - gdcmm::ImageChangePhotometricInterpretation, 433
- ~ImageChangePlanarConfiguration
 - gdcmm::ImageChangePlanarConfiguration, 436
- ~ImageChangeTransferSyntax
 - gdcmm::ImageChangeTransferSyntax, 439
- ~ImageCodec
 - gdcmm::ImageCodec, 443
- ~ImageConverter
 - gdcmm::ImageConverter, 448
- ~ImageFragmentSplitter
 - gdcmm::ImageFragmentSplitter, 451
- ~ImageReader
 - gdcmm::ImageReader, 457
- ~ImageRegionReader
 - gdcmm::ImageRegionReader, 460
- ~ImageToImageFilter
 - gdcmm::ImageToImageFilter, 463
- ~ImageWriter
 - gdcmm::ImageWriter, 465
- ~JPEG12Codec
 - gdcmm::JPEG12Codec, 485
- ~JPEG16Codec
 - gdcmm::JPEG16Codec, 487
- ~JPEG2000Codec
 - gdcmm::JPEG2000Codec, 489
- ~JPEG8Codec
 - gdcmm::JPEG8Codec, 493
- ~JPEGCodec
 - gdcmm::JPEGCodec, 496
- ~JPEGGLSCodec
 - gdcmm::JPEGGLSCodec, 500
- ~JSON
 - gdcmm::JSON, 503
- ~KAKADUCodec
 - gdcmm::KAKADUCodec, 505
- ~LookupTable
 - gdcmm::LookupTable, 510
- ~MD5
 - gdcmm::MD5, 517
- ~MemberCommand
 - gdcmm::MemberCommand, 528
- ~MeshPrimitive
 - gdcmm::MeshPrimitive, 532
- ~ModuleEntry
 - gdcmm::ModuleEntry, 542
- ~Object
 - gdcmm::Object, 572
- ~OpenSSLCryptographicMessageSyntax
 - gdcmm::OpenSSLCryptographicMessageSyntax, 576
- ~OpenSSL7CryptographicMessageSyntax
 - gdcmm::OpenSSL7CryptographicMessageSyntax, 579
- ~Orientation
 - gdcmm::Orientation, 582
- ~Overlay
 - gdcmm::Overlay, 585
- ~PDBHeader
 - gdcmm::PDBHeader, 598
- ~PDFCodec
 - gdcmm::PDFCodec, 600
- ~PGXCodec
 - gdcmm::PGXCodec, 605
- ~PNMCodec
 - gdcmm::PNMCodec, 627
- ~PVRGCodec
 - gdcmm::PVRGCodec, 650
- ~ParseException
 - gdcmm::ParseException, 590
- ~Parser
 - gdcmm::Parser, 592
- ~Pixmap
 - gdcmm::Pixmap, 615
- ~PixmapReader
 - gdcmm::PixmapReader, 619
- ~PixmapToPixmapFilter
 - gdcmm::PixmapToPixmapFilter, 621
- ~PixmapWriter
 - gdcmm::PixmapWriter, 624
- ~Preamble
 - gdcmm::Preamble, 629
- ~Printer
 - gdcmm::Printer, 641
- ~PrivateDict
 - gdcmm::PrivateDict, 643
- ~ProgressEvent
 - gdcmm::ProgressEvent, 648
- ~PythonFilter
 - gdcmm::PythonFilter, 651

- ~QueryBase
 - gdcm::QueryBase, [653](#)
- ~RAWCodec
 - gdcm::RAWCodec, [665](#)
- ~RLECodec
 - gdcm::RLECodec, [679](#)
- ~Reader
 - gdcm::Reader, [669](#)
- ~Region
 - gdcm::Region, [673](#)
- ~Rescaler
 - gdcm::Rescaler, [676](#)
- ~SHA1
 - gdcm::SHA1, [721](#)
- ~Scanner
 - gdcm::Scanner, [686](#)
- ~Segment
 - gdcm::Segment, [691](#)
- ~SegmentReader
 - gdcm::SegmentReader, [697](#)
- ~SegmentWriter
 - gdcm::SegmentWriter, [699](#)
- ~SegmentedPaletteColorLookupTable
 - gdcm::SegmentedPaletteColorLookupTable, [694](#)
- ~SerieHelper
 - gdcm::SerieHelper, [712](#)
- ~ServiceClassUser
 - gdcm::ServiceClassUser, [717](#)
- ~SimpleMemberCommand
 - gdcm::SimpleMemberCommand, [724](#)
- ~SimpleSubjectWatcher
 - gdcm::SimpleSubjectWatcher, [725](#)
- ~SmartPointer
 - gdcm::SmartPointer, [728](#)
- ~Sorter
 - gdcm::Sorter, [734](#)
- ~Spacing
 - gdcm::Spacing, [736](#)
- ~SplitMosaicFilter
 - gdcm::SplitMosaicFilter, [738](#)
- ~StreamImageReader
 - gdcm::StreamImageReader, [741](#)
- ~StreamImageWriter
 - gdcm::StreamImageWriter, [745](#)
- ~StrictScanner
 - gdcm::StrictScanner, [751](#)
- ~StringFilter
 - gdcm::StringFilter, [758](#)
- ~Subject
 - gdcm::Subject, [761](#)
- ~Surface
 - gdcm::Surface, [766](#)
- ~SurfaceReader
 - gdcm::SurfaceReader, [773](#)
- ~SurfaceWriter
 - gdcm::SurfaceWriter, [775](#)
- ~Table
 - gdcm::Table, [783](#)
- ~TableEntry
 - gdcm::TableEntry, [784](#)
- ~TableReader
 - gdcm::TableReader, [785](#)
- ~TableRow
 - gdcm::network::TableRow, [787](#)
- ~TagPath
 - gdcm::TagPath, [794](#)
- ~Testing
 - gdcm::Testing, [796](#)
- ~Trace
 - gdcm::Trace, [800](#)
- ~Transition
 - gdcm::network::Transition, [808](#)
- ~ULAction
 - gdcm::network::ULAction, [834](#)
- ~ULBasicCallback
 - gdcm::network::ULBasicCallback, [864](#)
- ~ULConnection
 - gdcm::network::ULConnection, [865](#)
- ~ULConnectionCallback
 - gdcm::network::ULConnectionCallback, [868](#)
- ~ULConnectionManager
 - gdcm::network::ULConnectionManager, [872](#)
- ~ULEvent
 - gdcm::network::ULEvent, [874](#)
- ~ULWritingCallback
 - gdcm::network::ULWritingCallback, [877](#)
- ~UserInformation
 - gdcm::network::UserInformation, [885](#)
- ~Validate
 - gdcm::Validate, [887](#)
- ~Value
 - gdcm::Value, [889](#)
- ~Version
 - gdcm::Version, [891](#)
- ~Writer
 - gdcm::Writer, [970](#)
- ~XMLDictReader
 - gdcm::XMLDictReader, [973](#)
- ~XMLPrinter
 - gdcm::XMLPrinter, [975](#)
- ~XMLPrivateDictReader
 - gdcm::XMLPrivateDictReader, [977](#)
- ~vtkGDCMImageReader
 - vtkGDCMImageReader, [909](#)
- ~vtkGDCMImageReader2
 - vtkGDCMImageReader2, [916](#)
- ~vtkGDCMImageWriter
 - vtkGDCMImageWriter, [921](#)

- ~vtkGDCMMedicalImageProperties
 - vtkGDCMMedicalImageProperties, [925](#)
- ~vtkGDCMPolyDataReader
 - vtkGDCMPolyDataReader, [927](#)
- ~vtkGDCMPolyDataWriter
 - vtkGDCMPolyDataWriter, [930](#)
- ~vtkGDCMTesting
 - vtkGDCMTesting, [933](#)
- ~vtkGDCMThreadedImageReader
 - vtkGDCMThreadedImageReader, [935](#)
- ~vtkGDCMThreadedImageReader2
 - vtkGDCMThreadedImageReader2, [938](#)
- ~vtkImageColorViewer
 - vtkImageColorViewer, [943](#)
- ~vtkImageMapToColors16
 - vtkImageMapToColors16, [949](#)
- ~vtkImageMapToWindowLevelColors2
 - vtkImageMapToWindowLevelColors2, [952](#)
- ~vtkImagePlanarComponentsToComponents
 - vtkImagePlanarComponentsToComponents, [954](#)
- ~vtkImageRGBToYBR
 - vtkImageRGBToYBR, [955](#)
- ~vtkImageYBRToRGB
 - vtkImageYBRToRGB, [957](#)
- ~vtkLookupTable16
 - vtkLookupTable16, [958](#)
- ~vtkRTStructSetProperties
 - vtkRTStructSetProperties, [961](#)
- AAabortPDU
 - gdcm::network::AAabortPDU, [146](#)
- AAAssociateACPDU
 - gdcm::network::AAAssociateACPDU, [149](#)
 - gdcm::network::AAAssociateRQPDU, [155](#)
- AAAssociateRJPDU
 - gdcm::network::AAAssociateRJPDU, [151](#)
- AAAssociateRQPDU
 - gdcm::network::AAAssociateACPDU, [149](#)
 - gdcm::network::AAAssociateRQPDU, [153](#)
- AE
 - gdcm::VR, [900](#)
- AECOMP
 - gdcm, [128](#)
- AES128_CIPHER
 - gdcm::CryptographicMessageSyntax, [269](#)
- AES192_CIPHER
 - gdcm::CryptographicMessageSyntax, [270](#)
- AES256_CIPHER
 - gdcm::CryptographicMessageSyntax, [270](#)
- ALGOType
 - gdcm::Segment, [691](#)
- ALGOType_END
 - gdcm::Segment, [691](#)
- ARGB
 - gdcm::PhotometricInterpretation, [607](#)
- ARTIMTimer
 - gdcm::network::ARTIMTimer, [173](#)
- ARReleaseRPPDU
 - gdcm::network::ARReleaseRPPDU, [170](#)
- ARReleaseRQPDU
 - gdcm::network::ARReleaseRQPDU, [171](#)
- AS
 - gdcm::VR, [900](#)
- ASComp
 - gdcm, [128](#)
- ASN1
 - gdcm::ASN1, [174](#)
- AT
 - gdcm::VR, [900](#)
- AUTOMATIC
 - gdcm::Segment, [691](#)
- AXIAL
 - gdcm::Orientation, [581](#)
- AbstractSyntax
 - gdcm::PresentationContext, [632](#)
 - gdcm::network::AbstractSyntax, [157](#)
- ActiveComponent
 - vtkImageMapToColors16, [950](#)
- Add
 - gdcm::GroupDict, [419](#)
- AddAcceptedPresentationContext
 - gdcm::network::ULConnection, [865](#)
- AddCSAHeaderDictEntry
 - gdcm::CSAHeaderDict, [280](#)
- AddContourReferencedFrameOfReference
 - vtkRTStructSetProperties, [961](#)
- AddDerivationDescription
 - gdcm::FileDerivation, [386](#)
- AddDictEntry
 - gdcm::Dict, [321](#)
 - gdcm::PrivateDict, [643](#)
- AddFile
 - gdcm::FileSet, [402](#), [403](#)
 - gdcm::SerieHelper, [713](#)
- AddFileName
 - gdcm::SerieHelper, [713](#)
- AddFragment
 - gdcm::SequenceOfFragments, [703](#)
- AddFromFile
 - gdcm::PresentationContextGenerator, [634](#)
- AddGroupLength
 - gdcm::DictConverter, [324](#)
- AddIOD
 - gdcm::IODs, [475](#)
- AddIODEntry
 - gdcm::IOD, [472](#)
- AddImageDirectoryRecord
 - gdcm::DICOMDIRGenerator, [319](#)

- AddInput
 - vtkImageColorViewer, [943](#)
- AddInputConnection
 - vtkImageColorViewer, [943](#)
- AddItem
 - gdcm::SequenceOfItems, [708](#)
- AddMacro
 - gdcm::Macros, [515](#)
 - gdcm::Module, [540](#)
- AddMacroEntry
 - gdcm::Macro, [514](#)
- AddModule
 - gdcm::Modules, [544](#)
- AddModuleEntry
 - gdcm::Module, [540](#)
 - gdcm::NestedModuleEntries, [559](#)
- AddNewUndefinedLengthItem
 - gdcm::SequenceOfItems, [708](#)
- AddObserver
 - gdcm::Subject, [761](#), [762](#)
- AddPatientDirectoryRecord
 - gdcm::DICOMDIRGenerator, [319](#)
- AddPresentationContext
 - gdcm::PresentationContextGenerator, [634](#)
 - gdcm::network::AAAssociateRQPDU, [153](#)
- AddPresentationContextAC
 - gdcm::network::AAAssociateACPDU, [149](#)
- AddPresentationDataValue
 - gdcm::network::PDataTFPDU, [594](#)
- AddPrimitiveData
 - gdcm::MeshPrimitive, [532](#)
- AddPrivateTag
 - gdcm::Scanner, [686](#)
 - gdcm::StrictScanner, [751](#)
- AddPurposeOfReferenceCodeSequence
 - gdcm::FileDerivation, [386](#)
- AddQueryDataSet
 - gdcm::BaseQuery, [207](#)
- AddReference
 - gdcm::FileDerivation, [386](#)
- AddReferencedFrameOfReference
 - vtkRTStructSetProperties, [962](#)
- AddRestriction
 - gdcm::SerieHelper, [713](#)
- AddRoleSelectionSub
 - gdcm::network::UserInformation, [885](#)
- AddSOPClassExtendedNegociationSub
 - gdcm::network::UserInformation, [885](#)
- AddSegment
 - gdcm::SegmentWriter, [699](#)
- AddSelect
 - gdcm::Sorter, [734](#)
- AddSeriesDirectoryRecord
 - gdcm::DICOMDIRGenerator, [319](#)
- AddSkipTag
 - gdcm::Scanner, [686](#)
 - gdcm::StrictScanner, [751](#)
- AddSourceImageSequence
 - gdcm::FileDerivation, [386](#)
- AddStructureSetROI
 - vtkRTStructSetProperties, [962](#)
- AddStructureSetROIObservation
 - vtkRTStructSetProperties, [962](#)
- AddStudyDirectoryRecord
 - gdcm::DICOMDIRGenerator, [319](#)
- AddSurface
 - gdcm::Segment, [691](#)
- AddTag
 - gdcm::Scanner, [686](#)
 - gdcm::StrictScanner, [751](#)
- AddTransferSyntax
 - gdcm::PresentationContext, [631](#)
 - gdcm::network::PresentationContextRQ, [636](#)
- AffectedSOPClassUID
 - gdcm::network::CEchoRQ, [241](#)
- Allocate
 - gdcm::LookupTable, [510](#)
- AmbulatoryECGWaveformStorage
 - gdcm::MediaStorage, [522](#)
 - gdcm::UIDs, [821](#)
- AnatomicRegion
 - gdcm::Segment, [692](#)
- AnonymizeEvent
 - gdcm::AnonymizeEvent, [159](#)
- Anonymizer
 - gdcm::Anonymizer, [162](#)
- Append
 - gdcm::ByteValue, [234](#)
 - gdcm::Global, [416](#)
- AppendFrameEncode
 - gdcm::ImageCodec, [443](#)
 - gdcm::JPEG2000Codec, [490](#)
 - gdcm::JPEGCodec, [496](#)
 - gdcm::JPEGLSCodec, [500](#)
 - gdcm::RLECodec, [679](#)
- AppendImplementationClassUID
 - gdcm::FileMetaInformation, [392](#)
- AppendRowEncode
 - gdcm::ImageCodec, [443](#)
 - gdcm::JPEG2000Codec, [490](#)
 - gdcm::JPEGCodec, [496](#)
 - gdcm::JPEGLSCodec, [500](#)
 - gdcm::RLECodec, [679](#)
- AppendToDataElement
 - gdcm::FileStreamer, [405](#)
- AppendToGroupDataElement
 - gdcm::FileStreamer, [405](#)
- ApplicationContext

- gdcm::network::ApplicationContext, 166
- Apply
 - gdcm::ImageApplyLookupTable, 431
- ApplyInverseVideo
 - vtkGDCMImageReader, 912
 - vtkGDCMImageReader2, 918
- ApplyLookupTable
 - vtkGDCMImageReader, 912
 - vtkGDCMImageReader2, 918
- ApplyPlanarConfiguration
 - vtkGDCMImageReader, 912
 - vtkGDCMImageReader2, 918
- ApplyShiftScale
 - vtkGDCMImageReader, 912
 - vtkGDCMImageReader2, 918
- ApplyYBRToRGB
 - vtkGDCMImageReader, 912
 - vtkGDCMImageReader2, 918
- AreOverlaysInPixelData
 - gdcm::Bitmap, 219
 - gdcm::Pixmap, 615
- Area
 - gdcm::BoxRegion, 227
 - gdcm::Region, 673
- ArrayIncludeMacrosType
 - gdcm::Macro, 514
 - gdcm::Module, 539
- ArrayType
 - gdcm::Attribute, 177
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 182
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 188
- AsynchronousOperationsWindowSub
 - gdcm::network::AsynchronousOperationsWindow←Sub, 174
- Attribute
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 188
 - gdcm::terminal, 143
- Audio
 - gdcm::MediaStorage, 523
- AudioCodec
 - gdcm::AudioCodec, 197
- AudioSRStorageTrialRetired
 - gdcm::UIDs, 822
- AutoPixelMinMax
 - gdcm::IconImageGenerator, 423
- BALCPPProtect
 - gdcm::Anonymizer, 162
- BLUE
 - gdcm::LookupTable, 510
- BOOL_FUNCTION_PFILE_PFILE_POINTER
 - gdcm, 128
- backslash
 - gdcm, 131
- BadBigEndian
 - gdcm::SwapCode, 776
- BadLittleEndian
 - gdcm::SwapCode, 776
- BaseQuery
 - gdcm::BaseQuery, 207
- BaseRootQuery
 - gdcm::BaseRootQuery, 210
- BasicAnnotationBoxSOPClass
 - gdcm::UIDs, 820
- BasicApplicationLevelConfidentialityProfile
 - gdcm::Anonymizer, 162
- BasicCodedEntry
 - gdcm::SegmentHelper::BasicCodedEntry, 213
- BasicColorImageBoxSOPClass
 - gdcm::UIDs, 820
- BasicColorPrintManagementMetaSOPClass
 - gdcm::UIDs, 820
- BasicFilmBoxSOPClass
 - gdcm::UIDs, 820
- BasicFilmSessionSOPClass
 - gdcm::UIDs, 820
- BasicGrayscaleImageBoxSOPClass
 - gdcm::UIDs, 820
- BasicGrayscalePrintManagementMetaSOPClass
 - gdcm::UIDs, 820
- BasicOffsetTable
 - gdcm::BasicOffsetTable, 215
- BasicPrintImageOverlayBoxSOPClassRetired
 - gdcm::UIDs, 821
- BasicStudyContentNotificationSOPClassRetired
 - gdcm::UIDs, 820
- BasicTextSR
 - gdcm::MediaStorage, 522
- BasicTextSRStorage
 - gdcm::UIDs, 822
- BasicVoiceAudioWaveformStorage
 - gdcm::MediaStorage, 522
 - gdcm::UIDs, 821
- Begin
 - gdcm::CSAHeaderDict, 280
 - gdcm::DataSet, 304
 - gdcm::Dict, 321
 - gdcm::IODs, 475
 - gdcm::Scanner, 686
 - gdcm::SequenceOfFragments, 703
 - gdcm::SequenceOfItems, 708
 - gdcm::StrictScanner, 751
- BigEndian
 - gdcm::SwapCode, 776
- BitSample

- gdcm::JPEGCodec, [498](#)
- gdcm::LookupTable, [512](#)
- Bitmap
 - gdcm::Bitmap, [219](#)
 - gdcm::JPEG2000Codec, [491](#)
 - gdcm::PixelFormat, [613](#)
- BitmapToBitmapFilter
 - gdcm::BitmapToBitmapFilter, [225](#)
- black
 - gdcm::terminal, [143](#)
- BlendingSoftcopyPresentationStateStorageSOPClass
 - gdcm::UIDs, [822](#)
- blink
 - gdcm::terminal, [143](#)
- blue
 - gdcm::terminal, [143](#)
- BoundingBox
 - gdcm::BoxRegion, [227](#)
- BoxRegion
 - gdcm::BoxRegion, [227](#)
- BreakConnection
 - gdcm::network::ULConnectionManager, [872](#)
- BreakConnectionNow
 - gdcm::network::ULConnectionManager, [872](#)
- BreastImagingRelevantPatientInformationQuery
 - gdcm::UIDs, [823](#)
- BreastTomosynthesisImageStorage
 - gdcm::MediaStorage, [523](#)
 - gdcm::UIDs, [825](#)
- bright
 - gdcm::terminal, [143](#)
- Build
 - vtkLookupTable16, [958](#)
- ByteBuffer
 - gdcm::ByteBuffer, [229](#)
- ByteSwap
 - gdcm::ByteSwapFilter, [231](#)
- ByteSwapFilter
 - gdcm::ByteSwapFilter, [231](#)
- ByteValue
 - gdcm::ByteValue, [233](#)
- bytes
 - gdcm::Tag, [793](#)
- C_CANCEL_RQ
 - gdcm::network::DIMSE, [333](#)
- C_ECHO_RQ
 - gdcm::network::DIMSE, [333](#)
- C_ECHO_RSP
 - gdcm::network::DIMSE, [333](#)
- C_FIND_RQ
 - gdcm::network::DIMSE, [332](#)
- C_FIND_RSP
 - gdcm::network::DIMSE, [332](#)
- C_GET_RQ
 - gdcm::network::DIMSE, [332](#)
- C_GET_RSP
 - gdcm::network::DIMSE, [332](#)
- C_MOVE_RQ
 - gdcm::network::DIMSE, [332](#)
- C_MOVE_RSP
 - gdcm::network::DIMSE, [333](#)
- C_STORE_RQ
 - gdcm::network::DIMSE, [332](#)
- C_STORE_RSP
 - gdcm::network::DIMSE, [332](#)
- CALIBRATED
 - gdcm::Spacing, [736](#)
- CAPICryptoFactory
 - gdcm::CryptoFactory, [268](#)
- CAPICryptoFactory
 - gdcm::CAPICryptoFactory, [237](#)
- CAPICryptographicMessageSyntax
 - gdcm::CAPICryptographicMessageSyntax, [239](#)
- CEcho
 - gdcm::CompositeNetworkFunctions, [261](#)
- CFind
 - gdcm::CompositeNetworkFunctions, [263](#)
- CM
 - gdcm::SegmentHelper::BasicCodedEntry, [213](#)
- CMYK
 - gdcm::PhotometricInterpretation, [607](#)
- cMaxEventID
 - gdcm::network, [141](#)
- cMaxStateID
 - gdcm::network, [141](#)
- CMove
 - gdcm::CompositeNetworkFunctions, [263](#)
- CONDENSED_STYLE
 - gdcm::Printer, [641](#)
- CONSOLE
 - gdcm::terminal, [143](#)
- CORONAL
 - gdcm::Orientation, [581](#)
- CS
 - gdcm::VR, [900](#)
- CSAElement
 - gdcm::CSAElement, [273](#)
- CSAHeader
 - gdcm::CSAHeader, [277](#)
 - gdcm::DataSet, [308](#)
- CSAHeaderDict
 - gdcm::CSAHeaderDict, [280](#)
- CSAHeaderDictEntry
 - gdcm::CSAHeaderDictEntry, [282](#)
- CSAHeaderType
 - gdcm::CSAHeader, [277](#)
- CSANonImageStorage

- gdcmm::MediaStorage, 522
- CSComp
 - gdcmm, 128
- CSD
 - gdcmm::SegmentHelper::BasicCodedEntry, 213
- CSV
 - gdcmm::SegmentHelper::BasicCodedEntry, 213
- CStore
 - gdcmm::CompositeNetworkFunctions, 264
- CT_private_ELE
 - gdcmm::TransferSyntax, 805
- CTImageStorage
 - gdcmm::MediaStorage, 521
 - gdcmm::UIDs, 821
- CV
 - gdcmm::SegmentHelper::BasicCodedEntry, 214
- CanCode
 - gdcmm::AudioCodec, 197
 - gdcmm::Coder, 252
 - gdcmm::ImageCodec, 443
 - gdcmm::JPEG2000Codec, 490
 - gdcmm::JPEGCodec, 496
 - gdcmm::JPEGLSCodec, 501
 - gdcmm::KAKADUCoder, 505
 - gdcmm::PDFCodec, 600
 - gdcmm::PGXCodec, 605
 - gdcmm::PNMCodec, 627
 - gdcmm::PVRGCodec, 650
 - gdcmm::RAWCodec, 665
 - gdcmm::RLECodec, 680
- CanDecode
 - gdcmm::AudioCodec, 198
 - gdcmm::Decoder, 312
 - gdcmm::DeltaEncodingCodec, 317
 - gdcmm::ImageCodec, 444
 - gdcmm::JPEG2000Codec, 490
 - gdcmm::JPEGCodec, 496
 - gdcmm::JPEGLSCodec, 501
 - gdcmm::KAKADUCoder, 505
 - gdcmm::PDFCodec, 601
 - gdcmm::PGXCodec, 605
 - gdcmm::PNMCodec, 627
 - gdcmm::PVRGCodec, 650
 - gdcmm::RAWCodec, 665
 - gdcmm::RLECodec, 680
- CanDisplay
 - gdcmm::VR, 902
- CanEmptyTag
 - gdcmm::Anonymizer, 162
- CanRead
 - gdcmm::Reader, 669
- CanReadFile
 - vtkGDCMImageReader, 910
 - vtkGDCMImageReader2, 916
- CanReadImage
 - gdcmm::StreamImageReader, 741
- CanStoreLossy
 - gdcmm::TransferSyntax, 805
- CanWriteFile
 - gdcmm::StreamImageWriter, 745
- CardiacElectrophysiologyWaveformStorage
 - gdcmm::MediaStorage, 522
 - gdcmm::UIDs, 821
- CardiacRelevantPatientInformationQuery
 - gdcmm::UIDs, 824
- Change
 - gdcmm::FileChangeTransferSyntax, 382
 - gdcmm::FileDecompressLookupTable, 384
 - gdcmm::FileExplicitFilter, 388
 - gdcmm::ImageChangePhotometricInterpretation, 433
 - gdcmm::ImageChangePlanarConfiguration, 436
 - gdcmm::ImageChangeTransferSyntax, 439
- ChangeFMI
 - gdcmm::FileExplicitFilter, 389
- ChangeMonochrome
 - gdcmm::ImageChangePhotometricInterpretation, 433
- CharacterDataHandler
 - gdcmm::TableReader, 785
 - gdcmm::XMLDictReader, 973
 - gdcmm::XMLPrivateDictReader, 977
- CheckDataElement
 - gdcmm::FileStreamer, 405
- CheckEvent
 - gdcmm::AnonymizeEvent, 159
 - gdcmm::DataEvent, 301
 - gdcmm::DataSetEvent, 310
 - gdcmm::Event, 366
 - gdcmm::FileNameEvent, 399
 - gdcmm::ProgressEvent, 648
- CheckFileMetaInformationOff
 - gdcmm::Writer, 970
- CheckFileMetaInformationOn
 - gdcmm::Writer, 970
- CheckTemplateFileName
 - gdcmm::FileStreamer, 405
- ChestCADSRStorage
 - gdcmm::UIDs, 823
- CipherTypes
 - gdcmm::CryptographicMessageSyntax, 269
- Clear
 - gdcmm::Bitmap, 219
 - gdcmm::ByteValue, 234
 - gdcmm::DataElement, 293
 - gdcmm::DataSet, 304
 - gdcmm::IOD, 472
 - gdcmm::IODs, 475
 - gdcmm::Item, 481
 - gdcmm::LookupTable, 510

- gdcm::Macro, [514](#)
- gdcm::Macros, [515](#)
- gdcm::Module, [540](#)
- gdcm::Modules, [544](#)
- gdcm::Preamble, [629](#)
- gdcm::SequenceOfFragments, [703](#)
- gdcm::SequenceOfItems, [708](#)
- gdcm::SerieHelper, [713](#)
- gdcm::Value, [889](#)
- vtkGDCMMedicalImageProperties, [925](#)
- vtkRTStructSetProperties, [962](#)
- ClearInternalUIDs
 - gdcm::Anonymizer, [163](#)
- ClearSkipTags
 - gdcm::Scanner, [687](#)
 - gdcm::StrictScanner, [752](#)
- ClearTags
 - gdcm::Scanner, [687](#)
 - gdcm::StrictScanner, [752](#)
- Clone
 - gdcm::BoxRegion, [228](#)
 - gdcm::ImageCodec, [444](#)
 - gdcm::JPEG2000Codec, [490](#)
 - gdcm::JPEGCodec, [496](#)
 - gdcm::JPEGLSCodec, [501](#)
 - gdcm::KAKADUCodec, [505](#)
 - gdcm::PGXCodec, [606](#)
 - gdcm::PNMCodec, [627](#)
 - gdcm::PVRGCodec, [650](#)
 - gdcm::RAWCodec, [665](#)
 - gdcm::RLECodec, [680](#)
 - gdcm::Region, [674](#)
- Code
 - gdcm::Coder, [252](#)
 - gdcm::JPEG2000Codec, [490](#)
 - gdcm::JPEGCodec, [496](#)
 - gdcm::JPEGLSCodec, [501](#)
 - gdcm::JSON, [503](#)
 - gdcm::KAKADUCodec, [505](#)
 - gdcm::PVRGCodec, [650](#)
 - gdcm::RAWCodec, [665](#)
 - gdcm::RLECodec, [680](#)
- CodeMeaning
 - gdcm::RealWorldValueMappingContent, [672](#)
- CodeString
 - gdcm::CodeString, [254](#), [255](#)
- CodeValue
 - gdcm::RealWorldValueMappingContent, [672](#)
- Color
 - gdcm::terminal, [143](#)
- ColorArray
 - gdcm::SurfaceHelper, [770](#)
- ColorSoftcopyPresentationStateStorageSOPClass
 - gdcm::UIDs, [822](#)
- Command
 - gdcm::Command, [257](#)
- CommandDataSet
 - gdcm::CommandDataSet, [259](#)
- CommandTypes
 - gdcm::network::DIMSE, [332](#)
- CompOperators
 - gdcm, [129](#)
- Compatible
 - gdcm::VM, [897](#)
 - gdcm::VR, [902](#)
- Component
 - gdcm::PersonName, [604](#)
- ComprehensiveSR
 - gdcm::MediaStorage, [522](#)
- ComprehensiveSRStorage
 - gdcm::UIDs, [822](#)
- ComprehensiveSRStorageTrialRetired
 - gdcm::UIDs, [822](#)
- CompressionTypes
 - vtkGDCMImageWriter, [921](#)
- Compute
 - gdcm::MD5, [517](#)
 - gdcm::SHA1, [721](#)
- ComputeBoundingBox
 - gdcm::BoxRegion, [228](#)
 - gdcm::Region, [674](#)
- ComputeBufferLength
 - gdcm::ImageRegionReader, [460](#)
- ComputeByteLength
 - gdcm::SequenceOfFragments, [703](#)
- ComputeDataElement
 - gdcm::DataSet, [304](#)
- ComputeDataSetMediaStorageSOPClass
 - gdcm::FileMetaInformation, [392](#)
- ComputeDataSetTransferSyntax
 - gdcm::FileMetaInformation, [392](#)
- ComputeDistAlongNormal
 - gdcm::DirectionCosines, [334](#)
- ComputeFile
 - gdcm::MD5, [517](#)
 - gdcm::SHA1, [721](#)
- ComputeFileMD5
 - gdcm::Testing, [796](#)
- ComputeGroupLength
 - gdcm::DataSet, [305](#)
- ComputeInterceptSlopePixelFormatType
 - gdcm::Rescaler, [676](#)
- ComputeLength
 - gdcm::ByteValue, [234](#)
 - gdcm::Fragment, [415](#)
 - gdcm::SequenceOfFragments, [703](#)
 - gdcm::SequenceOfItems, [708](#)
- ComputeLossyFlag

- gdcm::Bitmap, 219
- ComputeMD5
 - gdcm::Testing, 796
- ComputeMOSAICDimensions
 - gdcm::SplitMosaicFilter, 738
- ComputeMediaStorageFromModality
 - gdcm::ImageHelper, 452
- ComputeNumberOfSurfaces
 - gdcm::SurfaceWriter, 775
- ComputeOffsetTable
 - gdcm::JPEGCodec, 497
- ComputePixelAspectRatioFromPixelSpacing
 - gdcm::Spacing, 736
- ComputePixelTypeFromMinMax
 - gdcm::Rescaler, 676
- ComputeSpacingFromImagePositionPatient
 - gdcm::ImageHelper, 452
- ComputeVR
 - gdcm::DataSetHelper, 311
- ComputeZSpacing
 - gdcm::IPPSorter, 479
- ComputedRadiographyImageStorage
 - gdcm::MediaStorage, 521
 - gdcm::UIDs, 821
- ConcatenatePDVBlobs
 - gdcm::network::PresentationDataValue, 638
- ConcatenatePDVBlobsAsExplicit
 - gdcm::network::PresentationDataValue, 638
- Conditional
 - gdcm::Usage, 883
- const
 - gdcm::SOPClassUIDToIOD, 731
- const_iterator
 - gdcm::CodeString, 254
 - gdcm::LO, 507
 - gdcm::String, 755
- const_reference
 - gdcm::CodeString, 254
 - gdcm::LO, 507
 - gdcm::String, 755
- const_reverse_iterator
 - gdcm::CodeString, 254
 - gdcm::LO, 507
 - gdcm::String, 756
- ConstCharWrapper
 - gdcm::ConstCharWrapper, 265
- ConstIterator
 - gdcm::CSAHeaderDict, 280
 - gdcm::DataSet, 304
 - gdcm::Dict, 321
 - gdcm::Scanner, 686
 - gdcm::SequenceOfFragments, 702
 - gdcm::SequenceOfItems, 708
 - gdcm::StrictScanner, 751

- Construct
 - gdcm::BaseRootQuery, 210
- ConstructAbortPDU
 - gdcm::network::PDUFactory, 602
- ConstructCEchoRQ
 - gdcm::network::CompositeMessageFactory, 260
- ConstructCFindRQ
 - gdcm::network::CompositeMessageFactory, 260
- ConstructCMoveRQ
 - gdcm::network::CompositeMessageFactory, 260
- ConstructCStoreRQ
 - gdcm::network::CompositeMessageFactory, 260
- ConstructCStoreRSP
 - gdcm::network::CompositeMessageFactory, 260
- ConstructFromString
 - gdcm::TagPath, 794
- ConstructFromTagList
 - gdcm::TagPath, 794
- ConstructNAction
 - gdcm::network::NormalizedMessageFactory, 566
- ConstructNCreate
 - gdcm::network::NormalizedMessageFactory, 566
- ConstructNDelete
 - gdcm::network::NormalizedMessageFactory, 566
- ConstructNEventReport
 - gdcm::network::NormalizedMessageFactory, 566
- ConstructNGet
 - gdcm::network::NormalizedMessageFactory, 566
- ConstructNSet
 - gdcm::network::NormalizedMessageFactory, 566
- ConstructPDU
 - gdcm::network::PDUFactory, 602
- ConstructPDV
 - gdcm::network::BaseCompositeMessage, 201
 - gdcm::network::BaseNormalizedMessage, 203
 - gdcm::network::CEchoRQ, 241
 - gdcm::network::CFindRQ, 245
 - gdcm::network::CMoveRQ, 248
 - gdcm::network::CStoreRQ, 285
 - gdcm::network::CStoreRSP, 286
 - gdcm::network::NActionRQ, 550
 - gdcm::network::NCreateRQ, 553
 - gdcm::network::NDeleteRQ, 555
 - gdcm::network::NEventReportRQ, 560
 - gdcm::network::NGetRQ, 563
 - gdcm::network::NSetRQ, 569
- ConstructPDVByDataSet
 - gdcm::network::CEchoRSP, 242
 - gdcm::network::CFindCancelRQ, 243
 - gdcm::network::CFindRSP, 246
 - gdcm::network::CMoveCancelRq, 247
 - gdcm::network::CMoveRSP, 250
 - gdcm::network::NActionRSP, 552
 - gdcm::network::NCreateRSP, 554

- gdcm::network::NDeleteRSP, [557](#)
- gdcm::network::NEventReportRSP, [562](#)
- gdcm::network::NGetRSP, [564](#)
- gdcm::network::NSetRSP, [570](#)
- ConstructQuery
 - gdcm::CompositeNetworkFunctions, [264](#)
 - gdcm::NormalizedNetworkFunctions, [567](#)
- ConstructReleasePDU
 - gdcm::network::PDUFactory, [602](#)
- ConstructorType
 - gdcm::Dicts, [330](#)
- Convert
 - gdcm::DictConverter, [324](#)
 - gdcm::ImageConverter, [448](#)
- ConvertRGBToPaletteColor
 - gdcm::IconImageGenerator, [423](#)
- ConvertToCXX
 - gdcm::DictConverter, [324](#)
- ConvertToXML
 - gdcm::DictConverter, [324](#)
- Create
 - gdcm::Preamble, [629](#)
- CreateCEchoPDU
 - gdcm::network::PDUFactory, [602](#)
- CreateCFindPDU
 - gdcm::network::PDUFactory, [602](#)
- CreateCMSProvider
 - gdcm::CAPICryptoFactory, [237](#)
 - gdcm::CryptoFactory, [268](#)
 - gdcm::OpenSSLCryptoFactory, [574](#)
 - gdcm::OpenSSLPT7CryptoFactory, [578](#)
- CreateCMovePDU
 - gdcm::network::PDUFactory, [602](#)
- CreateCStoreRQPDU
 - gdcm::network::PDUFactory, [602](#)
- CreateCStoreRSPPDU
 - gdcm::network::PDUFactory, [602](#)
- CreateDefaultUniqueSeriesIdentifier
 - gdcm::SerieHelper, [713](#)
- CreateNActionPDU
 - gdcm::network::PDUFactory, [602](#)
- CreateNCreatePDU
 - gdcm::network::PDUFactory, [602](#)
- CreateNDeletePDU
 - gdcm::network::PDUFactory, [602](#)
- CreateNEventReportPDU
 - gdcm::network::PDUFactory, [602](#)
- CreateNGetPDU
 - gdcm::network::PDUFactory, [602](#)
- CreateNSetPDU
 - gdcm::network::PDUFactory, [602](#)
- CreateUniqueSeriesIdentifier
 - gdcm::SerieHelper, [713](#)
- Cross
 - gdcm::DirectionCosines, [334](#)
- CrossDot
 - gdcm::DirectionCosines, [334](#)
- CryptoFactory
 - gdcm::CryptoFactory, [268](#)
- CryptoLib
 - gdcm::CryptoFactory, [268](#)
- CryptographicMessageSyntax
 - gdcm::CryptographicMessageSyntax, [270](#)
- Curve
 - gdcm::Curve, [288](#)
 - vtkGDCMImageReader, [912](#)
 - vtkGDCMImageReader2, [918](#)
- Curves
 - gdcm::Pixmap, [616](#)
- cyan
 - gdcm::terminal, [143](#)
- DA
 - gdcm::VR, [900](#)
- DAComp
 - gdcm, [128](#)
- DATASET_FORMAT
 - gdcm::CSAHeader, [277](#)
- DEFAULT
 - gdcm::CryptoFactory, [268](#)
- DES3_CIPHER
 - gdcm::CryptographicMessageSyntax, [269](#)
- DETECTOR
 - gdcm::Spacing, [736](#)
- DICOMApplicationContextName
 - gdcm::UIDs, [820](#)
- DICOMControlledTerminology
 - gdcm::UIDs, [820](#)
- DICOMDIR
 - gdcm::DICOMDIR, [318](#)
- DICOMDIRGenerator
 - gdcm::DICOMDIRGenerator, [319](#)
- DICOMUIDRegistry
 - gdcm::UIDs, [820](#)
- DICT_DEBUG
 - gdcm::DictConverter, [324](#)
- DICT_DEFAULT
 - gdcm::DictConverter, [324](#)
- DICT_XML
 - gdcm::DictConverter, [324](#)
- DS
 - gdcm::VR, [901](#)
- DT
 - gdcm::VR, [901](#)
- DTComp
 - gdcm, [129](#)
- DataElement
 - gdcm::DataElement, [293](#)

- gdcmm::Value, [890](#)
- DataElementSet
 - gdcmm::DataSet, [304](#)
- DataElementType
 - gdcmm::ModuleEntry, [543](#)
- DataEvent
 - gdcmm::DataEvent, [300](#), [301](#)
- DataField
 - gdcmm::CSAElement, [275](#)
- DataPtr
 - gdcmm::CSAElement, [273](#)
- DataSetEvent
 - gdcmm::DataSetEvent, [310](#)
- DataSetHandled
 - gdcmm::network::ULConnectionCallback, [868](#)
- DataSetHandles
 - gdcmm::network::ULConnectionCallback, [868](#)
- DataSetMS
 - gdcmm::FileMetaInformation, [394](#)
- DataSetTS
 - gdcmm::FileMetaInformation, [394](#)
- DataWasPassed
 - vtkImageMapToColors16, [950](#)
- DebugOff
 - gdcmm::Trace, [800](#)
- DebugOn
 - gdcmm::Trace, [801](#)
- Decode
 - gdcmm::AudioCodec, [198](#)
 - gdcmm::Base64, [198](#)
 - gdcmm::Curve, [288](#)
 - gdcmm::Decoder, [312](#)
 - gdcmm::DeltaEncodingCodec, [317](#)
 - gdcmm::ImageCodec, [444](#)
 - gdcmm::JPEG2000Codec, [490](#)
 - gdcmm::JPEGCodec, [497](#)
 - gdcmm::JPEGLSCodec, [501](#)
 - gdcmm::JSON, [503](#)
 - gdcmm::KAKADUCodec, [505](#)
 - gdcmm::LookupTable, [510](#)
 - gdcmm::PDFCodec, [601](#)
 - gdcmm::PVRGCodec, [651](#)
 - gdcmm::RAWCodec, [665](#)
 - gdcmm::RLECodec, [680](#)
- DecodeByStreams
 - gdcmm::Decoder, [312](#)
 - gdcmm::ImageCodec, [444](#)
 - gdcmm::JPEG12Codec, [485](#)
 - gdcmm::JPEG16Codec, [487](#)
 - gdcmm::JPEG2000Codec, [490](#)
 - gdcmm::JPEG8Codec, [493](#)
 - gdcmm::JPEGCodec, [497](#)
 - gdcmm::RAWCodec, [665](#)
 - gdcmm::RLECodec, [680](#)
- DecodeBytes
 - gdcmm::RAWCodec, [666](#)
- DecodeExtent
 - gdcmm::JPEG2000Codec, [490](#)
 - gdcmm::JPEGCodec, [497](#)
 - gdcmm::JPEGLSCodec, [501](#)
 - gdcmm::RLECodec, [680](#)
- Decompress
 - gdcmm::Overlay, [586](#)
- Decrypt
 - gdcmm::CAPICryptographicMessageSyntax, [239](#)
 - gdcmm::CryptographicMessageSyntax, [270](#)
 - gdcmm::OpenSSLCryptographicMessageSyntax, [576](#)
 - gdcmm::OpenSSLP7CryptographicMessageSyntax, [579](#)
- DeepCopy
 - vtkRTStructSetProperties, [962](#)
- Default
 - gdcmm::FileMetaInformation, [392](#)
- DefinePixelExtent
 - gdcmm::StreamImageReader, [741](#)
 - gdcmm::StreamImageWriter, [746](#)
- DefineProperBufferLength
 - gdcmm::StreamImageReader, [742](#)
 - gdcmm::StreamImageWriter, [746](#)
- DefinedTerms
 - gdcmm::DefinedTerms, [313](#)
- DeflatedExplicitVRLittleEndian
 - gdcmm::TransferSyntax, [804](#)
 - gdcmm::UIDs, [818](#)
- DeformableSpatialRegistrationStorage
 - gdcmm::UIDs, [822](#)
- Defs
 - gdcmm::Defs, [314](#)
- DeleteDirectory
 - gdcmm::System, [779](#)
- DeltaEncodingCodec
 - gdcmm::DeltaEncodingCodec, [317](#)
- Derive
 - gdcmm::FileDerivation, [386](#)
- Description
 - gdcmm::ModuleEntry, [542](#)
- DescriptionField
 - gdcmm::ModuleEntry, [543](#)
- DetachedInterpretationManagementSOPClassRetired
 - gdcmm::UIDs, [820](#)
- DetachedPatientManagementMetaSOPClassRetired
 - gdcmm::UIDs, [820](#)
- DetachedPatientManagementSOPClass
 - gdcmm::MediaStorage, [522](#)
- DetachedPatientManagementSOPClassRetired
 - gdcmm::UIDs, [820](#)
- DetachedResultsManagementMetaSOPClassRetired
 - gdcmm::UIDs, [820](#)

- DetachedResultsManagementSOPClassRetired
 - gdcm::UIDs, [820](#)
- DetachedStudyManagementMetaSOPClassRetired
 - gdcm::UIDs, [820](#)
- DetachedStudyManagementSOPClass
 - gdcm::MediaStorage, [522](#)
- DetachedStudyManagementSOPClassRetired
 - gdcm::UIDs, [820](#)
- DetachedVisitManagementSOPClass
 - gdcm::MediaStorage, [522](#)
- DetachedVisitManagementSOPClassRetired
 - gdcm::UIDs, [820](#)
- DetailSRStorageTrialRetired
 - gdcm::UIDs, [822](#)
- DetermineEventByPDU
 - gdcm::network::PDUFactory, [602](#)
- dicomAETitle
 - gdcm::UIDs, [824](#)
- dicomApplicationCluster
 - gdcm::UIDs, [824](#)
- dicomAssociationAcceptor
 - gdcm::UIDs, [824](#)
- dicomAssociationInitiator
 - gdcm::UIDs, [824](#)
- dicomAuthorizedNodeCertificateReference
 - gdcm::UIDs, [824](#)
- dicomConfigurationRoot
 - gdcm::UIDs, [824](#)
- dicomDescription
 - gdcm::UIDs, [824](#)
- dicomDevice
 - gdcm::UIDs, [824](#)
- dicomDeviceName
 - gdcm::UIDs, [824](#)
- dicomDeviceSerialNumber
 - gdcm::UIDs, [824](#)
- dicomDevicesRoot
 - gdcm::UIDs, [824](#)
- dicomHostname
 - gdcm::UIDs, [824](#)
- dicomInstalled
 - gdcm::UIDs, [824](#)
- dicomInstitutionAddress
 - gdcm::UIDs, [824](#)
- dicomInstitutionDepartmentName
 - gdcm::UIDs, [824](#)
- dicomInstitutionName
 - gdcm::UIDs, [824](#)
- dicomIssuerOfPatientID
 - gdcm::UIDs, [824](#)
- dicomManufacturer
 - gdcm::UIDs, [824](#)
- dicomManufacturerModelName
 - gdcm::UIDs, [824](#)
- dicomNetworkAE
 - gdcm::UIDs, [824](#)
- dicomNetworkConnection
 - gdcm::UIDs, [825](#)
- dicomNetworkConnectionReference
 - gdcm::UIDs, [824](#)
- dicomPort
 - gdcm::UIDs, [824](#)
- dicomPreferredCalledAETitle
 - gdcm::UIDs, [824](#)
- dicomPreferredCallingAETitle
 - gdcm::UIDs, [824](#)
- dicomPrimaryDeviceType
 - gdcm::UIDs, [824](#)
- dicomRelatedDeviceReference
 - gdcm::UIDs, [824](#)
- dicomSOPClass
 - gdcm::UIDs, [824](#)
- dicomSoftwareVersion
 - gdcm::UIDs, [824](#)
- dicomStationName
 - gdcm::UIDs, [824](#)
- dicomSupportedCharacterSet
 - gdcm::UIDs, [824](#)
- dicomTLSCyphersuite
 - gdcm::UIDs, [824](#)
- dicomThisNodeCertificateReference
 - gdcm::UIDs, [824](#)
- dicomTransferCapability
 - gdcm::UIDs, [825](#)
- dicomTransferRole
 - gdcm::UIDs, [824](#)
- dicomTransferSyntax
 - gdcm::UIDs, [824](#)
- dicomUniqueAETitle
 - gdcm::UIDs, [825](#)
- dicomUniqueAETitlesRegistryRoot
 - gdcm::UIDs, [824](#)
- dicomVendorData
 - gdcm::UIDs, [824](#)
- Dict
 - gdcm::Dict, [321](#)
 - gdcm::DictEntry, [327](#)
- DictConverter
 - gdcm::DictConverter, [324](#)
- DictEntry
 - gdcm::DictEntry, [326](#)
- DictPrinter
 - gdcm::DictPrinter, [329](#)
- Dicts
 - gdcm::CSAHeaderDict, [281](#)
 - gdcm::Dict, [322](#)
 - gdcm::Dicts, [330](#)
 - gdcm::PrivateDict, [644](#)

- difference_type
 - gdcm::CodeString, [254](#)
 - gdcm::LO, [507](#)
 - gdcm::String, [756](#)
- DigitalIntraoralXRayImageStorageForPresentation
 - gdcm::UIDs, [821](#)
- DigitalIntraoralXRayImageStorageForProcessing
 - gdcm::MediaStorage, [521](#)
 - gdcm::UIDs, [821](#)
- DigitalIntraoralXrayImageStorageForPresentation
 - gdcm::MediaStorage, [521](#)
- DigitalMammographyImageStorageForPresentation
 - gdcm::MediaStorage, [521](#)
- DigitalMammographyImageStorageForProcessing
 - gdcm::MediaStorage, [521](#)
- DigitalMammographyXRayImageStorageForPresentation
 - gdcm::UIDs, [821](#)
- DigitalMammographyXRayImageStorageForProcessing
 - gdcm::UIDs, [821](#)
- DigitalXRayImageStorageForPresentation
 - gdcm::MediaStorage, [521](#)
 - gdcm::UIDs, [821](#)
- DigitalXRayImageStorageForProcessing
 - gdcm::MediaStorage, [521](#)
 - gdcm::UIDs, [821](#)
- dim
 - gdcm::terminal, [143](#)
- Dimensions
 - gdcm::Bitmap, [223](#)
 - gdcm::ImageCodec, [447](#)
- DirCosTolerance
 - gdcm::IPPSorter, [479](#)
- DirectionCosines
 - gdcm::DirectionCosines, [334](#)
 - vtkGDCMImageReader, [912](#)
 - vtkGDCMImageReader2, [918](#)
- Directory
 - gdcm::Directory, [336](#)
- DoByteSwap
 - gdcm::ImageCodec, [444](#)
- DolconImage
 - gdcm::PixmapWriter, [624](#)
- DoInvertMonochrome
 - gdcm::ImageCodec, [444](#)
- DoOverlayCleanup
 - gdcm::ImageCodec, [444](#)
- DoPaddedCompositePixelCode
 - gdcm::ImageCodec, [444](#)
- DoPlanarConfiguration
 - gdcm::ImageCodec, [444](#)
- DoSimpleCopy
 - gdcm::ImageCodec, [444](#)
- DoYBR
 - gdcm::ImageCodec, [444](#)
- Dot
 - gdcm::DirectionCosines, [334](#)
- DropDuplicatePositions
 - gdcm::IPPSorter, [479](#)
- Dumper
 - gdcm::Dumper, [341](#)
- DuplicateAttributeError
 - gdcm::Parser, [591](#)
- eAABORTPDUReturnedOpen
 - gdcm::network, [140](#)
- eAABORTRequest
 - gdcm::network, [140](#)
- eAASSOCIATE_RQPDUReturned
 - gdcm::network, [140](#)
- eAASSOCIATERequestLocalUser
 - gdcm::network, [140](#)
- eAASSOCIATEResponseAccept
 - gdcm::network, [140](#)
- eAASSOCIATEResponseReject
 - gdcm::network, [140](#)
- eARELEASE_RPPDUReturned
 - gdcm::network, [140](#)
- eARELEASE_RQPDUReturnedOpen
 - gdcm::network, [140](#)
- eARELEASERequest
 - gdcm::network, [140](#)
- eARELEASEResponse
 - gdcm::network, [140](#)
- eARTIMTimerExpired
 - gdcm::network, [141](#)
- eASSOCIATE_ACPDUReturned
 - gdcm::network, [140](#)
- eASSOCIATE_RJPDUReturned
 - gdcm::network, [140](#)
- eArabic
 - gdcm, [130](#)
- ECharSet
 - gdcm, [129](#)
- eCreateMMPS
 - gdcm, [130](#)
- eCyrillic
 - gdcm, [130](#)
- EDGE
 - gdcm::MeshPrimitive, [531](#)
- eEventDoesNotExist
 - gdcm::network, [141](#)
- EEventID
 - gdcm::network, [140](#)
- eFind
 - gdcm, [130](#)
- eGB18030
 - gdcm, [130](#)
- eGreek

- gdcM, 130
- eHebrew
 - gdcM, 130
- eImage
 - gdcM, 130
- eJapanese
 - gdcM, 130
- eJapaneseKanjiMultibyte
 - gdcM, 130
- eJapaneseSupplementaryKanjiMultibyte
 - gdcM, 130
- eKoreanHangulHanjaMultibyte
 - gdcM, 130
- eLatin1
 - gdcM, 129
- eLatin2
 - gdcM, 129
- eLatin3
 - gdcM, 129
- eLatin4
 - gdcM, 129
- eLatin5
 - gdcM, 130
- eMove
 - gdcM, 130
- ENQueryType
 - gdcM, 130
- ePDATATFPDU
 - gdcM::network, 140
- ePDATArequest
 - gdcM::network, 140
- ePatient
 - gdcM, 130
- ePatientRootType
 - gdcM, 130
- EQueryLevel
 - gdcM, 130
- EQueryType
 - gdcM, 130
- ERootType
 - gdcM, 130
- eSeries
 - gdcM, 130
- eSetMMPS
 - gdcM, 130
- eSta10ReleaseCollisionAc
 - gdcM::network, 141
- eSta11ReleaseCollisionRq
 - gdcM::network, 141
- eSta12ReleaseCollisionAcLocal
 - gdcM::network, 141
- eSta13AwaitingClose
 - gdcM::network, 141
- eSta1Idle
 - gdcM::network, 141
- eSta2Open
 - gdcM::network, 141
- eSta3WaitLocalAssoc
 - gdcM::network, 141
- eSta4LocalAssocDone
 - gdcM::network, 141
- eSta5WaitRemoteAssoc
 - gdcM::network, 141
- eSta6TransferReady
 - gdcM::network, 141
- eSta7WaitRelease
 - gdcM::network, 141
- eSta8WaitLocalRelease
 - gdcM::network, 141
- eSta9ReleaseCollisionRqLocal
 - gdcM::network, 141
- eStaDoesNotExist
 - gdcM::network, 141
- EStateID
 - gdcM::network, 141
- eStudy
 - gdcM, 130
- eStudyRootType
 - gdcM, 130
- eThai
 - gdcM, 130
- eTransportConnConfirmLocal
 - gdcM::network, 140
- eTransportConnIndicLocal
 - gdcM::network, 140
- eTransportConnectionClosed
 - gdcM::network, 140
- eUTF8
 - gdcM, 130
- eUnrecognizedPDUReceived
 - gdcM::network, 141
- eWLMFind
 - gdcM, 130
- elem
 - gdcM::SerieHelper::Rule, 683
- Element
 - gdcM::Element< TVR, VM::VM1_n >, 347
- Empty
 - gdcM::Anonymizer, 163
 - gdcM::BoxRegion, 228
 - gdcM::DataElement, 293
 - gdcM::FileAnonymizer, 379
 - gdcM::Region, 674
- EncapsulatedCDASStorage
 - gdcM::MediaStorage, 522
 - gdcM::UIDs, 823
- EncapsulatedDocument
 - gdcM::EncapsulatedDocument, 359

- EncapsulatedPDFStorage
 - gdcm::MediaStorage, [522](#)
 - gdcm::UIDs, [823](#)
- Encode
 - gdcm::Base64, [199](#)
- EncodeBuffer
 - gdcm::JPEG12Codec, [485](#)
 - gdcm::JPEG16Codec, [487](#)
 - gdcm::JPEG8Codec, [493](#)
 - gdcm::JPEGCodec, [497](#)
- EncodeBytes
 - gdcm::System, [779](#)
- Encrypt
 - gdcm::CAPICryptographicMessageSyntax, [239](#)
 - gdcm::CryptographicMessageSyntax, [270](#)
 - gdcm::OpenSSLCryptographicMessageSyntax, [576](#)
 - gdcm::OpenSSLP7CryptographicMessageSyntax, [580](#)
- End
 - gdcm::CSAHeaderDict, [281](#)
 - gdcm::DataSet, [305](#)
 - gdcm::Dict, [321](#)
 - gdcm::IODs, [475](#)
 - gdcm::Scanner, [687](#)
 - gdcm::SequenceOfFragments, [703](#)
 - gdcm::SequenceOfItems, [709](#)
 - gdcm::StrictScanner, [752](#)
- EndElement
 - gdcm::TableReader, [785](#)
 - gdcm::XMLDictReader, [973](#)
 - gdcm::XMLPrivateDictReader, [977](#)
- EndElementHandler
 - gdcm::Parser, [591](#)
- EndFilter
 - gdcm::SimpleSubjectWatcher, [725](#)
- EndWith
 - gdcm::Filename, [396](#)
- EnhancedCTImageStorage
 - gdcm::MediaStorage, [521](#)
 - gdcm::UIDs, [821](#)
- EnhancedMRImageStorage
 - gdcm::MediaStorage, [521](#)
 - gdcm::UIDs, [821](#)
- EnhancedPETImageStorage
 - gdcm::MediaStorage, [523](#)
- EnhancedSR
 - gdcm::MediaStorage, [522](#)
- EnhancedSRStorage
 - gdcm::UIDs, [822](#)
- EnhancedUSVolumeStorage
 - gdcm::MediaStorage, [523](#)
 - gdcm::UIDs, [825](#)
- EnhancedXAImageStorage
 - gdcm::MediaStorage, [523](#)
- gdcm::UIDs, [822](#)
- EnhancedXRImageStorage
 - gdcm::UIDs, [822](#)
- EnumeratedValues
 - gdcm::EnumeratedValues, [364](#)
- ErrorOff
 - gdcm::Trace, [801](#)
- ErrorOn
 - gdcm::Trace, [801](#)
- ErrorType
 - gdcm::Parser, [591](#)
- EstablishConnection
 - gdcm::network::ULConnectionManager, [872](#)
- EstablishConnectionMove
 - gdcm::network::ULConnectionManager, [872](#)
- Event
 - gdcm::Event, [366](#)
- Exception
 - gdcm::Exception, [368](#)
- Execute
 - gdcm::Command, [257](#)
 - gdcm::MemberCommand, [528](#)
 - gdcm::SimpleMemberCommand, [724](#)
- ExecuteData
 - vtkGDCMImageReader, [910](#)
 - vtkGDCMThreadedImageReader, [936](#)
- ExecuteInformation
 - vtkGDCMImageReader, [910](#)
 - vtkGDCMThreadedImageReader, [936](#)
- ExecuteQuery
 - gdcm::StringFilter, [758](#)
- Explicit
 - gdcm::TransferSyntax, [804](#)
- ExplicitVRBigEndian
 - gdcm::TransferSyntax, [804](#)
 - gdcm::UIDs, [818](#)
- ExplicitVRLittleEndian
 - gdcm::TransferSyntax, [804](#)
 - gdcm::UIDs, [818](#)
- Explore
 - gdcm::Directory, [336](#)
- Extract
 - gdcm::IconImageFilter, [421](#)
- ExtractIconImages
 - gdcm::IconImageFilter, [421](#)
- ExtractVeprolIconImages
 - gdcm::IconImageFilter, [421](#)
- F
 - gdcm::Printer, [642](#)
 - gdcm::Reader, [671](#)
 - gdcm::Validate, [888](#)
 - gdcm::XMLPrinter, [975](#)
- FACET

- gdcmmesh::MeshPrimitive, 531
- FD
 - gdcmmesh::VR, 901
- FL
 - gdcmmesh::VR, 901
- FLOAT16
 - gdcmmesh::PixelFormat, 611
- FLOAT32
 - gdcmmesh::PixelFormat, 611
- FLOAT64
 - gdcmmesh::PixelFormat, 611
- Fiducials
 - gdcmmesh::Fiducials, 374
- File
 - gdcmmesh::File, 376
- FileAnonymizer
 - gdcmmesh::FileAnonymizer, 379
- FileChangeTransferSyntax
 - gdcmmesh::FileChangeTransferSyntax, 382
 - gdcmmesh::ImageCodec, 446
- FileDecompressLookupTable
 - gdcmmesh::FileDecompressLookupTable, 384
- FileDerivation
 - gdcmmesh::FileDerivation, 386
- FileExists
 - gdcmmesh::System, 779
- FileExplicitFilter
 - gdcmmesh::FileExplicitFilter, 388
- FilesDirectory
 - gdcmmesh::System, 779
- FilesSymlink
 - gdcmmesh::System, 780
- FileList
 - gdcmmesh, 129
- FileMetaInformation
 - gdcmmesh::FileMetaInformation, 392
- FileName
 - vtkGDCMPolyDataReader, 928
- FileNameEvent
 - gdcmmesh::FileNameEvent, 399
- FileNameOrdering
 - gdcmmesh::SerieHelper, 713
- FileNames
 - vtkGDCMImageReader, 912
- FileSet
 - gdcmmesh::FileSet, 402
- FileSize
 - gdcmmesh::System, 780
- FileStreamer
 - gdcmmesh::FileStreamer, 405
- FileTime
 - gdcmmesh::System, 780
- FileType
 - gdcmmesh::FileSet, 402
- FileWithName
 - gdcmmesh::FileWithName, 408
- Filename
 - gdcmmesh::Filename, 396
- filename
 - gdcmmesh::FileWithName, 408
- FilenameGenerator
 - gdcmmesh::FilenameGenerator, 400
- FilenameType
 - gdcmmesh::DICOMDIRGenerator, 319
 - gdcmmesh::Directory, 336
 - gdcmmesh::FilenameGenerator, 400
- Filenames
 - gdcmmesh::Sorter, 735
- FilenamesType
 - gdcmmesh::DICOMDIRGenerator, 319
 - gdcmmesh::Directory, 336
 - gdcmmesh::FilenameGenerator, 400
- FileType
 - gdcmmesh::FileSet, 402
- Fill
 - gdcmmesh::ByteValue, 234
- FillFromDataSet
 - gdcmmesh::FileMetaInformation, 392
- FillMedicalImageInformation
 - vtkGDCMImageReader, 910
 - vtkGDCMImageReader2, 916
 - vtkGDCMPolyDataReader, 927
- FindCSAElementByName
 - gdcmmesh::CSAHeader, 277
- FindContext
 - gdcmmesh::network::ULConnection, 865
- FindDataElement
 - gdcmmesh::DataSet, 305
 - gdcmmesh::Item, 481
 - gdcmmesh::SequenceOfItems, 709
- FindDictEntry
 - gdcmmesh::PrivateDict, 643
- FindMacroEntry
 - gdcmmesh::Macro, 514
- FindModuleEntryInMacros
 - gdcmmesh::Module, 540
- FindNextDataElement
 - gdcmmesh::DataSet, 305
- FindPDBelementByName
 - gdcmmesh::PDBHeader, 598
- FindPatientRootQuery
 - gdcmmesh::FindPatientRootQuery, 410
- FindStudyRootQuery
 - gdcmmesh::FindStudyRootQuery, 412
- FirstRender
 - vtkImageColorViewer, 946
- ForceRescale
 - vtkGDCMImageReader, 912

- vtkGDCMImageReader2, [918](#)
- FormatDateTime
 - gdcm::System, [780](#)
- Fragment
 - gdcm::Fragment, [415](#)
- FragmentVector
 - gdcm::SequenceOfFragments, [702](#)
- FromString
 - gdcm::StringFilter, [758](#)
- FujiPrivateCRImageStorage
 - gdcm::MediaStorage, [523](#)
- GDCM_DIFFERENT
 - gdcm, [129](#)
- GDCM_DO_JOIN
 - gdcmStaticAssert.h, [1198](#)
- GDCM_DO_JOIN2
 - gdcmStaticAssert.h, [1198](#)
- GDCM_EQUAL
 - gdcm, [129](#)
- GDCM_EXPORT
 - gdcmWin32.h, [1254](#)
- GDCM_FUNCTION
 - gdcmTrace.h, [1221](#)
- GDCM_GREATER
 - gdcm, [129](#)
- GDCM_GREATEROREQUAL
 - gdcm, [129](#)
- GDCM_JOIN
 - gdcmStaticAssert.h, [1198](#)
- GDCM_LEGACY
 - gdcmLegacyMacro.h, [1104](#)
- GDCM_LEGACY_BODY
 - gdcmLegacyMacro.h, [1104](#)
- GDCM_LEGACY_REPLACED_BODY
 - gdcmLegacyMacro.h, [1105](#)
- GDCM_LESS
 - gdcm, [129](#)
- GDCM_LESOREQUAL
 - gdcm, [129](#)
- GDCM_STATIC_ASSERT
 - gdcm::Attribute, [177](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [182](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, [188](#)
 - gdcmStaticAssert.h, [1198](#)
- GDCMMACROENTRY_H
 - gdcmMacroEntry.h, [1109](#)
- GEMS
 - gdcm::Dicts, [330](#)
- GEPrivate3DModelStorage
 - gdcm::MediaStorage, [522](#)
- GRAY
 - gdcm::LookupTable, [510](#)
- GREEN
 - gdcm::LookupTable, [510](#)
- gdcm, [113](#)
 - AECComp, [128](#)
 - ASComp, [128](#)
 - BOOL_FUNCTION_PFILE_PFILE_POINTER, [128](#)
 - backslash, [131](#)
 - CSComp, [128](#)
 - CompOperators, [129](#)
 - DAComp, [128](#)
 - DTComp, [129](#)
 - eArabic, [130](#)
 - ECharSet, [129](#)
 - eCreateMMPS, [130](#)
 - eCyrillic, [130](#)
 - eFind, [130](#)
 - eGB18030, [130](#)
 - eGreek, [130](#)
 - eHebrew, [130](#)
 - eImage, [130](#)
 - eJapanese, [130](#)
 - eJapaneseKanjiMultibyte, [130](#)
 - eJapaneseSupplementaryKanjiMultibyte, [130](#)
 - eKoreanHangulHanjaMultibyte, [130](#)
 - eLatin1, [129](#)
 - eLatin2, [129](#)
 - eLatin3, [129](#)
 - eLatin4, [129](#)
 - eLatin5, [130](#)
 - eMove, [130](#)
 - ENQueryType, [130](#)
 - ePatient, [130](#)
 - ePatientRootType, [130](#)
 - EQueryLevel, [130](#)
 - EQueryType, [130](#)
 - ERootType, [130](#)
 - eSeries, [130](#)
 - eSetMMPS, [130](#)
 - eStudy, [130](#)
 - eStudyRootType, [130](#)
 - eThai, [130](#)
 - eUTF8, [130](#)
 - eWLMFind, [130](#)
- FileList, [129](#)
- GDCM_DIFFERENT, [129](#)
- GDCM_EQUAL, [129](#)
- GDCM_GREATER, [129](#)
- GDCM_GREATEROREQUAL, [129](#)
- GDCM_LESS, [129](#)
- GDCM_LESOREQUAL, [129](#)
- GetVRFromTag, [131](#)
- GlobalInstance, [135](#)
- IconImage, [129](#)

- LD_ALL, [131](#)
- LD_NOSEQ, [131](#)
- LD_NOSHADOW, [131](#)
- LD_NOSHADOWSEQ, [131](#)
- LOComp, [129](#)
- LTComp, [129](#)
- LodModeType, [130](#)
- MacroEntry, [129](#)
- NestedMacroEntries, [129](#)
- operator!=, [131](#)
- operator<<, [131–135](#)
- operator>>, [135](#)
- operator==, [135](#)
- PNComp, [129](#)
- SHComp, [129](#)
- STComp, [129](#)
- TMComp, [129](#)
- TYPETOENCODING, [135](#)
- to_string, [135](#)
- UIComp, [129](#)
- UTComp, [129](#)
- VRBINARY, [135](#)
- gdcm2pnm.dox, [979](#)
- gdcm2vtk.dox, [979](#)
- gdcm::ASN1, [173](#)
 - ~ASN1, [174](#)
 - ASN1, [174](#)
 - ParseDump, [174](#)
 - ParseDumpFile, [174](#)
 - TestPBKDF2, [174](#)
- gdcm::AbortEvent, [155](#)
- gdcm::AnonymizeEvent, [157](#)
 - ~AnonymizeEvent, [159](#)
 - AnonymizeEvent, [159](#)
 - CheckEvent, [159](#)
 - GetEventName, [159](#)
 - GetTag, [159](#)
 - MakeObject, [159](#)
 - Self, [159](#)
 - SetTag, [159](#)
 - Superclass, [159](#)
- gdcm::Anonymizer, [160](#)
 - ~Anonymizer, [162](#)
 - Anonymizer, [162](#)
 - BALCPPProtect, [162](#)
 - BasicApplicationLevelConfidentialityProfile, [162](#)
 - CanEmptyTag, [162](#)
 - ClearInternalUIDs, [163](#)
 - Empty, [163](#)
 - GetBasicApplicationLevelConfidentialityProfile↔
Attributes, [163](#)
 - GetCryptographicMessageSyntax, [163](#)
 - GetFile, [163](#)
 - New, [163](#)
 - RecurseDataSet, [163](#)
 - Remove, [163](#)
 - RemoveGroupLength, [163](#)
 - RemovePrivateTags, [163](#)
 - RemoveRetired, [164](#)
 - Replace, [164](#)
 - SetCryptographicMessageSyntax, [164](#)
 - SetFile, [164](#)
- gdcm::AnyEvent, [164](#)
- gdcm::ApplicationEntity, [167](#)
 - Internal, [168](#)
 - IsValid, [168](#)
 - MaxLength, [168](#)
 - MaxNumberOfComponents, [168](#)
 - Padding, [168](#)
 - Print, [168](#)
 - Separator, [168](#)
 - SetBlob, [168](#)
 - Squeeze, [168](#)
- gdcm::Attribute
 - ArrayType, [177](#)
 - GDCM_STATIC_ASSERT, [177](#)
 - GetAsDataElement, [177](#)
 - GetDictVM, [177](#)
 - GetDictVR, [177](#)
 - GetNumberOfValues, [177](#)
 - GetTag, [178](#)
 - GetVM, [178](#)
 - GetVR, [178](#)
 - GetValue, [178](#)
 - GetValues, [178](#)
 - Internal, [180](#)
 - operator!=, [178](#)
 - operator<, [178](#)
 - operator==, [178](#)
 - operator[], [179](#)
 - Print, [179](#)
 - Set, [179](#)
 - SetByteValue, [179](#)
 - SetByteValueNoSwap, [179](#)
 - SetFromDataElement, [179](#)
 - SetFromDataSet, [179](#)
 - SetValue, [179](#)
 - SetValues, [180](#)
 - VMType, [177](#)
- gdcm::Attribute< Group, Element, TVR, TVM >, [175](#)
- gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [180](#)
 - ArrayType, [182](#)
 - GDCM_STATIC_ASSERT, [182](#)
 - GetAsDataElement, [182](#)
 - GetDictVM, [182](#)
 - GetDictVR, [182](#)
 - GetNumberOfValues, [182](#)
 - GetTag, [182](#)

- GetVM, [183](#)
- GetVR, [183](#)
- GetValue, [182](#)
- GetValues, [182](#)
- Internal, [184](#)
- operator!=, [183](#)
- operator<, [183](#)
- operator==, [183](#)
- Print, [183](#)
- Set, [183](#)
- SetByteValue, [183](#)
- SetByteValueNoSwap, [183](#)
- SetFromDataElement, [183](#)
- SetFromDataSet, [183](#)
- SetValue, [184](#)
- VMType, [182](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >, [184](#)
- GetVM, [185](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >, [185](#)
- GetVM, [186](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, [187](#)
- ~Attribute, [188](#)
- ArrayType, [188](#)
- Attribute, [188](#)
- GDCM_STATIC_ASSERT, [188](#)
- GetAsDataElement, [188](#)
- GetDictVM, [188](#)
- GetDictVR, [188](#)
- GetNumberOfValues, [188](#)
- GetTag, [188](#)
- GetVM, [189](#)
- GetVR, [189](#)
- GetValue, [189](#)
- GetValues, [189](#)
- operator[], [189](#)
- Print, [189](#)
- Set, [189](#)
- SetByteValue, [189](#)
- SetFromDataElement, [189](#)
- SetFromDataSet, [189](#)
- SetNumberOfValues, [189](#)
- SetValue, [190](#)
- SetValues, [190](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >, [190](#)
- GetVM, [191](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >, [191](#)
- GetVM, [192](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >, [193](#)
- GetVM, [194](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >, [194](#)
- GetVM, [195](#)
- gdcmm::AudioCodec, [196](#)
- ~AudioCodec, [197](#)
- AudioCodec, [197](#)
- CanCode, [197](#)
- CanDecode, [198](#)
- Decode, [198](#)
- gdcmm::Base64, [198](#)
- Decode, [198](#)
- Encode, [199](#)
- GetDecodeLength, [199](#)
- GetEncodeLength, [199](#)
- gdcmm::BaseQuery, [205](#)
- ~BaseQuery, [207](#)
- AddQueryDataSet, [207](#)
- BaseQuery, [207](#)
- GetAbstractSyntaxUID, [207](#)
- GetQueryDataSet, [207](#)
- GetSOPInstanceUID, [208](#)
- mDataSet, [208](#)
- mHelpDescription, [208](#)
- mSopInstanceUID, [208](#)
- Print, [208](#)
- QueryFactory, [208](#)
- SetSOPInstanceUID, [208](#)
- SetSearchParameter, [208](#)
- ValidDataSet, [208](#)
- ValidateQuery, [208](#)
- WriteHelpFile, [208](#)
- WriteQuery, [208](#)
- gdcmm::BaseRootQuery, [208](#)
- ~BaseRootQuery, [210](#)
- BaseRootQuery, [210](#)
- Construct, [210](#)
- GetQueryLevelFromQueryRoot, [210](#)
- GetQueryLevelFromString, [210](#)
- GetQueryLevelString, [210](#)
- GetTagListByLevel, [210](#)
- InitializeDataSet, [211](#)
- mHelpDescription, [211](#)
- mImage, [211](#)
- mPatient, [211](#)
- mRootType, [211](#)
- mSeries, [211](#)
- mStudy, [211](#)
- QueryFactory, [211](#)
- ValidateQuery, [211](#)
- gdcmm::BasicOffsetTable, [214](#)
- BasicOffsetTable, [215](#)
- operator<<, [216](#)
- Read, [215](#)

- gdcmm::Bitmap, 216
 - ~Bitmap, 219
 - AreOverlaysInPixelData, 219
 - Bitmap, 219
 - Clear, 219
 - ComputeLossyFlag, 219
 - Dimensions, 223
 - GetBuffer, 219
 - GetBuffer2, 219
 - GetBufferLength, 219
 - GetColumns, 220
 - GetDataElement, 220
 - GetDimension, 220
 - GetDimensions, 220
 - GetLUT, 220
 - GetNeedByteSwap, 220
 - GetNumberOfDimensions, 220
 - GetPhotometricInterpretation, 220
 - GetPixelFormat, 220, 221
 - GetPlanarConfiguration, 221
 - GetRows, 221
 - GetTransferSyntax, 221
 - ImageChangeTransferSyntax, 223
 - IsEmpty, 221
 - IsLossy, 221
 - IsTransferSyntaxCompatible, 221
 - LUT, 223
 - LUTPtr, 219
 - LossyFlag, 223
 - NeedByteSwap, 223
 - NumberOfDimensions, 223
 - PF, 223
 - PI, 223
 - PixelData, 223
 - PixmapReader, 223
 - PlanarConfiguration, 223
 - Print, 221
 - SetColumns, 221
 - SetDataElement, 221
 - SetDimension, 221
 - SetDimensions, 222
 - SetLUT, 222
 - SetLossyFlag, 222
 - SetNeedByteSwap, 222
 - SetNumberOfDimensions, 222
 - SetPhotometricInterpretation, 222
 - SetPixelFormat, 222
 - SetPlanarConfiguration, 222
 - SetRows, 222
 - SetTransferSyntax, 223
 - TS, 224
 - TryJPEG2000Codec, 223
 - TryJPEG2000Codec2, 223
 - TryJPEGCodec, 223
 - TryJPEGCodec2, 223
 - TryJPEGLSCodec, 223
 - TryKAKADUCodec, 223
 - TryPVRGCodec, 223
 - TryRAWCodec, 223
 - TryRLECodec, 223
- gdcmm::BitmapToBitmapFilter, 224
 - ~BitmapToBitmapFilter, 225
 - BitmapToBitmapFilter, 225
 - GetOutput, 225
 - GetOutputAsBitmap, 225
 - Input, 225
 - Output, 225
 - SetInput, 225
- gdcmm::BoxRegion, 226
 - ~BoxRegion, 227
 - Area, 227
 - BoundingBox, 227
 - BoxRegion, 227
 - Clone, 228
 - ComputeBoundingBox, 228
 - Empty, 228
 - GetXMax, 228
 - GetXMin, 228
 - GetYMax, 228
 - GetYMin, 228
 - GetZMax, 228
 - GetZMin, 228
 - IsValid, 228
 - operator=, 228
 - Print, 228
 - SetDomain, 228
- gdcmm::ByteBuffer, 229
 - ByteBuffer, 229
 - Get, 229
 - GetStart, 229
 - ShiftEnd, 229
 - UpdatePosition, 229
- gdcmm::ByteSwap
 - Swap, 230
 - SwapFromSwapCodeIntoSystem, 230
 - SwapRange, 230
 - SwapRangeFromSwapCodeIntoSystem, 230
 - SystemIsBigEndian, 230
 - SystemIsLittleEndian, 230
- gdcmm::ByteSwap< T >, 230
- gdcmm::ByteSwapFilter, 231
 - ~ByteSwapFilter, 231
 - ByteSwap, 231
 - ByteSwapFilter, 231
 - SetByteSwapTag, 231
- gdcmm::ByteValue, 231
 - ~ByteValue, 234
 - Append, 234

- ByteValue, [233](#)
- Clear, [234](#)
- ComputeLength, [234](#)
- Fill, [234](#)
- GetBuffer, [234](#)
- GetLength, [234](#)
- GetPointer, [234](#)
- IsEmpty, [235](#)
- IsPrintable, [235](#)
- operator const std::vector< char > &, [235](#)
- operator=, [235](#)
- operator==, [235](#)
- Print, [235](#)
- PrintASCII, [235](#)
- PrintASCIIXML, [235](#)
- PrintGroupLength, [235](#)
- PrintHex, [235](#)
- PrintHexXML, [235](#)
- PrintPNXML, [235](#)
- Read, [235](#), [236](#)
- SetLength, [236](#)
- SetLengthOnly, [236](#)
- Write, [236](#)
- WriteBuffer, [236](#)
- gdcmm::CAPICryptoFactory, [236](#)
 - CAPICryptoFactory, [237](#)
 - CreateCMSProvider, [237](#)
- gdcmm::CAPICryptographicMessageSyntax, [237](#)
 - ~CAPICryptographicMessageSyntax, [239](#)
 - CAPICryptographicMessageSyntax, [239](#)
 - Decrypt, [239](#)
 - Encrypt, [239](#)
 - GetCipherType, [239](#)
 - GetInitialized, [239](#)
 - ParseCertificateFile, [239](#)
 - ParseKeyFile, [239](#)
 - SetCipherType, [239](#)
 - SetPassword, [239](#)
- gdcmm::CP246ExplicitDataElement, [265](#)
 - GetLength, [266](#)
 - Read, [266](#)
 - ReadPreValue, [267](#)
 - ReadValue, [267](#)
 - ReadWithLength, [267](#)
- gdcmm::CSAElement, [271](#)
 - CSAElement, [273](#)
 - DataField, [275](#)
 - DataPtr, [273](#)
 - GetByteValue, [273](#)
 - GetKey, [273](#)
 - GetName, [273](#)
 - GetNoOfItems, [273](#)
 - GetSyngoDT, [273](#)
 - GetVM, [274](#)
 - GetVR, [274](#)
 - GetValue, [273](#)
 - IsEmpty, [274](#)
 - KeyField, [275](#)
 - NameField, [275](#)
 - NoOfItemsField, [275](#)
 - operator<, [274](#)
 - operator<<, [275](#)
 - operator=, [274](#)
 - operator==, [274](#)
 - SetByteValue, [274](#)
 - SetKey, [274](#)
 - SetName, [274](#)
 - SetNoOfItems, [274](#)
 - SetSyngoDT, [274](#)
 - SetVM, [274](#)
 - SetVR, [274](#)
 - SetValue, [274](#)
 - SyngoDTField, [275](#)
 - VRField, [275](#)
 - ValueMultiplicityField, [275](#)
- gdcmm::CSAHeader, [275](#)
 - ~CSAHeader, [277](#)
 - CSAHeader, [277](#)
 - CSAHeaderType, [277](#)
 - DATASET_FORMAT, [277](#)
 - FindCSAElementByName, [277](#)
 - GetCSADataInfo, [278](#)
 - GetCSAEEnd, [278](#)
 - GetCSAElementByName, [278](#)
 - GetCSAImageHeaderInfoTag, [278](#)
 - GetCSASeriesHeaderInfoTag, [278](#)
 - GetDataSet, [278](#)
 - GetFormat, [279](#)
 - GetInterfile, [279](#)
 - INTERFILE, [277](#)
 - LoadFromDataElement, [279](#)
 - NOMAGIC, [277](#)
 - operator<<, [279](#)
 - Print, [279](#)
 - Read, [279](#)
 - SV10, [277](#)
 - UNKNOWN, [277](#)
 - Write, [279](#)
 - ZEROED_OUT, [277](#)
- gdcmm::CSAHeaderDict, [279](#)
 - AddCSAHeaderDictEntry, [280](#)
 - Begin, [280](#)
 - CSAHeaderDict, [280](#)
 - ConstIterator, [280](#)
 - Dicts, [281](#)
 - End, [281](#)
 - GetCSAHeaderDictEntry, [281](#)
 - IsEmpty, [281](#)

- Iterator, [280](#)
- LoadDefault, [281](#)
- MapCSAHeaderDictEntry, [280](#)
- operator<<, [281](#)
- gdcmm::CSAHeaderDictEntry, [281](#)
 - CSAHeaderDictEntry, [282](#)
 - GetDescription, [282](#)
 - GetName, [282](#)
 - GetVM, [282](#)
 - GetVR, [282](#)
 - operator<, [282](#)
 - operator<<, [283](#)
 - SetDescription, [282](#)
 - SetName, [283](#)
 - SetVM, [283](#)
 - SetVR, [283](#)
- gdcmm::CSAHeaderDictException, [283](#)
- gdcmm::CodeString, [253](#)
 - CodeString, [254](#), [255](#)
 - const_iterator, [254](#)
 - const_reference, [254](#)
 - const_reverse_iterator, [254](#)
 - difference_type, [254](#)
 - GetAsString, [255](#)
 - IsValid, [255](#)
 - iterator, [254](#)
 - operator!=, [255](#)
 - operator<<, [255](#)
 - operator==, [255](#)
 - pointer, [254](#)
 - reference, [254](#)
 - reverse_iterator, [254](#)
 - Size, [255](#)
 - size_type, [254](#)
 - TrimInternal, [255](#)
 - value_type, [254](#)
- gdcmm::Codec, [250](#)
- gdcmm::Coder, [251](#)
 - ~Coder, [252](#)
 - CanCode, [252](#)
 - Code, [252](#)
 - InternalCode, [253](#)
- gdcmm::Command, [255](#)
 - ~Command, [257](#)
 - Command, [257](#)
 - Execute, [257](#)
- gdcmm::CommandDataSet, [257](#)
 - ~CommandDataSet, [259](#)
 - CommandDataSet, [259](#)
 - Insert, [259](#)
 - operator<<, [259](#)
 - Read, [259](#)
 - Replace, [259](#)
 - Write, [259](#)
- gdcmm::CompositeNetworkFunctions, [260](#)
 - CEcho, [261](#)
 - CFind, [263](#)
 - CMove, [263](#)
 - CStore, [264](#)
 - ConstructQuery, [264](#)
 - KeyValuePairArrayType, [261](#)
 - KeyValuePairType, [261](#)
- gdcmm::ConstCharWrapper, [265](#)
 - ConstCharWrapper, [265](#)
 - operator const char *, [265](#)
- gdcmm::CryptoFactory, [267](#)
 - ~CryptoFactory, [268](#)
 - CAPI, [268](#)
 - CreateCMSProvider, [268](#)
 - CryptoFactory, [268](#)
 - CryptoLib, [268](#)
 - DEFAULT, [268](#)
 - GetFactoryInstance, [268](#)
 - OPENSSL, [268](#)
 - OPENSSL7, [268](#)
- gdcmm::CryptographicMessageSyntax, [269](#)
 - ~CryptographicMessageSyntax, [270](#)
 - AES128_CIPHER, [269](#)
 - AES192_CIPHER, [270](#)
 - AES256_CIPHER, [270](#)
 - CipherTypes, [269](#)
 - CryptographicMessageSyntax, [270](#)
 - DES3_CIPHER, [269](#)
 - Decrypt, [270](#)
 - Encrypt, [270](#)
 - GetCipherType, [270](#)
 - ParseCertificateFile, [270](#)
 - ParseKeyFile, [270](#)
 - SetCipherType, [270](#)
 - SetPassword, [270](#)
- gdcmm::Curve, [287](#)
 - ~Curve, [288](#)
 - Curve, [288](#)
 - Decode, [288](#)
 - GetAsPoints, [288](#)
 - GetCurveDataDescriptor, [288](#)
 - GetDataValueRepresentation, [289](#)
 - GetDimensions, [289](#)
 - GetGroup, [289](#)
 - GetNumberOfCurves, [289](#)
 - GetNumberOfPoints, [289](#)
 - GetTypeOfData, [289](#)
 - GetTypeOfDataDescription, [289](#)
 - IsEmpty, [289](#)
 - Print, [289](#)
 - SetCoordinateStartValue, [289](#)
 - SetCoordinateStepValue, [289](#)
 - SetCurve, [289](#)

- SetCurveDataDescriptor, [289](#)
- SetCurveDescription, [289](#)
- SetDataValueRepresentation, [289](#)
- SetDimensions, [289](#)
- SetGroup, [289](#)
- SetNumberOfPoints, [289](#)
- SetTypeOfData, [289](#)
- Update, [289](#)
- gdcmm::DICOMDIR, [317](#)
 - DICOMDIR, [318](#)
- gdcmm::DICOMDIRGenerator, [318](#)
 - ~DICOMDIRGenerator, [319](#)
 - AddImageDirectoryRecord, [319](#)
 - AddPatientDirectoryRecord, [319](#)
 - AddSeriesDirectoryRecord, [319](#)
 - AddStudyDirectoryRecord, [319](#)
 - DICOMDIRGenerator, [319](#)
 - FilenameType, [319](#)
 - FileNamesType, [319](#)
 - Generate, [319](#)
 - GetFile, [319](#)
 - GetScanner, [320](#)
 - SetDescriptor, [320](#)
 - SetFile, [320](#)
 - SetFileNames, [320](#)
 - SetRootDirectory, [320](#)
- gdcmm::DataElement, [289](#)
 - Clear, [293](#)
 - DataElement, [293](#)
 - Empty, [293](#)
 - GetByteValue, [293](#)
 - GetLength, [293](#)
 - GetSequenceOfFragments, [293](#)
 - GetTag, [294](#)
 - GetVL, [294](#)
 - GetVR, [295](#)
 - GetValue, [294](#)
 - GetValueAsSQ, [294](#)
 - IsEmpty, [295](#)
 - IsUndefinedLength, [295](#)
 - operator<, [295](#)
 - operator<<, [298](#)
 - operator=, [295](#)
 - operator==, [295](#)
 - Read, [295](#)
 - ReadOrSkip, [296](#)
 - ReadPreValue, [296](#)
 - ReadValue, [296](#)
 - ReadValueWithLength, [296](#)
 - ReadWithLength, [296](#)
 - SetByteValue, [296](#)
 - SetTag, [296](#)
 - SetVL, [297](#)
 - SetVLToUndefined, [297](#)
 - SetVR, [297](#)
 - SetValue, [296](#)
 - SetValueFieldLength, [297](#)
 - TagField, [298](#)
 - VRField, [298](#)
 - ValueField, [298](#)
 - ValueLengthField, [298](#)
 - ValuePtr, [292](#)
 - Write, [297](#)
- gdcmm::DataElementException, [298](#)
- gdcmm::DataEvent, [299](#)
 - ~DataEvent, [301](#)
 - CheckEvent, [301](#)
 - DataEvent, [300](#), [301](#)
 - GetData, [301](#)
 - GetDataLength, [301](#)
 - GetEventName, [301](#)
 - MakeObject, [301](#)
 - Self, [300](#)
 - SetData, [301](#)
 - Superclass, [300](#)
- gdcmm::DataSet, [301](#)
 - Begin, [304](#)
 - CSAHeader, [308](#)
 - Clear, [304](#)
 - ComputeDataElement, [304](#)
 - ComputeGroupLength, [305](#)
 - ConstIterator, [304](#)
 - DataElementSet, [304](#)
 - End, [305](#)
 - FindDataElement, [305](#)
 - FindNextDataElement, [305](#)
 - GetDEEnd, [306](#)
 - GetDES, [306](#)
 - GetDataElement, [305](#), [306](#)
 - GetLength, [306](#)
 - GetMediaStorage, [306](#)
 - GetPrivateCreator, [306](#)
 - Insert, [306](#)
 - InsertDataElement, [306](#)
 - IsEmpty, [306](#)
 - Iterator, [304](#)
 - operator<<, [308](#)
 - operator(), [307](#)
 - operator=, [307](#)
 - operator[], [307](#)
 - Print, [307](#)
 - Read, [307](#)
 - ReadNested, [307](#)
 - ReadSelectedPrivateTags, [307](#)
 - ReadSelectedPrivateTagsWithLength, [307](#)
 - ReadSelectedTags, [307](#)
 - ReadSelectedTagsWithLength, [307](#)
 - ReadUpToTag, [307](#)

- ReadUpToTagWithLength, [307](#)
- ReadWithLength, [307](#)
- Remove, [307](#)
- Replace, [307](#)
- ReplaceEmpty, [308](#)
- Size, [308](#)
- SizeType, [304](#)
- Write, [308](#)
- gdcm::DataSetEvent, [308](#)
 - ~DataSetEvent, [310](#)
 - CheckEvent, [310](#)
 - DataSetEvent, [310](#)
 - GetDataSet, [310](#)
 - GetEventName, [310](#)
 - MakeObject, [310](#)
 - Self, [310](#)
 - Superclass, [310](#)
- gdcm::DataSetHelper, [311](#)
 - ComputeVR, [311](#)
- gdcm::Decoder, [311](#)
 - ~Decoder, [312](#)
 - CanDecode, [312](#)
 - Decode, [312](#)
 - DecodeByStreams, [312](#)
- gdcm::DefinedTerms, [313](#)
 - DefinedTerms, [313](#)
- gdcm::Defs, [313](#)
 - ~Defs, [314](#)
 - Defs, [314](#)
 - GetIODFromFile, [314](#)
 - GetIODNameFromMediaStorage, [314](#)
 - GetIODs, [314](#), [315](#)
 - GetMacros, [315](#)
 - GetModules, [315](#)
 - GetTypeFromTag, [315](#)
 - Global, [315](#)
 - IsEmpty, [315](#)
 - LoadDefaults, [315](#)
 - LoadFromFile, [315](#)
 - Verify, [315](#)
- gdcm::DeltaEncodingCodec, [316](#)
 - ~DeltaEncodingCodec, [317](#)
 - CanDecode, [317](#)
 - Decode, [317](#)
 - DeltaEncodingCodec, [317](#)
- gdcm::Dict, [320](#)
 - AddDictEntry, [321](#)
 - Begin, [321](#)
 - ConstIterator, [321](#)
 - Dict, [321](#)
 - Dicts, [322](#)
 - End, [321](#)
 - GetDictEntry, [322](#)
 - GetDictEntryByKeyword, [322](#)
 - GetDictEntryByName, [322](#)
 - GetKeywordFromTag, [322](#)
 - IsEmpty, [322](#)
 - Iterator, [321](#)
 - LoadDefault, [322](#)
 - MapDictEntry, [321](#)
 - operator<<, [322](#)
- gdcm::DictConverter, [323](#)
 - ~DictConverter, [324](#)
 - AddGroupLength, [324](#)
 - Convert, [324](#)
 - ConvertToCXX, [324](#)
 - ConvertToXML, [324](#)
 - DICT_DEBUG, [324](#)
 - DICT_DEFAULT, [324](#)
 - DICT_XML, [324](#)
 - DictConverter, [324](#)
 - GetDictName, [324](#)
 - GetInputFilename, [324](#)
 - GetOutputFilename, [324](#)
 - GetOutputType, [324](#)
 - OutputTypes, [324](#)
 - ReadVM, [324](#)
 - ReadVR, [324](#)
 - Readuint16, [324](#)
 - SetDictName, [324](#)
 - SetInputFileName, [324](#)
 - SetOutputFileName, [324](#)
 - SetOutputType, [324](#)
 - WriteFooter, [324](#)
 - WriteHeader, [325](#)
- gdcm::DictEntry, [325](#)
 - Dict, [327](#)
 - DictEntry, [326](#)
 - GetKeyword, [326](#)
 - GetName, [326](#)
 - GetRetired, [326](#)
 - GetVM, [326](#)
 - GetVR, [326](#)
 - IsUnique, [327](#)
 - operator<<, [327](#)
 - SetElementXX, [327](#)
 - SetGroupXX, [327](#)
 - SetKeyword, [327](#)
 - SetName, [327](#)
 - SetRetired, [327](#)
 - SetVM, [327](#)
 - SetVR, [327](#)
- gdcm::DictPrinter, [327](#)
 - ~DictPrinter, [329](#)
 - DictPrinter, [329](#)
 - Print, [329](#)
 - PrintDataElement2, [329](#)
 - PrintDataSet2, [329](#)

- gdcmm::Dicts, [329](#)
 - ~Dicts, [330](#)
 - ConstructorType, [330](#)
 - Dicts, [330](#)
 - GEMS, [330](#)
 - GetCSAHeaderDict, [331](#)
 - GetConstructorString, [330](#)
 - GetDictEntry, [331](#)
 - GetPrivateDict, [331](#)
 - GetPublicDict, [331](#)
 - Global, [331](#)
 - IsEmpty, [331](#)
 - LoadDefaults, [331](#)
 - operator<<, [331](#)
 - PHILIPS, [330](#)
 - SIEMENS, [330](#)
- gdcmm::DirectionCosines, [333](#)
 - ~DirectionCosines, [334](#)
 - ComputeDistAlongNormal, [334](#)
 - Cross, [334](#)
 - CrossDot, [334](#)
 - DirectionCosines, [334](#)
 - Dot, [334](#)
 - IsValid, [334](#)
 - Normalize, [334](#)
 - operator const double *, [335](#)
 - Print, [335](#)
 - SetFromString, [335](#)
- gdcmm::Directory, [335](#)
 - ~Directory, [336](#)
 - Directory, [336](#)
 - Explore, [336](#)
 - FilenameType, [336](#)
 - FilenamesType, [336](#)
 - GetDirectories, [336](#)
 - GetFilenames, [337](#)
 - GetToplevel, [337](#)
 - Load, [337](#)
 - operator<<, [337](#)
 - Print, [337](#)
- gdcmm::DirectoryHelper, [338](#)
 - GetCTImageSeriesUIDs, [338](#)
 - GetFilenamesFromSeriesUIDs, [338](#)
 - GetFrameOfReference, [338](#)
 - GetMRImageSeriesUIDs, [338](#)
 - GetRTStructSeriesUIDs, [338](#)
 - GetSOPClassUID, [339](#)
 - GetSeriesUIDsBySOPClassUID, [339](#)
 - GetStringValueFromTag, [339](#)
 - LoadImageFromFiles, [339](#)
 - RetrieveSOPInstanceUIDFromIndex, [339](#)
 - RetrieveSOPInstanceUIDFromZPosition, [339](#)
- gdcmm::DummyValueGenerator, [339](#)
 - Generate, [339](#)
- gdcmm::Dumper, [340](#)
 - ~Dumper, [341](#)
 - Dumper, [341](#)
- gdcmm::Element
 - GetAsDataElement, [344](#)
 - GetLength, [344](#)
 - GetVM, [344](#)
 - GetVR, [344](#)
 - GetValue, [344](#)
 - GetValues, [344](#)
 - Internal, [344](#)
 - operator[], [344](#)
 - Print, [344](#)
 - Read, [344](#)
 - Set, [344](#)
 - SetFromDataElement, [344](#)
 - SetNoSwap, [344](#)
 - SetValue, [344](#)
 - Type, [344](#)
 - Write, [344](#)
- gdcmm::Element< TVR, TVM >, [342](#)
- gdcmm::Element< TVR, VM::VM1_2 >, [345](#)
 - Parent, [346](#)
 - SetLength, [346](#)
- gdcmm::Element< TVR, VM::VM1_n >, [346](#)
 - ~Element, [347](#)
 - Element, [347](#)
 - GetAsDataElement, [347](#)
 - GetLength, [347](#)
 - GetVM, [347](#)
 - GetVR, [347](#)
 - GetValue, [347](#)
 - operator=, [347](#)
 - operator[], [348](#)
 - Print, [348](#)
 - Read, [348](#)
 - Set, [348](#)
 - SetArray, [348](#)
 - SetFromDataElement, [348](#)
 - SetLength, [348](#)
 - SetNoSwap, [348](#)
 - SetValue, [348](#)
 - Type, [347](#)
 - Write, [348](#)
 - WriteASCII, [348](#)
- gdcmm::Element< TVR, VM::VM2_2n >, [348](#)
 - Parent, [350](#)
 - SetLength, [350](#)
- gdcmm::Element< TVR, VM::VM2_n >, [350](#)
 - Parent, [351](#)
 - SetLength, [351](#)
- gdcmm::Element< TVR, VM::VM3_3n >, [351](#)
 - Parent, [353](#)
 - SetLength, [353](#)

- gdcmm::Element< TVR, VM::VM3_n >, 353
 - Parent, 354
 - SetLength, 354
- gdcmm::Element< VR::AS, VM::VM5 >, 354
 - GetLength, 355
 - Internal, 355
 - Print, 355
- gdcmm::Element< VR::OB, VM::VM1 >, 355
- gdcmm::Element< VR::OW, VM::VM1 >, 356
- gdcmm::ElementDisableCombinations< TVR, TVM >, 358
- gdcmm::ElementDisableCombinations< VR::OB, VM::V←M1_n >, 359
- gdcmm::ElementDisableCombinations< VR::OW, VM::V←M1_n >, 359
- gdcmm::EncapsulatedDocument, 359
 - EncapsulatedDocument, 359
- gdcmm::EncodingImplementation< T >, 360
- gdcmm::EncodingImplementation< VR::VRASCII >, 360
 - Read, 360
 - ReadComputeLength, 360
 - ReadNoSwap, 361
 - Write, 361
- gdcmm::EncodingImplementation< VR::VRBINARY >, 361
 - Read, 361
 - ReadComputeLength, 361
 - ReadNoSwap, 361
 - Write, 362
- gdcmm::EndEvent, 362
- gdcmm::EnumeratedValues, 363
 - EnumeratedValues, 364
- gdcmm::Event, 364
 - ~Event, 366
 - CheckEvent, 366
 - Event, 366
 - GetEventName, 366
 - MakeObject, 366
 - Print, 366
- gdcmm::Exception, 367
 - ~Exception, 368
 - Exception, 368
 - GetDescription, 368
 - what, 368
- gdcmm::ExitEvent, 368
- gdcmm::ExplicitDataElement, 370
 - GetLength, 371
 - Read, 371
 - ReadPreValue, 371
 - ReadValue, 371
 - ReadWithLength, 371
 - Write, 371
- gdcmm::ExplicitImplicitDataElement, 371
 - GetLength, 373
 - Read, 373
 - ReadPreValue, 373
 - ReadValue, 373
 - ReadWithLength, 373
- gdcmm::Fiducials, 373
 - Fiducials, 374
- gdcmm::File, 374
 - ~File, 376
 - File, 376
 - GetDataSet, 376
 - GetHeader, 376
 - operator<<, 377
 - Read, 376
 - SetDataSet, 376
 - SetHeader, 377
 - Write, 377
- gdcmm::FileAnonymizer, 377
 - ~FileAnonymizer, 379
 - Empty, 379
 - FileAnonymizer, 379
 - Remove, 379
 - Replace, 379
 - SetInputFileName, 379
 - SetOutputFileName, 380
 - Write, 380
- gdcmm::FileChangeTransferSyntax, 380
 - ~FileChangeTransferSyntax, 382
 - Change, 382
 - FileChangeTransferSyntax, 382
 - GetCodec, 382
 - New, 382
 - SetInputFileName, 382
 - SetOutputFileName, 383
 - SetTransferSyntax, 383
- gdcmm::FileDecompressLookupTable, 383
 - ~FileDecompressLookupTable, 384
 - Change, 384
 - FileDecompressLookupTable, 384
 - GetFile, 385
 - GetPixmap, 385
 - SetFile, 385
 - SetPixmap, 385
- gdcmm::FileDerivation, 385
 - ~FileDerivation, 386
 - AddDerivationDescription, 386
 - AddPurposeOfReferenceCodeSequence, 386
 - AddReference, 386
 - AddSourceImageSequence, 386
 - Derive, 386
 - FileDerivation, 386
 - GetFile, 386, 387
 - SetDerivationCodeSequenceCodeValue, 387
 - SetDerivationDescription, 387
 - SetFile, 387
 - SetPurposeOfReferenceCodeSequenceCodeValue, 387

- gdcmm::FileExplicitFilter, 387
 - ~FileExplicitFilter, 388
 - Change, 388
 - ChangeFMI, 389
 - FileExplicitFilter, 388
 - GetFile, 389
 - ProcessDataSet, 389
 - SetChangePrivateTags, 389
 - SetFile, 389
 - SetRecomputeItemLength, 389
 - SetRecomputeSequenceLength, 389
 - SetUseVRUN, 389
- gdcmm::FileMetaInformation, 389
 - ~FileMetaInformation, 392
 - AppendImplementationClassUID, 392
 - ComputeDataSetMediaStorageSOPClass, 392
 - ComputeDataSetTransferSyntax, 392
 - DataSetMS, 394
 - DataSetTS, 394
 - Default, 392
 - FileMetaInformation, 392
 - FillFromDataSet, 392
 - GetDataSetTransferSyntax, 392
 - GetFileMetaInformationVersion, 392
 - GetFullLength, 393
 - GetGDCMImplementationClassUID, 393
 - GetGDCMImplementationVersionName, 393
 - GetGDCMSourceApplicationEntityTitle, 393
 - GetImplementationClassUID, 393
 - GetImplementationVersionName, 393
 - GetMediaStorage, 393
 - GetMediaStorageAsString, 393
 - GetMetaInformationTS, 393
 - GetPreamble, 393
 - GetSourceApplicationEntityTitle, 393
 - Insert, 393
 - IsValid, 393
 - MetaInformationTS, 394
 - operator<<, 394
 - Read, 393
 - ReadCompat, 393
 - ReadCompatInternal, 393
 - Replace, 393
 - SetDataSetTransferSyntax, 394
 - SetImplementationClassUID, 394
 - SetImplementationVersionName, 394
 - SetPreamble, 394
 - SetSourceApplicationEntityTitle, 394
 - Write, 394
- gdcmm::FileNameEvent, 397
 - ~FileNameEvent, 399
 - CheckEvent, 399
 - FileNameEvent, 399
 - GetEventName, 399
 - GetFileName, 399
 - MakeObject, 399
 - Self, 399
 - SetFileName, 399
 - Superclass, 399
- gdcmm::FileSet, 402
 - AddFile, 402, 403
 - FileSet, 402
 - FileType, 402
 - FilesType, 402
 - GetFiles, 403
 - operator<<, 403
 - SetFiles, 403
- gdcmm::FileStreamer, 403
 - ~FileStreamer, 405
 - AppendToDataElement, 405
 - AppendToGroupDataElement, 405
 - CheckDataElement, 405
 - CheckTemplateFileName, 405
 - FileStreamer, 405
 - New, 405
 - ReserveDataElement, 405
 - ReserveGroupDataElement, 406
 - SetOutputFileName, 406
 - SetTemplateFileName, 406
 - StartDataElement, 406
 - StartGroupDataElement, 406
 - StopDataElement, 406
 - StopGroupDataElement, 406
- gdcmm::FileWithName, 407
 - FileWithName, 408
 - filename, 408
- gdcmm::Filename, 395
 - EndWith, 396
 - Filename, 396
 - GetExtension, 396
 - GetFileName, 396
 - GetName, 396
 - GetPath, 396
 - IsEmpty, 396
 - IsIdentical, 396
 - Join, 396
 - operator const char *, 396
 - ToUnixSlashes, 396
 - ToWindowsSlashes, 397
- gdcmm::FilenameGenerator, 399
 - ~FilenameGenerator, 401
 - FilenameGenerator, 400
 - FilenameType, 400
 - FilenamesType, 400
 - Generate, 401
 - GetFilename, 401
 - GetFilenames, 401
 - GetNumberOfFilenames, 401

- GetPattern, [401](#)
- GetPrefix, [401](#)
- SetNumberOfFileNames, [401](#)
- SetPattern, [401](#)
- SetPrefix, [401](#)
- SizeType, [400](#)
- gdcmm::FindPatientRootQuery, [408](#)
 - FindPatientRootQuery, [410](#)
 - GetAbstractSyntaxUID, [410](#)
 - GetTagListByLevel, [410](#)
 - InitializeDataSet, [410](#)
 - QueryFactory, [410](#)
 - ValidateQuery, [410](#)
- gdcmm::FindStudyRootQuery, [411](#)
 - FindStudyRootQuery, [412](#)
 - GetAbstractSyntaxUID, [412](#)
 - GetTagListByLevel, [412](#)
 - InitializeDataSet, [412](#)
 - QueryFactory, [413](#)
 - ValidateQuery, [412](#)
- gdcmm::Fragment, [413](#)
 - ComputeLength, [415](#)
 - Fragment, [415](#)
 - GetLength, [415](#)
 - operator<<, [415](#)
 - Read, [415](#)
 - ReadBacktrack, [415](#)
 - ReadPreValue, [415](#)
 - ReadValue, [415](#)
 - Write, [415](#)
- gdcmm::Global, [415](#)
 - ~Global, [416](#)
 - Append, [416](#)
 - GetDefs, [417](#)
 - GetDicts, [417](#)
 - GetInstance, [417](#)
 - Global, [416](#)
 - LoadResourcesFiles, [417](#)
 - Locate, [417](#)
 - operator<<, [418](#)
 - Prepend, [417](#)
- gdcmm::GroupDict, [418](#)
 - ~GroupDict, [419](#)
 - Add, [419](#)
 - GetAbbreviation, [419](#)
 - GetName, [419](#)
 - GroupDict, [419](#)
 - GroupStringVector, [419](#)
 - Insert, [419](#)
 - operator<<, [419](#)
 - Size, [419](#)
- gdcmm::IOD, [471](#)
 - AddIODEntry, [472](#)
 - Clear, [472](#)
 - GetIODEntry, [472](#)
 - GetNumberOfIODs, [472](#)
 - GetTypeFromTag, [472](#)
 - IOD, [471](#)
 - MapIODEntry, [471](#)
 - operator<<, [472](#)
 - SizeType, [471](#)
- gdcmm::IODEntry, [472](#)
 - GetIE, [473](#)
 - GetName, [473](#)
 - GetRef, [473](#)
 - GetUsage, [473](#)
 - GetUsageType, [474](#)
 - IODEntry, [473](#)
 - operator<<, [474](#)
 - SetIE, [474](#)
 - SetName, [474](#)
 - SetRef, [474](#)
 - SetUsage, [474](#)
- gdcmm::IODs, [474](#)
 - AddIOD, [475](#)
 - Begin, [475](#)
 - Clear, [475](#)
 - End, [475](#)
 - GetIOD, [475](#)
 - IODMapType, [475](#)
 - IODMapTypeConstIterator, [475](#)
 - IODName, [475](#)
 - IODs, [475](#)
 - operator<<, [475](#)
- gdcmm::IPPSorter, [476](#)
 - ComputeZSpacing, [479](#)
 - DirCosTolerance, [479](#)
 - DropDuplicatePositions, [479](#)
 - GetDirectionCosinesTolerance, [478](#)
 - GetZSpacing, [478](#)
 - GetZSpacingTolerance, [478](#)
 - IPPSorter, [478](#)
 - SetComputeZSpacing, [478](#)
 - SetDirectionCosinesTolerance, [478](#)
 - SetDropDuplicatePositions, [478](#)
 - SetZSpacingTolerance, [478](#)
 - Sort, [479](#)
 - ZSpacing, [479](#)
 - ZTolerance, [479](#)
- gdcmm::IconImageFilter, [419](#)
 - ~IconImageFilter, [421](#)
 - Extract, [421](#)
 - ExtractIconImages, [421](#)
 - ExtractVeprolIconImages, [421](#)
 - GetFile, [421](#)
 - GetIconImage, [421](#)
 - GetNumberOfIconImages, [421](#)
 - IconImageFilter, [421](#)

- SetFile, [421](#)
- gdcmm::IconImageGenerator, [422](#)
 - ~IconImageGenerator, [423](#)
 - AutoPixelMinMax, [423](#)
 - ConvertRGBToPaletteColor, [423](#)
 - Generate, [423](#)
 - GetIconImage, [423](#)
 - GetPixmap, [423](#)
 - IconImageGenerator, [423](#)
 - SetOutputDimensions, [423](#)
 - SetOutsideValuePixel, [423](#)
 - SetPixelMinMax, [424](#)
 - SetPixmap, [424](#)
- gdcmm::Image, [425](#)
 - ~Image, [427](#)
 - GetDirectionCosines, [427](#)
 - GetIntercept, [427](#)
 - GetOrigin, [427](#)
 - GetSlope, [427](#)
 - GetSpacing, [427](#)
 - Image, [427](#)
 - Print, [427](#)
 - SetDirectionCosines, [428](#)
 - SetIntercept, [428](#)
 - SetOrigin, [428](#)
 - SetSlope, [428](#)
 - SetSpacing, [428](#)
- gdcmm::ImageApplyLookupTable, [428](#)
 - ~ImageApplyLookupTable, [431](#)
 - Apply, [431](#)
 - ImageApplyLookupTable, [431](#)
- gdcmm::ImageChangePhotometricInterpretation, [431](#)
 - ~ImageChangePhotometricInterpretation, [433](#)
 - Change, [433](#)
 - ChangeMonochrome, [433](#)
 - GetPhotometricInterpretation, [433](#)
 - ImageChangePhotometricInterpretation, [433](#)
 - RGB2YBR, [433](#)
 - SetPhotometricInterpretation, [433](#)
 - YBR2RGB, [433](#)
- gdcmm::ImageChangePlanarConfiguration, [434](#)
 - ~ImageChangePlanarConfiguration, [436](#)
 - Change, [436](#)
 - GetPlanarConfiguration, [436](#)
 - ImageChangePlanarConfiguration, [436](#)
 - RGBPixelsToRGBPlanes, [436](#)
 - RGBPlanesToRGBPixels, [436](#)
 - SetPlanarConfiguration, [436](#)
- gdcmm::ImageChangeTransferSyntax, [437](#)
 - ~ImageChangeTransferSyntax, [439](#)
 - Change, [439](#)
 - GetTransferSyntax, [439](#)
 - ImageChangeTransferSyntax, [439](#)
 - SetCompressIconImage, [439](#)
- SetForce, [440](#)
- SetTransferSyntax, [440](#)
- SetUserCodec, [440](#)
- TryJPEG2000Codec, [440](#)
- TryJPEGCodec, [440](#)
- TryJPEGLSCodec, [440](#)
- TryRAWCodec, [440](#)
- TryRLECodec, [440](#)
- gdcmm::ImageCodec, [441](#)
 - ~ImageCodec, [443](#)
 - AppendFrameEncode, [443](#)
 - AppendRowEncode, [443](#)
 - CanCode, [443](#)
 - CanDecode, [444](#)
 - Clone, [444](#)
 - Decode, [444](#)
 - DecodeByStreams, [444](#)
 - Dimensions, [447](#)
 - DoByteSwap, [444](#)
 - DoInvertMonochrome, [444](#)
 - DoOverlayCleanup, [444](#)
 - DoPaddedCompositePixelCode, [444](#)
 - DoPlanarConfiguration, [444](#)
 - DoSimpleCopy, [444](#)
 - DoYBR, [444](#)
 - FileChangeTransferSyntax, [446](#)
 - GetDimensions, [444](#)
 - GetHeaderInfo, [445](#)
 - GetLUT, [445](#)
 - GetLossyFlag, [445](#)
 - GetNeedByteSwap, [445](#)
 - GetNumberOfDimensions, [445](#)
 - GetPhotometricInterpretation, [445](#)
 - GetPixelFormat, [445](#)
 - GetPlanarConfiguration, [445](#)
 - ImageChangePhotometricInterpretation, [446](#)
 - ImageCodec, [443](#)
 - IsFrameEncoder, [445](#)
 - IsLossy, [445](#)
 - IsRowEncoder, [445](#)
 - IsValid, [445](#)
 - LUT, [447](#)
 - LUTPtr, [443](#)
 - LossyFlag, [447](#)
 - NeedByteSwap, [447](#)
 - NeedOverlayCleanup, [447](#)
 - NumberOfDimensions, [447](#)
 - PF, [447](#)
 - PI, [447](#)
 - PlanarConfiguration, [447](#)
 - RequestPaddedCompositePixelCode, [447](#)
 - RequestPlanarConfiguration, [447](#)
 - SetDimensions, [445](#)
 - SetLUT, [446](#)

- SetLossyFlag, 446
- SetNeedByteSwap, 446
- SetNeedOverlayCleanup, 446
- SetNumberOfDimensions, 446
- SetPhotometricInterpretation, 446
- SetPixelFormat, 446
- SetPlanarConfiguration, 446
- StartEncode, 446
- StopEncode, 446
- gdcm::ImageConverter, 447
 - ~ImageConverter, 448
 - Convert, 448
 - GetOutput, 448
 - ImageConverter, 448
 - SetInput, 448
- gdcm::ImageFragmentSplitter, 448
 - ~ImageFragmentSplitter, 451
 - GetFragmentSizeMax, 451
 - ImageFragmentSplitter, 451
 - SetForce, 451
 - SetFragmentSizeMax, 451
 - Split, 451
- gdcm::ImageHelper, 451
 - ComputeMediaStorageFromModality, 452
 - ComputeSpacingFromImagePositionPatient, 452
 - GetDimensionsValue, 453
 - GetDirectionCosinesFromDataSet, 453
 - GetDirectionCosinesValue, 453
 - GetForcePixelSpacing, 453
 - GetForceRescaleInterceptSlope, 453
 - GetLUT, 453
 - GetOriginValue, 453
 - GetPhotometricInterpretationValue, 453
 - GetPixelFormatValue, 453
 - GetPlanarConfigurationValue, 453
 - GetPointerFromElement, 453
 - GetRealWorldValueMappingContent, 453
 - GetRescaleInterceptSlopeValue, 453
 - GetSpacingTagFromMediaStorage, 454
 - GetSpacingValue, 454
 - GetZSpacingTagFromMediaStorage, 454
 - SetDimensionsValue, 454
 - SetDirectionCosinesValue, 454
 - SetForcePixelSpacing, 454
 - SetForceRescaleInterceptSlope, 454
 - SetOriginValue, 454
 - SetRescaleInterceptSlopeValue, 454
 - SetSpacingValue, 454
- gdcm::ImageReader, 455
 - ~ImageReader, 457
 - GetImage, 457
 - ImageReader, 457
 - Read, 457
 - ReadACRNEMAIImage, 458
 - ReadImage, 458
- gdcm::ImageRegionReader, 458
 - ~ImageRegionReader, 460
 - ComputeBufferLength, 460
 - GetRegion, 460
 - ImageRegionReader, 460
 - Read, 460
 - ReadInformation, 460
 - ReadIntoBuffer, 460
 - SetRegion, 461
- gdcm::ImageToImageFilter, 461
 - ~ImageToImageFilter, 463
 - GetInput, 463
 - GetOutput, 463
 - ImageToImageFilter, 462
- gdcm::ImageWriter, 463
 - ~ImageWriter, 465
 - GetImage, 465
 - ImageWriter, 465
 - Write, 465
- gdcm::ImplicitDataElement, 468
 - GetLength, 469
 - Read, 469
 - ReadPreValue, 469
 - ReadValue, 469
 - ReadValueWithLength, 469
 - ReadWithLength, 469
 - Write, 469
- gdcm::InitializeEvent, 470
- gdcm::Item, 479
 - Clear, 481
 - FindDataElement, 481
 - GetDataElement, 481
 - GetLength, 482
 - GetNestedDataSet, 482
 - InsertDataElement, 482
 - Item, 481
 - operator<<, 482
 - Read, 482
 - SetNestedDataSet, 482
 - Write, 482
- gdcm::IterationEvent, 482
- gdcm::JPEG12Codec, 484
 - ~JPEG12Codec, 485
 - DecodeByStreams, 485
 - EncodeBuffer, 485
 - GetHeaderInfo, 485
 - InternalCode, 485
 - IsStateSuspension, 485
 - JPEG12Codec, 485
- gdcm::JPEG16Codec, 486
 - ~JPEG16Codec, 487
 - DecodeByStreams, 487
 - EncodeBuffer, 487

- GetHeaderInfo, [487](#)
- InternalCode, [487](#)
- IsStateSuspension, [487](#)
- JPEG16Codec, [487](#)
- gdcmm::JPEG2000Codec, [488](#)
 - ~JPEG2000Codec, [489](#)
 - AppendFrameEncode, [490](#)
 - AppendRowEncode, [490](#)
 - Bitmap, [491](#)
 - CanCode, [490](#)
 - CanDecode, [490](#)
 - Clone, [490](#)
 - Code, [490](#)
 - Decode, [490](#)
 - DecodeByStreams, [490](#)
 - DecodeExtent, [490](#)
 - GetHeaderInfo, [490](#)
 - GetQuality, [491](#)
 - GetRate, [491](#)
 - ImageRegionReader, [491](#)
 - IsFrameEncoder, [491](#)
 - IsRowEncoder, [491](#)
 - JPEG2000Codec, [489](#)
 - SetNumberOfResolutions, [491](#)
 - SetQuality, [491](#)
 - SetRate, [491](#)
 - SetReversible, [491](#)
 - SetTileSize, [491](#)
 - StartEncode, [491](#)
 - StopEncode, [491](#)
- gdcmm::JPEG8Codec, [491](#)
 - ~JPEG8Codec, [493](#)
 - DecodeByStreams, [493](#)
 - EncodeBuffer, [493](#)
 - GetHeaderInfo, [493](#)
 - InternalCode, [493](#)
 - IsStateSuspension, [493](#)
 - JPEG8Codec, [493](#)
- gdcmm::JPEGCodec, [494](#)
 - ~JPEGCodec, [496](#)
 - AppendFrameEncode, [496](#)
 - AppendRowEncode, [496](#)
 - BitSample, [498](#)
 - CanCode, [496](#)
 - CanDecode, [496](#)
 - Clone, [496](#)
 - Code, [496](#)
 - ComputeOffsetTable, [497](#)
 - Decode, [497](#)
 - DecodeByStreams, [497](#)
 - DecodeExtent, [497](#)
 - EncodeBuffer, [497](#)
 - GetHeaderInfo, [497](#)
 - GetLossless, [497](#)
 - GetQuality, [497](#)
 - ImageRegionReader, [498](#)
 - IsFrameEncoder, [497](#)
 - IsRowEncoder, [497](#)
 - IsStateSuspension, [497](#)
 - IsValid, [498](#)
 - JPEGCodec, [496](#)
 - Quality, [498](#)
 - SetBitSample, [498](#)
 - SetLossless, [498](#)
 - SetPixelFormat, [498](#)
 - SetQuality, [498](#)
 - StartEncode, [498](#)
 - StopEncode, [498](#)
- gdcmm::JPEGLSCodec, [498](#)
 - ~JPEGLSCodec, [500](#)
 - AppendFrameEncode, [500](#)
 - AppendRowEncode, [500](#)
 - CanCode, [501](#)
 - CanDecode, [501](#)
 - Clone, [501](#)
 - Code, [501](#)
 - Decode, [501](#)
 - DecodeExtent, [501](#)
 - GetBufferLength, [501](#)
 - GetHeaderInfo, [501](#)
 - GetLossless, [501](#)
 - ImageRegionReader, [502](#)
 - IsFrameEncoder, [501](#)
 - IsRowEncoder, [501](#)
 - JPEGLSCodec, [500](#)
 - SetBufferLength, [502](#)
 - SetLossless, [502](#)
 - SetLossyError, [502](#)
 - StartEncode, [502](#)
 - StopEncode, [502](#)
- gdcmm::JSON, [502](#)
 - ~JSON, [503](#)
 - Code, [503](#)
 - Decode, [503](#)
 - GetPrettyPrint, [503](#)
 - JSON, [503](#)
 - PrettyPrintOff, [503](#)
 - PrettyPrintOn, [503](#)
 - SetPrettyPrint, [503](#)
- gdcmm::KAKADUCodec, [503](#)
 - ~KAKADUCodec, [505](#)
 - CanCode, [505](#)
 - CanDecode, [505](#)
 - Clone, [505](#)
 - Code, [505](#)
 - Decode, [505](#)
 - KAKADUCodec, [505](#)
- gdcmm::LO, [505](#)

- const_iterator, [507](#)
- const_reference, [507](#)
- const_reverse_iterator, [507](#)
- difference_type, [507](#)
- IsValid, [507](#)
- iterator, [507](#)
- LO, [507](#)
- pointer, [507](#)
- reference, [507](#)
- reverse_iterator, [507](#)
- size_type, [507](#)
- Superclass, [507](#)
- value_type, [507](#)
- gdcm::LookupTable, [508](#)
 - ~LookupTable, [510](#)
 - Allocate, [510](#)
 - BLUE, [510](#)
 - BitSample, [512](#)
 - Clear, [510](#)
 - Decode, [510](#)
 - GRAY, [510](#)
 - GREEN, [510](#)
 - GetBitSample, [511](#)
 - GetBufferAsRGBA, [511](#)
 - GetLUT, [511](#)
 - GetLUTDescriptor, [511](#)
 - GetLUTLength, [511](#)
 - GetPointer, [511](#)
 - IncompleteLUT, [512](#)
 - InitializeBlueLUT, [511](#)
 - InitializeGreenLUT, [511](#)
 - InitializeLUT, [511](#)
 - InitializeRedLUT, [511](#)
 - Initialized, [511](#)
 - Internal, [512](#)
 - LookupTable, [510](#)
 - LookupTableType, [510](#)
 - Print, [511](#)
 - RED, [510](#)
 - SetBlueLUT, [512](#)
 - SetGreenLUT, [512](#)
 - SetLUT, [512](#)
 - SetRedLUT, [512](#)
 - UNKNOWN, [510](#)
 - WriteBufferAsRGBA, [512](#)
- gdcm::MD5, [517](#)
 - ~MD5, [517](#)
 - Compute, [517](#)
 - ComputeFile, [517](#)
 - MD5, [517](#)
- gdcm::Macro, [513](#)
 - AddMacroEntry, [514](#)
 - ArrayIncludeMacrosType, [514](#)
 - Clear, [514](#)
 - FindMacroEntry, [514](#)
 - GetMacroEntry, [514](#)
 - GetName, [514](#)
 - Macro, [514](#)
 - MapModuleEntry, [514](#)
 - operator<<, [514](#)
 - SetName, [514](#)
 - Verify, [514](#)
- gdcm::Macros, [515](#)
 - AddMacro, [515](#)
 - Clear, [515](#)
 - GetMacro, [516](#)
 - IsEmpty, [516](#)
 - Macros, [515](#)
 - ModuleMapType, [515](#)
 - operator<<, [516](#)
- gdcm::MediaStorage, [518](#)
 - AmbulatoryECGWaveformStorage, [522](#)
 - Audio, [523](#)
 - BasicTextSR, [522](#)
 - BasicVoiceAudioWaveformStorage, [522](#)
 - BreastTomosynthesisImageStorage, [523](#)
 - CSANonImageStorage, [522](#)
 - CTImageStorage, [521](#)
 - CardiacElectrophysiologyWaveformStorage, [522](#)
 - ComprehensiveSR, [522](#)
 - ComputedRadiographyImageStorage, [521](#)
 - DetachedPatientManagementSOPClass, [522](#)
 - DetachedStudyManagementSOPClass, [522](#)
 - DetachedVisitManagementSOPClass, [522](#)
 - DigitalIntraoralXRayImageStorageForProcessing, [521](#)
 - DigitalIntraoralXrayImageStorageForPresentation, [521](#)
 - DigitalMammographyImageStorageForPresentation, [521](#)
 - DigitalMammographyImageStorageForProcessing, [521](#)
 - DigitalXRayImageStorageForPresentation, [521](#)
 - DigitalXRayImageStorageForProcessing, [521](#)
 - EncapsulatedCDASStorage, [522](#)
 - EncapsulatedPDFStorage, [522](#)
 - EnhancedCTImageStorage, [521](#)
 - EnhancedMRIImageStorage, [521](#)
 - EnhancedPETImageStorage, [523](#)
 - EnhancedSR, [522](#)
 - EnhancedUSVolumeStorage, [523](#)
 - EnhancedXAImageStorage, [523](#)
 - FujiPrivateCRLImageStorage, [523](#)
 - GEPrivate3DModelStorage, [522](#)
 - GeneralECGWaveformStorage, [522](#)
 - GeneralElectricMagneticResonanceImageStorage, [522](#)
 - GetMSSString, [524](#)

- GetMSType, [524](#)
- GetModality, [524](#)
- GetModalityDimension, [524](#)
- GetNumberOfMSString, [524](#)
- GetNumberOfMSType, [524](#)
- GetNumberOfModality, [524](#)
- GetString, [524](#)
- GrayscaleSoftcopyPresentationStateStorageSOP↔
Class, [522](#)
- GuessFromModality, [524](#)
- HangingProtocolStorage, [523](#)
- HardcopyGrayscaleImageStorage, [522](#)
- HemodynamicWaveformStorage, [522](#)
- IsImage, [524](#)
- IsUndefined, [524](#)
- KeyObjectSelectionDocument, [522](#)
- LeadECGWaveformStorage, [522](#)
- MRImageStorage, [521](#)
- MRSpectroscopyStorage, [521](#)
- MS_END, [523](#)
- MSType, [521](#)
- MammographyCADSR, [522](#)
- MediaStorage, [524](#)
- MediaStorageDirectoryStorage, [521](#)
- ModalityPerformedProcedureStepSOPClass, [523](#)
- MultiframeGrayscaleByteSecondaryCaptureImage↔
Storage, [521](#)
- MultiframeGrayscaleWordSecondaryCapture↔
ImageStorage, [521](#)
- MultiframeSingleBitSecondaryCaptureImage↔
Storage, [521](#)
- MultiframeTrueColorSecondaryCaptureImage↔
Storage, [522](#)
- NoObject, [523](#)
- NuclearMedicineImageStorage, [522](#)
- NuclearMedicineImageStorageRetired, [521](#)
- ObjectEnd, [523](#)
- ObjectType, [523](#)
- operator MSType, [525](#)
- operator<<, [525](#)
- OphthalmicPhotography8BitImageStorage, [523](#)
- OphthalmicTomographyImageStorage, [523](#)
- PDF, [523](#)
- PETImageStorage, [522](#)
- Philips3D, [522](#)
- PhilipsPrivateMRSyntheticImageStorage, [523](#)
- RTDoseStorage, [522](#)
- RTImageStorage, [522](#)
- RTIonBeamsTreatmentRecordStorage, [523](#)
- RTIonPlanStorage, [523](#)
- RTPlanStorage, [522](#)
- RTStructureSetStorage, [522](#)
- RTTreatmentSummaryRecordStorage, [523](#)
- RawDataStorage, [522](#)
- SecondaryCaptureImageStorage, [521](#)
- Segmentation, [523](#)
- SegmentationStorage, [523](#)
- SetFromDataSet, [525](#)
- SetFromFile, [525](#)
- SetFromHeader, [525](#)
- SetFromModality, [525](#)
- SetFromSourceImageSequence, [525](#)
- SpacialFiducialsStorage, [522](#)
- SpacialRegistrationStorage, [522](#)
- StandaloneCurveStorage, [522](#)
- StandaloneModalityLUTStorage, [522](#)
- StandaloneOverlayStorage, [522](#)
- StandaloneVOILUTStorage, [522](#)
- StudyComponentManagementSOPClass, [522](#)
- SurfaceSegmentationStorage, [523](#)
- ToshibaPrivateDataStorage, [522](#)
- URI, [523](#)
- UltrasoundImageStorage, [521](#)
- UltrasoundImageStorageRetired, [521](#)
- UltrasoundMultiFrameImageStorage, [521](#)
- UltrasoundMultiFrameImageStorageRetired, [521](#)
- VLEndoscopicImageStorage, [523](#)
- VLMicroscopicImageStorage, [523](#)
- VLPhotographicImageStorage, [523](#)
- VLWholeSlideMicroscopyImageStorage, [523](#)
- Video, [523](#)
- VideoEndoscopicImageStorage, [522](#)
- VideoPhotographicImageStorage, [523](#)
- Waveform, [523](#)
- XRay3DAngiographicImageStorage, [523](#)
- XRayAngiographicBiPlaneImageStorageRetired, [522](#)
- XRayAngiographicImageStorage, [522](#)
- XRayRadiationDoseSR, [523](#)
- XRayRadiofluoroscopicImageStorage, [522](#)
- gdcm::MemberCommand
 - ~MemberCommand, [528](#)
 - Execute, [528](#)
 - m_ConstMemberFunction, [528](#)
 - m_MemberFunction, [529](#)
 - m_This, [529](#)
 - MemberCommand, [528](#)
 - New, [528](#)
 - Self, [527](#)
 - SetCallbackFunction, [528](#)
 - TConstMemberFunctionPointer, [527](#)
 - TMemberFunctionPointer, [527](#)
- gdcm::MemberCommand< T >, [525](#)
- gdcm::MeshPrimitive, [529](#)
 - ~MeshPrimitive, [532](#)
 - AddPrimitiveData, [532](#)
 - EDGE, [531](#)
 - FACET, [531](#)
 - GetMPType, [532](#)

- GetMPTypeString, [532](#)
- GetNumberOfPrimitivesData, [532](#)
- GetPrimitiveData, [532](#)
- GetPrimitiveType, [532](#)
- GetPrimitivesData, [532](#)
- LINE, [531](#)
- MPTType, [531](#)
- MPTType_END, [531](#)
- MeshPrimitive, [532](#)
- PrimitiveData, [532](#)
- PrimitiveType, [532](#)
- PrimitivesData, [531](#)
- SetPrimitiveData, [532](#)
- SetPrimitiveType, [532](#)
- SetPrimitivesData, [532](#)
- TRIANGLE, [531](#)
- TRIANGLE_FAN, [531](#)
- TRIANGLE_STRIP, [531](#)
- VERTEX, [531](#)
- gdcmm::ModalityPerformedProcedureStepCreateQuery, [533](#)
 - GetAbstractSyntaxUID, [535](#)
 - GetRequiredDataSet, [535](#)
 - ModalityPerformedProcedureStepCreateQuery, [534](#)
 - QueryFactory, [535](#)
 - ValidateQuery, [535](#)
- gdcmm::ModalityPerformedProcedureStepSetQuery, [535](#)
 - GetAbstractSyntaxUID, [537](#)
 - GetRequiredDataSet, [537](#)
 - ModalityPerformedProcedureStepSetQuery, [537](#)
 - QueryFactory, [537](#)
 - ValidateQuery, [537](#)
- gdcmm::ModifiedEvent, [537](#)
- gdcmm::Module, [539](#)
 - AddMacro, [540](#)
 - AddModuleEntry, [540](#)
 - ArrayIncludeMacrosType, [539](#)
 - Clear, [540](#)
 - FindModuleEntryInMacros, [540](#)
 - GetModuleEntryInMacros, [540](#)
 - GetName, [540](#)
 - MapModuleEntry, [539](#)
 - Module, [540](#)
 - operator<<, [540](#)
 - SetName, [540](#)
 - Verify, [540](#)
- gdcmm::ModuleEntry, [540](#)
 - ~ModuleEntry, [542](#)
 - DataElementType, [543](#)
 - Description, [542](#)
 - DescriptionField, [543](#)
 - GetDescription, [542](#)
 - GetName, [542](#)
 - GetType, [542](#)
 - ModuleEntry, [542](#)
 - Name, [543](#)
 - operator<<, [543](#)
 - SetDescription, [543](#)
 - SetName, [543](#)
 - SetType, [543](#)
- gdcmm::Modules, [543](#)
 - AddModule, [544](#)
 - Clear, [544](#)
 - GetModule, [544](#)
 - IsEmpty, [544](#)
 - ModuleMapType, [544](#)
 - Modules, [544](#)
 - operator<<, [544](#)
- gdcmm::MovePatientRootQuery, [545](#)
 - GetAbstractSyntaxUID, [546](#)
 - GetTagListByLevel, [546](#)
 - InitializeDataSet, [546](#)
 - MovePatientRootQuery, [546](#)
 - QueryFactory, [547](#)
 - ValidateQuery, [546](#)
- gdcmm::MoveStudyRootQuery, [547](#)
 - GetAbstractSyntaxUID, [548](#)
 - GetTagListByLevel, [548](#)
 - InitializeDataSet, [549](#)
 - MoveStudyRootQuery, [548](#)
 - QueryFactory, [549](#)
 - ValidateQuery, [549](#)
- gdcmm::NestedModuleEntries, [557](#)
 - AddModuleEntry, [559](#)
 - GetModuleEntry, [559](#)
 - GetNumberOfModuleEntries, [559](#)
 - NestedModuleEntries, [559](#)
 - operator<<, [559](#)
 - SizeType, [559](#)
- gdcmm::NoEvent, [565](#)
- gdcmm::NormalizedNetworkFunctions, [566](#)
 - ConstructQuery, [567](#)
 - NAction, [567](#)
 - NCreate, [567](#)
 - NDelete, [567](#)
 - NEventReport, [567](#)
 - NGet, [568](#)
 - NSet, [568](#)
- gdcmm::Object, [571](#)
 - ~Object, [572](#)
 - Object, [572](#)
 - operator<<, [573](#)
 - operator=, [572](#)
 - Print, [572](#)
 - Register, [572](#)
 - SmartPointer, [573](#)
 - UnRegister, [573](#)
- gdcmm::OpenSSLCryptoFactory, [573](#)

- CreateCMSProvider, [574](#)
- InitOpenSSL, [574](#)
- OpenSSLCryptoFactory, [574](#)
- gdcmm::OpenSSLCryptographicMessageSyntax, [574](#)
 - ~OpenSSLCryptographicMessageSyntax, [576](#)
 - Decrypt, [576](#)
 - Encrypt, [576](#)
 - GetCipherType, [576](#)
 - OpenSSLCryptographicMessageSyntax, [576](#)
 - ParseCertificateFile, [576](#)
 - ParseKeyFile, [576](#)
 - SetCipherType, [576](#)
 - SetPassword, [576](#)
- gdcmm::OpenSSLP7CryptoFactory, [577](#)
 - CreateCMSProvider, [578](#)
 - OpenSSLP7CryptoFactory, [577](#)
- gdcmm::OpenSSLP7CryptographicMessageSyntax, [578](#)
 - ~OpenSSLP7CryptographicMessageSyntax, [579](#)
 - Decrypt, [579](#)
 - Encrypt, [580](#)
 - GetCipherType, [580](#)
 - OpenSSLP7CryptographicMessageSyntax, [579](#)
 - ParseCertificateFile, [580](#)
 - ParseKeyFile, [580](#)
 - SetCipherType, [580](#)
 - SetPassword, [580](#)
- gdcmm::Orientation, [580](#)
 - ~Orientation, [582](#)
 - AXIAL, [581](#)
 - CORONAL, [581](#)
 - GetLabel, [582](#)
 - GetMajorAxisFromPatientRelativeDirectionCosine, [582](#)
 - GetObliquityThresholdCosineValue, [582](#)
 - GetType, [582](#)
 - OBLIQUE, [581](#)
 - operator<<, [582](#)
 - Orientation, [582](#)
 - OrientationType, [581](#)
 - Print, [582](#)
 - SAGITTAL, [581](#)
 - SetObliquityThresholdCosineValue, [582](#)
 - UNKNOWN, [581](#)
- gdcmm::Overlay, [583](#)
 - ~Overlay, [585](#)
 - Decompress, [586](#)
 - GetBitPosition, [586](#)
 - GetBitsAllocated, [586](#)
 - GetColumns, [586](#)
 - GetDescription, [586](#)
 - GetGroup, [586](#)
 - GetOrigin, [586](#)
 - GetOverlayData, [586](#)
 - GetOverlayTypeAsString, [586](#)
 - GetOverlayTypeFromString, [586](#)
 - GetRows, [586](#)
 - GetType, [586](#)
 - GetTypeAsEnum, [586](#)
 - GetUnpackBuffer, [587](#)
 - GetUnpackBufferLength, [587](#)
 - GrabOverlayFromPixelData, [587](#)
 - Graphics, [585](#)
 - Invalid, [585](#)
 - IsEmpty, [587](#)
 - IsInPixelData, [587](#)
 - IsZero, [587](#)
 - operator=, [587](#)
 - Overlay, [585](#)
 - OverlayType, [585](#)
 - Print, [587](#)
 - ROI, [585](#)
 - SetBitPosition, [587](#)
 - SetBitsAllocated, [587](#)
 - SetColumns, [587](#)
 - SetDescription, [587](#)
 - SetFrameOrigin, [588](#)
 - SetGroup, [588](#)
 - SetNumberOfFrames, [588](#)
 - SetOrigin, [588](#)
 - SetOverlay, [588](#)
 - SetRows, [588](#)
 - SetType, [588](#)
 - Update, [588](#)
- gdcmm::PDBelement, [595](#)
 - GetName, [596](#)
 - GetValue, [596](#)
 - NameField, [597](#)
 - operator<<, [596](#)
 - operator==, [596](#)
 - PDBelement, [596](#)
 - SetName, [596](#)
 - SetValue, [596](#)
 - ValueField, [597](#)
- gdcmm::PDBHeader, [597](#)
 - ~PDBHeader, [598](#)
 - FindPDBelementByName, [598](#)
 - GetPDBeEnd, [598](#)
 - GetPDBelementByName, [598](#)
 - GetPDBInfoTag, [598](#)
 - LoadFromDataElement, [598](#)
 - operator<<, [599](#)
 - PDBHeader, [598](#)
 - Print, [598](#)
- gdcmm::PDFCodec, [599](#)
 - ~PDFCodec, [600](#)
 - CanCode, [600](#)
 - CanDecode, [601](#)
 - Decode, [601](#)

- PDFCodec, 600
- gdcm::PGXCodec, 604
 - ~PGXCodec, 605
 - CanCode, 605
 - CanDecode, 605
 - Clone, 606
 - GetHeaderInfo, 606
 - PGXCodec, 605
 - Read, 606
 - Write, 606
- gdcm::PNMCodec, 625
 - ~PNMCodec, 627
 - CanCode, 627
 - CanDecode, 627
 - Clone, 627
 - GetBufferLength, 627
 - GetHeaderInfo, 627
 - PNMCodec, 627
 - Read, 627
 - SetBufferLength, 628
 - Write, 628
- gdcm::PVRGCodec, 649
 - ~PVRGCodec, 650
 - CanCode, 650
 - CanDecode, 650
 - Clone, 650
 - Code, 650
 - Decode, 651
 - PVRGCodec, 650
 - SetLossyFlag, 651
- gdcm::ParseException, 588
 - ~ParseException, 590
 - GetLastElement, 590
 - operator=, 590
 - ParseException, 590
 - SetLastElement, 590
- gdcm::Parser, 590
 - ~Parser, 592
 - DuplicateAttributeError, 591
 - EndElementHandler, 591
 - ErrorType, 591
 - GetBuffer, 592
 - GetCurrentByteIndex, 592
 - GetErrorCode, 592
 - GetErrorString, 592
 - GetUserData, 592
 - JunkAfterDocElementError, 591
 - NoElementsError, 591
 - NoError, 591
 - NoMemoryError, 591
 - Parse, 592
 - ParseBuffer, 592
 - Parser, 592
 - Process, 592
 - SetElementHandler, 592
 - SetUserData, 592
 - StartElementHandler, 591
 - SyntaxError, 591
 - TagMismatchError, 591
 - UndefinedEntityError, 592
 - UnexpectedStateError, 592
- gdcm::Patient, 592
 - Patient, 593
- gdcm::PersonName, 603
 - Component, 604
 - GetMaxLength, 603
 - GetNumberOfComponents, 603
 - MaxLength, 604
 - MaxNumberOfComponents, 604
 - Padding, 604
 - Print, 603
 - Separator, 604
 - SetBlob, 603
 - SetComponents, 603
- gdcm::PhotometricInterpretation, 606
 - ARGB, 607
 - CMYK, 607
 - GetPIString, 608
 - GetPIType, 608
 - GetSamplesPerPixel, 608
 - GetString, 608
 - GetType, 608
 - HSV, 607
 - IsLossless, 608
 - IsLossy, 608
 - IsRetired, 608
 - IsSameColorSpace, 608
 - MONOCHROME1, 607
 - MONOCHROME2, 607
 - operator PIType, 608
 - operator<<, 608
 - PALETTE_COLOR, 607
 - PI_END, 607
 - PIType, 607
 - PhotometricInterpretation, 608
 - RGB, 607
 - UNKNOWN, 607
 - YBR_FULL, 607
 - YBR_FULL_422, 607
 - YBR_ICT, 607
 - YBR_PARTIAL_420, 607
 - YBR_PARTIAL_422, 607
 - YBR_RCT, 607
- gdcm::PixelFormat, 608
 - Bitmap, 613
 - FLOAT16, 611
 - FLOAT32, 611
 - FLOAT64, 611

- GetBitsAllocated, 611
- GetBitsStored, 611
- GetHighBit, 611
- GetMax, 611
- GetMin, 611
- GetPixelRepresentation, 612
- GetPixelSize, 612
- GetSamplesPerPixel, 612
- GetScalarType, 612
- GetScalarTypeAsString, 612
- INT12, 611
- INT16, 611
- INT32, 611
- INT64, 611
- INT8, 610
- IsCompatible, 612
- IsValid, 612
- operator ScalarType, 612
- operator!=, 612
- operator<<, 613
- operator==, 612, 613
- PixelFormat, 611
- Print, 613
- SINGLEBIT, 611
- ScalarType, 610
- SetBitsAllocated, 613
- SetBitsStored, 613
- SetHighBit, 613
- SetPixelRepresentation, 613
- SetSamplesPerPixel, 613
- SetScalarType, 613
- UINT12, 610
- UINT16, 611
- UINT32, 611
- UINT64, 611
- UINT8, 610
- UNKNOWN, 611
- Validate, 613
- gdcmm::Pixmap, 614
 - ~Pixmap, 615
 - AreOverlaysInPixelData, 615
 - Curves, 616
 - GetCurve, 616
 - GetIconImage, 616
 - GetNumberOfCurves, 616
 - GetNumberOfOverlays, 616
 - GetOverlay, 616
 - Icon, 616
 - Overlays, 616
 - Pixmap, 615
 - Print, 616
 - RemoveOverlay, 616
 - SetIconImage, 616
 - SetNumberOfCurves, 616
 - SetNumberOfOverlays, 616
 - gdcmm::PixmapReader, 617
 - ~PixmapReader, 619
 - GetPixmap, 619
 - PixelData, 620
 - PixmapReader, 619
 - Read, 619
 - ReadACRNEMAIImage, 619
 - ReadImage, 619
 - ReadImageInternal, 619
 - gdcmm::PixmapToPixmapFilter, 620
 - ~PixmapToPixmapFilter, 621
 - GetInput, 622
 - GetOutput, 622
 - GetOutputAsPixmap, 622
 - PixmapToPixmapFilter, 621
 - gdcmm::PixmapWriter, 622
 - ~PixmapWriter, 624
 - DolconImage, 624
 - GetImage, 624
 - GetPixmap, 624
 - PixelData, 625
 - PixmapWriter, 624
 - PrepareWrite, 624
 - SetImage, 624
 - SetPixmap, 625
 - Write, 625
 - gdcmm::Preamble, 628
 - ~Preamble, 629
 - Clear, 629
 - Create, 629
 - GetInternal, 629
 - GetLength, 629
 - IsEmpty, 629
 - IsValid, 629
 - operator<<, 629
 - operator=, 629
 - Preamble, 629
 - Print, 629
 - Read, 629
 - Remove, 629
 - Valid, 629
 - Write, 629
 - gdcmm::PresentationContext, 629
 - AbstractSyntax, 632
 - AddTransferSyntax, 631
 - GetAbstractSyntax, 631
 - GetNumberOfTransferSyntaxes, 631
 - GetPresentationContextID, 631
 - GetTransferSyntax, 631
 - ID, 632
 - operator==, 631
 - PresentationContext, 631
 - Print, 631

- SetAbstractSyntax, [631](#)
- SetPresentationContextID, [631](#)
- SizeType, [631](#)
- TransferSyntaxArrayType, [631](#)
- TransferSyntaxes, [632](#)
- gdcmm::PresentationContextGenerator, [633](#)
 - AddFromFile, [634](#)
 - AddPresentationContext, [634](#)
 - GenerateFromFilenames, [635](#)
 - GenerateFromUID, [635](#)
 - GetDefaultTransferSyntax, [635](#)
 - GetPresentationContexts, [635](#)
 - PresentationContextArrayType, [634](#)
 - PresentationContextGenerator, [634](#)
 - SetDefaultTransferSyntax, [635](#)
 - SetMergeModeToAbstractSyntax, [635](#)
 - SetMergeModeToTransferSyntax, [635](#)
 - SizeType, [634](#)
- gdcmm::Printer, [639](#)
 - ~Printer, [641](#)
 - CONDENSED_STYLE, [641](#)
 - F, [642](#)
 - GetPrintStyle, [641](#)
 - MaxPrintLength, [642](#)
 - Print, [641](#)
 - PrintDataElement, [641](#)
 - PrintDataSet, [642](#)
 - PrintSQ, [642](#)
 - PrintStyle, [642](#)
 - PrintStyles, [641](#)
 - Printer, [641](#)
 - SetColor, [642](#)
 - SetFile, [642](#)
 - SetStyle, [642](#)
 - VERBOSE_STYLE, [641](#)
 - XML, [641](#)
- gdcmm::PrivateDict, [642](#)
 - ~PrivateDict, [643](#)
 - AddDictEntry, [643](#)
 - Dicts, [644](#)
 - FindDictEntry, [643](#)
 - GetDictEntry, [643](#)
 - IsEmpty, [643](#)
 - LoadDefault, [643](#)
 - operator<<, [644](#)
 - PrintXML, [643](#)
 - PrivateDict, [643](#)
 - RemoveDictEntry, [643](#)
- gdcmm::PrivateTag, [644](#)
 - GetAsDataElement, [646](#)
 - GetOwner, [646](#)
 - operator<, [646](#)
 - operator<<, [646](#)
 - PrivateTag, [645](#)
 - ReadFromCommaSeparatedString, [646](#)
 - SetOwner, [646](#)
- gdcmm::ProgressEvent, [646](#)
 - ~ProgressEvent, [648](#)
 - CheckEvent, [648](#)
 - GetEventName, [648](#)
 - GetProgress, [648](#)
 - MakeObject, [648](#)
 - ProgressEvent, [648](#)
 - Self, [648](#)
 - SetProgress, [648](#)
 - Superclass, [648](#)
- gdcmm::PythonFilter, [651](#)
 - ~PythonFilter, [651](#)
 - GetFile, [651](#), [652](#)
 - PythonFilter, [651](#)
 - SetDicts, [652](#)
 - SetFile, [652](#)
 - ToPyObject, [652](#)
 - UseDictAlways, [652](#)
- gdcmm::QueryBase, [652](#)
 - ~QueryBase, [653](#)
 - GetAllRequiredTags, [653](#)
 - GetAllTags, [653](#)
 - GetHierarchicalSearchTags, [653](#)
 - GetName, [653](#)
 - GetOptionalTags, [653](#)
 - GetQueryLevel, [653](#)
 - GetRequiredTags, [654](#)
 - GetUniqueTags, [654](#)
- gdcmm::QueryFactory, [654](#)
 - GetCharacterFromCurrentLocale, [655](#)
 - ListCharSets, [655](#)
 - ProduceCharacterSetDataElement, [655](#)
 - ProduceQuery, [655](#)
- gdcmm::QueryImage, [655](#)
 - GetHierarchicalSearchTags, [656](#)
 - GetName, [657](#)
 - GetOptionalTags, [657](#)
 - GetQueryLevel, [657](#)
 - GetRequiredTags, [657](#)
 - GetUniqueTags, [657](#)
- gdcmm::QueryPatient, [657](#)
 - GetHierarchicalSearchTags, [658](#)
 - GetName, [659](#)
 - GetOptionalTags, [659](#)
 - GetQueryLevel, [659](#)
 - GetRequiredTags, [659](#)
 - GetUniqueTags, [659](#)
- gdcmm::QuerySeries, [659](#)
 - GetHierarchicalSearchTags, [660](#)
 - GetName, [661](#)
 - GetOptionalTags, [661](#)
 - GetQueryLevel, [661](#)

- GetRequiredTags, 661
- GetUniqueTags, 661
- gdcmm::QueryStudy, 661
 - GetHierarchicalSearchTags, 662
 - GetName, 663
 - GetOptionalTags, 663
 - GetQueryLevel, 663
 - GetRequiredTags, 663
 - GetUniqueTags, 663
- gdcmm::RAWCodec, 663
 - ~RAWCodec, 665
 - CanCode, 665
 - CanDecode, 665
 - Clone, 665
 - Code, 665
 - Decode, 665
 - DecodeByStreams, 665
 - DecodeBytes, 666
 - GetHeaderInfo, 666
 - RAWCodec, 665
- gdcmm::RLECodec, 677
 - ~RLECodec, 679
 - AppendFrameEncode, 679
 - AppendRowEncode, 679
 - CanCode, 680
 - CanDecode, 680
 - Clone, 680
 - Code, 680
 - Decode, 680
 - DecodeByStreams, 680
 - DecodeExtent, 680
 - GetBufferLength, 680
 - GetHeaderInfo, 680
 - ImageRegionReader, 681
 - IsFrameEncoder, 680
 - IsRowEncoder, 680
 - RLECodec, 679
 - SetBufferLength, 681
 - SetLength, 681
 - StartEncode, 681
 - StopEncode, 681
- gdcmm::Reader, 666
 - ~Reader, 669
 - CanRead, 669
 - F, 671
 - GetFile, 669
 - GetStreamCurrentPosition, 669
 - GetStreamPtr, 669
 - Read, 669
 - ReadDataSet, 670
 - ReadMetaInformation, 670
 - ReadPreamble, 670
 - ReadSelectedPrivateTags, 670
 - ReadSelectedTags, 670
 - ReadUpToTag, 670
 - Reader, 669
 - SetFile, 670
 - SetFileName, 670
 - SetStream, 671
 - StreamImageReader, 671
- gdcmm::RealWorldValueMappingContent, 671
 - CodeMeaning, 672
 - CodeValue, 672
 - RealWorldValueIntercept, 672
 - RealWorldValueSlope, 672
- gdcmm::Region, 672
 - ~Region, 673
 - Area, 673
 - Clone, 674
 - ComputeBoundingBox, 674
 - Empty, 674
 - IsValid, 674
 - Print, 674
 - Region, 673
- gdcmm::Rescaler, 674
 - ~Rescaler, 676
 - ComputeInterceptSlopePixelType, 676
 - ComputePixelTypeFromMinMax, 676
 - GetIntercept, 676
 - GetSlope, 676
 - InverseRescale, 676
 - InverseRescaleFunctionIntoBestFit, 676
 - Rescale, 676
 - RescaleFunctionIntoBestFit, 676
 - Rescaler, 676
 - SetIntercept, 676
 - SetMinMaxForPixelType, 676
 - SetPixelFormat, 677
 - SetSlope, 677
 - SetTargetPixelType, 677
 - SetUseTargetPixelType, 677
- gdcmm::SHA1, 720
 - ~SHA1, 721
 - Compute, 721
 - ComputeFile, 721
 - SHA1, 721
- gdcmm::SOPClassUIDToIOD, 730
 - const, 731
 - GetIOD, 731
 - GetIODFromSOPClassUID, 731
 - GetNumberOfSOPClassToIOD, 731
 - GetSOPClassUIDFromIOD, 731
 - GetSOPClassUIDToIOD, 731
 - GetSOPClassUIDToIODs, 731
- gdcmm::STATIC_ASSERTION_FAILURE< true >, 740
 - value, 740
- gdcmm::STATIC_ASSERTION_FAILURE< x >, 740
- gdcmm::Scanner, 683

- ~Scanner, [686](#)
- AddPrivateTag, [686](#)
- AddSkipTag, [686](#)
- AddTag, [686](#)
- Begin, [686](#)
- ClearSkipTags, [687](#)
- ClearTags, [687](#)
- ConstIterator, [686](#)
- End, [687](#)
- GetAllFileNamesFromTagToValue, [687](#)
- GetFilenameFromTagToValue, [687](#)
- GetFileNames, [687](#)
- GetKeys, [687](#)
- GetMapping, [687](#)
- GetMappingFromTagToValue, [687](#)
- GetMappings, [687](#)
- GetOrderedValues, [687](#)
- GetValue, [687](#)
- GetValues, [688](#)
- IsKey, [688](#)
- MappingType, [686](#)
- New, [688](#)
- operator<<, [689](#)
- Print, [688](#)
- ProcessPublicTag, [688](#)
- Scan, [688](#)
- Scanner, [686](#)
- TagToValue, [686](#)
- TagToValueValueType, [686](#)
- ValuesType, [686](#)
- gdcmm::Scanner::Itstr, [512](#)
- operator(), [512](#)
- gdcmm::Segment, [689](#)
- ~Segment, [691](#)
- ALGOType, [691](#)
- ALGOType_END, [691](#)
- AUTOMATIC, [691](#)
- AddSurface, [691](#)
- AnatomicRegion, [692](#)
- GetALGOType, [691](#)
- GetALGOTypeString, [691](#)
- GetAnatomicRegion, [691](#)
- GetPropertyCategory, [691](#)
- GetPropertyType, [691](#)
- GetSegmentAlgorithmName, [691](#)
- GetSegmentAlgorithmType, [692](#)
- GetSegmentDescription, [692](#)
- GetSegmentLabel, [692](#)
- GetSegmentNumber, [692](#)
- GetSurface, [692](#)
- GetSurfaceCount, [692](#)
- GetSurfaces, [692](#)
- MANUAL, [691](#)
- PropertyCategory, [692](#)
- PropertyType, [692](#)
- Segment, [691](#)
- SegmentAlgorithmName, [692](#)
- SegmentAlgorithmType, [692](#)
- SegmentDescription, [692](#)
- SegmentLabel, [692](#)
- SegmentNumber, [693](#)
- SetAnatomicRegion, [692](#)
- SetPropertyCategory, [692](#)
- SetPropertyType, [692](#)
- SetSegmentAlgorithmName, [692](#)
- SetSegmentAlgorithmType, [692](#)
- SetSegmentDescription, [692](#)
- SetSegmentLabel, [692](#)
- SetSegmentNumber, [692](#)
- SetSurfaceCount, [692](#)
- SurfaceCount, [693](#)
- SurfaceVector, [691](#)
- Surfaces, [693](#)
- gdcmm::SegmentHelper, [142](#)
- gdcmm::SegmentHelper::BasicCodedEntry, [212](#)
- BasicCodedEntry, [213](#)
- CM, [213](#)
- CSD, [213](#)
- CSV, [213](#)
- CV, [214](#)
- IsEmpty, [213](#)
- gdcmm::SegmentReader, [695](#)
- ~SegmentReader, [697](#)
- GetSegments, [697](#)
- Read, [697](#)
- ReadSegment, [697](#)
- ReadSegments, [697](#)
- SegmentMap, [697](#)
- SegmentReader, [697](#)
- SegmentVector, [697](#)
- Segments, [697](#)
- gdcmm::SegmentWriter, [698](#)
- ~SegmentWriter, [699](#)
- AddSegment, [699](#)
- GetNumberOfSegments, [699](#)
- GetSegment, [699](#)
- GetSegments, [699](#)
- PrepareWrite, [699](#)
- SegmentVector, [699](#)
- SegmentWriter, [699](#)
- Segments, [700](#)
- SetNumberOfSegments, [699](#)
- SetSegments, [699](#)
- Write, [699](#)
- gdcmm::SegmentedPaletteColorLookupTable, [693](#)
- ~SegmentedPaletteColorLookupTable, [694](#)
- Print, [694](#)
- SegmentedPaletteColorLookupTable, [694](#)

- SetLUT, 694
- gdcmm::SequenceOfFragments, 700
 - AddFragment, 703
 - Begin, 703
 - Clear, 703
 - ComputeByteLength, 703
 - ComputeLength, 703
 - ConstIterator, 702
 - End, 703
 - FragmentVector, 702
 - GetBuffer, 703
 - GetFragBuffer, 703
 - GetFragment, 703
 - GetLength, 703
 - GetNumberOfFragments, 703
 - GetTable, 703, 704
 - Iterator, 702
 - New, 704
 - operator==, 704
 - Print, 704
 - Read, 704
 - ReadPreValue, 704
 - ReadValue, 704
 - SequenceOfFragments, 702
 - SetLength, 704
 - SizeType, 702
 - Write, 704
 - WriteBuffer, 704
- gdcmm::SequenceOfItems, 705
 - AddItem, 708
 - AddNewUndefinedLengthItem, 708
 - Begin, 708
 - Clear, 708
 - ComputeLength, 708
 - ConstIterator, 708
 - End, 709
 - FindDataElement, 709
 - GetItem, 709
 - GetLength, 709
 - GetNumberOfItems, 709
 - IsUndefinedLength, 709
 - ItemVector, 708
 - Items, 710
 - Iterator, 708
 - New, 709
 - operator=, 709
 - operator==, 709
 - Print, 709
 - Read, 710
 - RemoveItemByIndex, 710
 - SequenceLengthField, 710
 - SequenceOfItems, 708
 - SetLength, 710
 - SetLengthToUndefined, 710
 - SetNumberOfItems, 710
 - SizeType, 708
 - Write, 710
- gdcmm::SerieHelper, 711
 - ~SerieHelper, 712
 - AddFile, 713
 - AddFileName, 713
 - AddRestriction, 713
 - Clear, 713
 - CreateDefaultUniqueSeriesIdentifier, 713
 - CreateUniqueSeriesIdentifier, 713
 - FileNameOrdering, 713
 - GetFirstSingleSerieUIDFileSet, 713
 - GetNextSingleSerieUIDFileSet, 713
 - ImagePositionPatientOrdering, 713
 - ItFileSetHt, 713
 - OrderFileList, 713
 - SerieHelper, 712
 - SerieRestrictions, 712
 - SetDirectory, 713
 - SetLoadMode, 713
 - SetUseSeriesDetails, 713
 - SingleSerieUIDFileSetHT, 713
 - SingleSerieUIDFileSetmap, 712
 - UserOrdering, 713
- gdcmm::SerieHelper::Rule, 682
 - elem, 683
 - group, 683
 - op, 683
 - value, 683
- gdcmm::Series, 713
 - Series, 714
- gdcmm::ServiceClassUser, 715
 - ~ServiceClassUser, 717
 - GetAETitle, 717
 - GetCalledAETitle, 717
 - GetTimeout, 718
 - InitializeConnection, 718
 - IsPresentationContextAccepted, 718
 - New, 718
 - SendEcho, 718
 - SendFind, 718
 - SendMove, 718
 - SendStore, 718, 719
 - ServiceClassUser, 717
 - SetAETitle, 719
 - SetCalledAETitle, 719
 - SetHostname, 719
 - SetPort, 719
 - SetPortSCP, 719
 - SetPresentationContexts, 719
 - SetTimeout, 719
 - StartAssociation, 720
 - StopAssociation, 720

- gdcmm::SimpleMemberCommand
 - ~SimpleMemberCommand, 724
 - Execute, 724
 - m_MemberFunction, 724
 - m_This, 724
 - New, 724
 - Self, 723
 - SetCallbackFunction, 724
 - SimpleMemberCommand, 724
 - TMemberFunctionPointer, 723
- gdcmm::SimpleMemberCommand< T >, 721
- gdcmm::SimpleSubjectWatcher, 725
 - ~SimpleSubjectWatcher, 725
 - EndFilter, 725
 - ShowAbort, 726
 - ShowAnonymization, 726
 - ShowData, 726
 - ShowDataSet, 726
 - ShowFileName, 726
 - ShowIteration, 726
 - ShowProgress, 726
 - SimpleSubjectWatcher, 725
 - StartFilter, 726
 - TestAbortOff, 726
 - TestAbortOn, 726
- gdcmm::SmartPointer
 - ~SmartPointer, 728
 - GetPointer, 728
 - operator ObjectType *, 729
 - operator*, 729
 - operator->, 729
 - operator=, 729
 - SmartPointer, 728
- gdcmm::SmartPointer< ObjectType >, 726
- gdcmm::Sorter, 731
 - ~Sorter, 734
 - AddSelect, 734
 - FileNames, 735
 - GetFileNames, 734
 - operator<<, 735
 - Print, 734
 - Selection, 735
 - SelectionMap, 733
 - SetSortFunction, 734
 - Sort, 734
 - SortFunc, 735
 - SortFunction, 733
 - Sorter, 734
 - StableSort, 734
- gdcmm::Spacing, 735
 - ~Spacing, 736
 - CALIBRATED, 736
 - ComputePixelAspectRatioFromPixelSpacing, 736
 - DETECTOR, 736
 - MAGNIFIED, 736
 - Spacing, 736
 - SpacingType, 736
 - UNKNOWN, 736
- gdcmm::Spectroscopy, 737
 - Spectroscopy, 737
- gdcmm::SplitMosaicFilter, 737
 - ~SplitMosaicFilter, 738
 - ComputeMOSAICDimensions, 738
 - GetFile, 738
 - GetImage, 738
 - SetFile, 738
 - SetImage, 738
 - Split, 738
 - SplitMosaicFilter, 738
- gdcmm::StartEvent, 738
- gdcmm::StreamImageReader, 740
 - ~StreamImageReader, 741
 - CanReadImage, 741
 - DefinePixelExtent, 741
 - DefineProperBufferLength, 742
 - GetDimensionsValueForResolution, 742
 - GetFile, 742
 - Read, 742
 - ReadImageInformation, 742
 - SetFileName, 742
 - SetStream, 743
 - StreamImageReader, 741
- gdcmm::StreamImageWriter, 743
 - ~StreamImageWriter, 745
 - CanWriteFile, 745
 - DefinePixelExtent, 746
 - DefineProperBufferLength, 746
 - mElementOffsets, 747
 - mElementOffsets1, 747
 - mWriter, 748
 - mXMax, 748
 - mXMin, 748
 - mYMax, 748
 - mYMin, 748
 - mZMax, 748
 - mZMin, 748
 - mspFile, 748
 - SetFile, 746
 - SetFileName, 746
 - SetStream, 746
 - StreamImageWriter, 745
 - Write, 746
 - WriteImageInformation, 747
 - WriteImageSubregionRAW, 747
 - WriteRawHeader, 747
- gdcmm::StrictScanner, 748
 - ~StrictScanner, 751
 - AddPrivateTag, 751

- AddSkipTag, [751](#)
- AddTag, [751](#)
- Begin, [751](#)
- ClearSkipTags, [752](#)
- ClearTags, [752](#)
- ConstIterator, [751](#)
- End, [752](#)
- GetAllFileNamesFromTagToValue, [752](#)
- GetFilenameFromTagToValue, [752](#)
- GetFileNames, [752](#)
- GetKeys, [752](#)
- GetMapping, [752](#)
- GetMappingFromTagToValue, [752](#)
- GetMappings, [752](#)
- GetOrderedValues, [752](#)
- GetValue, [752](#)
- GetValues, [753](#)
- IsKey, [753](#)
- MappingType, [751](#)
- New, [753](#)
- operator<<, [753](#)
- Print, [753](#)
- ProcessPublicTag, [753](#)
- Scan, [753](#)
- StrictScanner, [751](#)
- TagToValue, [751](#)
- TagToValueValueType, [751](#)
- ValueType, [751](#)
- gdcmm::StrictScanner::Itstr, [513](#)
- operator(), [513](#)
- gdcmm::String
 - const_iterator, [755](#)
 - const_reference, [755](#)
 - const_reverse_iterator, [756](#)
 - difference_type, [756](#)
 - IsValid, [756](#)
 - iterator, [756](#)
 - operator const char *, [756](#)
 - pointer, [756](#)
 - reference, [756](#)
 - reverse_iterator, [756](#)
 - size_type, [756](#)
 - String, [756](#)
 - Trim, [757](#)
 - Truncate, [757](#)
 - value_type, [756](#)
- gdcmm::String< TDelimiter, TMaxLength, TPadChar >, [754](#)
- gdcmm::StringFilter, [757](#)
- ~StringFilter, [758](#)
- ExecuteQuery, [758](#)
- FromString, [758](#)
- GetFile, [758](#)
- SetDicts, [758](#)
- SetFile, [758](#)
- StringFilter, [758](#)
- ToString, [758](#), [759](#)
- ToStringPair, [759](#)
- UseDictAlways, [759](#)
- gdcmm::Study, [759](#)
- Study, [760](#)
- gdcmm::Subject, [760](#)
- ~Subject, [761](#)
- AddObserver, [761](#), [762](#)
- GetCommand, [762](#)
- HasObserver, [762](#)
- InvokeEvent, [762](#)
- RemoveAllObservers, [762](#)
- RemoveObserver, [762](#)
- Subject, [761](#)
- gdcmm::Surface, [762](#)
- ~Surface, [766](#)
- GetAlgorithmFamily, [766](#)
- GetAlgorithmName, [766](#)
- GetAlgorithmVersion, [766](#)
- GetAxisOfRotation, [766](#)
- GetCenterOfRotation, [766](#)
- GetFiniteVolume, [766](#)
- GetManifold, [766](#)
- GetMaximumPointDistance, [766](#)
- GetMeanPointDistance, [766](#)
- GetMeshPrimitive, [766](#)
- GetNumberOfSurfacePoints, [766](#)
- GetNumberOfVectors, [767](#)
- GetPointCoordinatesData, [767](#)
- GetPointPositionAccuracy, [767](#)
- GetPointsBoundingBoxCoordinates, [767](#)
- GetProcessingAlgorithm, [767](#)
- GetRecommendedDisplayCIELabValue, [767](#)
- GetRecommendedDisplayGrayscaleValue, [767](#)
- GetRecommendedPresentationOpacity, [767](#)
- GetRecommendedPresentationType, [767](#)
- GetSTATES, [767](#)
- GetSTATESString, [767](#)
- GetSurfaceComments, [767](#)
- GetSurfaceNumber, [767](#)
- GetSurfaceProcessing, [767](#)
- GetSurfaceProcessingDescription, [767](#)
- GetSurfaceProcessingRatio, [767](#)
- GetVIEWType, [768](#)
- GetVIEWTypeString, [768](#)
- GetVectorAccuracy, [767](#)
- GetVectorCoordinateData, [767](#)
- GetVectorDimensionality, [768](#)
- NO, [765](#)
- POINTS, [766](#)
- STATES, [765](#)
- STATES_END, [765](#)
- SURFACE, [766](#)

- SetAlgorithmFamily, 768
- SetAlgorithmName, 768
- SetAlgorithmVersion, 768
- SetAxisOfRotation, 768
- SetCenterOfRotation, 768
- SetFiniteVolume, 768
- SetManifold, 768
- SetMaximumPointDistance, 768
- SetMeanPointDistance, 768
- SetMeshPrimitive, 768
- SetNumberOfSurfacePoints, 768
- SetNumberOfVectors, 768
- SetPointCoordinatesData, 768
- SetPointPositionAccuracy, 768
- SetPointsBoundingBoxCoordinates, 768
- SetProcessingAlgorithm, 768
- SetRecommendedDisplayCIELabValue, 768
- SetRecommendedDisplayGrayscaleValue, 768
- SetRecommendedPresentationOpacity, 768
- SetRecommendedPresentationType, 768
- SetSurfaceComments, 768
- SetSurfaceNumber, 769
- SetSurfaceProcessing, 769
- SetSurfaceProcessingDescription, 769
- SetSurfaceProcessingRatio, 769
- SetVectorAccuracy, 769
- SetVectorCoordinateData, 769
- SetVectorDimensionality, 769
- Surface, 766
- UNKNOWN, 765
- VIEWType, 765
- VIEWType_END, 766
- WIREFRAME, 766
- YES, 765
- gdcmm::SurfaceHelper, 769
 - ColorArray, 770
 - RGBToRecommendedDisplayCIELab, 770
 - RGBToRecommendedDisplayGrayscale, 771
 - RecommendedDisplayCIELabToRGB, 770
- gdcmm::SurfaceReader, 771
 - ~SurfaceReader, 773
 - GetNumberOfSurfaces, 773
 - Read, 773
 - ReadPointMacro, 773
 - ReadSurface, 773
 - ReadSurfaces, 773
 - SurfaceReader, 773
- gdcmm::SurfaceWriter, 774
 - ~SurfaceWriter, 775
 - ComputeNumberOfSurfaces, 775
 - GetNumberOfSurfaces, 775
 - NumberOfSurfaces, 775
 - PrepareWrite, 775
 - PrepareWritePointMacro, 775
 - SetNumberOfSurfaces, 775
 - SurfaceWriter, 775
 - Write, 775
- gdcmm::SwapCode, 775
 - BadBigEndian, 776
 - BadLittleEndian, 776
 - BigEndian, 776
 - GetIndex, 777
 - GetSwapCodeString, 777
 - LittleEndian, 776
 - operator SwapCode::SwapCodeType, 777
 - operator<<, 777
 - SwapCode, 777
 - SwapCodeType, 776
 - Unknown, 776
- gdcmm::SwapperDoOp, 777
 - Swap, 777
 - SwapArray, 777
- gdcmm::SwapperNoOp, 777
 - Swap, 778
 - SwapArray, 778
- gdcmm::System, 778
 - DeleteDirectory, 779
 - EncodeBytes, 779
 - FileExists, 779
 - FileIsDirectory, 779
 - FileIsSymlink, 780
 - FileSize, 780
 - FileTime, 780
 - FormatDateTime, 780
 - GetCWD, 781
 - GetCurrentDateTime, 780
 - GetCurrentModuleFileName, 780
 - GetCurrentProcessFileName, 780
 - GetCurrentResourcesDirectory, 780
 - GetHostName, 781
 - GetLastSystemError, 781
 - GetLocaleCharset, 781
 - GetPermissions, 781
 - GetTimezoneOffsetFromUTC, 781
 - MakeDirectory, 781
 - ParseDateTime, 781
 - RemoveFile, 782
 - SetPermissions, 782
 - StrCaseCmp, 782
 - StrNCaseCmp, 782
 - StrSep, 782
 - StrTokR, 782
- gdcmm::Table, 782
 - ~Table, 783
 - GetTableEntry, 783
 - InsertEntry, 783
 - MapTableEntry, 783
 - operator<<, 783

- Table, [783](#)
- gdcmm::TableEntry, [783](#)
 - ~TableEntry, [784](#)
 - TableEntry, [784](#)
- gdcmm::TableReader, [784](#)
 - ~TableReader, [785](#)
 - CharacterDataHandler, [785](#)
 - EndElement, [785](#)
 - GetDefs, [785](#)
 - GetFilename, [785](#)
 - HandleIOD, [785](#)
 - HandleIODEntry, [785](#)
 - HandleMacro, [785](#)
 - HandleMacroEntry, [785](#)
 - HandleMacroEntryDescription, [785](#)
 - HandleModule, [785](#)
 - HandleModuleEntry, [785](#)
 - HandleModuleEntryDescription, [785](#)
 - HandleModuleInclude, [786](#)
 - Read, [786](#)
 - SetFilename, [786](#)
 - StartElement, [786](#)
 - TableReader, [785](#)
- gdcmm::Tag, [787](#)
 - bytes, [793](#)
 - GetElement, [789](#)
 - GetElementTag, [790](#)
 - GetGroup, [790](#)
 - GetLength, [790](#)
 - GetPrivateCreator, [790](#)
 - IsGroupLength, [790](#)
 - IsGroupXX, [790](#)
 - IsIllegal, [790](#)
 - IsPrivate, [790](#)
 - IsPrivateCreator, [791](#)
 - IsPublic, [791](#)
 - operator!=, [791](#)
 - operator<, [791](#)
 - operator<<, [793](#)
 - operator<=, [791](#)
 - operator>>, [793](#)
 - operator=, [791](#)
 - operator==, [791](#)
 - operator[], [791](#)
 - PrintAsContinuousString, [791](#)
 - PrintAsContinuousUpperCaseString, [792](#)
 - PrintAsPipeSeparatedString, [792](#)
 - Read, [792](#)
 - ReadFromCommaSeparatedString, [792](#)
 - ReadFromContinuousString, [792](#)
 - ReadFromPipeSeparatedString, [792](#)
 - SetElement, [792](#)
 - SetElementTag, [792](#), [793](#)
 - SetGroup, [793](#)
 - SetPrivateCreator, [793](#)
 - Tag, [789](#)
 - tag, [793](#)
 - tags, [793](#)
 - Write, [793](#)
- gdcmm::TagPath, [794](#)
 - ~TagPath, [794](#)
 - ConstructFromString, [794](#)
 - ConstructFromTagList, [794](#)
 - IsValid, [794](#)
 - Print, [794](#)
 - Push, [795](#)
 - TagPath, [794](#)
- gdcmm::Testing, [795](#)
 - ~Testing, [796](#)
 - ComputeFileMD5, [796](#)
 - ComputeMD5, [796](#)
 - GetDataExtraRoot, [797](#)
 - GetDataRoot, [797](#)
 - GetFileName, [797](#)
 - GetFileNames, [797](#)
 - GetLossyFlagFromFile, [797](#)
 - GetMD5DataImage, [797](#)
 - GetMD5DataImages, [797](#)
 - GetMD5FromBrokenFile, [797](#)
 - GetMD5FromFile, [797](#)
 - GetMediaStorageDataFile, [798](#)
 - GetMediaStorageDataFiles, [798](#)
 - GetMediaStorageFromFile, [798](#)
 - GetNumberOfFileNames, [798](#)
 - GetNumberOfMD5DataImages, [798](#)
 - GetNumberOfMediaStorageDataFiles, [798](#)
 - GetPixelSpacingDataRoot, [798](#)
 - GetSelectedPrivateGroupOffsetFromFile, [798](#)
 - GetSelectedTagsOffsetFromFile, [798](#)
 - GetSourceDirectory, [798](#)
 - GetStreamOffsetFromFile, [798](#)
 - GetTempDirectory, [798](#)
 - GetTempDirectoryW, [799](#)
 - GetTempFilename, [799](#)
 - GetTempFilenameW, [799](#)
 - MD5DataImagesType, [796](#)
 - MediaStorageDataFilesType, [796](#)
 - Print, [799](#)
 - Testing, [796](#)
- gdcmm::Trace, [799](#)
 - ~Trace, [800](#)
 - DebugOff, [800](#)
 - DebugOn, [801](#)
 - ErrorOff, [801](#)
 - ErrorOn, [801](#)
 - GetDebugFlag, [801](#)
 - GetDebugStream, [801](#)
 - GetErrorFlag, [801](#)

- GetErrorStream, [801](#)
- GetStream, [801](#)
- GetWarningFlag, [801](#)
- GetWarningStream, [801](#)
- SetDebug, [801](#)
- SetDebugStream, [801](#)
- SetError, [801](#)
- SetErrorStream, [801](#)
- SetStream, [802](#)
- SetStreamToFile, [802](#)
- SetWarning, [802](#)
- SetWarningStream, [802](#)
- Trace, [800](#)
- WarningOff, [802](#)
- WarningOn, [802](#)
- gdcmm::TransferSyntax, [802](#)
 - CT_private_ELE, [805](#)
 - CanStoreLossy, [805](#)
 - DeflatedExplicitVRLittleEndian, [804](#)
 - Explicit, [804](#)
 - ExplicitVRBigEndian, [804](#)
 - ExplicitVRLittleEndian, [804](#)
 - GetNegociatedType, [805](#)
 - GetString, [805](#)
 - GetSwapCode, [805](#)
 - GetTSString, [805](#)
 - GetTSType, [806](#)
 - Implicit, [804](#)
 - ImplicitVRBigEndianACRNEMA, [805](#)
 - ImplicitVRBigEndianPrivateGE, [804](#)
 - ImplicitVRLittleEndian, [804](#)
 - IsEncapsulated, [806](#)
 - IsEncoded, [806](#)
 - IsExplicit, [806](#)
 - IsImplicit, [806](#)
 - IsLossless, [806](#)
 - IsLossy, [806](#)
 - IsValid, [806](#)
 - JPEG2000, [805](#)
 - JPEG2000Lossless, [805](#)
 - JPEG2000Part2, [805](#)
 - JPEG2000Part2Lossless, [805](#)
 - JPEGBaselineProcess1, [805](#)
 - JPEGExtendedProcess2_4, [805](#)
 - JPEGExtendedProcess3_5, [805](#)
 - JPEGFullProgressionProcess10_12, [805](#)
 - JPEGLSLossless, [805](#)
 - JPEGLSNearLossless, [805](#)
 - JPEGLosslessProcess14, [805](#)
 - JPEGLosslessProcess14_1, [805](#)
 - JPEGSpectralSelectionProcess6_8, [805](#)
 - JPIPRereferenced, [805](#)
 - MPEG2MainProfile, [805](#)
 - MPEG2MainProfileHighLevel, [805](#)
 - MPEG4AVCH264BDcompatibleHighProfileLevel4↔_1, [805](#)
 - MPEG4AVCH264HighProfileLevel4_1, [805](#)
 - NegociatedType, [804](#)
 - operator TSType, [806](#)
 - operator<<, [806](#)
 - RLELossless, [805](#)
 - TS_END, [805](#)
 - TSType, [804](#)
 - TransferSyntax, [805](#)
 - Unknown, [804](#)
- gdcmm::Type, [809](#)
 - GetTypeString, [810](#)
 - GetTypeType, [810](#)
 - operator TypeType, [810](#)
 - operator<<, [810](#)
 - T1, [810](#)
 - T1C, [810](#)
 - T2, [810](#)
 - T2C, [810](#)
 - T3, [810](#)
 - Type, [810](#)
 - TypeType, [810](#)
 - UNKNOWN, [810](#)
- gdcmm::UI, [811](#)
 - Internal, [811](#)
 - operator<<, [811](#)
- gdcmm::UIDGenerator, [811](#)
 - Generate, [812](#)
 - GenerateUUID, [812](#)
 - GetGDCMUID, [812](#)
 - GetRoot, [812](#)
 - IsValid, [813](#)
 - SetRoot, [813](#)
 - UIDGenerator, [812](#)
- gdcmm::UIDs, [813](#)
 - AmbulatoryECGWaveformStorage, [821](#)
 - AudioSRStorageTrialRetired, [822](#)
 - BasicAnnotationBoxSOPClass, [820](#)
 - BasicColorImageBoxSOPClass, [820](#)
 - BasicColorPrintManagementMetaSOPClass, [820](#)
 - BasicFilmBoxSOPClass, [820](#)
 - BasicFilmSessionSOPClass, [820](#)
 - BasicGrayscaleImageBoxSOPClass, [820](#)
 - BasicGrayscalePrintManagementMetaSOPClass, [820](#)
 - BasicPrintImageOverlayBoxSOPClassRetired, [821](#)
 - BasicStudyContentNotificationSOPClassRetired, [820](#)
 - BasicTextSRStorage, [822](#)
 - BasicVoiceAudioWaveformStorage, [821](#)
 - BlendingSoftcopyPresentationStateStorageSOP↔Class, [822](#)
 - BreastImagingRelevantPatientInformationQuery, [823](#)

- BreastTomosynthesisImageStorage, [825](#)
- CTImageStorage, [821](#)
- CardiacElectrophysiologyWaveformStorage, [821](#)
- CardiacRelevantPatientInformationQuery, [824](#)
- ChestCADSRStorage, [823](#)
- ColorSoftcopyPresentationStateStorageSOPClass, [822](#)
- ComprehensiveSRStorage, [822](#)
- ComprehensiveSRStorageTrialRetired, [822](#)
- ComputedRadiographyImageStorage, [821](#)
- DICOMApplicationContextName, [820](#)
- DICOMControlledTerminology, [820](#)
- DICOMUIDRegistry, [820](#)
- DeflatedExplicitVRLittleEndian, [818](#)
- DeformableSpatialRegistrationStorage, [822](#)
- DetachedInterpretationManagementSOPClass↔
Retired, [820](#)
- DetachedPatientManagementMetaSOPClass↔
Retired, [820](#)
- DetachedPatientManagementSOPClassRetired, [820](#)
- DetachedResultsManagementMetaSOPClass↔
Retired, [820](#)
- DetachedResultsManagementSOPClassRetired, [820](#)
- DetachedStudyManagementMetaSOPClassRetired, [820](#)
- DetachedStudyManagementSOPClassRetired, [820](#)
- DetachedVisitManagementSOPClassRetired, [820](#)
- DetailSRStorageTrialRetired, [822](#)
- dicomAETitle, [824](#)
- dicomApplicationCluster, [824](#)
- dicomAssociationAcceptor, [824](#)
- dicomAssociationInitiator, [824](#)
- dicomAuthorizedNodeCertificateReference, [824](#)
- dicomConfigurationRoot, [824](#)
- dicomDescription, [824](#)
- dicomDevice, [824](#)
- dicomDeviceName, [824](#)
- dicomDeviceSerialNumber, [824](#)
- dicomDevicesRoot, [824](#)
- dicomHostname, [824](#)
- dicomInstalled, [824](#)
- dicomInstitutionAddress, [824](#)
- dicomInstitutionDepartmentName, [824](#)
- dicomInstitutionName, [824](#)
- dicomIssuerOfPatientID, [824](#)
- dicomManufacturer, [824](#)
- dicomManufacturerModelName, [824](#)
- dicomNetworkAE, [824](#)
- dicomNetworkConnection, [825](#)
- dicomNetworkConnectionReference, [824](#)
- dicomPort, [824](#)
- dicomPreferredCalledAETitle, [824](#)
- dicomPreferredCallingAETitle, [824](#)
- dicomPrimaryDeviceType, [824](#)
- dicomRelatedDeviceReference, [824](#)
- dicomSOPClass, [824](#)
- dicomSoftwareVersion, [824](#)
- dicomStationName, [824](#)
- dicomSupportedCharacterSet, [824](#)
- dicomTLSCyphersuite, [824](#)
- dicomThisNodeCertificateReference, [824](#)
- dicomTransferCapability, [825](#)
- dicomTransferRole, [824](#)
- dicomTransferSyntax, [824](#)
- dicomUniqueAETitle, [825](#)
- dicomUniqueAETitlesRegistryRoot, [824](#)
- dicomVendorData, [824](#)
- DigitalIntraoralXRayImageStorageForPresentation, [821](#)
- DigitalIntraoralXRayImageStorageForProcessing, [821](#)
- DigitalMammographyXRayImageStorageFor↔
Presentation, [821](#)
- DigitalMammographyXRayImageStorageFor↔
Processing, [821](#)
- DigitalXRayImageStorageForPresentation, [821](#)
- DigitalXRayImageStorageForProcessing, [821](#)
- EncapsulatedCDASStorage, [823](#)
- EncapsulatedPDFStorage, [823](#)
- EnhancedCTImageStorage, [821](#)
- EnhancedMRIImageStorage, [821](#)
- EnhancedSRStorage, [822](#)
- EnhancedUSVolumeStorage, [825](#)
- EnhancedXAImageStorage, [822](#)
- EnhancedXRFImageStorage, [822](#)
- ExplicitVRBigEndian, [818](#)
- ExplicitVRLittleEndian, [818](#)
- GeneralECGWaveformStorage, [821](#)
- GeneralPurposePerformedProcedureStepSOP↔
Class, [823](#)
- GeneralPurposeScheduledProcedureStepSOP↔
Class, [823](#)
- GeneralPurposeWorklistInformationModelFIND, [823](#)
- GeneralPurposeWorklistManagementMetaSOP↔
Class, [823](#)
- GeneralRelevantPatientInformationQuery, [823](#)
- GetName, [831](#)
- GetNumberOfTransferSyntaxStrings, [831](#)
- GetString, [832](#)
- GetTransferSyntaxString, [832](#)
- GetTransferSyntaxStrings, [832](#)
- GetUIDName, [832](#)
- GetUIDString, [832](#)
- GrayscaleSoftcopyPresentationStateStorageSOP↔
Class, [822](#)
- HangingProtocolInformationModelFIND, [824](#)
- HangingProtocolInformationModelMOVE, [824](#)

- HangingProtocolStorage, [824](#)
- HardcopyColorImageStorageSOPClassRetired, [821](#)
- HardcopyGrayscaleImageStorageSOPClassRetired, [821](#)
- HemodynamicWaveformStorage, [821](#)
- ICBM452T1FrameofReference, [820](#)
- ICBMSingleSubjectMRIFrameofReference, [820](#)
- ImageOverlayBoxSOPClassRetired, [821](#)
- ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM, [818](#)
- InstanceAvailabilityNotificationSOPClass, [823](#)
- JPEG2000ImageCompression, [819](#)
- JPEG2000ImageCompressionLosslessOnly, [819](#)
- JPEG2000Part2MulticomponentImageCompression, [819](#)
- JPEG2000Part2MulticomponentImageCompressionLosslessOnly, [819](#)
- JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImageCompression, [818](#)
- JPEGExtendedHierarchicalProcess1618Retired, [819](#)
- JPEGExtendedHierarchicalProcess1719Retired, [819](#)
- JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG12BitImageCompressionProcess4only, [818](#)
- JPEGExtendedProcess35Retired, [818](#)
- JPEGFullProgressionHierarchicalProcess2426Retired, [819](#)
- JPEGFullProgressionHierarchicalProcess2527Retired, [819](#)
- JPEGFullProgressionNonHierarchicalProcess1012Retired, [818](#)
- JPEGFullProgressionNonHierarchicalProcess1113Retired, [818](#)
- JPEGLSLosslessImageCompression, [819](#)
- JPEGLSLossyNearLosslessImageCompression, [819](#)
- JPEGLosslessHierarchicalProcess28Retired, [819](#)
- JPEGLosslessHierarchicalProcess29Retired, [819](#)
- JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLosslessJPEGImageCompression, [819](#)
- JPEGLosslessNonHierarchicalProcess14, [818](#)
- JPEGLosslessNonHierarchicalProcess15Retired, [819](#)
- JPEGSpectralSelectionHierarchicalProcess2022Retired, [819](#)
- JPEGSpectralSelectionHierarchicalProcess2123Retired, [819](#)
- JPEGSpectralSelectionNonHierarchicalProcess68Retired, [818](#)
- JPEGSpectralSelectionNonHierarchicalProcess79Retired, [818](#)
- JPIPReferenced, [819](#)
- JPIPReferencedDeflate, [819](#)
- KeyObjectSelectionDocumentStorage, [823](#)
- MPEG2MainProfileMainLevel, [819](#)
- MRImageStorage, [821](#)
- MRSpectroscopyStorage, [821](#)
- MammographyCADSRStorage, [822](#)
- MediaCreationManagementSOPClassUID, [821](#)
- MediaStorageDirectoryStorage, [819](#)
- ModalityPerformedProcedureStepNotificationSOPClass, [820](#)
- ModalityPerformedProcedureStepRetrieveSOPClass, [820](#)
- ModalityPerformedProcedureStepSOPClass, [820](#)
- ModalityWorklistInformationModelFIND, [823](#)
- MultiframeGrayscaleByteSecondaryCaptureImageStorage, [821](#)
- MultiframeGrayscaleWordSecondaryCaptureImageStorage, [821](#)
- MultiframeSingleBitSecondaryCaptureImageStorage, [821](#)
- MultiframeTrueColorSecondaryCaptureImageStorage, [821](#)
- NuclearMedicineImageStorage, [822](#)
- NuclearMedicineImageStorageRetired, [821](#)
- operator TSType, [832](#)
- OphthalmicPhotography16BitImageStorage, [822](#)
- OphthalmicPhotography8BitImageStorage, [822](#)
- OphthalmicTomographyImageStorage, [822](#)
- PatientRootQueryRetrieveInformationModelFIND, [823](#)
- PatientRootQueryRetrieveInformationModelGET, [823](#)
- PatientRootQueryRetrieveInformationModelMOVE, [823](#)
- PatientStudyOnlyQueryRetrieveInformationModelFINDRetired, [823](#)
- PatientStudyOnlyQueryRetrieveInformationModelGETRetired, [823](#)
- PatientStudyOnlyQueryRetrieveInformationModelMOVERetired, [823](#)
- PositronEmissionTomographyImageStorage, [823](#)
- PresentationLUTSOPClass, [821](#)
- PrintJobSOPClass, [820](#)
- PrintQueueManagementSOPClassRetired, [821](#)
- PrintQueueSOPInstanceRetired, [821](#)
- PrinterConfigurationRetrieveSOPClass, [820](#)
- PrinterConfigurationRetrieveSOPInstance, [820](#)
- PrinterSOPClass, [820](#)
- PrinterSOPInstance, [820](#)
- ProceduralEventLoggingSOPClass, [820](#)
- ProceduralEventLoggingSOPInstance, [820](#)
- ProcedureLogStorage, [822](#)
- ProductCharacteristicsQuerySOPClass, [824](#)
- PseudoColorSoftcopyPresentationStateStorageSOPClass, [822](#)

- PullPrintRequestSOPClassRetired, [821](#)
- PullStoredPrintManagementMetaSOPClassRetired, [821](#)
- RFC2557MIMEencapsulation, [819](#)
- RLELossless, [819](#)
- RTBeamsDeliveryInstructionStorageSupplement74↔
FrozenDraft, [823](#)
- RTBeamsTreatmentRecordStorage, [823](#)
- RTBrachyTreatmentRecordStorage, [823](#)
- RTConventionalMachineVerificationSupplement74↔
FrozenDraft, [823](#)
- RTDoseStorage, [823](#)
- RTImageStorage, [823](#)
- RTIonBeamsTreatmentRecordStorage, [823](#)
- RTIonMachineVerificationSupplement74FrozenDraft, [823](#)
- RTIonPlanStorage, [823](#)
- RTPlanStorage, [823](#)
- RTStructureSetStorage, [823](#)
- RTTreatmentSummaryRecordStorage, [823](#)
- RawDataStorage, [822](#)
- RealWorldValueMappingStorage, [822](#)
- ReferencedColorPrintManagementMetaSOPClass↔
Retired, [820](#)
- ReferencedGrayscalePrintManagementMetaSOP↔
ClassRetired, [820](#)
- ReferencedImageBoxSOPClassRetired, [820](#)
- SPM2AVG152PDFFrameofReference, [819](#)
- SPM2AVG152T1FrameofReference, [819](#)
- SPM2AVG152T2FrameofReference, [819](#)
- SPM2AVG305T1FrameofReference, [819](#)
- SPM2BRAINMASKFrameofReference, [819](#)
- SPM2CSFFFrameofReference, [819](#)
- SPM2EPIFrameofReference, [819](#)
- SPM2FILT1FrameofReference, [819](#)
- SPM2GRAYFrameofReference, [819](#)
- SPM2PDFFrameofReference, [819](#)
- SPM2PETFrameofReference, [819](#)
- SPM2SINGLESUBJT1FrameofReference, [819](#)
- SPM2SPECTFrameofReference, [819](#)
- SPM2T1FrameofReference, [819](#)
- SPM2T2FrameofReference, [819](#)
- SPM2TRANSMFrameofReference, [819](#)
- SPM2WHITEFrameofReference, [819](#)
- SecondaryCaptureImageStorage, [821](#)
- SegmentationStorage, [822](#)
- SetFromUID, [832](#)
- SpatialFiducialsStorage, [822](#)
- SpatialRegistrationStorage, [822](#)
- StandaloneCurveStorageRetired, [821](#)
- StandaloneModalityLUTStorageRetired, [822](#)
- StandaloneOverlayStorageRetired, [821](#)
- StandalonePETCurveStorageRetired, [823](#)
- StandaloneVOILUTStorageRetired, [822](#)
- StereometricRelationshipStorage, [822](#)
- StorageCommitmentPullModelSOPClassRetired, [820](#)
- StorageCommitmentPullModelSOPInstanceRetired, [820](#)
- StorageCommitmentPushModelSOPClass, [820](#)
- StorageCommitmentPushModelSOPInstance, [820](#)
- StorageServiceClass, [820](#)
- StoredPrintStorageSOPClassRetired, [821](#)
- StudyComponentManagementSOPClassRetired, [820](#)
- StudyRootQueryRetrieveInformationModelFIND, [823](#)
- StudyRootQueryRetrieveInformationModelGET, [823](#)
- StudyRootQueryRetrieveInformationModelMOVE, [823](#)
- SubstanceAdministrationLoggingSOPClass, [820](#)
- SubstanceAdministrationLoggingSOPInstance, [820](#)
- SubstanceApprovalQuerySOPClass, [824](#)
- SurfaceSegmentationStorage, [825](#)
- TSName, [818](#)
- TSType, [825](#)
- TalairachBrainAtlasFrameofReference, [819](#)
- TextSRStorageTrialRetired, [822](#)
- TransferSyntaxStringsType, [818](#)
- uid_1_2_840_10008_15_0_3_1, [830](#)
- uid_1_2_840_10008_15_0_3_10, [830](#)
- uid_1_2_840_10008_15_0_3_11, [830](#)
- uid_1_2_840_10008_15_0_3_12, [831](#)
- uid_1_2_840_10008_15_0_3_13, [831](#)
- uid_1_2_840_10008_15_0_3_14, [831](#)
- uid_1_2_840_10008_15_0_3_15, [831](#)
- uid_1_2_840_10008_15_0_3_16, [831](#)
- uid_1_2_840_10008_15_0_3_17, [831](#)
- uid_1_2_840_10008_15_0_3_18, [831](#)
- uid_1_2_840_10008_15_0_3_19, [831](#)
- uid_1_2_840_10008_15_0_3_2, [830](#)
- uid_1_2_840_10008_15_0_3_20, [831](#)
- uid_1_2_840_10008_15_0_3_21, [831](#)
- uid_1_2_840_10008_15_0_3_22, [831](#)
- uid_1_2_840_10008_15_0_3_23, [831](#)
- uid_1_2_840_10008_15_0_3_24, [831](#)
- uid_1_2_840_10008_15_0_3_25, [831](#)
- uid_1_2_840_10008_15_0_3_26, [831](#)
- uid_1_2_840_10008_15_0_3_27, [831](#)
- uid_1_2_840_10008_15_0_3_28, [831](#)
- uid_1_2_840_10008_15_0_3_29, [831](#)
- uid_1_2_840_10008_15_0_3_3, [830](#)
- uid_1_2_840_10008_15_0_3_30, [831](#)
- uid_1_2_840_10008_15_0_3_31, [831](#)
- uid_1_2_840_10008_15_0_3_4, [830](#)
- uid_1_2_840_10008_15_0_3_5, [830](#)
- uid_1_2_840_10008_15_0_3_6, [830](#)
- uid_1_2_840_10008_15_0_3_7, [830](#)
- uid_1_2_840_10008_15_0_3_8, [830](#)

uid_1_2_840_10008_15_0_3_9, 830
uid_1_2_840_10008_15_0_4_1, 831
uid_1_2_840_10008_15_0_4_2, 831
uid_1_2_840_10008_15_0_4_3, 831
uid_1_2_840_10008_15_0_4_4, 831
uid_1_2_840_10008_15_0_4_5, 831
uid_1_2_840_10008_15_0_4_6, 831
uid_1_2_840_10008_15_0_4_7, 831
uid_1_2_840_10008_15_0_4_8, 831
uid_1_2_840_10008_1_1, 825
uid_1_2_840_10008_1_2, 825
uid_1_2_840_10008_1_20_1, 826
uid_1_2_840_10008_1_20_1_1, 826
uid_1_2_840_10008_1_20_2, 826
uid_1_2_840_10008_1_20_2_1, 826
uid_1_2_840_10008_1_2_1, 825
uid_1_2_840_10008_1_2_1_99, 825
uid_1_2_840_10008_1_2_2, 825
uid_1_2_840_10008_1_2_4_100, 826
uid_1_2_840_10008_1_2_4_101, 831
uid_1_2_840_10008_1_2_4_102, 831
uid_1_2_840_10008_1_2_4_103, 831
uid_1_2_840_10008_1_2_4_50, 825
uid_1_2_840_10008_1_2_4_51, 825
uid_1_2_840_10008_1_2_4_52, 825
uid_1_2_840_10008_1_2_4_53, 825
uid_1_2_840_10008_1_2_4_54, 825
uid_1_2_840_10008_1_2_4_55, 825
uid_1_2_840_10008_1_2_4_56, 825
uid_1_2_840_10008_1_2_4_57, 825
uid_1_2_840_10008_1_2_4_58, 825
uid_1_2_840_10008_1_2_4_59, 825
uid_1_2_840_10008_1_2_4_60, 825
uid_1_2_840_10008_1_2_4_61, 825
uid_1_2_840_10008_1_2_4_62, 825
uid_1_2_840_10008_1_2_4_63, 825
uid_1_2_840_10008_1_2_4_64, 825
uid_1_2_840_10008_1_2_4_65, 825
uid_1_2_840_10008_1_2_4_66, 825
uid_1_2_840_10008_1_2_4_70, 825
uid_1_2_840_10008_1_2_4_80, 825
uid_1_2_840_10008_1_2_4_81, 825
uid_1_2_840_10008_1_2_4_90, 825
uid_1_2_840_10008_1_2_4_91, 825
uid_1_2_840_10008_1_2_4_92, 825
uid_1_2_840_10008_1_2_4_93, 825
uid_1_2_840_10008_1_2_4_94, 825
uid_1_2_840_10008_1_2_4_95, 826
uid_1_2_840_10008_1_2_5, 826
uid_1_2_840_10008_1_2_6_1, 826
uid_1_2_840_10008_1_2_6_2, 826
uid_1_2_840_10008_1_3_10, 826
uid_1_2_840_10008_1_40, 826
uid_1_2_840_10008_1_40_1, 826
uid_1_2_840_10008_1_42, 826
uid_1_2_840_10008_1_42_1, 826
uid_1_2_840_10008_1_4_1_1, 826
uid_1_2_840_10008_1_4_1_10, 826
uid_1_2_840_10008_1_4_1_11, 826
uid_1_2_840_10008_1_4_1_12, 826
uid_1_2_840_10008_1_4_1_13, 826
uid_1_2_840_10008_1_4_1_14, 826
uid_1_2_840_10008_1_4_1_15, 826
uid_1_2_840_10008_1_4_1_16, 826
uid_1_2_840_10008_1_4_1_17, 826
uid_1_2_840_10008_1_4_1_18, 826
uid_1_2_840_10008_1_4_1_2, 826
uid_1_2_840_10008_1_4_1_3, 826
uid_1_2_840_10008_1_4_1_4, 826
uid_1_2_840_10008_1_4_1_5, 826
uid_1_2_840_10008_1_4_1_6, 826
uid_1_2_840_10008_1_4_1_7, 826
uid_1_2_840_10008_1_4_1_8, 826
uid_1_2_840_10008_1_4_1_9, 826
uid_1_2_840_10008_1_4_2_1, 826
uid_1_2_840_10008_1_4_2_2, 826
uid_1_2_840_10008_1_9, 826
uid_1_2_840_10008_2_16_4, 826
uid_1_2_840_10008_2_6_1, 826
uid_1_2_840_10008_3_1_1_1, 826
uid_1_2_840_10008_3_1_2_1_1, 826
uid_1_2_840_10008_3_1_2_1_4, 826
uid_1_2_840_10008_3_1_2_2_1, 826
uid_1_2_840_10008_3_1_2_3_1, 826
uid_1_2_840_10008_3_1_2_3_2, 827
uid_1_2_840_10008_3_1_2_3_3, 827
uid_1_2_840_10008_3_1_2_3_4, 827
uid_1_2_840_10008_3_1_2_3_5, 827
uid_1_2_840_10008_3_1_2_5_1, 827
uid_1_2_840_10008_3_1_2_5_4, 827
uid_1_2_840_10008_3_1_2_5_5, 827
uid_1_2_840_10008_3_1_2_6_1, 827
uid_1_2_840_10008_4_2, 827
uid_1_2_840_10008_5_1_1_1, 827
uid_1_2_840_10008_5_1_1_14, 827
uid_1_2_840_10008_5_1_1_15, 827
uid_1_2_840_10008_5_1_1_16, 827
uid_1_2_840_10008_5_1_1_16_376, 827
uid_1_2_840_10008_5_1_1_17, 827
uid_1_2_840_10008_5_1_1_17_376, 827
uid_1_2_840_10008_5_1_1_18, 827
uid_1_2_840_10008_5_1_1_18_1, 827
uid_1_2_840_10008_5_1_1_2, 827
uid_1_2_840_10008_5_1_1_22, 827
uid_1_2_840_10008_5_1_1_23, 827
uid_1_2_840_10008_5_1_1_24, 827
uid_1_2_840_10008_5_1_1_24_1, 827
uid_1_2_840_10008_5_1_1_25, 827

uid_1_2_840_10008_5_1_1_26, [827](#)
 uid_1_2_840_10008_5_1_1_27, [827](#)
 uid_1_2_840_10008_5_1_1_29, [827](#)
 uid_1_2_840_10008_5_1_1_30, [827](#)
 uid_1_2_840_10008_5_1_1_31, [827](#)
 uid_1_2_840_10008_5_1_1_32, [827](#)
 uid_1_2_840_10008_5_1_1_33, [827](#)
 uid_1_2_840_10008_5_1_1_4, [827](#)
 uid_1_2_840_10008_5_1_1_4_1, [827](#)
 uid_1_2_840_10008_5_1_1_4_2, [827](#)
 uid_1_2_840_10008_5_1_1_9, [827](#)
 uid_1_2_840_10008_5_1_1_9_1, [827](#)
 uid_1_2_840_10008_5_1_4_1_1_1, [827](#)
 uid_1_2_840_10008_5_1_4_1_1_10, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_104_1, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_104_2, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_11, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_11_1, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_11_2, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_11_3, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_11_4, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_128, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_129, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_12_1, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_12_1_1, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_12_2, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_12_2_1, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_12_3, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_13_1_1, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_13_1_2, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_13_1_3, [831](#)
 uid_1_2_840_10008_5_1_4_1_1_1_1, [827](#)
 uid_1_2_840_10008_5_1_4_1_1_1_1_1, [827](#)
 uid_1_2_840_10008_5_1_4_1_1_1_2, [827](#)
 uid_1_2_840_10008_5_1_4_1_1_1_2_1, [827](#)
 uid_1_2_840_10008_5_1_4_1_1_1_3, [827](#)
 uid_1_2_840_10008_5_1_4_1_1_1_3_1, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_2, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_20, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_2_1, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_3, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_3_1, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_4, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_481_1, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_481_2, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_481_3, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_481_4, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_481_5, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_481_6, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_481_7, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_481_8, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_481_9, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_4_1, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_4_2, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_5, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_6, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_66, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_66_1, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_66_2, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_66_3, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_66_4, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_66_5, [831](#)
 uid_1_2_840_10008_5_1_4_1_1_67, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_6_1, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_6_2, [831](#)
 uid_1_2_840_10008_5_1_4_1_1_7, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_1, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_1_1, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_2, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_2_1, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_3, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_4, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_4_1, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_5_1, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_5_2, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_5_3, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_5_4, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_6, [831](#)
 uid_1_2_840_10008_5_1_4_1_1_77_2, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_7_1, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_7_2, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_7_3, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_7_4, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_8, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_88_1, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_88_11, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_88_2, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_88_22, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_88_3, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_88_33, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_88_4, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_88_40, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_88_50, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_88_59, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_88_65, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_88_67, [829](#)
 uid_1_2_840_10008_5_1_4_1_1_9, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_9_1, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_9_1_1, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_9_1_2, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_9_1_3, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_9_2_1, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_9_3_1, [828](#)
 uid_1_2_840_10008_5_1_4_1_1_9_4_1, [828](#)
 uid_1_2_840_10008_5_1_4_1_2_1_1, [829](#)
 uid_1_2_840_10008_5_1_4_1_2_1_2, [830](#)
 uid_1_2_840_10008_5_1_4_1_2_1_3, [830](#)

- uid_1_2_840_10008_5_1_4_1_2_2_1, [830](#)
- uid_1_2_840_10008_5_1_4_1_2_2_2, [830](#)
- uid_1_2_840_10008_5_1_4_1_2_2_3, [830](#)
- uid_1_2_840_10008_5_1_4_1_2_3_1, [830](#)
- uid_1_2_840_10008_5_1_4_1_2_3_2, [830](#)
- uid_1_2_840_10008_5_1_4_1_2_3_3, [830](#)
- uid_1_2_840_10008_5_1_4_31, [830](#)
- uid_1_2_840_10008_5_1_4_32, [830](#)
- uid_1_2_840_10008_5_1_4_32_1, [830](#)
- uid_1_2_840_10008_5_1_4_32_2, [830](#)
- uid_1_2_840_10008_5_1_4_32_3, [830](#)
- uid_1_2_840_10008_5_1_4_33, [830](#)
- uid_1_2_840_10008_5_1_4_34_1, [830](#)
- uid_1_2_840_10008_5_1_4_34_2, [830](#)
- uid_1_2_840_10008_5_1_4_34_3, [830](#)
- uid_1_2_840_10008_5_1_4_34_4, [830](#)
- uid_1_2_840_10008_5_1_4_34_4_1, [830](#)
- uid_1_2_840_10008_5_1_4_34_4_2, [830](#)
- uid_1_2_840_10008_5_1_4_34_4_3, [830](#)
- uid_1_2_840_10008_5_1_4_34_4_4, [830](#)
- uid_1_2_840_10008_5_1_4_34_5, [830](#)
- uid_1_2_840_10008_5_1_4_37_1, [830](#)
- uid_1_2_840_10008_5_1_4_37_2, [830](#)
- uid_1_2_840_10008_5_1_4_37_3, [830](#)
- uid_1_2_840_10008_5_1_4_38_1, [830](#)
- uid_1_2_840_10008_5_1_4_38_2, [830](#)
- uid_1_2_840_10008_5_1_4_38_3, [830](#)
- uid_1_2_840_10008_5_1_4_41, [830](#)
- uid_1_2_840_10008_5_1_4_42, [830](#)
- UltrasoundImageStorage, [821](#)
- UltrasoundImageStorageRetired, [821](#)
- UltrasoundMultiframeImageStorage, [821](#)
- UltrasoundMultiframeImageStorageRetired, [821](#)
- UnifiedProcedureStepEventSOPClass, [823](#)
- UnifiedProcedureStepPullSOPClass, [823](#)
- UnifiedProcedureStepPushSOPClass, [823](#)
- UnifiedProcedureStepWatchSOPClass, [823](#)
- UnifiedWorklistandProcedureStepSOPInstance, [823](#)
- UnifiedWorklistandProcedureStepServiceClass, [823](#)
- VLEndoscopicImageStorage, [822](#)
- VLImageStorageTrialRetired, [822](#)
- VLMicroscopicImageStorage, [822](#)
- VLMultiframeImageStorageTrialRetired, [822](#)
- VLPhotographicImageStorage, [822](#)
- VLSlideCoordinatesMicroscopicImageStorage, [822](#)
- VLWholeSlideMicroscopyImageStorage, [825](#)
- VOILUTBoxSOPClass, [821](#)
- VerificationSOPClass, [818](#)
- VideoEndoscopicImageStorage, [822](#)
- VideoMicroscopicImageStorage, [822](#)
- VideoPhotographicImageStorage, [822](#)
- WaveformStorageTrialRetired, [821](#)
- XMLEncoding, [819](#)
- XRay3DAngiographicImageStorage, [822](#)
- XRay3DCraniofacialImageStorage, [822](#)
- XRayAngiographicBiPlaneImageStorageRetired, [822](#)
- XRayAngiographicImageStorage, [822](#)
- XRayRadiationDoseSRStorage, [823](#)
- XRayRadiofluoroscopicImageStorage, [822](#)
- gdcm::UNExplicitDataElement, [877](#)
 - GetLength, [878](#)
 - Read, [878](#)
 - ReadPreValue, [879](#)
 - ReadValue, [879](#)
 - ReadWithLength, [879](#)
- gdcm::UNExplicitImplicitDataElement, [879](#)
 - GetLength, [880](#)
 - Read, [880](#)
 - ReadPreValue, [881](#)
 - ReadValue, [881](#)
- gdcm::UUIDGenerator, [886](#)
 - Generate, [886](#)
 - IsValid, [886](#)
- gdcm::Unpacker12Bits, [881](#)
 - Pack, [881](#)
 - Unpack, [881](#)
- gdcm::Usage, [882](#)
 - Conditional, [883](#)
 - GetUsageString, [883](#)
 - GetUsageType, [883](#)
 - Invalid, [883](#)
 - Mandatory, [883](#)
 - operator UsageType, [883](#)
 - operator<<, [883](#)
 - Usage, [883](#)
 - UsageType, [883](#)
 - UserOption, [883](#)
- gdcm::UserEvent, [883](#)
- gdcm::VL, [892](#)
 - GetLength, [893](#)
 - GetVL16Max, [893](#)
 - GetVL32Max, [893](#)
 - IsOdd, [893](#)
 - IsUndefined, [893](#)
 - operator uint32_t, [893](#)
 - operator<<, [894](#)
 - operator++, [893](#)
 - operator+==, [893](#)
 - Read, [893](#)
 - Read16, [893](#)
 - SetToUndefined, [893](#)
 - Type, [893](#)
 - VL, [893](#)
 - Write, [894](#)
 - Write16, [894](#)
- gdcm::VM, [894](#)
 - Compatible, [897](#)
 - GetIndex, [897](#)

- GetLength, [897](#)
- GetNumberOfElementsFromArray, [897](#)
- GetVMString, [897](#)
- GetVMType, [897](#)
- GetVMTypeFromLength, [898](#)
- IsValid, [898](#)
- operator VMType, [898](#)
- operator<<, [898](#)
- VM, [897](#)
- VM0, [896](#)
- VM1, [896](#)
- VM10, [896](#)
- VM12, [896](#)
- VM16, [896](#)
- VM18, [896](#)
- VM1_2, [897](#)
- VM1_3, [897](#)
- VM1_32, [897](#)
- VM1_4, [897](#)
- VM1_5, [897](#)
- VM1_8, [897](#)
- VM1_99, [897](#)
- VM1_n, [897](#)
- VM2, [896](#)
- VM24, [896](#)
- VM256, [897](#)
- VM28, [896](#)
- VM2_2n, [897](#)
- VM2_n, [897](#)
- VM3, [896](#)
- VM30_30n, [897](#)
- VM32, [896](#)
- VM35, [896](#)
- VM3_3n, [897](#)
- VM3_4, [897](#)
- VM3_n, [897](#)
- VM4, [896](#)
- VM47_47n, [897](#)
- VM4_4n, [897](#)
- VM5, [896](#)
- VM6, [896](#)
- VM6_6n, [897](#)
- VM7_7n, [897](#)
- VM8, [896](#)
- VM9, [896](#)
- VM99, [897](#)
- VM_END, [897](#)
- VMType, [896](#)
- gdcm::VMToLength< T >, [898](#)
- gdcm::VR, [898](#)
 - AE, [900](#)
 - AS, [900](#)
 - AT, [900](#)
 - CS, [900](#)
 - CanDisplay, [902](#)
 - Compatible, [902](#)
 - DA, [900](#)
 - DS, [901](#)
 - DT, [901](#)
 - FD, [901](#)
 - FL, [901](#)
 - GetLength, [902](#)
 - GetSize, [902](#)
 - GetSizeof, [902](#)
 - GetVRString, [902](#)
 - GetVRStringFromFile, [902](#)
 - GetVRType, [902](#)
 - GetVRTypeFromFile, [902](#)
 - INVALID, [900](#)
 - IS, [901](#)
 - IsASCII, [902](#)
 - IsASCII2, [902](#)
 - IsBinary, [902](#)
 - IsBinary2, [902](#)
 - IsDual, [902](#)
 - IsSwap, [902](#)
 - IsVRFile, [902](#)
 - IsValid, [902](#)
 - LO, [901](#)
 - LT, [901](#)
 - OB, [901](#)
 - OB_OW, [901](#)
 - OD, [901](#)
 - OF, [901](#)
 - OW, [901](#)
 - operator VRTType, [902](#)
 - operator<<, [903](#)
 - PN, [901](#)
 - Read, [903](#)
 - SH, [901](#)
 - SL, [901](#)
 - SQ, [901](#)
 - SS, [901](#)
 - ST, [901](#)
 - TM, [901](#)
 - UI, [901](#)
 - UL, [901](#)
 - UN, [901](#)
 - US, [901](#)
 - US_SS, [901](#)
 - US_SS_OW, [901](#)
 - UT, [901](#)
 - VL16, [901](#)
 - VL32, [901](#)
 - VR, [902](#)
 - VR_END, [901](#)
 - VR_VM1, [901](#)
 - VRALL, [901](#)

- VRASCII, [901](#)
- VRBINARY, [901](#)
- VRType, [900](#)
- Write, [903](#)
- gdcmm::VR16ExplicitDataElement, [903](#)
 - GetLength, [904](#)
 - Read, [904](#)
 - ReadPreValue, [905](#)
 - ReadValue, [905](#)
 - ReadWithLength, [905](#)
- gdcmm::VRToEncoding< T >, [905](#)
- gdcmm::VRToType< T >, [905](#)
- gdcmm::VRVLSIZE< 0 >, [906](#)
 - Read, [906](#)
 - Write, [906](#)
- gdcmm::VRVLSIZE< 1 >, [906](#)
 - Read, [906](#)
 - Write, [906](#)
- gdcmm::VRVLSIZE< T >, [906](#)
- gdcmm::Validate, [886](#)
 - ~Validate, [887](#)
 - F, [888](#)
 - GetValidatedFile, [887](#)
 - SetFile, [887](#)
 - V, [888](#)
 - Validate, [887](#)
 - Validation, [888](#)
- gdcmm::Value, [888](#)
 - ~Value, [889](#)
 - Clear, [889](#)
 - DataElement, [890](#)
 - GetLength, [889](#)
 - operator==, [889](#)
 - SetLength, [889](#)
 - SetLengthOnly, [890](#)
 - Value, [889](#)
- gdcmm::ValueIO
 - Read, [890](#)
 - Write, [890](#)
- gdcmm::ValueIO< TDE, TSwap, TType >, [890](#)
- gdcmm::Version, [891](#)
 - ~Version, [891](#)
 - GetBuildVersion, [891](#)
 - GetMajorVersion, [891](#)
 - GetMinorVersion, [891](#)
 - GetVersion, [891](#)
 - operator<<, [892](#)
 - Print, [891](#)
 - Version, [891](#)
- gdcmm::WLMFindQuery, [964](#)
 - GetAbstractSyntaxUID, [966](#)
 - GetTagListByLevel, [966](#)
 - GetValidDataSet, [966](#)
 - InitializeDataSet, [966](#)
 - QueryFactory, [967](#)
 - ValidateQuery, [966](#)
 - WLMFindQuery, [966](#)
- gdcmm::Waveform, [964](#)
 - Waveform, [964](#)
- gdcmm::Writer, [967](#)
 - ~Writer, [970](#)
 - CheckFileMetaInformationOff, [970](#)
 - CheckFileMetaInformationOn, [970](#)
 - GetFile, [970](#)
 - GetStreamPtr, [970](#)
 - Ofstream, [971](#)
 - SetCheckFileMetaInformation, [970](#)
 - SetFile, [970](#)
 - SetFileName, [971](#)
 - SetStream, [971](#)
 - SetWriteDataSetOnly, [971](#)
 - Stream, [971](#)
 - StreamImageWriter, [971](#)
 - Write, [971](#)
 - Writer, [970](#)
- gdcmm::XMLDictReader, [972](#)
 - ~XMLDictReader, [973](#)
 - CharacterDataHandler, [973](#)
 - EndElement, [973](#)
 - GetDict, [973](#)
 - HandleDescription, [973](#)
 - HandleEntry, [973](#)
 - StartElement, [973](#)
 - XMLDictReader, [973](#)
- gdcmm::XMLPrinter, [973](#)
 - ~XMLPrinter, [975](#)
 - F, [975](#)
 - GetPrintStyle, [975](#)
 - HandleBulkData, [975](#)
 - LOADBULKDATA, [975](#)
 - OnlyUUID, [975](#)
 - Print, [975](#)
 - PrintDataElement, [975](#)
 - PrintDataSet, [975](#)
 - PrintSQ, [975](#)
 - PrintStyle, [975](#)
 - PrintStyles, [975](#)
 - SetFile, [975](#)
 - SetStyle, [975](#)
 - XMLPrinter, [975](#)
- gdcmm::XMLPrivateDictReader, [976](#)
 - ~XMLPrivateDictReader, [977](#)
 - CharacterDataHandler, [977](#)
 - EndElement, [977](#)
 - GetPrivateDict, [977](#)
 - HandleDescription, [977](#)
 - HandleEntry, [977](#)
 - StartElement, [977](#)

- XMLPrivateDictReader, 977
- gdcmm::ignore_char, 424
 - ignore_char, 424
 - m_char, 424
- gdcmm::network, 136
 - cMaxEventID, 141
 - cMaxStateID, 141
 - eAABORTPDUReturnedOpen, 140
 - eAABORTRequest, 140
 - eAASSOCIATE_RQPDUreceived, 140
 - eAASSOCIATERequestLocalUser, 140
 - eAASSOCIATEResponseAccept, 140
 - eAASSOCIATEResponseReject, 140
 - eARELEASE_RPPDUReceived, 140
 - eARELEASE_RQPDUReceivedOpen, 140
 - eARELEASERequest, 140
 - eARELEASEResponse, 140
 - eARTIMTimerExpired, 141
 - eASSOCIATE_ACPDUreceived, 140
 - eASSOCIATE_RJPDUreceived, 140
 - eEventDoesNotExist, 141
 - EEventID, 140
 - ePDATATFPDU, 140
 - ePDATArequest, 140
 - eSta10ReleaseCollisionAc, 141
 - eSta11ReleaseCollisionRq, 141
 - eSta12ReleaseCollisionAcLocal, 141
 - eSta13AwaitingClose, 141
 - eSta1Idle, 141
 - eSta2Open, 141
 - eSta3WaitLocalAssoc, 141
 - eSta4LocalAssocDone, 141
 - eSta5WaitRemoteAssoc, 141
 - eSta6TransferReady, 141
 - eSta7WaitRelease, 141
 - eSta8WaitLocalRelease, 141
 - eSta9ReleaseCollisionRqLocal, 141
 - eStaDoesNotExist, 141
 - EStateID, 141
 - eTransportConnConfirmLocal, 140
 - eTransportConnIndicLocal, 140
 - eTransportConnectionClosed, 140
 - eUnrecognizedPDUReturned, 141
 - GetStateIndex, 141
- gdcmm::network::AAbortPDU, 145
 - AAabortPDU, 146
 - IsLastFragment, 146
 - Print, 146
 - Read, 146
 - SetReason, 146
 - SetSource, 147
 - Size, 147
 - Write, 147
- gdcmm::network::AAssociateACPDU, 147
 - AAssociateACPDU, 149
 - AAssociateRQPDU, 149
 - AddPresentationContextAC, 149
 - GetNumberOfPresentationContextAC, 149
 - GetPresentationContextAC, 149
 - GetUserInformation, 149
 - InitFromRQ, 149
 - IsLastFragment, 149
 - Print, 149
 - Read, 149
 - SetCalledAETitle, 149
 - SetCallingAETitle, 149
 - Size, 149
 - SizeType, 149
 - Write, 149
- gdcmm::network::AAssociateRJPDU, 150
 - AAssociateRJPDU, 151
 - IsLastFragment, 151
 - Print, 151
 - Read, 151
 - Size, 151
 - Write, 151
- gdcmm::network::AAssociateRQPDU, 151
 - AAssociateACPDU, 155
 - AAssociateRQPDU, 153
 - AddPresentationContext, 153
 - GetCalledAETitle, 153
 - GetCallingAETitle, 154
 - GetNumberOfPresentationContext, 154
 - GetPresentationContext, 154
 - GetPresentationContextByAbstractSyntax, 154
 - GetPresentationContextByID, 154
 - GetPresentationContexts, 154
 - GetReserved43_74, 154
 - GetUserInformation, 154
 - IsAETitleValid, 154
 - IsLastFragment, 154
 - PresentationContextArrayType, 153
 - Print, 154
 - Read, 154
 - SetCalledAETitle, 154
 - SetCallingAETitle, 154
 - SetUserInformation, 154
 - Size, 155
 - SizeType, 153
 - Write, 155
- gdcmm::network::ARTIMTimer, 172
 - ARTIMTimer, 173
 - GetElapsedTime, 173
 - GetHasExpired, 173
 - GetTimeout, 173
 - SetTimeout, 173
 - Start, 173
 - Stop, 173

- gdcmm::network::AReleaseRPPDU, 169
 - AReleaseRPPDU, 170
 - IsLastFragment, 170
 - Print, 170
 - Read, 170
 - Size, 170
 - Write, 170
- gdcmm::network::AReleaseRQPDU, 170
 - AReleaseRQPDU, 171
 - IsLastFragment, 172
 - Print, 172
 - Read, 172
 - Size, 172
 - Write, 172
- gdcmm::network::AbstractSyntax, 156
 - AbstractSyntax, 157
 - GetAsDataElement, 157
 - GetName, 157
 - operator==, 157
 - Print, 157
 - Read, 157
 - SetName, 157
 - SetNameFromUID, 157
 - Size, 157
 - Write, 157
- gdcmm::network::ApplicationContext, 166
 - ApplicationContext, 166
 - GetName, 167
 - Print, 167
 - Read, 167
 - SetName, 167
 - Size, 167
 - Write, 167
- gdcmm::network::AsynchronousOperationsWindowSub, 174
 - AsynchronousOperationsWindowSub, 174
 - Print, 174
 - Read, 175
 - Size, 175
 - Write, 175
- gdcmm::network::BaseCompositeMessage, 199
 - ~BaseCompositeMessage, 201
 - ConstructPDV, 201
- gdcmm::network::BaseNormalizedMessage, 201
 - ~BaseNormalizedMessage, 203
 - ConstructPDV, 203
- gdcmm::network::BasePDU, 203
 - ~BasePDU, 205
 - IsLastFragment, 205
 - Print, 205
 - Read, 205
 - Size, 205
 - Write, 205
- gdcmm::network::CEchoRQ, 240
 - AffectedSOPClassUID, 241
 - ConstructPDV, 241
 - MessageID, 241
- gdcmm::network::CEchoRSP, 241
 - ConstructPDVByDataSet, 242
- gdcmm::network::CFind, 242
- gdcmm::network::CFindCancelRQ, 243
 - ConstructPDVByDataSet, 243
- gdcmm::network::CFindRQ, 244
 - ConstructPDV, 245
- gdcmm::network::CFindRSP, 245
 - ConstructPDVByDataSet, 246
- gdcmm::network::CMoveCancelRq, 246
 - ConstructPDVByDataSet, 247
- gdcmm::network::CMoveRQ, 248
 - ConstructPDV, 248
- gdcmm::network::CMoveRSP, 249
 - ConstructPDVByDataSet, 250
- gdcmm::network::CStoreRQ, 284
 - ConstructPDV, 285
- gdcmm::network::CStoreRSP, 285
 - ConstructPDV, 286
- gdcmm::network::CompositeMessageFactory, 259
 - ConstructCEchoRQ, 260
 - ConstructCFindRQ, 260
 - ConstructCMoveRQ, 260
 - ConstructCStoreRQ, 260
 - ConstructCStoreRSP, 260
- gdcmm::network::DIMSE, 332
 - C_CANCEL_RQ, 333
 - C_ECHO_RQ, 333
 - C_ECHO_RSP, 333
 - C_FIND_RQ, 332
 - C_FIND_RSP, 332
 - C_GET_RQ, 332
 - C_GET_RSP, 332
 - C_MOVE_RQ, 332
 - C_MOVE_RSP, 333
 - C_STORE_RQ, 332
 - C_STORE_RSP, 332
 - CommandTypes, 332
 - N_ACTION_RQ, 333
 - N_ACTION_RSP, 333
 - N_CREATE_RQ, 333
 - N_CREATE_RSP, 333
 - N_DELETE_RQ, 333
 - N_DELETE_RSP, 333
 - N_EVENT_REPORT_RQ, 333
 - N_EVENT_REPORT_RSP, 333
 - N_GET_RQ, 333
 - N_GET_RSP, 333
 - N_SET_RQ, 333
 - N_SET_RSP, 333
- gdcmm::network::ImplementationClassUIDSub, 466

- ImplementationClassUIDSub, 466
- Print, 466
- Read, 466
- Size, 466
- Write, 466
- gdcmm::network::ImplementationUIDSub, 466
 - ImplementationUIDSub, 467
 - Write, 467
- gdcmm::network::ImplementationVersionNameSub, 467
 - ImplementationVersionNameSub, 467
 - Print, 467
 - Read, 467
 - Size, 467
 - Write, 467
- gdcmm::network::MaximumLengthSub, 516
 - GetMaximumLength, 516
 - MaximumLengthSub, 516
 - Print, 516
 - Read, 516
 - SetMaximumLength, 516
 - Size, 517
 - Write, 517
- gdcmm::network::NActionRQ, 549
 - ConstructPDV, 550
- gdcmm::network::NActionRSP, 551
 - ConstructPDVByDataSet, 552
- gdcmm::network::NCreateRQ, 552
 - ConstructPDV, 553
- gdcmm::network::NCreateRSP, 553
 - ConstructPDVByDataSet, 554
- gdcmm::network::NDeleteRQ, 555
 - ConstructPDV, 555
- gdcmm::network::NDeleteRSP, 556
 - ConstructPDVByDataSet, 557
- gdcmm::network::NEventReportRQ, 559
 - ConstructPDV, 560
- gdcmm::network::NEventReportRSP, 561
 - ConstructPDVByDataSet, 562
- gdcmm::network::NGetRQ, 562
 - ConstructPDV, 563
- gdcmm::network::NGetRSP, 563
 - ConstructPDVByDataSet, 564
- gdcmm::network::NSetRQ, 568
 - ConstructPDV, 569
- gdcmm::network::NSetRSP, 569
 - ConstructPDVByDataSet, 570
- gdcmm::network::NormalizedMessageFactory, 565
 - ConstructNAction, 566
 - ConstructNCreate, 566
 - ConstructNDelete, 566
 - ConstructNEventReport, 566
 - ConstructNGet, 566
 - ConstructNSet, 566
- gdcmm::network::PDUFactory, 601
 - ConstructAbortPDU, 602
 - ConstructPDU, 602
 - ConstructReleasePDU, 602
 - CreateCEchoPDU, 602
 - CreateCFindPDU, 602
 - CreateCMovePDU, 602
 - CreateCStoreRQPDU, 602
 - CreateCStoreRSPPDU, 602
 - CreateNActionPDU, 602
 - CreateNCreatePDU, 602
 - CreateNDeletePDU, 602
 - CreateNEventReportPDU, 602
 - CreateNGetPDU, 602
 - CreateNSetPDU, 602
 - DetermineEventByPDU, 602
 - GetPDVs, 602
- gdcmm::network::PDataTFPDU, 593
 - AddPresentationDataValue, 594
 - GetNumberOfPresentationDataValues, 594
 - GetPresentationDataValue, 594
 - IsLastFragment, 594
 - PDataTFPDU, 594
 - Print, 594
 - Read, 594
 - ReadInto, 594
 - Size, 595
 - SizeType, 594
 - Write, 595
- gdcmm::network::PresentationContextAC, 632
 - GetPresentationContextID, 633
 - GetReason, 633
 - GetTransferSyntax, 633
 - PresentationContextAC, 633
 - Print, 633
 - Read, 633
 - SetPresentationContextID, 633
 - SetReason, 633
 - SetTransferSyntax, 633
 - Size, 633
 - Write, 633
- gdcmm::network::PresentationContextRQ, 635
 - AddTransferSyntax, 636
 - GetAbstractSyntax, 636, 637
 - GetNumberOfTransferSyntaxes, 637
 - GetPresentationContextID, 637
 - GetTransferSyntax, 637
 - GetTransferSyntaxes, 637
 - operator==, 637
 - PresentationContextRQ, 636
 - Print, 637
 - Read, 637
 - SetAbstractSyntax, 637
 - SetPresentationContextID, 637
 - Size, 637

- SizeType, 636
- Write, 637
- gdcmm::network::PresentationDataValue, 637
 - ConcatenatePDVBlobs, 638
 - ConcatenatePDVBlobsAsExplicit, 638
 - GetBlob, 638
 - GetIsCommand, 638
 - GetIsLastFragment, 638
 - GetMessageHeader, 638
 - GetPresentationContextID, 638
 - PresentationDataValue, 638
 - Print, 638
 - Read, 638
 - ReadInto, 638
 - SetBlob, 639
 - SetCommand, 639
 - SetDataSet, 639
 - SetLastFragment, 639
 - SetMessageHeader, 639
 - SetPresentationContextID, 639
 - Size, 639
 - Write, 639
- gdcmm::network::RoleSelectionSub, 681
 - Print, 682
 - Read, 682
 - RoleSelectionSub, 682
 - SetTuple, 682
 - Size, 682
 - Write, 682
- gdcmm::network::SOPClassExtendedNegociationSub, 729
 - Print, 730
 - Read, 730
 - SOPClassExtendedNegociationSub, 730
 - SetTuple, 730
 - Size, 730
 - Write, 730
- gdcmm::network::ServiceClassApplicationInformation, 714
 - Print, 714
 - Read, 714
 - ServiceClassApplicationInformation, 714
 - SetTuple, 714
 - Size, 715
 - Write, 715
- gdcmm::network::TableRow, 786
 - ~TableRow, 787
 - TableRow, 787
 - transitions, 787
- gdcmm::network::TransferSyntaxSub, 806
 - GetName, 807
 - operator==, 807
 - Print, 807
 - Read, 807
 - SetName, 807
 - SetNameFromUID, 807
 - Size, 807
 - TransferSyntaxSub, 807
 - Write, 807
- gdcmm::network::Transition, 807
 - ~Transition, 808
 - mAction, 809
 - mEnd, 809
 - MakeNew, 808
 - Transition, 808
- gdcmm::network::ULAction, 832
 - ~ULAction, 834
 - PerformAction, 834
 - ULAction, 834
- gdcmm::network::ULActionAA1, 835
 - PerformAction, 835
- gdcmm::network::ULActionAA2, 836
 - PerformAction, 836
- gdcmm::network::ULActionAA3, 837
 - PerformAction, 837
- gdcmm::network::ULActionAA4, 838
 - PerformAction, 838
- gdcmm::network::ULActionAA5, 839
 - PerformAction, 839
- gdcmm::network::ULActionAA6, 840
 - PerformAction, 840
- gdcmm::network::ULActionAA7, 841
 - PerformAction, 841
- gdcmm::network::ULActionAA8, 842
 - PerformAction, 842
- gdcmm::network::ULActionAE1, 843
 - PerformAction, 843
- gdcmm::network::ULActionAE2, 844
 - PerformAction, 844
- gdcmm::network::ULActionAE3, 845
 - PerformAction, 845
- gdcmm::network::ULActionAE4, 846
 - PerformAction, 846
- gdcmm::network::ULActionAE5, 847
 - PerformAction, 847
- gdcmm::network::ULActionAE6, 848
 - PerformAction, 848
- gdcmm::network::ULActionAE7, 849
 - PerformAction, 849
- gdcmm::network::ULActionAE8, 850
 - PerformAction, 850
- gdcmm::network::ULActionAR1, 851
 - PerformAction, 851
- gdcmm::network::ULActionAR10, 852
 - PerformAction, 852
- gdcmm::network::ULActionAR2, 853
 - PerformAction, 853
- gdcmm::network::ULActionAR3, 854
 - PerformAction, 854
- gdcmm::network::ULActionAR4, 855
 - PerformAction, 855

- PerformAction, [855](#)
- gdcmm::network::ULActionAR5, [856](#)
 - PerformAction, [856](#)
- gdcmm::network::ULActionAR6, [857](#)
 - PerformAction, [857](#)
- gdcmm::network::ULActionAR7, [858](#)
 - PerformAction, [858](#)
- gdcmm::network::ULActionAR8, [859](#)
 - PerformAction, [859](#)
- gdcmm::network::ULActionAR9, [860](#)
 - PerformAction, [860](#)
- gdcmm::network::ULActionDT1, [861](#)
 - PerformAction, [861](#)
- gdcmm::network::ULActionDT2, [862](#)
 - PerformAction, [862](#)
- gdcmm::network::ULBasicCallback, [863](#)
 - ~ULBasicCallback, [864](#)
 - GetDataSets, [864](#)
 - GetResponses, [864](#)
 - HandleDataSet, [864](#)
 - HandleResponse, [864](#)
 - ULBasicCallback, [864](#)
- gdcmm::network::ULConnection, [864](#)
 - ~ULConnection, [865](#)
 - AddAcceptedPresentationContext, [865](#)
 - FindContext, [865](#)
 - GetAcceptedPresentationContexts, [866](#)
 - GetConnectionInfo, [866](#)
 - GetMaxPDUSize, [866](#)
 - GetPresentationContextACByID, [866](#)
 - GetPresentationContextIDFromPresentationContext, [866](#)
 - GetPresentationContextRQByID, [866](#)
 - GetPresentationContexts, [866](#)
 - GetProtocol, [866](#)
 - GetState, [866](#)
 - GetTimer, [866](#)
 - InitializeConnection, [866](#)
 - InitializeIncomingConnection, [866](#)
 - SetMaxPDUSize, [866](#)
 - SetPresentationContexts, [866](#)
 - SetState, [866](#)
 - StopProtocol, [866](#)
 - ULActionAE6, [866](#)
 - ULConnection, [865](#)
 - ULConnectionManager, [867](#)
- gdcmm::network::ULConnectionCallback, [867](#)
 - ~ULConnectionCallback, [868](#)
 - DataSetHandled, [868](#)
 - DataSetHandles, [868](#)
 - HandleDataSet, [868](#)
 - HandleResponse, [868](#)
 - mImplicit, [868](#)
 - ResetHandledDataSet, [868](#)
 - SetImplicitFlag, [868](#)
 - ULConnectionCallback, [868](#)
- gdcmm::network::ULConnectionInfo, [868](#)
 - GetCalledAETitle, [869](#)
 - GetCalledComputerName, [869](#)
 - GetCalledIPAddress, [869](#)
 - GetCalledIPPort, [869](#)
 - GetCallingAETitle, [869](#)
 - GetMaxPDULength, [869](#)
 - Initialize, [869](#)
 - SetMaxPDULength, [869](#)
 - ULConnectionInfo, [869](#)
- gdcmm::network::ULConnectionManager, [870](#)
 - ~ULConnectionManager, [872](#)
 - BreakConnection, [872](#)
 - BreakConnectionNow, [872](#)
 - EstablishConnection, [872](#)
 - EstablishConnectionMove, [872](#)
 - mConnection, [873](#)
 - mSecondaryConnection, [873](#)
 - mTransitions, [873](#)
 - RunEventLoop, [872](#)
 - RunMoveEventLoop, [872](#)
 - SendEcho, [872](#)
 - SendFind, [872](#)
 - SendMove, [872](#)
 - SendNAction, [872](#), [873](#)
 - SendNCreate, [873](#)
 - SendNDelete, [873](#)
 - SendNEventReport, [873](#)
 - SendNGet, [873](#)
 - SendNSet, [873](#)
 - SendStore, [873](#)
 - ULConnectionManager, [872](#)
- gdcmm::network::ULEvent, [874](#)
 - ~ULEvent, [874](#)
 - GetDataSetPos, [874](#)
 - GetEvent, [874](#)
 - GetIStream, [874](#)
 - GetPDUs, [874](#)
 - SetEvent, [874](#)
 - SetPDU, [874](#)
 - ULEvent, [874](#)
- gdcmm::network::ULTransitionTable, [875](#)
 - HandleEvent, [875](#)
 - PrintTable, [875](#)
 - ULTransitionTable, [875](#)
- gdcmm::network::ULWritingCallback, [875](#)
 - ~ULWritingCallback, [877](#)
 - HandleDataSet, [877](#)
 - HandleResponse, [877](#)
 - SetDirectory, [877](#)
 - ULWritingCallback, [876](#)
- gdcmm::network::UserInformation, [885](#)

- ~UserInformation, [885](#)
- AddRoleSelectionSub, [885](#)
- AddSOPClassExtendedNegociationSub, [885](#)
- GetMaximumLengthSub, [885](#)
- operator=, [885](#)
- Print, [885](#)
- Read, [885](#)
- Size, [885](#)
- UserInformation, [885](#)
- Write, [886](#)
- gdcmm::static_assert_test< x >, [740](#)
- gdcmm::terminal, [142](#)
 - Attribute, [143](#)
 - black, [143](#)
 - blink, [143](#)
 - blue, [143](#)
 - bright, [143](#)
 - CONSOLE, [143](#)
 - Color, [143](#)
 - cyan, [143](#)
 - dim, [143](#)
 - green, [143](#)
 - hidden, [143](#)
 - magenta, [143](#)
 - Mode, [143](#)
 - red, [143](#)
 - reset, [143](#)
 - reverse, [143](#)
 - setattrtribute, [143](#)
 - setbgcolor, [143](#)
 - setfgcolor, [143](#)
 - setmode, [143](#)
 - underline, [143](#)
 - VT100, [143](#)
 - white, [143](#)
 - yellow, [143](#)
- gdcmmAAabortPDU.h, [979](#)
- gdcmmAAAssociateACPDU.h, [980](#)
- gdcmmAAAssociateRJPDU.h, [980](#)
- gdcmmAAAssociateRQPDU.h, [981](#)
- gdcmmARTIMTimer.h, [989](#)
- gdcmmAReleaseRPPDU.h, [987](#)
- gdcmmAReleaseRQPDU.h, [988](#)
- gdcmmASN1.h, [990](#)
- gdcmmAbstractSyntax.h, [982](#)
- gdcmmAnonymizeEvent.h, [983](#)
- gdcmmAnonymizer.h, [984](#)
- gdcmmApplicationContext.h, [985](#)
- gdcmmApplicationEntity.h, [986](#)
- gdcmmAssertAlwaysMacro
 - gdcmmTrace.h, [1221](#)
- gdcmmAssertMacro
 - gdcmmTrace.h, [1221](#)
- gdcmmAsynchronousOperationsWindowSub.h, [990](#)
- gdcmmAttribute.h, [991](#)
- gdcmmAudioCodec.h, [993](#)
- gdcmmBase64.h, [993](#)
- gdcmmBaseCompositeMessage.h, [994](#)
- gdcmmBaseNormalizedMessage.h, [995](#)
- gdcmmBasePDU.h, [996](#)
- gdcmmBaseQuery.h, [997](#)
- gdcmmBaseRootQuery.h, [998](#)
- gdcmmBasicOffsetTable.h, [1000](#)
- gdcmmBitmap.h, [1001](#)
- gdcmmBitmapToBitmapFilter.h, [1002](#)
- gdcmmBoxRegion.h, [1003](#)
- gdcmmByteBuffer.h, [1003](#)
- gdcmmByteSwap.h, [1004](#)
- gdcmmByteSwapFilter.h, [1005](#)
- gdcmmByteValue.h, [1006](#)
- gdcmmCAPICryptoFactory.h, [1007](#)
- gdcmmCAPICryptographicMessageSyntax.h, [1007](#)
- gdcmmCEchoMessages.h, [1008](#)
- gdcmmCFindMessages.h, [1009](#)
- gdcmmCMoveMessages.h, [1010](#)
- gdcmmCP246ExplicitDataElement.h, [1019](#)
- gdcmmCSAElement.h, [1021](#)
- gdcmmCSAHeader.h, [1023](#)
- gdcmmCSAHeaderDict.h, [1023](#)
- gdcmmCSAHeaderDictEntry.h, [1025](#)
- gdcmmCStoreMessages.h, [1026](#)
- gdcmmCodeString.h, [1013](#)
- gdcmmCodec.h, [1011](#)
- gdcmmCoder.h, [1012](#)
- gdcmmCommand.h, [1014](#)
- gdcmmCommandDataSet.h, [1016](#)
- gdcmmCompositeMessageFactory.h, [1016](#)
- gdcmmCompositeNetworkFunctions.h, [1017](#)
- gdcmmConstCharWrapper.h, [1018](#)
- gdcmmCryptoFactory.h, [1019](#)
- gdcmmCryptographicMessageSyntax.h, [1020](#)
- gdcmmCurve.h, [1027](#)
- gdcmmDICOMDIR.h, [1037](#)
- gdcmmDICOMDIRGenerator.h, [1038](#)
- gdcmmDIMSE.h, [1045](#)
- gdcmmDataElement.h, [1028](#)
- gdcmmDataEvent.h, [1029](#)
- gdcmmDataSet.h, [1030](#)
- gdcmmDataSetEvent.h, [1031](#)
- gdcmmDataSetHelper.h, [1032](#)
- gdcmmDebugMacro
 - gdcmmTrace.h, [1222](#)
- gdcmmDecoder.h, [1033](#)
- gdcmmDefinedTerms.h, [1034](#)
- gdcmmDeflateStream.h, [1035](#)
- gdcmmDefs.h, [1035](#)
- gdcmmDeltaEncodingCodec.h, [1037](#)
- gdcmmDict.h, [1039](#)

- gdcmDictConverter.h, 1041
- gdcmDictEntry.h, 1041
- gdcmDictPrinter.h, 1043
- gdcmDicts.h, 1043
- gdcmDirectionCosines.h, 1045
- gdcmDirectory.h, 1046
- gdcmDirectoryHelper.h, 1047
- gdcmDummyValueGenerator.h, 1048
- gdcmDumper.h, 1048
- gdcmElement.h, 1049
 - VRDS16ILLEGAL, 1051
- gdcmEncapsulatedDocument.h, 1051
- gdcmEnumeratedValues.h, 1052
- gdcmErrorMacro
 - gdcmTrace.h, 1222
- gdcmEvent.h, 1052
 - gdcmEventMacro, 1054
- gdcmEventMacro
 - gdcmEvent.h, 1054
- gdcmException.h, 1054
- gdcmExplicitDataElement.h, 1055
- gdcmExplicitImplicitDataElement.h, 1056
- gdcmFiducials.h, 1056
- gdcmFile.h, 1057
- gdcmFileAnonymizer.h, 1058
- gdcmFileChangeTransferSyntax.h, 1059
- gdcmFileDecompressLookupTable.h, 1059
- gdcmFileDerivation.h, 1060
- gdcmFileExplicitFilter.h, 1061
- gdcmFileMetaInformation.h, 1061
- gdcmFileNameEvent.h, 1063
- gdcmFileSet.h, 1065
- gdcmFileStreamer.h, 1066
- gdcmFilename.h, 1062
- gdcmFilenameGenerator.h, 1064
- gdcmFindPatientRootQuery.h, 1067
- gdcmFindStudyRootQuery.h, 1068
- gdcmFragment.h, 1069
- gdcmGlobal.h, 1071
- gdcmGroupDict.h, 1072
- gdcmIOD.h, 1090
- gdcmIODEntry.h, 1091
- gdcmIODs.h, 1093
- gdcmIPPSorter.h, 1095
- gdcmIconImage.h, 1072
- gdcmIconImageFilter.h, 1073
- gdcmIconImageGenerator.h, 1074
- gdcmImage.h, 1075
- gdcmImageApplyLookupTable.h, 1076
- gdcmImageChangePhotometricInterpretation.h, 1077
- gdcmImageChangePlanarConfiguration.h, 1077
- gdcmImageChangeTransferSyntax.h, 1078
- gdcmImageCodec.h, 1079
- gdcmImageConverter.h, 1080
- gdcmImageFragmentSplitter.h, 1081
- gdcmImageHelper.h, 1082
- gdcmImageReader.h, 1083
- gdcmImageRegionReader.h, 1083
- gdcmImageToImageFilter.h, 1084
- gdcmImageWriter.h, 1085
- gdcmImplementationClassUIDSub.h, 1086
- gdcmImplementationUIDSub.h, 1087
- gdcmImplementationVersionNameSub.h, 1088
- gdcmImplicitDataElement.h, 1089
- gdcmItem.h, 1096
- gdcmJPEG12Codec.h, 1097
- gdcmJPEG16Codec.h, 1097
- gdcmJPEG2000Codec.h, 1098
- gdcmJPEG8Codec.h, 1099
- gdcmJPEGCodec.h, 1100
- gdcmJPEGLSCodec.h, 1101
- gdcmJSON.h, 1102
- gdcmKAKADUCodec.h, 1103
- gdcmLO.h, 1105
- gdcmLegacyMacro.h, 1104
 - GDCM_LEGACY, 1104
 - GDCM_LEGACY_BODY, 1104
 - GDCM_LEGACY_REPLACED_BODY, 1105
- gdcmLookupTable.h, 1105
- gdcmMD5.h, 1112
- gdcmMacro.h, 1106
- gdcmMacroEntry.h, 1108
 - GDCMMACROENTRY_H, 1109
- gdcmMacros.h, 1109
- gdcmMaximumLengthSub.h, 1111
- gdcmMediaStorage.h, 1113
- gdcmMeshPrimitive.h, 1114
- gdcmModalityPerformedProcedureStepCreateQuery.h, 1115
- gdcmModalityPerformedProcedureStepSetQuery.h, 1116
- gdcmModule.h, 1117
- gdcmModuleEntry.h, 1118
- gdcmModules.h, 1120
- gdcmMovePatientRootQuery.h, 1121
- gdcmMoveStudyRootQuery.h, 1122
- gdcmNActionMessages.h, 1123
- gdcmNCreateMessages.h, 1124
- gdcmNDeleteMessages.h, 1125
- gdcmNEventReportMessages.h, 1129
- gdcmNGetMessages.h, 1129
- gdcmNSetMessages.h, 1132
- gdcmNestedModuleEntries.h, 1125
- gdcmNetworkEvents.h, 1127
- gdcmNetworkStateID.h, 1128
- gdcmNormalizedMessageFactory.h, 1130
- gdcmNormalizedNetworkFunctions.h, 1131
- gdcmObject.h, 1132
- gdcmOpenSSLCryptoFactory.h, 1133

gdcmOpenSSLCryptographicMessageSyntax.h, 1134
gdcmOpenSSL7CryptoFactory.h, 1135
gdcmOpenSSL7CryptographicMessageSyntax.h, 1136
gdcmOrientation.h, 1138
gdcmOverlay.h, 1138
gdcmPDBElement.h, 1143
gdcmPDBHeader.h, 1145
gdcmPDFCodec.h, 1145
gdcmPDUFactory.h, 1146
gdcmPDataTFPDU.h, 1142
gdcmPGXCodec.h, 1147
gdcmPNMCodec.h, 1154
gdcmPVRGCodec.h, 1164
gdcmParseException.h, 1139
gdcmParser.h, 1141
gdcmPatient.h, 1141
gdcmPersonName.h, 1147
gdcmPhotometricInterpretation.h, 1148
gdcmPixelFormat.h, 1149
gdcmPixmap.h, 1150
gdcmPixmapReader.h, 1151
gdcmPixmapToPixmapFilter.h, 1152
gdcmPixmapWriter.h, 1153
gdcmPreamble.h, 1155
gdcmPresentationContext.h, 1156
gdcmPresentationContextAC.h, 1157
gdcmPresentationContextGenerator.h, 1159
gdcmPresentationContextRQ.h, 1159
gdcmPresentationDataValue.h, 1160
gdcmPrinter.h, 1161
gdcmPrivateTag.h, 1162
gdcmProgressEvent.h, 1164
gdcmPythonFilter.h, 1165
gdcmQueryBase.h, 1166
gdcmQueryFactory.h, 1167
gdcmQueryImage.h, 1168
gdcmQueryPatient.h, 1169
gdcmQuerySeries.h, 1170
gdcmQueryStudy.h, 1170
gdcmRAWCodec.h, 1171
gdcmRLECodec.h, 1175
gdcmReader.h, 1172
gdcmRegion.h, 1174
gdcmRescaler.h, 1175
gdcmRoleSelectionSub.h, 1176
gdcmSHA1.h, 1189
gdcmSOPClassExtendedNegociationSub.h, 1192
gdcmSOPClassUIDToIOD.h, 1193
gdcmScanner.h, 1177
gdcmSegment.h, 1178
gdcmSegmentHelper.h, 1180
gdcmSegmentReader.h, 1182
gdcmSegmentWriter.h, 1183
gdcmSegmentedPaletteColorLookupTable.h, 1179
gdcmSequenceOfFragments.h, 1184
gdcmSequenceOfItems.h, 1185
gdcmSerieHelper.h, 1185
gdcmSeries.h, 1187
gdcmServiceClassApplicationInformation.h, 1188
gdcmServiceClassUser.h, 1189
gdcmSimpleSubjectWatcher.h, 1190
gdcmSmartPointer.h, 1191
gdcmSorter.h, 1194
gdcmSpacing.h, 1196
gdcmSpectroscopy.h, 1196
gdcmSplitMosaicFilter.h, 1197
gdcmStaticAssert.h, 1198
 GDCM_DO_JOIN, 1198
 GDCM_DO_JOIN2, 1198
 GDCM_JOIN, 1198
 GDCM_STATIC_ASSERT, 1198
gdcmStreamImageReader.h, 1199
gdcmStreamImageWriter.h, 1199
gdcmStrictScanner.h, 1200
gdcmString.h, 1201
gdcmStringFilter.h, 1202
gdcmStudy.h, 1203
gdcmSubject.h, 1204
gdcmSurface.h, 1205
gdcmSurfaceHelper.h, 1206
gdcmSurfaceReader.h, 1207
gdcmSurfaceWriter.h, 1208
gdcmSwapCode.h, 1208
gdcmSwapper.h, 1209
gdcmSystem.h, 1210
gdcmTable.h, 1211
gdcmTableEntry.h, 1212
gdcmTableReader.h, 1213
gdcmTag.h, 1215
gdcmTagPath.h, 1215
gdcmTagToVR.h, 1216
gdcmTerminal.h, 1217
gdcmTestDriver.h, 1219
gdcmTesting.h, 1219
gdcmTrace.h, 1220
 GDCM_FUNCTION, 1221
 gdcmAssertAlwaysMacro, 1221
 gdcmAssertMacro, 1221
 gdcmDebugMacro, 1222
 gdcmErrorMacro, 1222
 gdcmWarningMacro, 1222
gdcmTransferSyntax.h, 1223
gdcmTransferSyntaxSub.h, 1224
gdcmType.h, 1225
gdcmTypes.h, 1226
gdcmUIDGenerator.h, 1227
gdcmUIDs.h, 1228
gdcmULAction.h, 1228

- gdcmULActionAA.h, [1229](#)
- gdcmULActionAE.h, [1230](#)
- gdcmULActionAR.h, [1231](#)
- gdcmULActionDT.h, [1232](#)
- gdcmULBasicCallback.h, [1232](#)
- gdcmULConnection.h, [1233](#)
- gdcmULConnectionCallback.h, [1234](#)
- gdcmULConnectionInfo.h, [1235](#)
- gdcmULConnectionManager.h, [1237](#)
- gdcmULEvent.h, [1237](#)
- gdcmULTransitionTable.h, [1238](#)
- gdcmULWritingCallback.h, [1240](#)
- gdcmUNExplicitDataElement.h, [1240](#)
- gdcmUNExplicitImplicitDataElement.h, [1241](#)
- gdcmUUIDGenerator.h, [1245](#)
- gdcmUnpacker12Bits.h, [1241](#)
- gdcmUsage.h, [1242](#)
- gdcmUserInformation.h, [1244](#)
- gdcmVL.h, [1249](#)
- gdcmVM.h, [1250](#)
 - TYPETOLENGTH, [1251](#)
- gdcmVR.h, [1251](#)
 - TYPETOENCODING, [1253](#)
 - VRTypeTemplateCase, [1253](#)
- gdcmVR16ExplicitDataElement.h, [1253](#)
- gdcmValidate.h, [1245](#)
- gdcmValue.h, [1246](#)
- gdcmValueIO.h, [1247](#)
- gdcmVersion.h, [1248](#)
- gdcmWLMFindQuery.h, [1255](#)
- gdcmWarningMacro
 - gdcmTrace.h, [1222](#)
- gdcmWaveform.h, [1254](#)
- gdcmWin32.h, [1254](#)
 - GDCM_EXPORT, [1254](#)
- gdcmWriter.h, [1255](#)
- gdcmXMLDictReader.h, [1257](#)
- gdcmXMLPrinter.h, [1257](#)
- gdcmXMLPrivateDictReader.h, [1258](#)
- gdcmanon.dox, [983](#)
- gdcmconv.dox, [1019](#)
- gdcmdiff.dox, [1044](#)
- gdcmdump.dox, [1048](#)
- gdcmgendir.dox, [1071](#)
- gdcmimg.dox, [1086](#)
- gdcminfo.dox, [1089](#)
- gdcmpap3.dox, [1139](#)
- gdcmpdf.dox, [1145](#)
- gdcmraw.dox, [1171](#)
- gdcmscanner.dox, [1177](#)
- gdcm SCU.dox, [1178](#)
- gdcm tar.dox, [1217](#)
- gdcmviewer.dox, [1249](#)
- gdcmxml.dox, [1256](#)
- GeneralECGWaveformStorage
 - gdcm::MediaStorage, [522](#)
 - gdcm::UIDs, [821](#)
- GeneralElectricMagneticResonanceImageStorage
 - gdcm::MediaStorage, [522](#)
- GeneralPurposePerformedProcedureStepSOPClass
 - gdcm::UIDs, [823](#)
- GeneralPurposeScheduledProcedureStepSOPClass
 - gdcm::UIDs, [823](#)
- GeneralPurposeWorklistInformationModelFIND
 - gdcm::UIDs, [823](#)
- GeneralPurposeWorklistManagementMetaSOPClass
 - gdcm::UIDs, [823](#)
- GeneralRelevantPatientInformationQuery
 - gdcm::UIDs, [823](#)
- Generate
 - gdcm::DICOmdirGenerator, [319](#)
 - gdcm::DummyValueGenerator, [339](#)
 - gdcm::FilenameGenerator, [401](#)
 - gdcm::IconImageGenerator, [423](#)
 - gdcm::UIDGenerator, [812](#)
 - gdcm::UUIDGenerator, [886](#)
- GenerateFromFilenames
 - gdcm::PresentationContextGenerator, [635](#)
- GenerateFromUID
 - gdcm::PresentationContextGenerator, [635](#)
- GenerateUUID
 - gdcm::UIDGenerator, [812](#)
- Get
 - gdcm::ByteBuffer, [229](#)
- GetAETitle
 - gdcm::ServiceClassUser, [717](#)
- GetALGOType
 - gdcm::Segment, [691](#)
- GetALGOTypeString
 - gdcm::Segment, [691](#)
- GetAbbreviation
 - gdcm::GroupDict, [419](#)
- GetAbstractSyntax
 - gdcm::PresentationContext, [631](#)
 - gdcm::network::PresentationContextRQ, [636](#), [637](#)
- GetAbstractSyntaxUID
 - gdcm::BaseQuery, [207](#)
 - gdcm::FindPatientRootQuery, [410](#)
 - gdcm::FindStudyRootQuery, [412](#)
 - gdcm::ModalityPerformedProcedureStepCreate↔Query, [535](#)
 - gdcm::ModalityPerformedProcedureStepSetQuery, [537](#)
 - gdcm::MovePatientRootQuery, [546](#)
 - gdcm::MoveStudyRootQuery, [548](#)
 - gdcm::WLMFindQuery, [966](#)
- GetAcceptedPresentationContexts
 - gdcm::network::ULConnection, [866](#)

- GetAlgorithmFamily
 - gdcm::Surface, [766](#)
- GetAlgorithmName
 - gdcm::Surface, [766](#)
- GetAlgorithmVersion
 - gdcm::Surface, [766](#)
- GetAllFilenamesFromTagToValue
 - gdcm::Scanner, [687](#)
 - gdcm::StrictScanner, [752](#)
- GetAllRequiredTags
 - gdcm::QueryBase, [653](#)
- GetAllTags
 - gdcm::QueryBase, [653](#)
- GetAnatomicRegion
 - gdcm::Segment, [691](#)
- GetAsDataElement
 - gdcm::Attribute, [177](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [182](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, [188](#)
 - gdcm::Element, [344](#)
 - gdcm::Element< TVR, VM::VM1_n >, [347](#)
 - gdcm::PrivateTag, [646](#)
 - gdcm::network::AbstractSyntax, [157](#)
- GetAsPoints
 - gdcm::Curve, [288](#)
- GetAsString
 - gdcm::CodeString, [255](#)
- GetAxisOfRotation
 - gdcm::Surface, [766](#)
- GetBasicApplicationLevelConfidentialityProfileAttributes
 - gdcm::Anonymizer, [163](#)
- GetBitPosition
 - gdcm::Overlay, [586](#)
- GetBitSample
 - gdcm::LookupTable, [511](#)
- GetBitsAllocated
 - gdcm::Overlay, [586](#)
 - gdcm::PixelFormat, [611](#)
- GetBitsStored
 - gdcm::PixelFormat, [611](#)
- GetBlob
 - gdcm::network::PresentationDataValue, [638](#)
- GetBuffer
 - gdcm::Bitmap, [219](#)
 - gdcm::ByteValue, [234](#)
 - gdcm::Parser, [592](#)
 - gdcm::SequenceOfFragments, [703](#)
- GetBuffer2
 - gdcm::Bitmap, [219](#)
- GetBufferAsRGBA
 - gdcm::LookupTable, [511](#)
- GetBufferLength
 - gdcm::Bitmap, [219](#)
 - gdcm::JPEGLSCodec, [501](#)
 - gdcm::PNMCodec, [627](#)
 - gdcm::RLECodec, [680](#)
- GetBuildVersion
 - gdcm::Version, [891](#)
- GetByteValue
 - gdcm::CSAElement, [273](#)
 - gdcm::DataElement, [293](#)
- GetCSADataInfo
 - gdcm::CSAHeader, [278](#)
- GetCSAEEnd
 - gdcm::CSAHeader, [278](#)
- GetCSAElementByName
 - gdcm::CSAHeader, [278](#)
- GetCSAHeaderDict
 - gdcm::Dicts, [331](#)
- GetCSAHeaderDictEntry
 - gdcm::CSAHeaderDict, [281](#)
- GetCSAImageHeaderInfoTag
 - gdcm::CSAHeader, [278](#)
- GetCSASeriesHeaderInfoTag
 - gdcm::CSAHeader, [278](#)
- GetCTImageSeriesUIDs
 - gdcm::DirectoryHelper, [338](#)
- GetCWD
 - gdcm::System, [781](#)
- GetCalledAETitle
 - gdcm::ServiceClassUser, [717](#)
 - gdcm::network::AAssociateRQPDU, [153](#)
 - gdcm::network::ULConnectionInfo, [869](#)
- GetCalledComputerName
 - gdcm::network::ULConnectionInfo, [869](#)
- GetCalledIPAddress
 - gdcm::network::ULConnectionInfo, [869](#)
- GetCalledIPPort
 - gdcm::network::ULConnectionInfo, [869](#)
- GetCallingAETitle
 - gdcm::network::AAssociateRQPDU, [154](#)
 - gdcm::network::ULConnectionInfo, [869](#)
- GetCenterOfRotation
 - gdcm::Surface, [766](#)
- GetCharacterFromCurrentLocale
 - gdcm::QueryFactory, [655](#)
- GetCipherType
 - gdcm::CAPICryptographicMessageSyntax, [239](#)
 - gdcm::CryptographicMessageSyntax, [270](#)
 - gdcm::OpenSSLCryptographicMessageSyntax, [576](#)
 - gdcm::OpenSSL7CryptographicMessageSyntax, [580](#)
- GetCodec
 - gdcm::FileChangeTransferSyntax, [382](#)
- GetColorLevel
 - vtkImageColorViewer, [943](#)

- GetColorWindow
 - vtkImageColorViewer, [943](#)
- GetColumns
 - gdcm::Bitmap, [220](#)
 - gdcm::Overlay, [586](#)
- GetCommand
 - gdcm::Subject, [762](#)
- GetConnectionInfo
 - gdcm::network::ULConnection, [866](#)
- GetConstructorString
 - gdcm::Dicts, [330](#)
- GetContourReferencedFrameOfReferenceClassUID
 - vtkRTStructSetProperties, [962](#)
- GetContourReferencedFrameOfReferenceInstanceUID
 - vtkRTStructSetProperties, [962](#)
- GetCryptographicMessageSyntax
 - gdcm::Anonymizer, [163](#)
- GetCurrentByteIndex
 - gdcm::Parser, [592](#)
- GetCurrentDateTime
 - gdcm::System, [780](#)
- GetCurrentModuleFileName
 - gdcm::System, [780](#)
- GetCurrentProcessFileName
 - gdcm::System, [780](#)
- GetCurrentResourcesDirectory
 - gdcm::System, [780](#)
- GetCurve
 - gdcm::Pixmap, [616](#)
- GetCurveDataDescriptor
 - gdcm::Curve, [288](#)
- GetDEEnd
 - gdcm::DataSet, [306](#)
- GetDES
 - gdcm::DataSet, [306](#)
- GetData
 - gdcm::DataEvent, [301](#)
- GetDataElement
 - gdcm::Bitmap, [220](#)
 - gdcm::DataSet, [305](#), [306](#)
 - gdcm::Item, [481](#)
- GetDataExtraRoot
 - gdcm::Testing, [797](#)
- GetDataLength
 - gdcm::DataEvent, [301](#)
- GetDataRoot
 - gdcm::Testing, [797](#)
- GetDataSet
 - gdcm::CSAHeader, [278](#)
 - gdcm::DataSetEvent, [310](#)
 - gdcm::File, [376](#)
- GetDataSetPos
 - gdcm::network::ULEvent, [874](#)
- GetDataSetTransferSyntax
 - gdcm::FileMetaInformation, [392](#)
- GetDataSets
 - gdcm::network::ULBasicCallback, [864](#)
- GetDataValueRepresentation
 - gdcm::Curve, [289](#)
- GetDebugFlag
 - gdcm::Trace, [801](#)
- GetDebugStream
 - gdcm::Trace, [801](#)
- GetDecodeLength
 - gdcm::Base64, [199](#)
- GetDefaultTransferSyntax
 - gdcm::PresentationContextGenerator, [635](#)
- GetDefs
 - gdcm::Global, [417](#)
 - gdcm::TableReader, [785](#)
- GetDescription
 - gdcm::CSAHeaderDictEntry, [282](#)
 - gdcm::Exception, [368](#)
 - gdcm::ModuleEntry, [542](#)
 - gdcm::Overlay, [586](#)
- GetDescriptiveName
 - vtkGDCMImageReader, [910](#)
 - vtkGDCMImageReader2, [916](#)
 - vtkGDCMImageWriter, [921](#)
- GetDict
 - gdcm::XMLDictReader, [973](#)
- GetDictEntry
 - gdcm::Dict, [322](#)
 - gdcm::Dicts, [331](#)
 - gdcm::PrivateDict, [643](#)
- GetDictEntryByKeyword
 - gdcm::Dict, [322](#)
- GetDictEntryByName
 - gdcm::Dict, [322](#)
- GetDictName
 - gdcm::DictConverter, [324](#)
- GetDictVM
 - gdcm::Attribute, [177](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [182](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, [188](#)
- GetDictVR
 - gdcm::Attribute, [177](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [182](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, [188](#)
- GetDicts
 - gdcm::Global, [417](#)
- GetDimension
 - gdcm::Bitmap, [220](#)
- GetDimensions

- gdcm::Bitmap, [220](#)
- gdcm::Curve, [289](#)
- gdcm::ImageCodec, [444](#)
- GetDimensionsValue
 - gdcm::ImageHelper, [453](#)
- GetDimensionsValueForResolution
 - gdcm::StreamImageReader, [742](#)
- GetDirectionCosines
 - gdcm::Image, [427](#)
- GetDirectionCosinesFromDataSet
 - gdcm::ImageHelper, [453](#)
- GetDirectionCosinesTolerance
 - gdcm::IPPSorter, [478](#)
- GetDirectionCosinesValue
 - gdcm::ImageHelper, [453](#)
- GetDirectories
 - gdcm::Directory, [336](#)
- GetElapsedTime
 - gdcm::network::ARTIMTimer, [173](#)
- GetElement
 - gdcm::Tag, [789](#)
- GetElementTag
 - gdcm::Tag, [790](#)
- GetEncodeLength
 - gdcm::Base64, [199](#)
- GetErrorCode
 - gdcm::Parser, [592](#)
- GetErrorFlag
 - gdcm::Trace, [801](#)
- GetErrorStream
 - gdcm::Trace, [801](#)
- GetErrorString
 - gdcm::Parser, [592](#)
- GetEvent
 - gdcm::network::ULEvent, [874](#)
- GetEventName
 - gdcm::AnonymizeEvent, [159](#)
 - gdcm::DataEvent, [301](#)
 - gdcm::DataSetEvent, [310](#)
 - gdcm::Event, [366](#)
 - gdcm::FileNameEvent, [399](#)
 - gdcm::ProgressEvent, [648](#)
- GetExtension
 - gdcm::Filename, [396](#)
- GetFactoryInstance
 - gdcm::CryptoFactory, [268](#)
- GetFile
 - gdcm::Anonymizer, [163](#)
 - gdcm::DICOMDIRGenerator, [319](#)
 - gdcm::FileDecompressLookupTable, [385](#)
 - gdcm::FileDerivation, [386](#), [387](#)
 - gdcm::FileExplicitFilter, [389](#)
 - gdcm::IconImageFilter, [421](#)
 - gdcm::PythonFilter, [651](#), [652](#)
 - gdcm::Reader, [669](#)
 - gdcm::SplitMosaicFilter, [738](#)
 - gdcm::StreamImageReader, [742](#)
 - gdcm::StringFilter, [758](#)
 - gdcm::Writer, [970](#)
 - vtkGDCMMedicalImageProperties, [925](#)
- GetFileExtensions
 - vtkGDCMImageReader, [910](#)
 - vtkGDCMImageReader2, [916](#)
 - vtkGDCMImageWriter, [921](#)
- GetFileMetaInformationVersion
 - gdcm::FileMetaInformation, [392](#)
- GetFileName
 - gdcm::FileNameEvent, [399](#)
 - gdcm::Filename, [396](#)
 - gdcm::Testing, [797](#)
 - vtkGDCMImageWriter, [921](#)
 - vtkGDCMThreadedImageReader2, [938](#)
- GetFileNames
 - gdcm::Testing, [797](#)
- GetFilename
 - gdcm::FilenameGenerator, [401](#)
 - gdcm::TableReader, [785](#)
- GetFilenameFromTagToValue
 - gdcm::Scanner, [687](#)
 - gdcm::StrictScanner, [752](#)
- GetFilenames
 - gdcm::Directory, [337](#)
 - gdcm::FilenameGenerator, [401](#)
 - gdcm::Scanner, [687](#)
 - gdcm::Sorter, [734](#)
 - gdcm::StrictScanner, [752](#)
- GetFilenamesFromSeriesUIDs
 - gdcm::DirectoryHelper, [338](#)
- GetFiles
 - gdcm::FileSet, [403](#)
- GetFiniteVolume
 - gdcm::Surface, [766](#)
- GetFirstSingleSerieUIDFileSet
 - gdcm::SerieHelper, [713](#)
- GetForcePixelSpacing
 - gdcm::ImageHelper, [453](#)
- GetForceRescaleInterceptSlope
 - gdcm::ImageHelper, [453](#)
- GetFormat
 - gdcm::CSAHeader, [279](#)
- GetFragBuffer
 - gdcm::SequenceOfFragments, [703](#)
- GetFragment
 - gdcm::SequenceOfFragments, [703](#)
- GetFragmentSizeMax
 - gdcm::ImageFragmentSplitter, [451](#)
- GetFrameOfReference
 - gdcm::DirectoryHelper, [338](#)

- GetFullLength
 - gdcm::FileMetaInformation, [393](#)
- GetGDCMDataRoot
 - vtkGDCMTesting, [933](#)
- GetGDCMImplementationClassUID
 - gdcm::FileMetaInformation, [393](#)
- GetGDCMImplementationVersionName
 - gdcm::FileMetaInformation, [393](#)
- GetGDCMSourceApplicationEntityTitle
 - gdcm::FileMetaInformation, [393](#)
- GetGDCMUID
 - gdcm::UIDGenerator, [812](#)
- GetGroup
 - gdcm::Curve, [289](#)
 - gdcm::Overlay, [586](#)
 - gdcm::Tag, [790](#)
- GetHasExpired
 - gdcm::network::ARTIMTimer, [173](#)
- GetHeader
 - gdcm::File, [376](#)
- GetHeaderInfo
 - gdcm::ImageCodec, [445](#)
 - gdcm::JPEG12Codec, [485](#)
 - gdcm::JPEG16Codec, [487](#)
 - gdcm::JPEG2000Codec, [490](#)
 - gdcm::JPEG8Codec, [493](#)
 - gdcm::JPEGCodec, [497](#)
 - gdcm::JPEGLSCodec, [501](#)
 - gdcm::PGXCodec, [606](#)
 - gdcm::PNMCodec, [627](#)
 - gdcm::RAWCodec, [666](#)
 - gdcm::RLECodec, [680](#)
- GetHierarchicalSearchTags
 - gdcm::QueryBase, [653](#)
 - gdcm::QueryImage, [656](#)
 - gdcm::QueryPatient, [658](#)
 - gdcm::QuerySeries, [660](#)
 - gdcm::QueryStudy, [662](#)
- GetHighBit
 - gdcm::PixelFormat, [611](#)
- GetHostName
 - gdcm::System, [781](#)
- GetIE
 - gdcm::IODEntry, [473](#)
- GetIOD
 - gdcm::IODs, [475](#)
 - gdcm::SOPClassUIDToIOD, [731](#)
- GetIODEntry
 - gdcm::IOD, [472](#)
- GetIODFromFile
 - gdcm::Defs, [314](#)
- GetIODFromSOPClassUID
 - gdcm::SOPClassUIDToIOD, [731](#)
- GetIODNameFromMediaStorage
 - gdcm::Defs, [314](#)
- GetIODs
 - gdcm::Defs, [314](#), [315](#)
- GetIStream
 - gdcm::network::ULEvent, [874](#)
- GetIconImage
 - gdcm::IconImageFilter, [421](#)
 - gdcm::IconImageGenerator, [423](#)
 - gdcm::Pixmap, [616](#)
 - vtkGDCMImageReader, [910](#)
 - vtkGDCMImageReader2, [916](#)
- GetIconImagePort
 - vtkGDCMImageReader2, [916](#)
- GetImage
 - gdcm::ImageReader, [457](#)
 - gdcm::ImageWriter, [465](#)
 - gdcm::PixmapWriter, [624](#)
 - gdcm::SplitMosaicFilter, [738](#)
- GetImplementationClassUID
 - gdcm::FileMetaInformation, [393](#)
- GetImplementationVersionName
 - gdcm::FileMetaInformation, [393](#)
- GetIndex
 - gdcm::SwapCode, [777](#)
 - gdcm::VM, [897](#)
- GetInitialized
 - gdcm::CAPICryptographicMessageSyntax, [239](#)
- GetInput
 - gdcm::ImageToImageFilter, [463](#)
 - gdcm::PixmapToPixmapFilter, [622](#)
 - vtkImageColorViewer, [943](#)
- GetInputFilename
 - gdcm::DictConverter, [324](#)
- GetInstance
 - gdcm::Global, [417](#)
- GetIntercept
 - gdcm::Image, [427](#)
 - gdcm::Rescaler, [676](#)
- GetInterfile
 - gdcm::CSAHeader, [279](#)
- GetInternal
 - gdcm::Preamble, [629](#)
- GetIsCommand
 - gdcm::network::PresentationDataValue, [638](#)
- GetIsLastFragment
 - gdcm::network::PresentationDataValue, [638](#)
- GetItem
 - gdcm::SequenceOfItems, [709](#)
- GetKey
 - gdcm::CSAElement, [273](#)
- GetKeys
 - gdcm::Scanner, [687](#)
 - gdcm::StrictScanner, [752](#)
- GetKeyword

- gdcmm::DictEntry, 326
- GetKeywordFromTag
 - gdcmm::Dict, 322
- GetLUT
 - gdcmm::Bitmap, 220
 - gdcmm::ImageCodec, 445
 - gdcmm::ImageHelper, 453
 - gdcmm::LookupTable, 511
- GetLUTDescriptor
 - gdcmm::LookupTable, 511
- GetLUTLength
 - gdcmm::LookupTable, 511
- GetLabel
 - gdcmm::Orientation, 582
- GetLastElement
 - gdcmm::ParseException, 590
- GetLastSystemError
 - gdcmm::System, 781
- GetLength
 - gdcmm::ByteValue, 234
 - gdcmm::CP246ExplicitDataElement, 266
 - gdcmm::DataElement, 293
 - gdcmm::DataSet, 306
 - gdcmm::Element, 344
 - gdcmm::Element< TVR, VM::VM1_n >, 347
 - gdcmm::Element< VR::AS, VM::VM5 >, 355
 - gdcmm::ExplicitDataElement, 371
 - gdcmm::ExplicitImplicitDataElement, 373
 - gdcmm::Fragment, 415
 - gdcmm::ImplicitDataElement, 469
 - gdcmm::Item, 482
 - gdcmm::Preamble, 629
 - gdcmm::SequenceOfFragments, 703
 - gdcmm::SequenceOfItems, 709
 - gdcmm::Tag, 790
 - gdcmm::UNExplicitDataElement, 878
 - gdcmm::UNExplicitImplicitDataElement, 880
 - gdcmm::VL, 893
 - gdcmm::VM, 897
 - gdcmm::VR, 902
 - gdcmm::VR16ExplicitDataElement, 904
 - gdcmm::Value, 889
- GetLocaleCharset
 - gdcmm::System, 781
- GetLossless
 - gdcmm::JPEGCodec, 497
 - gdcmm::JPEGLSCodec, 501
- GetLossyFlag
 - gdcmm::ImageCodec, 445
- GetLossyFlagFromFile
 - gdcmm::Testing, 797
- GetMD5DataImage
 - gdcmm::Testing, 797
- GetMD5DataImages
 - gdcmm::Testing, 797
- GetMD5FromBrokenFile
 - gdcmm::Testing, 797
- GetMD5FromFile
 - gdcmm::Testing, 797
- GetMD5MetaImage
 - vtkGDCMTesting, 933
- GetMHDMD5FromFile
 - vtkGDCMTesting, 933
- GetMPType
 - gdcmm::MeshPrimitive, 532
- GetMPTypeString
 - gdcmm::MeshPrimitive, 532
- GetMRImageSeriesUIDs
 - gdcmm::DirectoryHelper, 338
- GetMSString
 - gdcmm::MediaStorage, 524
- GetMSType
 - gdcmm::MediaStorage, 524
- GetMTime
 - vtkImageMapToColors16, 949
- GetMacro
 - gdcmm::Macros, 516
- GetMacroEntry
 - gdcmm::Macro, 514
- GetMacros
 - gdcmm::Defs, 315
- GetMajorAxisFromPatientRelativeDirectionCosine
 - gdcmm::Orientation, 582
- GetMajorVersion
 - gdcmm::Version, 891
- GetManifold
 - gdcmm::Surface, 766
- GetMapping
 - gdcmm::Scanner, 687
 - gdcmm::StrictScanner, 752
- GetMappingFromTagToValue
 - gdcmm::Scanner, 687
 - gdcmm::StrictScanner, 752
- GetMappings
 - gdcmm::Scanner, 687
 - gdcmm::StrictScanner, 752
- GetMax
 - gdcmm::PixelFormat, 611
- GetMaxLength
 - gdcmm::PersonName, 603
- GetMaxPDULength
 - gdcmm::network::ULConnectionInfo, 869
- GetMaxPDUSize
 - gdcmm::network::ULConnection, 866
- GetMaximumLength
 - gdcmm::network::MaximumLengthSub, 516
- GetMaximumLengthSub
 - gdcmm::network::UserInformation, 885

- GetMaximumPointDistance
 - gdcm::Surface, 766
- GetMeanPointDistance
 - gdcm::Surface, 766
- GetMediaStorage
 - gdcm::DataSet, 306
 - gdcm::FileMetaInformation, 393
- GetMediaStorageAsString
 - gdcm::FileMetaInformation, 393
- GetMediaStorageDataFile
 - gdcm::Testing, 798
- GetMediaStorageDataFiles
 - gdcm::Testing, 798
- GetMediaStorageFromFile
 - gdcm::Testing, 798
- GetMeshPrimitive
 - gdcm::Surface, 766
- GetMessageHeader
 - gdcm::network::PresentationDataValue, 638
- GetMetaInformationTS
 - gdcm::FileMetaInformation, 393
- GetMin
 - gdcm::PixelFormat, 611
- GetMinorVersion
 - gdcm::Version, 891
- GetModality
 - gdcm::MediaStorage, 524
- GetModalityDimension
 - gdcm::MediaStorage, 524
- GetModule
 - gdcm::Modules, 544
- GetModuleEntry
 - gdcm::NestedModuleEntries, 559
- GetModuleEntryInMacros
 - gdcm::Module, 540
- GetModules
 - gdcm::Defs, 315
- GetName
 - gdcm::CSAElement, 273
 - gdcm::CSAHeaderDictEntry, 282
 - gdcm::DictEntry, 326
 - gdcm::Filename, 396
 - gdcm::GroupDict, 419
 - gdcm::IODEntry, 473
 - gdcm::Macro, 514
 - gdcm::Module, 540
 - gdcm::ModuleEntry, 542
 - gdcm::PDBelement, 596
 - gdcm::QueryBase, 653
 - gdcm::QueryImage, 657
 - gdcm::QueryPatient, 659
 - gdcm::QuerySeries, 661
 - gdcm::QueryStudy, 663
 - gdcm::UIDs, 831
 - gdcm::network::AbstractSyntax, 157
 - gdcm::network::ApplicationContext, 167
 - gdcm::network::TransferSyntaxSub, 807
- GetNeedByteSwap
 - gdcm::Bitmap, 220
 - gdcm::ImageCodec, 445
- GetNegotiatedType
 - gdcm::TransferSyntax, 805
- GetNestedDataSet
 - gdcm::Item, 482
- GetNextSingleSerieUIDFileSet
 - gdcm::SerieHelper, 713
- GetNoOfItems
 - gdcm::CSAElement, 273
- GetNumberOfComponents
 - gdcm::PersonName, 603
- GetNumberOfContourReferencedFrameOfReferences
 - vtkRTStructSetProperties, 962
- GetNumberOfCurves
 - gdcm::Curve, 289
 - gdcm::Pixmap, 616
- GetNumberOfDimensions
 - gdcm::Bitmap, 220
 - gdcm::ImageCodec, 445
- GetNumberOfElementsFromArray
 - gdcm::VM, 897
- GetNumberOfFileNames
 - gdcm::Testing, 798
- GetNumberOfFilenames
 - gdcm::FilenameGenerator, 401
- GetNumberOfFragments
 - gdcm::SequenceOfFragments, 703
- GetNumberOfIODs
 - gdcm::IOD, 472
- GetNumberOfIconImages
 - gdcm::IconImageFilter, 421
- GetNumberOfItems
 - gdcm::SequenceOfItems, 709
- GetNumberOfMD5DataImages
 - gdcm::Testing, 798
- GetNumberOfMD5MetaImages
 - vtkGDCMTesting, 933
- GetNumberOfMSString
 - gdcm::MediaStorage, 524
- GetNumberOfMSType
 - gdcm::MediaStorage, 524
- GetNumberOfMediaStorageDataFiles
 - gdcm::Testing, 798
- GetNumberOfModality
 - gdcm::MediaStorage, 524
- GetNumberOfModuleEntries
 - gdcm::NestedModuleEntries, 559
- GetNumberOfOverlays
 - gdcm::Pixmap, 616

- GetNumberOfPoints
 - gdcm::Curve, [289](#)
- GetNumberOfPresentationContext
 - gdcm::network::AAssociateRQPDU, [154](#)
- GetNumberOfPresentationContextAC
 - gdcm::network::AAssociateACPDU, [149](#)
- GetNumberOfPresentationDataValues
 - gdcm::network::PDataTFPDU, [594](#)
- GetNumberOfPrimitivesData
 - gdcm::MeshPrimitive, [532](#)
- GetNumberOfReferencedFrameOfReferences
 - vtkRTStructSetProperties, [962](#)
- GetNumberOfSOPClassToIOD
 - gdcm::SOPClassUIDToIOD, [731](#)
- GetNumberOfSegments
 - gdcm::SegmentWriter, [699](#)
- GetNumberOfStructureSetROIs
 - vtkRTStructSetProperties, [962](#)
- GetNumberOfSurfacePoints
 - gdcm::Surface, [766](#)
- GetNumberOfSurfaces
 - gdcm::SurfaceReader, [773](#)
 - gdcm::SurfaceWriter, [775](#)
- GetNumberOfTransferSyntaxStrings
 - gdcm::UIDs, [831](#)
- GetNumberOfTransferSyntaxes
 - gdcm::PresentationContext, [631](#)
 - gdcm::network::PresentationContextRQ, [637](#)
- GetNumberOfValues
 - gdcm::Attribute, [177](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [182](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, [188](#)
- GetNumberOfVectors
 - gdcm::Surface, [767](#)
- GetObliquityThresholdCosineValue
 - gdcm::Orientation, [582](#)
- GetOffScreenRendering
 - vtkImageColorViewer, [943](#)
- GetOptionalTags
 - gdcm::QueryBase, [653](#)
 - gdcm::QueryImage, [657](#)
 - gdcm::QueryPatient, [659](#)
 - gdcm::QuerySeries, [661](#)
 - gdcm::QueryStudy, [663](#)
- GetOrderedValues
 - gdcm::Scanner, [687](#)
 - gdcm::StrictScanner, [752](#)
- GetOrigin
 - gdcm::Image, [427](#)
 - gdcm::Overlay, [586](#)
- GetOriginValue
 - gdcm::ImageHelper, [453](#)
- GetOutput
 - gdcm::ImageConverter, [448](#)
- GetOutput
 - gdcm::BitmapToBitmapFilter, [225](#)
 - gdcm::ImageToImageFilter, [463](#)
 - gdcm::PixmapToPixmapFilter, [622](#)
- GetOutputAsBitmap
 - gdcm::BitmapToBitmapFilter, [225](#)
- GetOutputAsPixmap
 - gdcm::PixmapToPixmapFilter, [622](#)
- GetOutputFilename
 - gdcm::DictConverter, [324](#)
- GetOutputType
 - gdcm::DictConverter, [324](#)
- GetOverlay
 - gdcm::Pixmap, [616](#)
 - vtkGDCMImageReader, [910](#)
 - vtkGDCMImageReader2, [916](#)
- GetOverlayData
 - gdcm::Overlay, [586](#)
- GetOverlayPort
 - vtkGDCMImageReader2, [916](#)
- GetOverlayTypeAsString
 - gdcm::Overlay, [586](#)
- GetOverlayTypeFromString
 - gdcm::Overlay, [586](#)
- GetOverlayVisibility
 - vtkImageColorViewer, [944](#)
- GetOwner
 - gdcm::PrivateTag, [646](#)
- GetPDBEEnd
 - gdcm::PDBHeader, [598](#)
- GetPDBElementByName
 - gdcm::PDBHeader, [598](#)
- GetPDBInfoTag
 - gdcm::PDBHeader, [598](#)
- GetPDUs
 - gdcm::network::ULEvent, [874](#)
- GetPDVs
 - gdcm::network::PDUFactory, [602](#)
- GetPIString
 - gdcm::PhotometricInterpretation, [608](#)
- GetPIType
 - gdcm::PhotometricInterpretation, [608](#)
- GetPath
 - gdcm::Filename, [396](#)
- GetPattern
 - gdcm::FilenameGenerator, [401](#)
- GetPermissions
 - gdcm::System, [781](#)
- GetPhotometricInterpretation
 - gdcm::Bitmap, [220](#)
 - gdcm::ImageChangePhotometricInterpretation, [433](#)
 - gdcm::ImageCodec, [445](#)

- GetPhotometricInterpretationValue
 - gdcm::ImageHelper, [453](#)
- GetPixelFormat
 - gdcm::Bitmap, [220](#), [221](#)
 - gdcm::ImageCodec, [445](#)
- GetPixelFormatValue
 - gdcm::ImageHelper, [453](#)
- GetPixelRepresentation
 - gdcm::PixelFormat, [612](#)
- GetPixelSize
 - gdcm::PixelFormat, [612](#)
- GetPixelSpacingDataRoot
 - gdcm::Testing, [798](#)
- GetPixmap
 - gdcm::FileDecompressLookupTable, [385](#)
 - gdcm::IconImageGenerator, [423](#)
 - gdcm::PixmapReader, [619](#)
 - gdcm::PixmapWriter, [624](#)
- GetPlanarConfiguration
 - gdcm::Bitmap, [221](#)
 - gdcm::ImageChangePlanarConfiguration, [436](#)
 - gdcm::ImageCodec, [445](#)
- GetPlanarConfigurationValue
 - gdcm::ImageHelper, [453](#)
- GetPointCoordinatesData
 - gdcm::Surface, [767](#)
- GetPointPositionAccuracy
 - gdcm::Surface, [767](#)
- GetPointer
 - gdcm::ByteValue, [234](#)
 - gdcm::LookupTable, [511](#)
 - gdcm::SmartPointer, [728](#)
 - vtkLookupTable16, [959](#)
- GetPointerFromElement
 - gdcm::ImageHelper, [453](#)
- GetPointsBoundingBoxCoordinates
 - gdcm::Surface, [767](#)
- GetPosition
 - vtkImageColorViewer, [944](#)
- GetPreamble
 - gdcm::FileMetaInformation, [393](#)
- GetPrefix
 - gdcm::FilenameGenerator, [401](#)
- GetPresentationContext
 - gdcm::network::AAssociateRQPDU, [154](#)
- GetPresentationContextAC
 - gdcm::network::AAssociateACPDU, [149](#)
- GetPresentationContextACByID
 - gdcm::network::ULConnection, [866](#)
- GetPresentationContextByAbstractSyntax
 - gdcm::network::AAssociateRQPDU, [154](#)
- GetPresentationContextByID
 - gdcm::network::AAssociateRQPDU, [154](#)
- GetPresentationContextID
 - gdcm::PresentationContext, [631](#)
 - gdcm::network::PresentationContextAC, [633](#)
 - gdcm::network::PresentationContextRQ, [637](#)
 - gdcm::network::PresentationDataValue, [638](#)
- GetPresentationContextIDFromPresentationContext
 - gdcm::network::ULConnection, [866](#)
- GetPresentationContextRQByID
 - gdcm::network::ULConnection, [866](#)
- GetPresentationContexts
 - gdcm::PresentationContextGenerator, [635](#)
 - gdcm::network::AAssociateRQPDU, [154](#)
 - gdcm::network::ULConnection, [866](#)
- GetPresentationDataValue
 - gdcm::network::PDataTFPDU, [594](#)
- GetPrettyPrint
 - gdcm::JSON, [503](#)
- GetPrimitiveData
 - gdcm::MeshPrimitive, [532](#)
- GetPrimitiveType
 - gdcm::MeshPrimitive, [532](#)
- GetPrimitivesData
 - gdcm::MeshPrimitive, [532](#)
- GetPrintStyle
 - gdcm::Printer, [641](#)
 - gdcm::XMLPrinter, [975](#)
- GetPrivateCreator
 - gdcm::DataSet, [306](#)
 - gdcm::Tag, [790](#)
- GetPrivateDict
 - gdcm::Dicts, [331](#)
 - gdcm::XMLPrivateDictReader, [977](#)
- GetProcessingAlgorithm
 - gdcm::Surface, [767](#)
- GetProgress
 - gdcm::ProgressEvent, [648](#)
- GetPropertyCategory
 - gdcm::Segment, [691](#)
- GetPropertyType
 - gdcm::Segment, [691](#)
- GetProtocol
 - gdcm::network::ULConnection, [866](#)
- GetPublicDict
 - gdcm::Dicts, [331](#)
- GetQuality
 - gdcm::JPEG2000Codec, [491](#)
 - gdcm::JPEGCodec, [497](#)
- GetQueryDataSet
 - gdcm::BaseQuery, [207](#)
- GetQueryLevel
 - gdcm::QueryBase, [653](#)
 - gdcm::QueryImage, [657](#)
 - gdcm::QueryPatient, [659](#)
 - gdcm::QuerySeries, [661](#)
 - gdcm::QueryStudy, [663](#)

- GetQueryLevelFromQueryRoot
 - gdcm::BaseRootQuery, [210](#)
- GetQueryLevelFromString
 - gdcm::BaseRootQuery, [210](#)
- GetQueryLevelString
 - gdcm::BaseRootQuery, [210](#)
- GetRAWMD5FromFile
 - vtkGDCMTesting, [933](#)
- GetRTStructSeriesUIDs
 - gdcm::DirectoryHelper, [338](#)
- GetRate
 - gdcm::JPEG2000Codec, [491](#)
- GetRealWorldValueMappingContent
 - gdcm::ImageHelper, [453](#)
- GetReason
 - gdcm::network::PresentationContextAC, [633](#)
- GetRecommendedDisplayCIELabValue
 - gdcm::Surface, [767](#)
- GetRecommendedDisplayGrayscaleValue
 - gdcm::Surface, [767](#)
- GetRecommendedPresentationOpacity
 - gdcm::Surface, [767](#)
- GetRecommendedPresentationType
 - gdcm::Surface, [767](#)
- GetRef
 - gdcm::IODEntry, [473](#)
- GetReferencedFrameOfReferenceClassUID
 - vtkRTStructSetProperties, [962](#)
- GetReferencedFrameOfReferenceInstanceUID
 - vtkRTStructSetProperties, [962](#)
- GetRegion
 - gdcm::ImageRegionReader, [460](#)
- GetRequiredDataSet
 - gdcm::ModalityPerformedProcedureStepCreate↔
Query, [535](#)
 - gdcm::ModalityPerformedProcedureStepSetQuery,
[537](#)
- GetRequiredTags
 - gdcm::QueryBase, [654](#)
 - gdcm::QueryImage, [657](#)
 - gdcm::QueryPatient, [659](#)
 - gdcm::QuerySeries, [661](#)
 - gdcm::QueryStudy, [663](#)
- GetRescaleInterceptSlopeValue
 - gdcm::ImageHelper, [453](#)
- GetReserved43_74
 - gdcm::network::AAssociateRQPDU, [154](#)
- GetResponses
 - gdcm::network::ULBasicCallback, [864](#)
- GetRetired
 - gdcm::DictEntry, [326](#)
- GetRoot
 - gdcm::UIDGenerator, [812](#)
- GetRows
 - gdcm::Bitmap, [221](#)
 - gdcm::Overlay, [586](#)
- GetSOPClassUID
 - gdcm::DirectoryHelper, [339](#)
- GetSOPClassUIDFromIOD
 - gdcm::SOPClassUIDToIOD, [731](#)
- GetSOPClassUIDToIOD
 - gdcm::SOPClassUIDToIOD, [731](#)
- GetSOPClassUIDToIODs
 - gdcm::SOPClassUIDToIOD, [731](#)
- GetSOPInstanceUID
 - gdcm::BaseQuery, [208](#)
- GetSTATES
 - gdcm::Surface, [767](#)
- GetSTATESString
 - gdcm::Surface, [767](#)
- GetSamplesPerPixel
 - gdcm::PhotometricInterpretation, [608](#)
 - gdcm::PixelFormat, [612](#)
- GetScalarType
 - gdcm::PixelFormat, [612](#)
- GetScalarTypeAsString
 - gdcm::PixelFormat, [612](#)
- GetScanner
 - gdcm::DICOMDIRGenerator, [320](#)
- GetSegment
 - gdcm::SegmentWriter, [699](#)
- GetSegmentAlgorithmName
 - gdcm::Segment, [691](#)
- GetSegmentAlgorithmType
 - gdcm::Segment, [692](#)
- GetSegmentDescription
 - gdcm::Segment, [692](#)
- GetSegmentLabel
 - gdcm::Segment, [692](#)
- GetSegmentNumber
 - gdcm::Segment, [692](#)
- GetSegments
 - gdcm::SegmentReader, [697](#)
 - gdcm::SegmentWriter, [699](#)
- GetSelectedPrivateGroupOffsetFromFile
 - gdcm::Testing, [798](#)
- GetSelectedTagsOffsetFromFile
 - gdcm::Testing, [798](#)
- GetSequenceOfFragments
 - gdcm::DataElement, [293](#)
- GetSeriesUIDsBySOPClassUID
 - gdcm::DirectoryHelper, [339](#)
- GetSize
 - gdcm::VR, [902](#)
 - vtkImageColorViewer, [944](#)
- GetSizeof
 - gdcm::VR, [902](#)
- GetSliceMax

- vtkImageColorViewer, [944](#)
- GetSliceMin
 - vtkImageColorViewer, [944](#)
- GetSliceRange
 - vtkImageColorViewer, [944](#)
- GetSlope
 - gdcm::Image, [427](#)
 - gdcm::Rescaler, [676](#)
- GetSourceApplicationEntityTitle
 - gdcm::FileMetaInformation, [393](#)
- GetSourceDirectory
 - gdcm::Testing, [798](#)
- GetSpacing
 - gdcm::Image, [427](#)
- GetSpacingTagFromMediaStorage
 - gdcm::ImageHelper, [454](#)
- GetSpacingValue
 - gdcm::ImageHelper, [454](#)
- GetStart
 - gdcm::ByteBuffer, [229](#)
- GetState
 - gdcm::network::ULConnection, [866](#)
- GetStateIndex
 - gdcm::network, [141](#)
- GetStream
 - gdcm::Trace, [801](#)
- GetStreamCurrentPosition
 - gdcm::Reader, [669](#)
- GetStreamOffsetFromFile
 - gdcm::Testing, [798](#)
- GetStreamPtr
 - gdcm::Reader, [669](#)
 - gdcm::Writer, [970](#)
- GetString
 - gdcm::MediaStorage, [524](#)
 - gdcm::PhotometricInterpretation, [608](#)
 - gdcm::TransferSyntax, [805](#)
 - gdcm::UIDs, [832](#)
- GetStringValueFromTag
 - gdcm::DirectoryHelper, [339](#)
- GetStructureSetObservationNumber
 - vtkRTStructSetProperties, [962](#)
- GetStructureSetROIDescription
 - vtkRTStructSetProperties, [962](#)
- GetStructureSetROIGenerationAlgorithm
 - vtkRTStructSetProperties, [962](#)
- GetStructureSetROIName
 - vtkRTStructSetProperties, [962](#)
- GetStructureSetROINumber
 - vtkRTStructSetProperties, [962](#)
- GetStructureSetROIObservationLabel
 - vtkRTStructSetProperties, [962](#)
- GetStructureSetROIRefFrameRefUID
 - vtkRTStructSetProperties, [962](#)
- GetStructureSetRTROIInterpretedType
 - vtkRTStructSetProperties, [962](#)
- GetSurface
 - gdcm::Segment, [692](#)
- GetSurfaceComments
 - gdcm::Surface, [767](#)
- GetSurfaceCount
 - gdcm::Segment, [692](#)
- GetSurfaceNumber
 - gdcm::Surface, [767](#)
- GetSurfaceProcessing
 - gdcm::Surface, [767](#)
- GetSurfaceProcessingDescription
 - gdcm::Surface, [767](#)
- GetSurfaceProcessingRatio
 - gdcm::Surface, [767](#)
- GetSurfaces
 - gdcm::Segment, [692](#)
- GetSwapCode
 - gdcm::TransferSyntax, [805](#)
- GetSwapCodeString
 - gdcm::SwapCode, [777](#)
- GetSyngoDT
 - gdcm::CSAElement, [273](#)
- GetTSString
 - gdcm::TransferSyntax, [805](#)
- GetTSType
 - gdcm::TransferSyntax, [806](#)
- GetTable
 - gdcm::SequenceOfFragments, [703](#), [704](#)
- GetTableEntry
 - gdcm::Table, [783](#)
- GetTag
 - gdcm::AnonymizeEvent, [159](#)
 - gdcm::Attribute, [178](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [182](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, [188](#)
 - gdcm::DataElement, [294](#)
- GetTagListByLevel
 - gdcm::BaseRootQuery, [210](#)
 - gdcm::FindPatientRootQuery, [410](#)
 - gdcm::FindStudyRootQuery, [412](#)
 - gdcm::MovePatientRootQuery, [546](#)
 - gdcm::MoveStudyRootQuery, [548](#)
 - gdcm::WLMFindQuery, [966](#)
- GetTempDirectory
 - gdcm::Testing, [798](#)
- GetTempDirectoryW
 - gdcm::Testing, [799](#)
- GetTempFilename
 - gdcm::Testing, [799](#)
- GetTempFilenameW

- gdcm::Testing, 799
- GetTimeout
 - gdcm::ServiceClassUser, 718
 - gdcm::network::ARTIMTimer, 173
- GetTimer
 - gdcm::network::ULConnection, 866
- GetTimezoneOffsetFromUTC
 - gdcm::System, 781
- GetToplevel
 - gdcm::Directory, 337
- GetTransferSyntax
 - gdcm::Bitmap, 221
 - gdcm::ImageChangeTransferSyntax, 439
 - gdcm::PresentationContext, 631
 - gdcm::network::PresentationContextAC, 633
 - gdcm::network::PresentationContextRQ, 637
- GetTransferSyntaxString
 - gdcm::UIDs, 832
- GetTransferSyntaxStrings
 - gdcm::UIDs, 832
- GetTransferSyntaxes
 - gdcm::network::PresentationContextRQ, 637
- GetType
 - gdcm::ModuleEntry, 542
 - gdcm::Orientation, 582
 - gdcm::Overlay, 586
 - gdcm::PhotometricInterpretation, 608
- GetTypeAsEnum
 - gdcm::Overlay, 586
- GetTypeFromTag
 - gdcm::Defs, 315
 - gdcm::IOD, 472
- GetTypeOfData
 - gdcm::Curve, 289
- GetTypeOfDataDescription
 - gdcm::Curve, 289
- GetTypeString
 - gdcm::Type, 810
- GetTypeType
 - gdcm::Type, 810
- GetUIDName
 - gdcm::UIDs, 832
- GetUIDString
 - gdcm::UIDs, 832
- GetUniqueTags
 - gdcm::QueryBase, 654
 - gdcm::QueryImage, 657
 - gdcm::QueryPatient, 659
 - gdcm::QuerySeries, 661
 - gdcm::QueryStudy, 663
- GetUnpackBuffer
 - gdcm::Overlay, 587
- GetUnpackBufferLength
 - gdcm::Overlay, 587
- GetUsage
 - gdcm::IODEntry, 473
- GetUsageString
 - gdcm::Usage, 883
- GetUsageType
 - gdcm::IODEntry, 474
 - gdcm::Usage, 883
- GetUserData
 - gdcm::Parser, 592
- GetUserInformation
 - gdcm::network::AAssociateACPDU, 149
 - gdcm::network::AAssociateRQPDU, 154
- GetVIEWType
 - gdcm::Surface, 768
- GetVIEWTypeString
 - gdcm::Surface, 768
- GetVL
 - gdcm::DataElement, 294
- GetVL16Max
 - gdcm::VL, 893
- GetVL32Max
 - gdcm::VL, 893
- GetVM
 - gdcm::Attribute, 178
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 183
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 185
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 186
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 189
 - gdcm::Attribute< Group, Element, TVR, VM::VM2↵_2n >, 191
 - gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 192
 - gdcm::Attribute< Group, Element, TVR, VM::VM3↵_3n >, 194
 - gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 195
 - gdcm::CSAElement, 274
 - gdcm::CSAHeaderDictEntry, 282
 - gdcm::DictEntry, 326
 - gdcm::Element, 344
 - gdcm::Element< TVR, VM::VM1_n >, 347
- GetVMString
 - gdcm::VM, 897
- GetVMType
 - gdcm::VM, 897
- GetVMTypeFromLength
 - gdcm::VM, 898
- GetVR
 - gdcm::Attribute, 178

- gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 183
- gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 189
- gdcm::CSAElement, 274
- gdcm::CSAHeaderDictEntry, 282
- gdcm::DataElement, 295
- gdcm::DictEntry, 326
- gdcm::Element, 344
- gdcm::Element< TVR, VM::VM1_n >, 347
- GetVRFromTag
 - gdcm, 131
- GetVRString
 - gdcm::VR, 902
- GetVRStringFromFile
 - gdcm::VR, 902
- GetVRType
 - gdcm::VR, 902
- GetVRTypeFromFile
 - gdcm::VR, 902
- GetVTKDataRoot
 - vtkGDCMTesting, 933
- GetValidDataSet
 - gdcm::WLMFindQuery, 966
- GetValidatedFile
 - gdcm::Validate, 887
- GetValue
 - gdcm::Attribute, 178
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 182
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 189
 - gdcm::CSAElement, 273
 - gdcm::DataElement, 294
 - gdcm::Element, 344
 - gdcm::Element< TVR, VM::VM1_n >, 347
 - gdcm::PDBelement, 596
 - gdcm::Scanner, 687
 - gdcm::StrictScanner, 752
- GetValueAsSQ
 - gdcm::DataElement, 294
- GetValues
 - gdcm::Attribute, 178
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 182
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 189
 - gdcm::Element, 344
 - gdcm::Scanner, 688
 - gdcm::StrictScanner, 753
- GetVectorAccuracy
 - gdcm::Surface, 767
- GetVectorCoordinateData
 - gdcm::Surface, 767
- GetVectorDimensionality
 - gdcm::Surface, 768
- GetVersion
 - gdcm::Version, 891
- GetWarningFlag
 - gdcm::Trace, 801
- GetWarningStream
 - gdcm::Trace, 801
- GetWindowName
 - vtkImageColorViewer, 944
- GetXMax
 - gdcm::BoxRegion, 228
- GetXMin
 - gdcm::BoxRegion, 228
- GetYMax
 - gdcm::BoxRegion, 228
- GetYMin
 - gdcm::BoxRegion, 228
- GetZMax
 - gdcm::BoxRegion, 228
- GetZMin
 - gdcm::BoxRegion, 228
- GetZSpacing
 - gdcm::IPPSorter, 478
- GetZSpacingTagFromMediaStorage
 - gdcm::ImageHelper, 454
- GetZSpacingTolerance
 - gdcm::IPPSorter, 478
- Global
 - gdcm::Defs, 315
 - gdcm::Dicts, 331
 - gdcm::Global, 416
- GlobalInstance
 - gdcm, 135
- GrabOverlayFromPixelData
 - gdcm::Overlay, 587
- Graphics
 - gdcm::Overlay, 585
- GrayscaleSoftcopyPresentationStateStorageSOPClass
 - gdcm::MediaStorage, 522
 - gdcm::UIDs, 822
- green
 - gdcm::terminal, 143
- group
 - gdcm::SerieHelper::Rule, 683
- GroupDict
 - gdcm::GroupDict, 419
- GroupStringVector
 - gdcm::GroupDict, 419
- GuessFromModality
 - gdcm::MediaStorage, 524
- HSV
 - gdcm::PhotometricInterpretation, 607

- HandleBulkData
 - gdcm::XMLPrinter, [975](#)
- HandleDataSet
 - gdcm::network::ULBasicCallback, [864](#)
 - gdcm::network::ULConnectionCallback, [868](#)
 - gdcm::network::ULWritingCallback, [877](#)
- HandleDescription
 - gdcm::XMLDictReader, [973](#)
 - gdcm::XMLPrivateDictReader, [977](#)
- HandleEntry
 - gdcm::XMLDictReader, [973](#)
 - gdcm::XMLPrivateDictReader, [977](#)
- HandleEvent
 - gdcm::network::ULTransitionTable, [875](#)
- HandleIOD
 - gdcm::TableReader, [785](#)
- HandleIODEntry
 - gdcm::TableReader, [785](#)
- HandleMacro
 - gdcm::TableReader, [785](#)
- HandleMacroEntry
 - gdcm::TableReader, [785](#)
- HandleMacroEntryDescription
 - gdcm::TableReader, [785](#)
- HandleModule
 - gdcm::TableReader, [785](#)
- HandleModuleEntry
 - gdcm::TableReader, [785](#)
- HandleModuleEntryDescription
 - gdcm::TableReader, [785](#)
- HandleModuleInclude
 - gdcm::TableReader, [786](#)
- HandleResponse
 - gdcm::network::ULBasicCallback, [864](#)
 - gdcm::network::ULConnectionCallback, [868](#)
 - gdcm::network::ULWritingCallback, [877](#)
- HangingProtocolInformationModelFIND
 - gdcm::UIDs, [824](#)
- HangingProtocolInformationModelMOVE
 - gdcm::UIDs, [824](#)
- HangingProtocolStorage
 - gdcm::MediaStorage, [523](#)
 - gdcm::UIDs, [824](#)
- HardcopyColorImageStorageSOPClassRetired
 - gdcm::UIDs, [821](#)
- HardcopyGrayscaleImageStorage
 - gdcm::MediaStorage, [522](#)
- HardcopyGrayscaleImageStorageSOPClassRetired
 - gdcm::UIDs, [821](#)
- HasObserver
 - gdcm::Subject, [762](#)
- HemodynamicWaveformStorage
 - gdcm::MediaStorage, [522](#)
 - gdcm::UIDs, [821](#)
- hidden
 - gdcm::terminal, [143](#)
- ICBM452T1FrameofReference
 - gdcm::UIDs, [820](#)
- ICBMSingleSubjectMRIFrameofReference
 - gdcm::UIDs, [820](#)
- ID
 - gdcm::PresentationContext, [632](#)
- INT12
 - gdcm::PixelFormat, [611](#)
- INT16
 - gdcm::PixelFormat, [611](#)
- INT32
 - gdcm::PixelFormat, [611](#)
- INT64
 - gdcm::PixelFormat, [611](#)
- INT8
 - gdcm::PixelFormat, [610](#)
- INTERFILE
 - gdcm::CSAHeader, [277](#)
- INVALID
 - gdcm::VR, [900](#)
- IOD
 - gdcm::IOD, [471](#)
- IODEntry
 - gdcm::IODEntry, [473](#)
- IODMapType
 - gdcm::IODs, [475](#)
- IODMapTypeConstIterator
 - gdcm::IODs, [475](#)
- IODName
 - gdcm::IODs, [475](#)
- IODs
 - gdcm::IODs, [475](#)
- IPPSorter
 - gdcm::IPPSorter, [478](#)
- IS
 - gdcm::VR, [901](#)
- Icon
 - gdcm::Pixmap, [616](#)
- IconDataScalarType
 - vtkGDCMImageReader, [912](#)
 - vtkGDCMImageReader2, [918](#)
- IconImage
 - gdcm, [129](#)
- IconImageDataExtent
 - vtkGDCMImageReader, [912](#)
 - vtkGDCMImageReader2, [918](#)
- IconImageFilter
 - gdcm::IconImageFilter, [421](#)
- IconImageGenerator
 - gdcm::IconImageGenerator, [423](#)
- IconNumberOfScalarComponents

- vtkGDCMImageReader, 912
 - vtkGDCMImageReader2, 918
- ignore_char
 - gdcm::ignore_char, 424
- Image
 - gdcm::Image, 427
- ImageActor
 - vtkImageColorViewer, 946
- ImageApplyLookupTable
 - gdcm::ImageApplyLookupTable, 431
- ImageChangePhotometricInterpretation
 - gdcm::ImageChangePhotometricInterpretation, 433
 - gdcm::ImageCodec, 446
- ImageChangePlanarConfiguration
 - gdcm::ImageChangePlanarConfiguration, 436
- ImageChangeTransferSyntax
 - gdcm::Bitmap, 223
 - gdcm::ImageChangeTransferSyntax, 439
- ImageCodec
 - gdcm::ImageCodec, 443
- ImageConverter
 - gdcm::ImageConverter, 448
- ImageFormat
 - vtkGDCMImageReader, 912
 - vtkGDCMImageReader2, 918
- ImageFragmentSplitter
 - gdcm::ImageFragmentSplitter, 451
- ImageOrientationPatient
 - vtkGDCMImageReader, 912
 - vtkGDCMImageReader2, 918
- ImageOverlayBoxSOPClassRetired
 - gdcm::UIDs, 821
- ImagePositionPatient
 - vtkGDCMImageReader, 912
 - vtkGDCMImageReader2, 918
- ImagePositionPatientOrdering
 - gdcm::SerieHelper, 713
- ImageReader
 - gdcm::ImageReader, 457
- ImageRegionReader
 - gdcm::ImageRegionReader, 460
 - gdcm::JPEG2000Codec, 491
 - gdcm::JPEGCodec, 498
 - gdcm::JPEGLSCCodec, 502
 - gdcm::RLECodec, 681
- ImageToImageFilter
 - gdcm::ImageToImageFilter, 462
- ImageWriter
 - gdcm::ImageWriter, 465
- ImplementationClassUIDSub
 - gdcm::network::ImplementationClassUIDSub, 466
- ImplementationUIDSub
 - gdcm::network::ImplementationUIDSub, 467
- ImplementationVersionNameSub
 - gdcm::network::ImplementationVersionNameSub, 467
- Implicit
 - gdcm::TransferSyntax, 804
- ImplicitVRBigEndianACRNEMA
 - gdcm::TransferSyntax, 805
- ImplicitVRBigEndianPrivateGE
 - gdcm::TransferSyntax, 804
- ImplicitVRLittleEndian
 - gdcm::TransferSyntax, 804
- ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM
 - gdcm::UIDs, 818
- IncompleteLUT
 - gdcm::LookupTable, 512
- InitFromRQ
 - gdcm::network::AAssociateACPDU, 149
- InitOpenSSL
 - gdcm::OpenSSLCryptoFactory, 574
- Initialize
 - gdcm::network::ULConnectionInfo, 869
- InitializeBlueLUT
 - gdcm::LookupTable, 511
- InitializeConnection
 - gdcm::ServiceClassUser, 718
 - gdcm::network::ULConnection, 866
- InitializeDataSet
 - gdcm::BaseRootQuery, 211
 - gdcm::FindPatientRootQuery, 410
 - gdcm::FindStudyRootQuery, 412
 - gdcm::MovePatientRootQuery, 546
 - gdcm::MoveStudyRootQuery, 549
 - gdcm::WLMFindQuery, 966
- InitializeGreenLUT
 - gdcm::LookupTable, 511
- InitializeIncomingConnection
 - gdcm::network::ULConnection, 866
- InitializeLUT
 - gdcm::LookupTable, 511
- InitializeRTStructSet
 - vtkGDCMPolyDataWriter, 930
- InitializeRedLUT
 - gdcm::LookupTable, 511
- Initialized
 - gdcm::LookupTable, 511
- Input
 - gdcm::BitmapToBitmapFilter, 225
- Insert
 - gdcm::CommandDataSet, 259
 - gdcm::DataSet, 306
 - gdcm::FileMetaInformation, 393
 - gdcm::GroupDict, 419
- InsertDataElement
 - gdcm::DataSet, 306
 - gdcm::Item, 482

- InsertEntry
 - gdcm::Table, [783](#)
- InstallPipeline
 - vtkImageColorViewer, [944](#)
- InstanceAvailabilityNotificationSOPClass
 - gdcm::UIDs, [823](#)
- Interactor
 - vtkImageColorViewer, [946](#)
- InteractorStyle
 - vtkImageColorViewer, [946](#)
- Internal
 - gdcm::ApplicationEntity, [168](#)
 - gdcm::Attribute, [180](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [184](#)
 - gdcm::Element, [344](#)
 - gdcm::Element< VR::AS, VM::VM5 >, [355](#)
 - gdcm::LookupTable, [512](#)
 - gdcm::UI, [811](#)
- InternalCode
 - gdcm::Coder, [253](#)
 - gdcm::JPEG12Codec, [485](#)
 - gdcm::JPEG16Codec, [487](#)
 - gdcm::JPEG8Codec, [493](#)
- Internals
 - vtkRTStructSetProperties, [963](#)
- Invalid
 - gdcm::Overlay, [585](#)
 - gdcm::Usage, [883](#)
- InverseRescale
 - gdcm::Rescaler, [676](#)
- InverseRescaleFunctionIntoBestFit
 - gdcm::Rescaler, [676](#)
- InvokeEvent
 - gdcm::Subject, [762](#)
- IsAETitleValid
 - gdcm::network::AAssociateRQPDU, [154](#)
- IsASCII
 - gdcm::VR, [902](#)
- IsASCII2
 - gdcm::VR, [902](#)
- IsBinary
 - gdcm::VR, [902](#)
- IsBinary2
 - gdcm::VR, [902](#)
- IsCompatible
 - gdcm::PixelFormat, [612](#)
- IsDual
 - gdcm::VR, [902](#)
- IsEmpty
 - gdcm::Bitmap, [221](#)
 - gdcm::ByteValue, [235](#)
 - gdcm::CSAElement, [274](#)
 - gdcm::CSAHeaderDict, [281](#)
 - gdcm::Curve, [289](#)
 - gdcm::DataElement, [295](#)
 - gdcm::DataSet, [306](#)
 - gdcm::Defs, [315](#)
 - gdcm::Dict, [322](#)
 - gdcm::Dicts, [331](#)
 - gdcm::Filename, [396](#)
 - gdcm::Macros, [516](#)
 - gdcm::Modules, [544](#)
 - gdcm::Overlay, [587](#)
 - gdcm::Preamble, [629](#)
 - gdcm::PrivateDict, [643](#)
 - gdcm::SegmentHelper::BasicCodedEntry, [213](#)
- IsEncapsulated
 - gdcm::TransferSyntax, [806](#)
- IsEncoded
 - gdcm::TransferSyntax, [806](#)
- IsExplicit
 - gdcm::TransferSyntax, [806](#)
- IsFrameEncoder
 - gdcm::ImageCodec, [445](#)
 - gdcm::JPEG2000Codec, [491](#)
 - gdcm::JPEGCodec, [497](#)
 - gdcm::JPEGLSCodec, [501](#)
 - gdcm::RLECodec, [680](#)
- IsGroupLength
 - gdcm::Tag, [790](#)
- IsGroupXX
 - gdcm::Tag, [790](#)
- IsIdentical
 - gdcm::Filename, [396](#)
- IsIllegal
 - gdcm::Tag, [790](#)
- IsImage
 - gdcm::MediaStorage, [524](#)
- IsImplicit
 - gdcm::TransferSyntax, [806](#)
- IsInPixelData
 - gdcm::Overlay, [587](#)
- IsKey
 - gdcm::Scanner, [688](#)
 - gdcm::StrictScanner, [753](#)
- IsLastFragment
 - gdcm::network::AAAbortPDU, [146](#)
 - gdcm::network::AAssociateACPD, [149](#)
 - gdcm::network::AAssociateRJPDU, [151](#)
 - gdcm::network::AAssociateRQPDU, [154](#)
 - gdcm::network::AReleaseRPPDU, [170](#)
 - gdcm::network::AReleaseRQPDU, [172](#)
 - gdcm::network::BasePDU, [205](#)
 - gdcm::network::PDataTFPDU, [594](#)
- IsLossless
 - gdcm::PhotometricInterpretation, [608](#)
 - gdcm::TransferSyntax, [806](#)

- IsLossy
 - gdcm::Bitmap, 221
 - gdcm::ImageCodec, 445
 - gdcm::PhotometricInterpretation, 608
 - gdcm::TransferSyntax, 806
- IsOdd
 - gdcm::VL, 893
- IsPresentationContextAccepted
 - gdcm::ServiceClassUser, 718
- IsPrintable
 - gdcm::ByteValue, 235
- IsPrivate
 - gdcm::Tag, 790
- IsPrivateCreator
 - gdcm::Tag, 791
- IsPublic
 - gdcm::Tag, 791
- IsRetired
 - gdcm::PhotometricInterpretation, 608
- IsRowEncoder
 - gdcm::ImageCodec, 445
 - gdcm::JPEG2000Codec, 491
 - gdcm::JPEGCodec, 497
 - gdcm::JPEGLSCodec, 501
 - gdcm::RLECodec, 680
- IsSameColorSpace
 - gdcm::PhotometricInterpretation, 608
- IsStateSuspension
 - gdcm::JPEG12Codec, 485
 - gdcm::JPEG16Codec, 487
 - gdcm::JPEG8Codec, 493
 - gdcm::JPEGCodec, 497
- IsSwap
 - gdcm::VR, 902
- IsTransferSyntaxCompatible
 - gdcm::Bitmap, 221
- IsUndefined
 - gdcm::MediaStorage, 524
 - gdcm::VL, 893
- IsUndefinedLength
 - gdcm::DataElement, 295
 - gdcm::SequenceOfItems, 709
- IsUnique
 - gdcm::DictEntry, 327
- IsVRFile
 - gdcm::VR, 902
- IsValid
 - gdcm::ApplicationEntity, 168
 - gdcm::BoxRegion, 228
 - gdcm::CodeString, 255
 - gdcm::DirectionCosines, 334
 - gdcm::FileMetaInformation, 393
 - gdcm::ImageCodec, 445
 - gdcm::JPEGCodec, 498
 - gdcm::LO, 507
 - gdcm::PixelFormat, 612
 - gdcm::Preamble, 629
 - gdcm::Region, 674
 - gdcm::String, 756
 - gdcm::TagPath, 794
 - gdcm::TransferSyntax, 806
 - gdcm::UIDGenerator, 813
 - gdcm::UUIIDGenerator, 886
 - gdcm::VM, 898
 - gdcm::VR, 902
- IsZero
 - gdcm::Overlay, 587
- ItFileSetHt
 - gdcm::SerieHelper, 713
- Item
 - gdcm::Item, 481
- ItemVector
 - gdcm::SequenceOfItems, 708
- Items
 - gdcm::SequenceOfItems, 710
- Iterator
 - gdcm::CSAHeaderDict, 280
 - gdcm::DataSet, 304
 - gdcm::Dict, 321
 - gdcm::SequenceOfFragments, 702
 - gdcm::SequenceOfItems, 708
- iterator
 - gdcm::CodeString, 254
 - gdcm::LO, 507
 - gdcm::String, 756
- JPEG12Codec
 - gdcm::JPEG12Codec, 485
- JPEG16Codec
 - gdcm::JPEG16Codec, 487
- JPEG2000
 - gdcm::TransferSyntax, 805
- JPEG2000_COMPRESSION
 - vtkGDCMImageWriter, 921
- JPEG2000Codec
 - gdcm::JPEG2000Codec, 489
- JPEG2000ImageCompression
 - gdcm::UIDs, 819
- JPEG2000ImageCompressionLosslessOnly
 - gdcm::UIDs, 819
- JPEG2000Lossless
 - gdcm::TransferSyntax, 805
- JPEG2000Part2
 - gdcm::TransferSyntax, 805
- JPEG2000Part2Lossless
 - gdcm::TransferSyntax, 805
- JPEG2000Part2MulticomponentImageCompression
 - gdcm::UIDs, 819

- JPEG2000Part2MulticomponentImageCompression↔
 - LosslessOnly
 - gdcm::UIDs, [819](#)
- JPEG8Codec
 - gdcm::JPEG8Codec, [493](#)
- JPEG_COMPRESSION
 - vtkGDCMImageWriter, [921](#)
- JPEGBaselineProcess1
 - gdcm::TransferSyntax, [805](#)
- JPEGBaselineProcess1DefaultTransferSyntaxforLossyJ↔
 - PEG8BitImageCompression
 - gdcm::UIDs, [818](#)
- JPEGCodec
 - gdcm::JPEGCodec, [496](#)
- JPEGExtendedHierarchicalProcess1618Retired
 - gdcm::UIDs, [819](#)
- JPEGExtendedHierarchicalProcess1719Retired
 - gdcm::UIDs, [819](#)
- JPEGExtendedProcess24DefaultTransferSyntaxfor↔
 - LossyJPEG12BitImageCompressionProcess4only
 - gdcm::UIDs, [818](#)
- JPEGExtendedProcess2_4
 - gdcm::TransferSyntax, [805](#)
- JPEGExtendedProcess35Retired
 - gdcm::UIDs, [818](#)
- JPEGExtendedProcess3_5
 - gdcm::TransferSyntax, [805](#)
- JPEGFullProgressionHierarchicalProcess2426Retired
 - gdcm::UIDs, [819](#)
- JPEGFullProgressionHierarchicalProcess2527Retired
 - gdcm::UIDs, [819](#)
- JPEGFullProgressionNonHierarchicalProcess1012↔
 - Retired
 - gdcm::UIDs, [818](#)
- JPEGFullProgressionNonHierarchicalProcess1113↔
 - Retired
 - gdcm::UIDs, [818](#)
- JPEGFullProgressionProcess10_12
 - gdcm::TransferSyntax, [805](#)
- JPEGLS_COMPRESSION
 - vtkGDCMImageWriter, [921](#)
- JPEGLSCodec
 - gdcm::JPEGLSCodec, [500](#)
- JPEGLSLossless
 - gdcm::TransferSyntax, [805](#)
- JPEGLSLosslessImageCompression
 - gdcm::UIDs, [819](#)
- JPEGLSLossyNearLosslessImageCompression
 - gdcm::UIDs, [819](#)
- JPEGLSNearLossless
 - gdcm::TransferSyntax, [805](#)
- JPEGLosslessHierarchicalProcess28Retired
 - gdcm::UIDs, [819](#)
- JPEGLosslessHierarchicalProcess29Retired
 - gdcm::UIDs, [819](#)
- JPEGLosslessNonHierarchicalFirstOrderPrediction↔
 - Process14SelectionValue1DefaultTransfer↔
 - SyntaxforLosslessJPEGImageCompression
 - gdcm::UIDs, [819](#)
- JPEGLosslessNonHierarchicalProcess14
 - gdcm::UIDs, [818](#)
- JPEGLosslessNonHierarchicalProcess15Retired
 - gdcm::UIDs, [819](#)
- JPEGLosslessProcess14
 - gdcm::TransferSyntax, [805](#)
- JPEGLosslessProcess14_1
 - gdcm::TransferSyntax, [805](#)
- JPEGSpectralSelectionHierarchicalProcess2022Retired
 - gdcm::UIDs, [819](#)
- JPEGSpectralSelectionHierarchicalProcess2123Retired
 - gdcm::UIDs, [819](#)
- JPEGSpectralSelectionNonHierarchicalProcess68Retired
 - gdcm::UIDs, [818](#)
- JPEGSpectralSelectionNonHierarchicalProcess79Retired
 - gdcm::UIDs, [818](#)
- JPEGSpectralSelectionProcess6_8
 - gdcm::TransferSyntax, [805](#)
- JPIPReferenced
 - gdcm::TransferSyntax, [805](#)
 - gdcm::UIDs, [819](#)
- JPIPReferencedDeflate
 - gdcm::UIDs, [819](#)
- JSON
 - gdcm::JSON, [503](#)
- Join
 - gdcm::Filename, [396](#)
- JunkAfterDocElementError
 - gdcm::Parser, [591](#)
- KAKADUCodec
 - gdcm::KAKADUCodec, [505](#)
- KeyField
 - gdcm::CSAElement, [275](#)
- KeyObjectSelectionDocument
 - gdcm::MediaStorage, [522](#)
- KeyObjectSelectionDocumentStorage
 - gdcm::UIDs, [823](#)
- KeyValuePairArrayType
 - gdcm::CompositeNetworkFunctions, [261](#)
- KeyValuePairType
 - gdcm::CompositeNetworkFunctions, [261](#)
- LD_ALL
 - gdcm, [131](#)
- LD_NOSEQ
 - gdcm, [131](#)
- LD_NOSHADOW
 - gdcm, [131](#)
- LD_NOSHADOWSEQ

- gdcM, [131](#)
- LINE
 - gdcM::MeshPrimitive, [531](#)
- LO
 - gdcM::LO, [507](#)
 - gdcM::VR, [901](#)
- LOADBULKDATA
 - gdcM::XMLPrinter, [975](#)
- LOComp
 - gdcM, [129](#)
- LT
 - gdcM::VR, [901](#)
- LTComp
 - gdcM, [129](#)
- LUT
 - gdcM::Bitmap, [223](#)
 - gdcM::ImageCodec, [447](#)
- LUTPtr
 - gdcM::Bitmap, [219](#)
 - gdcM::ImageCodec, [443](#)
- LeadECGWaveformStorage
 - gdcM::MediaStorage, [522](#)
- Level
 - vtkImageMapToWindowLevelColors2, [952](#)
- ListCharSets
 - gdcM::QueryFactory, [655](#)
- LittleEndian
 - gdcM::SwapCode, [776](#)
- Load
 - gdcM::Directory, [337](#)
- LoadDefault
 - gdcM::CSAHeaderDict, [281](#)
 - gdcM::Dict, [322](#)
 - gdcM::PrivateDict, [643](#)
- LoadDefaults
 - gdcM::Defs, [315](#)
 - gdcM::Dicts, [331](#)
- LoadFromDataElement
 - gdcM::CSAHeader, [279](#)
 - gdcM::PDBHeader, [598](#)
- LoadFromFile
 - gdcM::Defs, [315](#)
- LoadIconImage
 - vtkGDCMImageReader, [913](#)
 - vtkGDCMImageReader2, [918](#)
- LoadImageFromFiles
 - gdcM::DirectoryHelper, [339](#)
- LoadOverlays
 - vtkGDCMImageReader, [913](#)
 - vtkGDCMImageReader2, [918](#)
- LoadResourcesFiles
 - gdcM::Global, [417](#)
- LoadSingleFile
 - vtkGDCMImageReader, [910](#)
- vtkGDCMImageReader2, [916](#)
- Locate
 - gdcM::Global, [417](#)
- LodModeType
 - gdcM, [130](#)
- LookupTable
 - gdcM::LookupTable, [510](#)
 - vtkImageMapToColors16, [950](#)
- LookupTableType
 - gdcM::LookupTable, [510](#)
- LossyFlag
 - gdcM::Bitmap, [223](#)
 - gdcM::ImageCodec, [447](#)
 - vtkGDCMImageReader, [913](#)
 - vtkGDCMImageReader2, [918](#)
- m_ConstMemberFunction
 - gdcM::MemberCommand, [528](#)
- m_MemberFunction
 - gdcM::MemberCommand, [529](#)
 - gdcM::SimpleMemberCommand, [724](#)
- m_This
 - gdcM::MemberCommand, [529](#)
 - gdcM::SimpleMemberCommand, [724](#)
- m_char
 - gdcM::ignore_char, [424](#)
- MAGNIFIED
 - gdcM::Spacing, [736](#)
- MANUAL
 - gdcM::Segment, [691](#)
- mAction
 - gdcM::network::Transition, [809](#)
- mConnection
 - gdcM::network::ULConnectionManager, [873](#)
- MD5
 - gdcM::MD5, [517](#)
- MD5DataImagesType
 - gdcM::Testing, [796](#)
- MD5MetalImagesType
 - vtkGDCMTesting, [933](#)
- mDataSet
 - gdcM::BaseQuery, [208](#)
- mElementOffsets
 - gdcM::StreamImageWriter, [747](#)
- mElementOffsets1
 - gdcM::StreamImageWriter, [747](#)
- mEnd
 - gdcM::network::Transition, [809](#)
- mHelpDescription
 - gdcM::BaseQuery, [208](#)
 - gdcM::BaseRootQuery, [211](#)
- mImage
 - gdcM::BaseRootQuery, [211](#)
- mImplicit

- gdcm::network::ULConnectionCallback, [868](#)
- MONOCHROME1
 - gdcm::PhotometricInterpretation, [607](#)
- MONOCHROME2
 - gdcm::PhotometricInterpretation, [607](#)
- MPEG2MainProfile
 - gdcm::TransferSyntax, [805](#)
- MPEG2MainProfileHighLevel
 - gdcm::TransferSyntax, [805](#)
- MPEG2MainProfileMainLevel
 - gdcm::UIDs, [819](#)
- MPEG4AVCH264BDcompatibleHighProfileLevel4_1
 - gdcm::TransferSyntax, [805](#)
- MPEG4AVCH264HighProfileLevel4_1
 - gdcm::TransferSyntax, [805](#)
- MPTType
 - gdcm::MeshPrimitive, [531](#)
- MPTType_END
 - gdcm::MeshPrimitive, [531](#)
- mPatient
 - gdcm::BaseRootQuery, [211](#)
- MRImageStorage
 - gdcm::MediaStorage, [521](#)
 - gdcm::UIDs, [821](#)
- MRSpectroscopyStorage
 - gdcm::MediaStorage, [521](#)
 - gdcm::UIDs, [821](#)
- mRootType
 - gdcm::BaseRootQuery, [211](#)
- MS_END
 - gdcm::MediaStorage, [523](#)
- MSType
 - gdcm::MediaStorage, [521](#)
- mSecondaryConnection
 - gdcm::network::ULConnectionManager, [873](#)
- mSeries
 - gdcm::BaseRootQuery, [211](#)
- mSopInstanceUID
 - gdcm::BaseQuery, [208](#)
- mStudy
 - gdcm::BaseRootQuery, [211](#)
- mTransitions
 - gdcm::network::ULConnectionManager, [873](#)
- mWriter
 - gdcm::StreamImageWriter, [748](#)
- mXMax
 - gdcm::StreamImageWriter, [748](#)
- mXMin
 - gdcm::StreamImageWriter, [748](#)
- mYMax
 - gdcm::StreamImageWriter, [748](#)
- mYMin
 - gdcm::StreamImageWriter, [748](#)
- mZMax
 - gdcm::StreamImageWriter, [748](#)
- gdcm::StreamImageWriter, [748](#)
- mZMin
 - gdcm::StreamImageWriter, [748](#)
- Macro
 - gdcm::Macro, [514](#)
- MacroEntry
 - gdcm, [129](#)
- Macros
 - gdcm::Macros, [515](#)
- magenta
 - gdcm::terminal, [143](#)
- MakeDirectory
 - gdcm::System, [781](#)
- MakeNew
 - gdcm::network::Transition, [808](#)
- MakeObject
 - gdcm::AnonymizeEvent, [159](#)
 - gdcm::DataEvent, [301](#)
 - gdcm::DataSetEvent, [310](#)
 - gdcm::Event, [366](#)
 - gdcm::FileNameEvent, [399](#)
 - gdcm::ProgressEvent, [648](#)
- MammographyCADSR
 - gdcm::MediaStorage, [522](#)
- MammographyCADSRStorage
 - gdcm::UIDs, [822](#)
- Mandatory
 - gdcm::Usage, [883](#)
- MapCSAHeaderDictEntry
 - gdcm::CSAHeaderDict, [280](#)
- MapDictEntry
 - gdcm::Dict, [321](#)
- MapIODEntry
 - gdcm::IOD, [471](#)
- MapModuleEntry
 - gdcm::Macro, [514](#)
 - gdcm::Module, [539](#)
- MapScalarsThroughTable2
 - vtkLookupTable16, [959](#)
- MapTableEntry
 - gdcm::Table, [783](#)
- MappingType
 - gdcm::Scanner, [686](#)
 - gdcm::StrictScanner, [751](#)
- MaxLength
 - gdcm::ApplicationEntity, [168](#)
 - gdcm::PersonName, [604](#)
- MaxNumberOfComponents
 - gdcm::ApplicationEntity, [168](#)
 - gdcm::PersonName, [604](#)
- MaxPrintLength
 - gdcm::Printer, [642](#)
- MaximumLengthSub
 - gdcm::network::MaximumLengthSub, [516](#)

- MediaCreationManagementSOPClassUID
 - gdcm::UIDs, [821](#)
- MediaStorage
 - gdcm::MediaStorage, [524](#)
- MediaStorageDataFilesType
 - gdcm::Testing, [796](#)
- MediaStorageDirectoryStorage
 - gdcm::MediaStorage, [521](#)
 - gdcm::UIDs, [819](#)
- MedicalImageProperties
 - vtkGDCMImageReader, [913](#)
 - vtkGDCMPolyDataReader, [928](#)
 - vtkGDCMPolyDataWriter, [931](#)
- MemberCommand
 - gdcm::MemberCommand, [528](#)
- MeshPrimitive
 - gdcm::MeshPrimitive, [532](#)
- MessageID
 - gdcm::network::CEchoRQ, [241](#)
- MetaInformationTS
 - gdcm::FileMetaInformation, [394](#)
- ModalityPerformedProcedureStepCreateQuery
 - gdcm::ModalityPerformedProcedureStepCreate↔Query, [534](#)
- ModalityPerformedProcedureStepNotificationSOPClass
 - gdcm::UIDs, [820](#)
- ModalityPerformedProcedureStepRetrieveSOPClass
 - gdcm::UIDs, [820](#)
- ModalityPerformedProcedureStepSOPClass
 - gdcm::MediaStorage, [523](#)
 - gdcm::UIDs, [820](#)
- ModalityPerformedProcedureStepSetQuery
 - gdcm::ModalityPerformedProcedureStepSetQuery, [537](#)
- ModalityWorklistInformationModelFIND
 - gdcm::UIDs, [823](#)
- Mode
 - gdcm::terminal, [143](#)
- Module
 - gdcm::Module, [540](#)
- ModuleEntry
 - gdcm::ModuleEntry, [542](#)
- ModuleMapType
 - gdcm::Macros, [515](#)
 - gdcm::Modules, [544](#)
- Modules
 - gdcm::Modules, [544](#)
- MovePatientRootQuery
 - gdcm::MovePatientRootQuery, [546](#)
- MoveStudyRootQuery
 - gdcm::MoveStudyRootQuery, [548](#)
- mspFile
 - gdcm::StreamImageWriter, [748](#)
- MultiframeGrayscaleByteSecondaryCaptureImage↔Storage
 - gdcm::MediaStorage, [521](#)
 - gdcm::UIDs, [821](#)
- MultiframeGrayscaleWordSecondaryCaptureImage↔Storage
 - gdcm::MediaStorage, [521](#)
 - gdcm::UIDs, [821](#)
- MultiframeSingleBitSecondaryCaptureImageStorage
 - gdcm::MediaStorage, [521](#)
 - gdcm::UIDs, [821](#)
- MultiframeTrueColorSecondaryCaptureImageStorage
 - gdcm::MediaStorage, [522](#)
 - gdcm::UIDs, [821](#)
- N_ACTION_RQ
 - gdcm::network::DIMSE, [333](#)
- N_ACTION_RSP
 - gdcm::network::DIMSE, [333](#)
- N_CREATE_RQ
 - gdcm::network::DIMSE, [333](#)
- N_CREATE_RSP
 - gdcm::network::DIMSE, [333](#)
- N_DELETE_RQ
 - gdcm::network::DIMSE, [333](#)
- N_DELETE_RSP
 - gdcm::network::DIMSE, [333](#)
- N_EVENT_REPORT_RQ
 - gdcm::network::DIMSE, [333](#)
- N_EVENT_REPORT_RSP
 - gdcm::network::DIMSE, [333](#)
- N_GET_RQ
 - gdcm::network::DIMSE, [333](#)
- N_GET_RSP
 - gdcm::network::DIMSE, [333](#)
- N_SET_RQ
 - gdcm::network::DIMSE, [333](#)
- N_SET_RSP
 - gdcm::network::DIMSE, [333](#)
- NAction
 - gdcm::NormalizedNetworkFunctions, [567](#)
- NCreate
 - gdcm::NormalizedNetworkFunctions, [567](#)
- NDelete
 - gdcm::NormalizedNetworkFunctions, [567](#)
- NEventReport
 - gdcm::NormalizedNetworkFunctions, [567](#)
- NGet
 - gdcm::NormalizedNetworkFunctions, [568](#)
- NO
 - gdcm::Surface, [765](#)
- NO_COMPRESSION
 - vtkGDCMImageWriter, [921](#)
- NOMAGIC

- gdcmm::CSAHeader, [277](#)
- NSet
 - gdcmm::NormalizedNetworkFunctions, [568](#)
- Name
 - gdcmm::ModuleEntry, [543](#)
- NameField
 - gdcmm::CSAElement, [275](#)
 - gdcmm::PDBelement, [597](#)
- NeedByteSwap
 - gdcmm::Bitmap, [223](#)
 - gdcmm::ImageCodec, [447](#)
- NeedOverlayCleanup
 - gdcmm::ImageCodec, [447](#)
- NegotiatedType
 - gdcmm::TransferSyntax, [804](#)
- NestedMacroEntries
 - gdcmm, [129](#)
- NestedModuleEntries
 - gdcmm::NestedModuleEntries, [559](#)
- New
 - gdcmm::Anonymizer, [163](#)
 - gdcmm::FileChangeTransferSyntax, [382](#)
 - gdcmm::FileStreamer, [405](#)
 - gdcmm::MemberCommand, [528](#)
 - gdcmm::Scanner, [688](#)
 - gdcmm::SequenceOfFragments, [704](#)
 - gdcmm::SequenceOfItems, [709](#)
 - gdcmm::ServiceClassUser, [718](#)
 - gdcmm::SimpleMemberCommand, [724](#)
 - gdcmm::StrictScanner, [753](#)
 - vtkGDCMImageReader, [910](#)
 - vtkGDCMImageReader2, [916](#)
 - vtkGDCMImageWriter, [921](#)
 - vtkGDCMMedicalImageProperties, [925](#)
 - vtkGDCMPolyDataReader, [927](#)
 - vtkGDCMPolyDataWriter, [930](#)
 - vtkGDCMTesting, [933](#)
 - vtkGDCMThreadedImageReader, [936](#)
 - vtkGDCMThreadedImageReader2, [938](#)
 - vtkImageColorViewer, [944](#)
 - vtkImageMapToColors16, [949](#)
 - vtkImageMapToWindowLevelColors2, [952](#)
 - vtkImagePlanarComponentsToComponents, [954](#)
 - vtkImageRGBToYBR, [955](#)
 - vtkImageYBRToRGB, [957](#)
 - vtkLookupTable16, [959](#)
 - vtkRTStructSetProperties, [962](#)
- NoElementsError
 - gdcmm::Parser, [591](#)
- NoError
 - gdcmm::Parser, [591](#)
- NoMemoryError
 - gdcmm::Parser, [591](#)
- NoObject
 - gdcmm::MediaStorage, [523](#)
- NoOfItemsField
 - gdcmm::CSAElement, [275](#)
- Normalize
 - gdcmm::DirectionCosines, [334](#)
- NuclearMedicineImageStorage
 - gdcmm::MediaStorage, [522](#)
 - gdcmm::UIDs, [822](#)
- NuclearMedicineImageStorageRetired
 - gdcmm::MediaStorage, [521](#)
 - gdcmm::UIDs, [821](#)
- NumberOfDimensions
 - gdcmm::Bitmap, [223](#)
 - gdcmm::ImageCodec, [447](#)
- NumberOfIconImages
 - vtkGDCMImageReader, [913](#)
 - vtkGDCMImageReader2, [918](#)
- NumberOfOverlays
 - vtkGDCMImageReader, [913](#)
 - vtkGDCMImageReader2, [918](#)
- NumberOfSurfaces
 - gdcmm::SurfaceWriter, [775](#)
- OB
 - gdcmm::VR, [901](#)
- OB_OW
 - gdcmm::VR, [901](#)
- OBLIQUE
 - gdcmm::Orientation, [581](#)
- OD
 - gdcmm::VR, [901](#)
- OF
 - gdcmm::VR, [901](#)
- OPENSSL
 - gdcmm::CryptoFactory, [268](#)
- OPENSSL7
 - gdcmm::CryptoFactory, [268](#)
- OW
 - gdcmm::VR, [901](#)
- Object
 - gdcmm::Object, [572](#)
- ObjectEnd
 - gdcmm::MediaStorage, [523](#)
- ObjectType
 - gdcmm::MediaStorage, [523](#)
- Ofstream
 - gdcmm::Writer, [971](#)
- OnlyUUID
 - gdcmm::XMLPrinter, [975](#)
- op
 - gdcmm::SerieHelper::Rule, [683](#)
- OpenSSLCryptoFactory
 - gdcmm::OpenSSLCryptoFactory, [574](#)
- OpenSSLCryptographicMessageSyntax

- gdcmm::OpenSSLCryptographicMessageSyntax, 576
- OpenSSL7CryptoFactory
 - gdcmm::OpenSSL7CryptoFactory, 577
- OpenSSL7CryptographicMessageSyntax
 - gdcmm::OpenSSL7CryptographicMessageSyntax, 579
- operator const char *
 - gdcmm::ConstCharWrapper, 265
 - gdcmm::Filename, 396
 - gdcmm::String, 756
- operator const double *
 - gdcmm::DirectionCosines, 335
- operator const std::vector< char > &
 - gdcmm::ByteValue, 235
- operator MStype
 - gdcmm::MediaStorage, 525
- operator ObjectType *
 - gdcmm::SmartPointer, 729
- operator PType
 - gdcmm::PhotometricInterpretation, 608
- operator ScalarType
 - gdcmm::PixelFormat, 612
- operator SwapCode::SwapCodeType
 - gdcmm::SwapCode, 777
- operator TStype
 - gdcmm::TransferSyntax, 806
 - gdcmm::UIDs, 832
- operator TypeType
 - gdcmm::Type, 810
- operator uint32_t
 - gdcmm::VL, 893
- operator UsageType
 - gdcmm::Usage, 883
- operator VMType
 - gdcmm::VM, 898
- operator VRType
 - gdcmm::VR, 902
- operator!=
 - gdcmm, 131
 - gdcmm::Attribute, 178
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 183
 - gdcmm::CodeString, 255
 - gdcmm::PixelFormat, 612
 - gdcmm::Tag, 791
- operator<
 - gdcmm::Attribute, 178
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 183
 - gdcmm::CSAElement, 274
 - gdcmm::CSAHeaderDictEntry, 282
 - gdcmm::DataElement, 295
 - gdcmm::PrivateTag, 646
 - gdcmm::Tag, 791
- operator<<
 - gdcmm, 131–135
 - gdcmm::BasicOffsetTable, 216
 - gdcmm::CSAElement, 275
 - gdcmm::CSAHeader, 279
 - gdcmm::CSAHeaderDict, 281
 - gdcmm::CSAHeaderDictEntry, 283
 - gdcmm::CodeString, 255
 - gdcmm::CommandDataSet, 259
 - gdcmm::DataElement, 298
 - gdcmm::DataSet, 308
 - gdcmm::Dict, 322
 - gdcmm::DictEntry, 327
 - gdcmm::Dicts, 331
 - gdcmm::Directory, 337
 - gdcmm::File, 377
 - gdcmm::FileMetaInformation, 394
 - gdcmm::FileSet, 403
 - gdcmm::Fragment, 415
 - gdcmm::Global, 418
 - gdcmm::GroupDict, 419
 - gdcmm::IOD, 472
 - gdcmm::IODEntry, 474
 - gdcmm::IODs, 475
 - gdcmm::Item, 482
 - gdcmm::Macro, 514
 - gdcmm::Macros, 516
 - gdcmm::MediaStorage, 525
 - gdcmm::Module, 540
 - gdcmm::ModuleEntry, 543
 - gdcmm::Modules, 544
 - gdcmm::NestedModuleEntries, 559
 - gdcmm::Object, 573
 - gdcmm::Orientation, 582
 - gdcmm::PDBelement, 596
 - gdcmm::PDBHeader, 599
 - gdcmm::PhotometricInterpretation, 608
 - gdcmm::PixelFormat, 613
 - gdcmm::Preamble, 629
 - gdcmm::PrivateDict, 644
 - gdcmm::PrivateTag, 646
 - gdcmm::Scanner, 689
 - gdcmm::Sorter, 735
 - gdcmm::StrictScanner, 753
 - gdcmm::SwapCode, 777
 - gdcmm::Table, 783
 - gdcmm::Tag, 793
 - gdcmm::TransferSyntax, 806
 - gdcmm::Type, 810
 - gdcmm::UI, 811
 - gdcmm::Usage, 883
 - gdcmm::VL, 894
 - gdcmm::VM, 898
 - gdcmm::VR, 903

- gdcmm::Version, [892](#)
- operator<=
 - gdcmm::Tag, [791](#)
- operator>>
 - gdcmm, [135](#)
 - gdcmm::Tag, [793](#)
- operator*
 - gdcmm::SmartPointer, [729](#)
- operator()
 - gdcmm::DataSet, [307](#)
 - gdcmm::Scanner::ltstr, [512](#)
 - gdcmm::StrictScanner::ltstr, [513](#)
- operator++
 - gdcmm::VL, [893](#)
- operator+=
 - gdcmm::VL, [893](#)
- operator->
 - gdcmm::SmartPointer, [729](#)
- operator=
 - gdcmm::BoxRegion, [228](#)
 - gdcmm::ByteValue, [235](#)
 - gdcmm::CSAElement, [274](#)
 - gdcmm::DataElement, [295](#)
 - gdcmm::DataSet, [307](#)
 - gdcmm::Element< TVR, VM::VM1_n >, [347](#)
 - gdcmm::Object, [572](#)
 - gdcmm::Overlay, [587](#)
 - gdcmm::ParseException, [590](#)
 - gdcmm::Preamble, [629](#)
 - gdcmm::SequenceOfItems, [709](#)
 - gdcmm::SmartPointer, [729](#)
 - gdcmm::Tag, [791](#)
 - gdcmm::network::UserInformation, [885](#)
- operator==
 - gdcmm, [135](#)
 - gdcmm::Attribute, [178](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [183](#)
 - gdcmm::ByteValue, [235](#)
 - gdcmm::CSAElement, [274](#)
 - gdcmm::CodeString, [255](#)
 - gdcmm::DataElement, [295](#)
 - gdcmm::PDBelement, [596](#)
 - gdcmm::PixelFormat, [612](#), [613](#)
 - gdcmm::PresentationContext, [631](#)
 - gdcmm::SequenceOfFragments, [704](#)
 - gdcmm::SequenceOfItems, [709](#)
 - gdcmm::Tag, [791](#)
 - gdcmm::Value, [889](#)
 - gdcmm::network::AbstractSyntax, [157](#)
 - gdcmm::network::PresentationContextRQ, [637](#)
 - gdcmm::network::TransferSyntaxSub, [807](#)
- operator[]
 - gdcmm::Attribute, [179](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, [189](#)
- gdcmm::DataSet, [307](#)
- gdcmm::Element, [344](#)
- gdcmm::Element< TVR, VM::VM1_n >, [348](#)
- gdcmm::Tag, [791](#)
- OphthalmicPhotography16BitImageStorage
 - gdcmm::UIDs, [822](#)
- OphthalmicPhotography8BitImageStorage
 - gdcmm::MediaStorage, [523](#)
 - gdcmm::UIDs, [822](#)
- OphthalmicTomographyImageStorage
 - gdcmm::MediaStorage, [523](#)
 - gdcmm::UIDs, [822](#)
- OrderFileList
 - gdcmm::SerieHelper, [713](#)
- Orientation
 - gdcmm::Orientation, [582](#)
- OrientationType
 - gdcmm::Orientation, [581](#)
- Output
 - gdcmm::BitmapToBitmapFilter, [225](#)
- OutputFormat
 - vtkImageMapToColors16, [950](#)
- OutputTypes
 - gdcmm::DictConverter, [324](#)
- Overlay
 - gdcmm::Overlay, [585](#)
- OverlayImageActor
 - vtkImageColorViewer, [947](#)
- OverlayType
 - gdcmm::Overlay, [585](#)
- Overlays
 - gdcmm::Pixmap, [616](#)
- PALETTE_COLOR
 - gdcmm::PhotometricInterpretation, [607](#)
- PDBelement
 - gdcmm::PDBelement, [596](#)
- PDBHeader
 - gdcmm::PDBHeader, [598](#)
- PDF
 - gdcmm::MediaStorage, [523](#)
- PDFCodec
 - gdcmm::PDFCodec, [600](#)
- PDataTFPDU
 - gdcmm::network::PDataTFPDU, [594](#)
- PETImageStorage
 - gdcmm::MediaStorage, [522](#)
- PF
 - gdcmm::Bitmap, [223](#)
 - gdcmm::ImageCodec, [447](#)
- PGXCodec
 - gdcmm::PGXCodec, [605](#)

- PHILIPS
 - gdcm::Dicts, [330](#)
- PI
 - gdcm::Bitmap, [223](#)
 - gdcm::ImageCodec, [447](#)
- PI_END
 - gdcm::PhotometricInterpretation, [607](#)
- PIType
 - gdcm::PhotometricInterpretation, [607](#)
- PN
 - gdcm::VR, [901](#)
- PNComp
 - gdcm, [129](#)
- PNMCodec
 - gdcm::PNMCodec, [627](#)
- POINTS
 - gdcm::Surface, [766](#)
- PVRGCodec
 - gdcm::PVRGCodec, [650](#)
- Pack
 - gdcm::Unpacker12Bits, [881](#)
- Padding
 - gdcm::ApplicationEntity, [168](#)
 - gdcm::PersonName, [604](#)
- Parent
 - gdcm::Element< TVR, VM::VM1_2 >, [346](#)
 - gdcm::Element< TVR, VM::VM2_2n >, [350](#)
 - gdcm::Element< TVR, VM::VM2_n >, [351](#)
 - gdcm::Element< TVR, VM::VM3_3n >, [353](#)
 - gdcm::Element< TVR, VM::VM3_n >, [354](#)
- Parse
 - gdcm::Parser, [592](#)
- ParseBuffer
 - gdcm::Parser, [592](#)
- ParseCertificateFile
 - gdcm::CAPICryptographicMessageSyntax, [239](#)
 - gdcm::CryptographicMessageSyntax, [270](#)
 - gdcm::OpenSSLCryptographicMessageSyntax, [576](#)
 - gdcm::OpenSSL7CryptographicMessageSyntax, [580](#)
- ParseDateTime
 - gdcm::System, [781](#)
- ParseDump
 - gdcm::ASN1, [174](#)
- ParseDumpFile
 - gdcm::ASN1, [174](#)
- ParseException
 - gdcm::ParseException, [590](#)
- ParseKeyFile
 - gdcm::CAPICryptographicMessageSyntax, [239](#)
 - gdcm::CryptographicMessageSyntax, [270](#)
 - gdcm::OpenSSLCryptographicMessageSyntax, [576](#)
 - gdcm::OpenSSL7CryptographicMessageSyntax, [580](#)
- Parser
 - gdcm::Parser, [592](#)
- PassAlphaToOutput
 - vtkImageMapToColors16, [950](#)
- Patient
 - gdcm::Patient, [593](#)
- PatientRootQueryRetrieveInformationModelFIND
 - gdcm::UIDs, [823](#)
- PatientRootQueryRetrieveInformationModelGET
 - gdcm::UIDs, [823](#)
- PatientRootQueryRetrieveInformationModelMOVE
 - gdcm::UIDs, [823](#)
- PatientStudyOnlyQueryRetrieveInformationModelFIND↔
 - Retired
 - gdcm::UIDs, [823](#)
- PatientStudyOnlyQueryRetrieveInformationModelGET↔
 - Retired
 - gdcm::UIDs, [823](#)
- PatientStudyOnlyQueryRetrieveInformationModelMOV↔
 - ERetired
 - gdcm::UIDs, [823](#)
- PerformAction
 - gdcm::network::ULAction, [834](#)
 - gdcm::network::ULActionAA1, [835](#)
 - gdcm::network::ULActionAA2, [836](#)
 - gdcm::network::ULActionAA3, [837](#)
 - gdcm::network::ULActionAA4, [838](#)
 - gdcm::network::ULActionAA5, [839](#)
 - gdcm::network::ULActionAA6, [840](#)
 - gdcm::network::ULActionAA7, [841](#)
 - gdcm::network::ULActionAA8, [842](#)
 - gdcm::network::ULActionAE1, [843](#)
 - gdcm::network::ULActionAE2, [844](#)
 - gdcm::network::ULActionAE3, [845](#)
 - gdcm::network::ULActionAE4, [846](#)
 - gdcm::network::ULActionAE5, [847](#)
 - gdcm::network::ULActionAE6, [848](#)
 - gdcm::network::ULActionAE7, [849](#)
 - gdcm::network::ULActionAE8, [850](#)
 - gdcm::network::ULActionAR1, [851](#)
 - gdcm::network::ULActionAR10, [852](#)
 - gdcm::network::ULActionAR2, [853](#)
 - gdcm::network::ULActionAR3, [854](#)
 - gdcm::network::ULActionAR4, [855](#)
 - gdcm::network::ULActionAR5, [856](#)
 - gdcm::network::ULActionAR6, [857](#)
 - gdcm::network::ULActionAR7, [858](#)
 - gdcm::network::ULActionAR8, [859](#)
 - gdcm::network::ULActionAR9, [860](#)
 - gdcm::network::ULActionDT1, [861](#)
 - gdcm::network::ULActionDT2, [862](#)
- Philips3D
 - gdcm::MediaStorage, [522](#)
- PhilipsPrivateMRSyntheticImageStorage

- gdcm::MediaStorage, [523](#)
- PhotometricInterpretation
 - gdcm::PhotometricInterpretation, [608](#)
- PixelData
 - gdcm::Bitmap, [223](#)
 - gdcm::PixmapReader, [620](#)
 - gdcm::PixmapWriter, [625](#)
- PixelFormat
 - gdcm::PixelFormat, [611](#)
- Pixmap
 - gdcm::Pixmap, [615](#)
- PixmapReader
 - gdcm::Bitmap, [223](#)
 - gdcm::PixmapReader, [619](#)
- PixmapToPixmapFilter
 - gdcm::PixmapToPixmapFilter, [621](#)
- PixmapWriter
 - gdcm::PixmapWriter, [624](#)
- PlanarConfiguration
 - gdcm::Bitmap, [223](#)
 - gdcm::ImageCodec, [447](#)
 - vtkGDCMImageReader, [913](#)
 - vtkGDCMImageReader2, [919](#)
- pointer
 - gdcm::CodeString, [254](#)
 - gdcm::LO, [507](#)
 - gdcm::String, [756](#)
- PositronEmissionTomographyImageStorage
 - gdcm::UIDs, [823](#)
- Preamble
 - gdcm::Preamble, [629](#)
- PrepareWrite
 - gdcm::PixmapWriter, [624](#)
 - gdcm::SegmentWriter, [699](#)
 - gdcm::SurfaceWriter, [775](#)
- PrepareWritePointMacro
 - gdcm::SurfaceWriter, [775](#)
- Prepend
 - gdcm::Global, [417](#)
- PresentationContext
 - gdcm::PresentationContext, [631](#)
- PresentationContextAC
 - gdcm::network::PresentationContextAC, [633](#)
- PresentationContextArrayType
 - gdcm::PresentationContextGenerator, [634](#)
 - gdcm::network::AAssociateRQPDU, [153](#)
- PresentationContextGenerator
 - gdcm::PresentationContextGenerator, [634](#)
- PresentationContextRQ
 - gdcm::network::PresentationContextRQ, [636](#)
- PresentationDataValue
 - gdcm::network::PresentationDataValue, [638](#)
- PresentationLUTSOPClass
 - gdcm::UIDs, [821](#)
- PrettyPrintOff
 - gdcm::JSON, [503](#)
- PrettyPrintOn
 - gdcm::JSON, [503](#)
- PrimitiveData
 - gdcm::MeshPrimitive, [532](#)
- PrimitiveType
 - gdcm::MeshPrimitive, [532](#)
- PrimitivesData
 - gdcm::MeshPrimitive, [531](#)
- Print
 - gdcm::ApplicationEntity, [168](#)
 - gdcm::Attribute, [179](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [183](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, [189](#)
 - gdcm::BaseQuery, [208](#)
 - gdcm::Bitmap, [221](#)
 - gdcm::BoxRegion, [228](#)
 - gdcm::ByteValue, [235](#)
 - gdcm::CSAHeader, [279](#)
 - gdcm::Curve, [289](#)
 - gdcm::DataSet, [307](#)
 - gdcm::DictPrinter, [329](#)
 - gdcm::DirectionCosines, [335](#)
 - gdcm::Directory, [337](#)
 - gdcm::Element, [344](#)
 - gdcm::Element< TVR, VM::VM1_n >, [348](#)
 - gdcm::Element< VR::AS, VM::VM5 >, [355](#)
 - gdcm::Event, [366](#)
 - gdcm::Image, [427](#)
 - gdcm::LookupTable, [511](#)
 - gdcm::Object, [572](#)
 - gdcm::Orientation, [582](#)
 - gdcm::Overlay, [587](#)
 - gdcm::PDBHeader, [598](#)
 - gdcm::PersonName, [603](#)
 - gdcm::PixelFormat, [613](#)
 - gdcm::Pixmap, [616](#)
 - gdcm::Preamble, [629](#)
 - gdcm::PresentationContext, [631](#)
 - gdcm::Printer, [641](#)
 - gdcm::Region, [674](#)
 - gdcm::Scanner, [688](#)
 - gdcm::SegmentedPaletteColorLookupTable, [694](#)
 - gdcm::SequenceOfFragments, [704](#)
 - gdcm::SequenceOfItems, [709](#)
 - gdcm::Sorter, [734](#)
 - gdcm::StrictScanner, [753](#)
 - gdcm::TagPath, [794](#)
 - gdcm::Testing, [799](#)
 - gdcm::Version, [891](#)
 - gdcm::XMLPrinter, [975](#)

- gdcm::network::AAAbortPDU, [146](#)
- gdcm::network::AAAssociateACPDU, [149](#)
- gdcm::network::AAAssociateRJPDU, [151](#)
- gdcm::network::AAAssociateRQPDU, [154](#)
- gdcm::network::AReleaseRPPDU, [170](#)
- gdcm::network::AReleaseRQPDU, [172](#)
- gdcm::network::AbstractSyntax, [157](#)
- gdcm::network::ApplicationContext, [167](#)
- gdcm::network::AsynchronousOperationsWindow↔
Sub, [174](#)
- gdcm::network::BasePDU, [205](#)
- gdcm::network::ImplementationClassUIDSub, [466](#)
- gdcm::network::ImplementationVersionNameSub,
[467](#)
- gdcm::network::MaximumLengthSub, [516](#)
- gdcm::network::PDataTFPDU, [594](#)
- gdcm::network::PresentationContextAC, [633](#)
- gdcm::network::PresentationContextRQ, [637](#)
- gdcm::network::PresentationDataValue, [638](#)
- gdcm::network::RoleSelectionSub, [682](#)
- gdcm::network::SOPClassExtendedNegociationSub,
[730](#)
- gdcm::network::ServiceClassApplicationInformation,
[714](#)
- gdcm::network::TransferSyntaxSub, [807](#)
- gdcm::network::UserInformation, [885](#)
- PrintASCII
 - gdcm::ByteValue, [235](#)
- PrintASCIIXML
 - gdcm::ByteValue, [235](#)
- PrintAsContinuousString
 - gdcm::Tag, [791](#)
- PrintAsContinuousUpperCaseString
 - gdcm::Tag, [792](#)
- PrintAsPipeSeparatedString
 - gdcm::Tag, [792](#)
- PrintDataElement
 - gdcm::Printer, [641](#)
 - gdcm::XMLPrinter, [975](#)
- PrintDataElement2
 - gdcm::DictPrinter, [329](#)
- PrintDataSet
 - gdcm::Printer, [642](#)
 - gdcm::XMLPrinter, [975](#)
- PrintDataSet2
 - gdcm::DictPrinter, [329](#)
- PrintGroupLength
 - gdcm::ByteValue, [235](#)
- PrintHex
 - gdcm::ByteValue, [235](#)
- PrintHexXML
 - gdcm::ByteValue, [235](#)
- PrintJobSOPClass
 - gdcm::UIDs, [820](#)
- PrintPNXML
 - gdcm::ByteValue, [235](#)
- PrintQueueManagementSOPClassRetired
 - gdcm::UIDs, [821](#)
- PrintQueueSOPInstanceRetired
 - gdcm::UIDs, [821](#)
- PrintSQ
 - gdcm::Printer, [642](#)
 - gdcm::XMLPrinter, [975](#)
- PrintSelf
 - vtkGDCMImageReader, [910](#)
 - vtkGDCMImageReader2, [916](#)
 - vtkGDCMImageWriter, [921](#)
 - vtkGDCMMedicalImageProperties, [925](#)
 - vtkGDCMPolyDataReader, [927](#)
 - vtkGDCMPolyDataWriter, [930](#)
 - vtkGDCMTesting, [933](#)
 - vtkGDCMThreadedImageReader, [936](#)
 - vtkGDCMThreadedImageReader2, [938](#)
 - vtkImageColorViewer, [944](#)
 - vtkImageMapToColors16, [949](#)
 - vtkImageMapToWindowLevelColors2, [952](#)
 - vtkImagePlanarComponentsToComponents, [954](#)
 - vtkImageRGBToYBR, [955](#)
 - vtkImageYBRToRGB, [957](#)
 - vtkLookupTable16, [959](#)
 - vtkRTStructSetProperties, [963](#)
- PrintStyle
 - gdcm::Printer, [642](#)
 - gdcm::XMLPrinter, [975](#)
- PrintStyles
 - gdcm::Printer, [641](#)
 - gdcm::XMLPrinter, [975](#)
- PrintTable
 - gdcm::network::ULTransitionTable, [875](#)
- PrintXML
 - gdcm::PrivateDict, [643](#)
- Printer
 - gdcm::Printer, [641](#)
- PrinterConfigurationRetrievalSOPClass
 - gdcm::UIDs, [820](#)
- PrinterConfigurationRetrievalSOPInstance
 - gdcm::UIDs, [820](#)
- PrinterSOPClass
 - gdcm::UIDs, [820](#)
- PrinterSOPInstance
 - gdcm::UIDs, [820](#)
- PrivateDict
 - gdcm::PrivateDict, [643](#)
- PrivateTag
 - gdcm::PrivateTag, [645](#)
- ProceduralEventLoggingSOPClass
 - gdcm::UIDs, [820](#)
- ProceduralEventLoggingSOPInstance

- gdcmm::UIDs, [820](#)
- ProcedureLogStorage
 - gdcmm::UIDs, [822](#)
- Process
 - gdcmm::Parser, [592](#)
- ProcessDataSet
 - gdcmm::FileExplicitFilter, [389](#)
- ProcessPublicTag
 - gdcmm::Scanner, [688](#)
 - gdcmm::StrictScanner, [753](#)
- ProcessRequest
 - vtkGDCMImageReader2, [916](#)
- ProduceCharacterSetDataElement
 - gdcmm::QueryFactory, [655](#)
- ProduceQuery
 - gdcmm::QueryFactory, [655](#)
- ProductCharacteristicsQuerySOPClass
 - gdcmm::UIDs, [824](#)
- ProgressEvent
 - gdcmm::ProgressEvent, [648](#)
- PropertyCategory
 - gdcmm::Segment, [692](#)
- PropertyType
 - gdcmm::Segment, [692](#)
- PseudoColorSoftcopyPresentationStateStorageSOP↔
 - Class
 - gdcmm::UIDs, [822](#)
- PullPrintRequestSOPClassRetired
 - gdcmm::UIDs, [821](#)
- PullStoredPrintManagementMetaSOPClassRetired
 - gdcmm::UIDs, [821](#)
- Push
 - gdcmm::TagPath, [795](#)
- PushBackFile
 - vtkGDCMMedicalImageProperties, [925](#)
- PythonFilter
 - gdcmm::PythonFilter, [651](#)
- Quality
 - gdcmm::JPEGCodec, [498](#)
- QueryFactory
 - gdcmm::BaseQuery, [208](#)
 - gdcmm::BaseRootQuery, [211](#)
 - gdcmm::FindPatientRootQuery, [410](#)
 - gdcmm::FindStudyRootQuery, [413](#)
 - gdcmm::ModalityPerformedProcedureStepCreate↔
 - Query, [535](#)
 - gdcmm::ModalityPerformedProcedureStepSetQuery,
 - [537](#)
 - gdcmm::MovePatientRootQuery, [547](#)
 - gdcmm::MoveStudyRootQuery, [549](#)
 - gdcmm::WLMFindQuery, [967](#)
- RAWCodec
 - gdcmm::RAWCodec, [665](#)
- README.txt, [1259](#)
- RED
 - gdcmm::LookupTable, [510](#)
- RFC2557MIMEencapsulation
 - gdcmm::UIDs, [819](#)
- RGB
 - gdcmm::PhotometricInterpretation, [607](#)
- RGB2YBR
 - gdcmm::ImageChangePhotometricInterpretation, [433](#)
- RGBPixelsToRGBPlanes
 - gdcmm::ImageChangePlanarConfiguration, [436](#)
- RGBPlanesToRGBPixels
 - gdcmm::ImageChangePlanarConfiguration, [436](#)
- RGBToRecommendedDisplayCIELab
 - gdcmm::SurfaceHelper, [770](#)
- RGBToRecommendedDisplayGrayscale
 - gdcmm::SurfaceHelper, [771](#)
- RLE_COMPRESSION
 - vtkGDCMImageWriter, [921](#)
- RLECodec
 - gdcmm::RLECodec, [679](#)
- RLELossless
 - gdcmm::TransferSyntax, [805](#)
 - gdcmm::UIDs, [819](#)
- ROI
 - gdcmm::Overlay, [585](#)
- RTBeamsDeliveryInstructionStorageSupplement74↔
 - FrozenDraft
 - gdcmm::UIDs, [823](#)
- RTBeamsTreatmentRecordStorage
 - gdcmm::UIDs, [823](#)
- RTBrachyTreatmentRecordStorage
 - gdcmm::UIDs, [823](#)
- RTConventionalMachineVerificationSupplement74↔
 - FrozenDraft
 - gdcmm::UIDs, [823](#)
- RTDoseStorage
 - gdcmm::MediaStorage, [522](#)
 - gdcmm::UIDs, [823](#)
- RTImageStorage
 - gdcmm::MediaStorage, [522](#)
 - gdcmm::UIDs, [823](#)
- RTIonBeamsTreatmentRecordStorage
 - gdcmm::MediaStorage, [523](#)
 - gdcmm::UIDs, [823](#)
- RTIonMachineVerificationSupplement74FrozenDraft
 - gdcmm::UIDs, [823](#)
- RTIonPlanStorage
 - gdcmm::MediaStorage, [523](#)
 - gdcmm::UIDs, [823](#)
- RTPlanStorage
 - gdcmm::MediaStorage, [522](#)
 - gdcmm::UIDs, [823](#)
- RTStructSetProperties

- vtkGDCMPolyDataReader, 928
 - vtkGDCMPolyDataWriter, 931
- RTStructureSetStorage
 - gdcm::MediaStorage, 522
 - gdcm::UIDs, 823
- RTTreatmentSummaryRecordStorage
 - gdcm::MediaStorage, 523
 - gdcm::UIDs, 823
- RawDataStorage
 - gdcm::MediaStorage, 522
 - gdcm::UIDs, 822
- Read
 - gdcm::BasicOffsetTable, 215
 - gdcm::ByteValue, 235, 236
 - gdcm::CP246ExplicitDataElement, 266
 - gdcm::CSAHeader, 279
 - gdcm::CommandDataSet, 259
 - gdcm::DataElement, 295
 - gdcm::DataSet, 307
 - gdcm::Element, 344
 - gdcm::Element< TVR, VM::VM1_n >, 348
 - gdcm::EncodingImplementation< VR::VRASCII >, 360
 - gdcm::EncodingImplementation< VR::VRBINARY >, 361
 - gdcm::ExplicitDataElement, 371
 - gdcm::ExplicitImplicitDataElement, 373
 - gdcm::File, 376
 - gdcm::FileMetaInformation, 393
 - gdcm::Fragment, 415
 - gdcm::ImageReader, 457
 - gdcm::ImageRegionReader, 460
 - gdcm::ImplicitDataElement, 469
 - gdcm::Item, 482
 - gdcm::PGXCodec, 606
 - gdcm::PNMCodec, 627
 - gdcm::PixmapReader, 619
 - gdcm::Preamble, 629
 - gdcm::Reader, 669
 - gdcm::SegmentReader, 697
 - gdcm::SequenceOfFragments, 704
 - gdcm::SequenceOfItems, 710
 - gdcm::StreamImageReader, 742
 - gdcm::SurfaceReader, 773
 - gdcm::TableReader, 786
 - gdcm::Tag, 792
 - gdcm::UNExplicitDataElement, 878
 - gdcm::UNExplicitImplicitDataElement, 880
 - gdcm::VL, 893
 - gdcm::VR, 903
 - gdcm::VR16ExplicitDataElement, 904
 - gdcm::VRVLSIZE< 0 >, 906
 - gdcm::VRVLSIZE< 1 >, 906
 - gdcm::ValueIO, 890
 - gdcm::network::AAAbortPDU, 146
 - gdcm::network::AAAssociateACPDU, 149
 - gdcm::network::AAAssociateRJPDU, 151
 - gdcm::network::AAAssociateRQPDU, 154
 - gdcm::network::AReleaseRPPDU, 170
 - gdcm::network::AReleaseRQPDU, 172
 - gdcm::network::AbstractSyntax, 157
 - gdcm::network::ApplicationContext, 167
 - gdcm::network::AsynchronousOperationsWindow↔ Sub, 175
 - gdcm::network::BasePDU, 205
 - gdcm::network::ImplementationClassUIDSub, 466
 - gdcm::network::ImplementationVersionNameSub, 467
 - gdcm::network::MaximumLengthSub, 516
 - gdcm::network::PDataTFPDU, 594
 - gdcm::network::PresentationContextAC, 633
 - gdcm::network::PresentationContextRQ, 637
 - gdcm::network::PresentationDataValue, 638
 - gdcm::network::RoleSelectionSub, 682
 - gdcm::network::SOPClassExtendedNegociationSub, 730
 - gdcm::network::ServiceClassApplicationInformation, 714
 - gdcm::network::TransferSyntaxSub, 807
 - gdcm::network::UserInformation, 885
- Read16
 - gdcm::VL, 893
- ReadACRNEMAImage
 - gdcm::ImageReader, 458
 - gdcm::PixmapReader, 619
- ReadBacktrack
 - gdcm::Fragment, 415
- ReadCompat
 - gdcm::FileMetaInformation, 393
- ReadCompatInternal
 - gdcm::FileMetaInformation, 393
- ReadComputeLength
 - gdcm::EncodingImplementation< VR::VRASCII >, 360
 - gdcm::EncodingImplementation< VR::VRBINARY >, 361
- ReadDataSet
 - gdcm::Reader, 670
- ReadFiles
 - vtkGDCMThreadedImageReader, 936
- ReadFromCommaSeparatedString
 - gdcm::PrivateTag, 646
 - gdcm::Tag, 792
- ReadFromContinuousString
 - gdcm::Tag, 792
- ReadFromPipeSeparatedString
 - gdcm::Tag, 792
- ReadImage

- gdcm::ImageReader, [458](#)
- gdcm::PixmapReader, [619](#)
- ReadImageInformation
 - gdcm::StreamImageReader, [742](#)
- ReadImageInternal
 - gdcm::PixmapReader, [619](#)
- ReadInformation
 - gdcm::ImageRegionReader, [460](#)
- ReadInto
 - gdcm::network::PDataTFPDU, [594](#)
 - gdcm::network::PresentationDataValue, [638](#)
- ReadIntoBuffer
 - gdcm::ImageRegionReader, [460](#)
- ReadMetaInformation
 - gdcm::Reader, [670](#)
- ReadNested
 - gdcm::DataSet, [307](#)
- ReadNoSwap
 - gdcm::EncodingImplementation< VR::VRASCII >, [361](#)
 - gdcm::EncodingImplementation< VR::VRBINARY >, [361](#)
- ReadOrSkip
 - gdcm::DataElement, [296](#)
- ReadPointMacro
 - gdcm::SurfaceReader, [773](#)
- ReadPreValue
 - gdcm::CP246ExplicitDataElement, [267](#)
 - gdcm::DataElement, [296](#)
 - gdcm::ExplicitDataElement, [371](#)
 - gdcm::ExplicitImplicitDataElement, [373](#)
 - gdcm::Fragment, [415](#)
 - gdcm::ImplicitDataElement, [469](#)
 - gdcm::SequenceOfFragments, [704](#)
 - gdcm::UNExplicitDataElement, [879](#)
 - gdcm::UNExplicitImplicitDataElement, [881](#)
 - gdcm::VR16ExplicitDataElement, [905](#)
- ReadPreamble
 - gdcm::Reader, [670](#)
- ReadSegment
 - gdcm::SegmentReader, [697](#)
- ReadSegments
 - gdcm::SegmentReader, [697](#)
- ReadSelectedPrivateTags
 - gdcm::DataSet, [307](#)
 - gdcm::Reader, [670](#)
- ReadSelectedPrivateTagsWithLength
 - gdcm::DataSet, [307](#)
- ReadSelectedTags
 - gdcm::DataSet, [307](#)
 - gdcm::Reader, [670](#)
- ReadSelectedTagsWithLength
 - gdcm::DataSet, [307](#)
- ReadSurface
 - gdcm::SurfaceReader, [773](#)
- ReadSurfaces
 - gdcm::SurfaceReader, [773](#)
- ReadUpToTag
 - gdcm::DataSet, [307](#)
 - gdcm::Reader, [670](#)
- ReadUpToTagWithLength
 - gdcm::DataSet, [307](#)
- ReadVM
 - gdcm::DictConverter, [324](#)
- ReadVR
 - gdcm::DictConverter, [324](#)
- ReadValue
 - gdcm::CP246ExplicitDataElement, [267](#)
 - gdcm::DataElement, [296](#)
 - gdcm::ExplicitDataElement, [371](#)
 - gdcm::ExplicitImplicitDataElement, [373](#)
 - gdcm::Fragment, [415](#)
 - gdcm::ImplicitDataElement, [469](#)
 - gdcm::SequenceOfFragments, [704](#)
 - gdcm::UNExplicitDataElement, [879](#)
 - gdcm::UNExplicitImplicitDataElement, [881](#)
 - gdcm::VR16ExplicitDataElement, [905](#)
- ReadValueWithLength
 - gdcm::DataElement, [296](#)
 - gdcm::ImplicitDataElement, [469](#)
- ReadWithLength
 - gdcm::CP246ExplicitDataElement, [267](#)
 - gdcm::DataElement, [296](#)
 - gdcm::DataSet, [307](#)
 - gdcm::ExplicitDataElement, [371](#)
 - gdcm::ExplicitImplicitDataElement, [373](#)
 - gdcm::ImplicitDataElement, [469](#)
 - gdcm::UNExplicitDataElement, [879](#)
 - gdcm::VR16ExplicitDataElement, [905](#)
- Reader
 - gdcm::Reader, [669](#)
- Readuint16
 - gdcm::DictConverter, [324](#)
- RealWorldValueIntercept
 - gdcm::RealWorldValueMappingContent, [672](#)
- RealWorldValueMappingStorage
 - gdcm::UIDs, [822](#)
- RealWorldValueSlope
 - gdcm::RealWorldValueMappingContent, [672](#)
- RecommendedDisplayCIELabToRGB
 - gdcm::SurfaceHelper, [770](#)
- RecurseDataSet
 - gdcm::Anonymizer, [163](#)
- red
 - gdcm::terminal, [143](#)
- reference
 - gdcm::CodeString, [254](#)
 - gdcm::LO, [507](#)

- gdcM::String, [756](#)
- ReferenceFrameOfReferenceUID
 - vtkRTStructSetProperties, [963](#)
- ReferenceSeriesInstanceUID
 - vtkRTStructSetProperties, [963](#)
- ReferencedColorPrintManagementMetaSOPClassRetired
 - gdcM::UIDs, [820](#)
- ReferencedGrayscalePrintManagementMetaSOPClassRetired
 - Retired
 - gdcM::UIDs, [820](#)
- ReferencedImageBoxSOPClassRetired
 - gdcM::UIDs, [820](#)
- Region
 - gdcM::Region, [673](#)
- Register
 - gdcM::Object, [572](#)
- Remove
 - gdcM::Anonymizer, [163](#)
 - gdcM::DataSet, [307](#)
 - gdcM::FileAnonymizer, [379](#)
 - gdcM::Preamble, [629](#)
- RemoveAllObservers
 - gdcM::Subject, [762](#)
- RemoveDictEntry
 - gdcM::PrivateDict, [643](#)
- RemoveFile
 - gdcM::System, [782](#)
- RemoveGroupLength
 - gdcM::Anonymizer, [163](#)
- RemoveItemByIndex
 - gdcM::SequenceOfItems, [710](#)
- RemoveObserver
 - gdcM::Subject, [762](#)
- RemoveOverlay
 - gdcM::Pixmap, [616](#)
- RemovePrivateTags
 - gdcM::Anonymizer, [163](#)
- RemoveRetired
 - gdcM::Anonymizer, [164](#)
- Render
 - vtkImageColorViewer, [944](#)
- RenderWindow
 - vtkImageColorViewer, [947](#)
- Renderer
 - vtkImageColorViewer, [947](#)
- Replace
 - gdcM::Anonymizer, [164](#)
 - gdcM::CommandDataSet, [259](#)
 - gdcM::DataSet, [307](#)
 - gdcM::FileAnonymizer, [379](#)
 - gdcM::FileMetaInformation, [393](#)
- ReplaceEmpty
 - gdcM::DataSet, [308](#)
- RequestData
 - vtkGDCMImageReader2, [916](#)
 - vtkGDCMPolyDataReader, [927](#)
 - vtkImageMapToColors16, [949](#)
 - vtkImageMapToWindowLevelColors2, [952](#)
 - vtkImagePlanarComponentsToComponents, [954](#)
- RequestData_HemodynamicWaveformStorage
 - vtkGDCMPolyDataReader, [927](#)
- RequestData_RTStructureSetStorage
 - vtkGDCMPolyDataReader, [927](#)
- RequestDataCompat
 - vtkGDCMImageReader, [910](#)
 - vtkGDCMImageReader2, [916](#)
 - vtkGDCMThreadedImageReader, [936](#)
- RequestInformation
 - vtkGDCMImageReader2, [916](#)
 - vtkGDCMPolyDataReader, [928](#)
 - vtkGDCMThreadedImageReader2, [938](#)
 - vtkImageMapToColors16, [949](#)
 - vtkImageMapToWindowLevelColors2, [952](#)
- RequestInformation_HemodynamicWaveformStorage
 - vtkGDCMPolyDataReader, [928](#)
- RequestInformation_RTStructureSetStorage
 - vtkGDCMPolyDataReader, [928](#)
- RequestInformationCompat
 - vtkGDCMImageReader, [910](#)
 - vtkGDCMImageReader2, [916](#)
- RequestPaddedCompositePixelCode
 - gdcM::ImageCodec, [447](#)
- RequestPlanarConfiguration
 - gdcM::ImageCodec, [447](#)
- Rescale
 - gdcM::Rescaler, [676](#)
- RescaleFunctionIntoBestFit
 - gdcM::Rescaler, [676](#)
- Rescaler
 - gdcM::Rescaler, [676](#)
- ReserveDataElement
 - gdcM::FileStreamer, [405](#)
- ReserveGroupDataElement
 - gdcM::FileStreamer, [406](#)
- reset
 - gdcM::terminal, [143](#)
- ResetHandledDataSet
 - gdcM::network::ULConnectionCallback, [868](#)
- RetrieveSOPInstanceUIDFromIndex
 - gdcM::DirectoryHelper, [339](#)
- RetrieveSOPInstanceUIDFromZPosition
 - gdcM::DirectoryHelper, [339](#)
- reverse
 - gdcM::terminal, [143](#)
- reverse_iterator
 - gdcM::CodeString, [254](#)
 - gdcM::LO, [507](#)
 - gdcM::String, [756](#)

- RoleSelectionSub
 - gdcm::network::RoleSelectionSub, [682](#)
- RunEventLoop
 - gdcm::network::ULConnectionManager, [872](#)
- RunMoveEventLoop
 - gdcm::network::ULConnectionManager, [872](#)
- SAGITTAL
 - gdcm::Orientation, [581](#)
- SH
 - gdcm::VR, [901](#)
- SHA1
 - gdcm::SHA1, [721](#)
- SHComp
 - gdcm, [129](#)
- SIEMENS
 - gdcm::Dicts, [330](#)
- SINGLEBIT
 - gdcm::PixelFormat, [611](#)
- SL
 - gdcm::VR, [901](#)
- SLICE_ORIENTATION_XY
 - vtkImageColorViewer, [943](#)
- SLICE_ORIENTATION_XZ
 - vtkImageColorViewer, [943](#)
- SLICE_ORIENTATION_YZ
 - vtkImageColorViewer, [943](#)
- SOPClassExtendedNegociationSub
 - gdcm::network::SOPClassExtendedNegociationSub, [730](#)
- SOPInstanceUID
 - vtkRTStructSetProperties, [964](#)
- SPM2AVG152PDFFrameofReference
 - gdcm::UIDs, [819](#)
- SPM2AVG152T1FrameofReference
 - gdcm::UIDs, [819](#)
- SPM2AVG152T2FrameofReference
 - gdcm::UIDs, [819](#)
- SPM2AVG305T1FrameofReference
 - gdcm::UIDs, [819](#)
- SPM2BRAINMASKFrameofReference
 - gdcm::UIDs, [819](#)
- SPM2CSFFFrameofReference
 - gdcm::UIDs, [819](#)
- SPM2EPIFrameofReference
 - gdcm::UIDs, [819](#)
- SPM2FILT1FrameofReference
 - gdcm::UIDs, [819](#)
- SPM2GRAYFrameofReference
 - gdcm::UIDs, [819](#)
- SPM2PDFFrameofReference
 - gdcm::UIDs, [819](#)
- SPM2PETFrameofReference
 - gdcm::UIDs, [819](#)
- SPM2SINGLESUBJT1FrameofReference
 - gdcm::UIDs, [819](#)
- SPM2SPECTFrameofReference
 - gdcm::UIDs, [819](#)
- SPM2T1FrameofReference
 - gdcm::UIDs, [819](#)
- SPM2T2FrameofReference
 - gdcm::UIDs, [819](#)
- SPM2TRANSMFrameofReference
 - gdcm::UIDs, [819](#)
- SPM2WHITEFrameofReference
 - gdcm::UIDs, [819](#)
- SQ
 - gdcm::VR, [901](#)
- SS
 - gdcm::VR, [901](#)
- ST
 - gdcm::VR, [901](#)
- STATES
 - gdcm::Surface, [765](#)
- STATES_END
 - gdcm::Surface, [765](#)
- STComp
 - gdcm, [129](#)
- SURFACE
 - gdcm::Surface, [766](#)
- SV10
 - gdcm::CSAHeader, [277](#)
- ScalarType
 - gdcm::PixelFormat, [610](#)
- Scale
 - vtkGDCMImageReader, [913](#)
 - vtkGDCMImageReader2, [919](#)
- Scan
 - gdcm::Scanner, [688](#)
 - gdcm::StrictScanner, [753](#)
- Scanner
 - gdcm::Scanner, [686](#)
- SecondaryCaptureImageStorage
 - gdcm::MediaStorage, [521](#)
 - gdcm::UIDs, [821](#)
- Segment
 - gdcm::Segment, [691](#)
- SegmentAlgorithmName
 - gdcm::Segment, [692](#)
- SegmentAlgorithmType
 - gdcm::Segment, [692](#)
- SegmentDescription
 - gdcm::Segment, [692](#)
- SegmentLabel
 - gdcm::Segment, [692](#)
- SegmentMap
 - gdcm::SegmentReader, [697](#)
- SegmentNumber

- gdcmm::Segment, 693
- SegmentReader
 - gdcmm::SegmentReader, 697
- SegmentVector
 - gdcmm::SegmentReader, 697
 - gdcmm::SegmentWriter, 699
- SegmentWriter
 - gdcmm::SegmentWriter, 699
- Segmentation
 - gdcmm::MediaStorage, 523
- SegmentationStorage
 - gdcmm::MediaStorage, 523
 - gdcmm::UIDs, 822
- SegmentedPaletteColorLookupTable
 - gdcmm::SegmentedPaletteColorLookupTable, 694
- Segments
 - gdcmm::SegmentReader, 697
 - gdcmm::SegmentWriter, 700
- Selection
 - gdcmm::Sorter, 735
- SelectionMap
 - gdcmm::Sorter, 733
- Self
 - gdcmm::AnonymizeEvent, 159
 - gdcmm::DataEvent, 300
 - gdcmm::DataSetEvent, 310
 - gdcmm::FileNameEvent, 399
 - gdcmm::MemberCommand, 527
 - gdcmm::ProgressEvent, 648
 - gdcmm::SimpleMemberCommand, 723
- SendEcho
 - gdcmm::ServiceClassUser, 718
 - gdcmm::network::ULConnectionManager, 872
- SendFind
 - gdcmm::ServiceClassUser, 718
 - gdcmm::network::ULConnectionManager, 872
- SendMove
 - gdcmm::ServiceClassUser, 718
 - gdcmm::network::ULConnectionManager, 872
- SendNAction
 - gdcmm::network::ULConnectionManager, 872, 873
- SendNCreate
 - gdcmm::network::ULConnectionManager, 873
- SendNDelete
 - gdcmm::network::ULConnectionManager, 873
- SendNEventReport
 - gdcmm::network::ULConnectionManager, 873
- SendNGet
 - gdcmm::network::ULConnectionManager, 873
- SendNSet
 - gdcmm::network::ULConnectionManager, 873
- SendStore
 - gdcmm::ServiceClassUser, 718, 719
 - gdcmm::network::ULConnectionManager, 873
- Separator
 - gdcmm::ApplicationEntity, 168
 - gdcmm::PersonName, 604
- SequenceLengthField
 - gdcmm::SequenceOfItems, 710
- SequenceOfFragments
 - gdcmm::SequenceOfFragments, 702
- SequenceOfItems
 - gdcmm::SequenceOfItems, 708
- SerieHelper
 - gdcmm::SerieHelper, 712
- SerieRestrictions
 - gdcmm::SerieHelper, 712
- Series
 - gdcmm::Series, 714
- SeriesInstanceUID
 - vtkRTStructSetProperties, 964
- ServiceClassApplicationInformation
 - gdcmm::network::ServiceClassApplicationInformation, 714
- ServiceClassUser
 - gdcmm::ServiceClassUser, 717
- Set
 - gdcmm::Attribute, 179
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 183
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, 189
 - gdcmm::Element, 344
 - gdcmm::Element< TVR, VM::VM1_n >, 348
- SetAETitle
 - gdcmm::ServiceClassUser, 719
- SetAbstractSyntax
 - gdcmm::PresentationContext, 631
 - gdcmm::network::PresentationContextRQ, 637
- SetAlgorithmFamily
 - gdcmm::Surface, 768
- SetAlgorithmName
 - gdcmm::Surface, 768
- SetAlgorithmVersion
 - gdcmm::Surface, 768
- SetAnatomicRegion
 - gdcmm::Segment, 692
- SetArray
 - gdcmm::Element< TVR, VM::VM1_n >, 348
- SetAxisOfRotation
 - gdcmm::Surface, 768
- SetBitPosition
 - gdcmm::Overlay, 587
- SetBitSample
 - gdcmm::JPEGCodec, 498
- SetBitsAllocated
 - gdcmm::Overlay, 587
 - gdcmm::PixelFormat, 613

- SetBitsStored
 - gdcm::PixelFormat, [613](#)
- SetBlob
 - gdcm::ApplicationEntity, [168](#)
 - gdcm::PersonName, [603](#)
 - gdcm::network::PresentationDataValue, [639](#)
- SetBlueLUT
 - gdcm::LookupTable, [512](#)
- SetBufferLength
 - gdcm::JPEGLSCodec, [502](#)
 - gdcm::PNMCodec, [628](#)
 - gdcm::RLECodec, [681](#)
- SetByteSwapTag
 - gdcm::ByteSwapFilter, [231](#)
- SetByteValue
 - gdcm::Attribute, [179](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [183](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, [189](#)
 - gdcm::CSAElement, [274](#)
 - gdcm::DataElement, [296](#)
- SetByteValueNoSwap
 - gdcm::Attribute, [179](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [183](#)
- SetCallbackFunction
 - gdcm::MemberCommand, [528](#)
 - gdcm::SimpleMemberCommand, [724](#)
- SetCalledAETitle
 - gdcm::ServiceClassUser, [719](#)
 - gdcm::network::AAssociateACPDU, [149](#)
 - gdcm::network::AAssociateRQPDU, [154](#)
- SetCallingAETitle
 - gdcm::network::AAssociateACPDU, [149](#)
 - gdcm::network::AAssociateRQPDU, [154](#)
- SetCenterOfRotation
 - gdcm::Surface, [768](#)
- SetChangePrivateTags
 - gdcm::FileExplicitFilter, [389](#)
- SetCheckFileMetaInformation
 - gdcm::Writer, [970](#)
- SetCipherType
 - gdcm::CAPICryptographicMessageSyntax, [239](#)
 - gdcm::CryptographicMessageSyntax, [270](#)
 - gdcm::OpenSSLCryptographicMessageSyntax, [576](#)
 - gdcm::OpenSSL7CryptographicMessageSyntax, [580](#)
- SetColor
 - gdcm::Printer, [642](#)
- SetColorLevel
 - vtkImageColorViewer, [944](#)
- SetColorWindow
 - vtkImageColorViewer, [944](#)
- SetColumns
 - gdcm::Bitmap, [221](#)
 - gdcm::Overlay, [587](#)
- SetCommand
 - gdcm::network::PresentationDataValue, [639](#)
- SetComponents
 - gdcm::PersonName, [603](#)
- SetCompressIconImage
 - gdcm::ImageChangeTransferSyntax, [439](#)
- SetComputeZSpacing
 - gdcm::IPPSorter, [478](#)
- SetCoordinateStartValue
 - gdcm::Curve, [289](#)
- SetCoordinateStepValue
 - gdcm::Curve, [289](#)
- SetCryptographicMessageSyntax
 - gdcm::Anonymizer, [164](#)
- SetCurve
 - gdcm::Curve, [289](#)
 - vtkGDCMImageReader, [910](#)
 - vtkGDCMImageReader2, [917](#)
- SetCurveDataDescriptor
 - gdcm::Curve, [289](#)
- SetCurveDescription
 - gdcm::Curve, [289](#)
- SetData
 - gdcm::DataEvent, [301](#)
- SetDataElement
 - gdcm::Bitmap, [221](#)
- SetDataSet
 - gdcm::File, [376](#)
 - gdcm::network::PresentationDataValue, [639](#)
- SetDataSetTransferSyntax
 - gdcm::FileMetaInformation, [394](#)
- SetDataValueRepresentation
 - gdcm::Curve, [289](#)
- SetDebug
 - gdcm::Trace, [801](#)
- SetDebugStream
 - gdcm::Trace, [801](#)
- SetDefaultTransferSyntax
 - gdcm::PresentationContextGenerator, [635](#)
- SetDerivationCodeSequenceCodeValue
 - gdcm::FileDerivation, [387](#)
- SetDerivationDescription
 - gdcm::FileDerivation, [387](#)
- SetDescription
 - gdcm::CSAHeaderDictEntry, [282](#)
 - gdcm::ModuleEntry, [543](#)
 - gdcm::Overlay, [587](#)
- SetDescriptor
 - gdcm::DICOMDIRGenerator, [320](#)
- SetDictName
 - gdcm::DictConverter, [324](#)

- SetDicts
 - gdcm::PythonFilter, [652](#)
 - gdcm::StringFilter, [758](#)
- SetDimension
 - gdcm::Bitmap, [221](#)
- SetDimensions
 - gdcm::Bitmap, [222](#)
 - gdcm::Curve, [289](#)
 - gdcm::ImageCodec, [445](#)
- SetDimensionsValue
 - gdcm::ImageHelper, [454](#)
- SetDirectionCosines
 - gdcm::Image, [428](#)
 - vtkGDCMImageWriter, [922](#)
- SetDirectionCosinesFromImageOrientationPatient
 - vtkGDCMImageWriter, [922](#)
- SetDirectionCosinesTolerance
 - gdcm::IPPSorter, [478](#)
- SetDirectionCosinesValue
 - gdcm::ImageHelper, [454](#)
- SetDirectory
 - gdcm::SerieHelper, [713](#)
 - gdcm::network::ULWritingCallback, [877](#)
- SetDisplayId
 - vtkImageColorViewer, [944](#)
- SetDomain
 - gdcm::BoxRegion, [228](#)
- SetDropDuplicatePositions
 - gdcm::IPPSorter, [478](#)
- SetElement
 - gdcm::Tag, [792](#)
- SetElementHandler
 - gdcm::Parser, [592](#)
- SetElementTag
 - gdcm::Tag, [792](#), [793](#)
- SetElementXX
 - gdcm::DictEntry, [327](#)
- SetError
 - gdcm::Trace, [801](#)
- SetErrorStream
 - gdcm::Trace, [801](#)
- SetEvent
 - gdcm::network::ULEvent, [874](#)
- SetFile
 - gdcm::Anonymizer, [164](#)
 - gdcm::DICOMDIRGenerator, [320](#)
 - gdcm::FileDecompressLookupTable, [385](#)
 - gdcm::FileDerivation, [387](#)
 - gdcm::FileExplicitFilter, [389](#)
 - gdcm::IconImageFilter, [421](#)
 - gdcm::Printer, [642](#)
 - gdcm::PythonFilter, [652](#)
 - gdcm::Reader, [670](#)
 - gdcm::SplitMosaicFilter, [738](#)
 - gdcm::StreamImageWriter, [746](#)
 - gdcm::StringFilter, [758](#)
 - gdcm::Validate, [887](#)
 - gdcm::Writer, [970](#)
 - gdcm::XMLPrinter, [975](#)
- SetFileName
 - gdcm::FileNameEvent, [399](#)
 - gdcm::Reader, [670](#)
 - gdcm::StreamImageReader, [742](#)
 - gdcm::StreamImageWriter, [746](#)
 - gdcm::Writer, [971](#)
 - vtkGDCMThreadedImageReader2, [938](#)
- SetFileNames
 - vtkGDCMImageReader, [910](#)
 - vtkGDCMImageWriter, [922](#)
 - vtkGDCMThreadedImageReader2, [938](#)
- SetFilePattern
 - vtkGDCMImageReader, [911](#)
 - vtkGDCMImageReader2, [917](#)
- SetFilePrefix
 - vtkGDCMImageReader, [911](#)
 - vtkGDCMImageReader2, [917](#)
- SetFilename
 - gdcm::TableReader, [786](#)
- SetFilenames
 - gdcm::DICOMDIRGenerator, [320](#)
- SetFiles
 - gdcm::FileSet, [403](#)
- SetFiniteVolume
 - gdcm::Surface, [768](#)
- SetForce
 - gdcm::ImageChangeTransferSyntax, [440](#)
 - gdcm::ImageFragmentSplitter, [451](#)
- SetForcePixelSpacing
 - gdcm::ImageHelper, [454](#)
- SetForceRescaleInterceptSlope
 - gdcm::ImageHelper, [454](#)
- SetFragmentSizeMax
 - gdcm::ImageFragmentSplitter, [451](#)
- SetFrameOrigin
 - gdcm::Overlay, [588](#)
- SetFromDataElement
 - gdcm::Attribute, [179](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [183](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, [189](#)
 - gdcm::Element, [344](#)
 - gdcm::Element< TVR, VM::VM1_n >, [348](#)
- SetFromDataSet
 - gdcm::Attribute, [179](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [183](#)

- gdcmm::Attribute< Group, Element, TVR, VM::VM1_n
>, 189
- gdcmm::MediaStorage, 525
- SetFromFile
 - gdcmm::MediaStorage, 525
- SetFromHeader
 - gdcmm::MediaStorage, 525
- SetFromModality
 - gdcmm::MediaStorage, 525
- SetFromSourceImageSequence
 - gdcmm::MediaStorage, 525
- SetFromString
 - gdcmm::DirectionCosines, 335
- SetFromUID
 - gdcmm::UIDs, 832
- SetGreenLUT
 - gdcmm::LookupTable, 512
- SetGroup
 - gdcmm::Curve, 289
 - gdcmm::Overlay, 588
 - gdcmm::Tag, 793
- SetGroupXX
 - gdcmm::DictEntry, 327
- SetHeader
 - gdcmm::File, 377
- SetHighBit
 - gdcmm::PixelFormat, 613
- SetHostname
 - gdcmm::ServiceClassUser, 719
- SetIE
 - gdcmm::IODEntry, 474
- SetIconImage
 - gdcmm::Pixmap, 616
- SetImage
 - gdcmm::PixmapWriter, 624
 - gdcmm::SplitMosaicFilter, 738
- SetImplementationClassUID
 - gdcmm::FileMetaInformation, 394
- SetImplementationVersionName
 - gdcmm::FileMetaInformation, 394
- SetImplicitFlag
 - gdcmm::network::ULConnectionCallback, 868
- SetInput
 - gdcmm::BitmapToBitmapFilter, 225
 - gdcmm::ImageConverter, 448
 - vtkImageColorViewer, 944
- SetInputConnection
 - vtkImageColorViewer, 944
- SetInputFileName
 - gdcmm::DictConverter, 324
 - gdcmm::FileAnonymizer, 379
 - gdcmm::FileChangeTransferSyntax, 382
- SetIntercept
 - gdcmm::Image, 428
- gdcmm::Rescaler, 676
- SetKey
 - gdcmm::CSAElement, 274
- SetKeyword
 - gdcmm::DictEntry, 327
- SetLUT
 - gdcmm::Bitmap, 222
 - gdcmm::ImageCodec, 446
 - gdcmm::LookupTable, 512
 - gdcmm::SegmentedPaletteColorLookupTable, 694
- SetLastElement
 - gdcmm::ParseException, 590
- SetLastFragment
 - gdcmm::network::PresentationDataValue, 639
- SetLength
 - gdcmm::ByteValue, 236
 - gdcmm::Element< TVR, VM::VM1_2 >, 346
 - gdcmm::Element< TVR, VM::VM1_n >, 348
 - gdcmm::Element< TVR, VM::VM2_2n >, 350
 - gdcmm::Element< TVR, VM::VM2_n >, 351
 - gdcmm::Element< TVR, VM::VM3_3n >, 353
 - gdcmm::Element< TVR, VM::VM3_n >, 354
 - gdcmm::RLECodec, 681
 - gdcmm::SequenceOfFragments, 704
 - gdcmm::SequenceOfItems, 710
 - gdcmm::Value, 889
- SetLengthOnly
 - gdcmm::ByteValue, 236
 - gdcmm::Value, 890
- SetLengthToUndefined
 - gdcmm::SequenceOfItems, 710
- SetLoadMode
 - gdcmm::SerieHelper, 713
- SetLookupTable
 - vtkImageMapToColors16, 949
- SetLossless
 - gdcmm::JPEGCodec, 498
 - gdcmm::JPEGLSCCodec, 502
- SetLossyError
 - gdcmm::JPEGLSCCodec, 502
- SetLossyFlag
 - gdcmm::Bitmap, 222
 - gdcmm::ImageCodec, 446
 - gdcmm::PVRGCodec, 651
- SetManifold
 - gdcmm::Surface, 768
- SetMaxPDULength
 - gdcmm::network::ULConnectionInfo, 869
- SetMaxPDUSize
 - gdcmm::network::ULConnection, 866
- SetMaximumLength
 - gdcmm::network::MaximumLengthSub, 516
- SetMaximumPointDistance
 - gdcmm::Surface, 768

- SetMeanPointDistance
 - gdcm::Surface, 768
- SetMedicalImageProperties
 - vtkGDCMImageReader, 911
 - vtkGDCMImageReader2, 917
 - vtkGDCMImageWriter, 922
 - vtkGDCMPolyDataWriter, 930
- SetMergeModeToAbstractSyntax
 - gdcm::PresentationContextGenerator, 635
- SetMergeModeToTransferSyntax
 - gdcm::PresentationContextGenerator, 635
- SetMeshPrimitive
 - gdcm::Surface, 768
- SetMessageHeader
 - gdcm::network::PresentationDataValue, 639
- SetMinMaxForPixelType
 - gdcm::Rescaler, 676
- SetName
 - gdcm::CSAElement, 274
 - gdcm::CSAHeaderDictEntry, 283
 - gdcm::DictEntry, 327
 - gdcm::IODEntry, 474
 - gdcm::Macro, 514
 - gdcm::Module, 540
 - gdcm::ModuleEntry, 543
 - gdcm::PDBelement, 596
 - gdcm::network::AbstractSyntax, 157
 - gdcm::network::ApplicationContext, 167
 - gdcm::network::TransferSyntaxSub, 807
- SetNameFromUID
 - gdcm::network::AbstractSyntax, 157
 - gdcm::network::TransferSyntaxSub, 807
- SetNeedByteSwap
 - gdcm::Bitmap, 222
 - gdcm::ImageCodec, 446
- SetNeedOverlayCleanup
 - gdcm::ImageCodec, 446
- SetNestedDataSet
 - gdcm::Item, 482
- SetNoOfItems
 - gdcm::CSAElement, 274
- SetNoSwap
 - gdcm::Element, 344
 - gdcm::Element< TVR, VM::VM1_n >, 348
- SetNumberOfCurves
 - gdcm::Pixmap, 616
- SetNumberOfDimensions
 - gdcm::Bitmap, 222
 - gdcm::ImageCodec, 446
- SetNumberOfFilenames
 - gdcm::FilenameGenerator, 401
- SetNumberOfFrames
 - gdcm::Overlay, 588
- SetNumberOfInputPorts
 - vtkGDCMPolyDataWriter, 930
- SetNumberOfItems
 - gdcm::SequenceOfItems, 710
- SetNumberOfOverlays
 - gdcm::Pixmap, 616
- SetNumberOfPoints
 - gdcm::Curve, 289
- SetNumberOfResolutions
 - gdcm::JPEG2000Codec, 491
- SetNumberOfSegments
 - gdcm::SegmentWriter, 699
- SetNumberOfSurfacePoints
 - gdcm::Surface, 768
- SetNumberOfSurfaces
 - gdcm::SurfaceWriter, 775
- SetNumberOfTableValues
 - vtkLookupTable16, 959
- SetNumberOfValues
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 189
- SetNumberOfVectors
 - gdcm::Surface, 768
- SetObliquityThresholdCosineValue
 - gdcm::Orientation, 582
- SetOffScreenRendering
 - vtkImageColorViewer, 945
- SetOrigin
 - gdcm::Image, 428
 - gdcm::Overlay, 588
- SetOriginValue
 - gdcm::ImageHelper, 454
- SetOutputDimensions
 - gdcm::IconImageGenerator, 423
- SetOutputFileName
 - gdcm::DictConverter, 324
 - gdcm::FileAnonymizer, 380
 - gdcm::FileChangeTransferSyntax, 383
 - gdcm::FileStreamer, 406
- SetOutputFormatToLuminance
 - vtkImageMapToColors16, 949
- SetOutputFormatToLuminanceAlpha
 - vtkImageMapToColors16, 949
- SetOutputFormatToRGB
 - vtkImageMapToColors16, 949
- SetOutputFormatToRGBA
 - vtkImageMapToColors16, 949
- SetOutputType
 - gdcm::DictConverter, 324
- SetOutsideValuePixel
 - gdcm::IconImageGenerator, 423
- SetOverlay
 - gdcm::Overlay, 588
- SetOverlayVisibility
 - vtkImageColorViewer, 945

- SetOwner
 - gdcm::PrivateTag, [646](#)
- SetPDU
 - gdcm::network::ULEvent, [874](#)
- SetParentId
 - vtkImageColorViewer, [945](#)
- SetPassword
 - gdcm::CAPICryptographicMessageSyntax, [239](#)
 - gdcm::CryptographicMessageSyntax, [270](#)
 - gdcm::OpenSSLCryptographicMessageSyntax, [576](#)
 - gdcm::OpenSSL7CryptographicMessageSyntax, [580](#)
- SetPattern
 - gdcm::FilenameGenerator, [401](#)
- SetPermissions
 - gdcm::System, [782](#)
- SetPhotometricInterpretation
 - gdcm::Bitmap, [222](#)
 - gdcm::ImageChangePhotometricInterpretation, [433](#)
 - gdcm::ImageCodec, [446](#)
- SetPixelFormat
 - gdcm::Bitmap, [222](#)
 - gdcm::ImageCodec, [446](#)
 - gdcm::JPEGCodec, [498](#)
 - gdcm::Rescaler, [677](#)
- SetPixelMinMax
 - gdcm::IconImageGenerator, [424](#)
- SetPixelRepresentation
 - gdcm::PixelFormat, [613](#)
- SetPixmap
 - gdcm::FileDecompressLookupTable, [385](#)
 - gdcm::IconImageGenerator, [424](#)
 - gdcm::PixmapWriter, [625](#)
- SetPlanarConfiguration
 - gdcm::Bitmap, [222](#)
 - gdcm::ImageChangePlanarConfiguration, [436](#)
 - gdcm::ImageCodec, [446](#)
- SetPointCoordinatesData
 - gdcm::Surface, [768](#)
- SetPointPositionAccuracy
 - gdcm::Surface, [768](#)
- SetPointsBoundingBoxCoordinates
 - gdcm::Surface, [768](#)
- SetPort
 - gdcm::ServiceClassUser, [719](#)
- SetPortSCP
 - gdcm::ServiceClassUser, [719](#)
- SetPosition
 - vtkImageColorViewer, [945](#)
- SetPreamble
 - gdcm::FileMetaInformation, [394](#)
- SetPrefix
 - gdcm::FilenameGenerator, [401](#)
- SetPresentationContextID
 - gdcm::PresentationContext, [631](#)
 - gdcm::network::PresentationContextAC, [633](#)
 - gdcm::network::PresentationContextRQ, [637](#)
 - gdcm::network::PresentationDataValue, [639](#)
- SetPresentationContexts
 - gdcm::ServiceClassUser, [719](#)
 - gdcm::network::ULConnection, [866](#)
- SetPrettyPrint
 - gdcm::JSON, [503](#)
- SetPrimitiveData
 - gdcm::MeshPrimitive, [532](#)
- SetPrimitiveType
 - gdcm::MeshPrimitive, [532](#)
- SetPrimitivesData
 - gdcm::MeshPrimitive, [532](#)
- SetPrivateCreator
 - gdcm::Tag, [793](#)
- SetProcessingAlgorithm
 - gdcm::Surface, [768](#)
- SetProgress
 - gdcm::ProgressEvent, [648](#)
- SetPropertyCategory
 - gdcm::Segment, [692](#)
- SetPropertyType
 - gdcm::Segment, [692](#)
- SetPurposeOfReferenceCodeSequenceCodeValue
 - gdcm::FileDerivation, [387](#)
- SetQuality
 - gdcm::JPEG2000Codec, [491](#)
 - gdcm::JPEGCodec, [498](#)
- SetRTStructSetProperties
 - vtkGDCMPolyDataWriter, [931](#)
- SetRate
 - gdcm::JPEG2000Codec, [491](#)
- SetReason
 - gdcm::network::AAAbortPDU, [146](#)
 - gdcm::network::PresentationContextAC, [633](#)
- SetRecommendedDisplayCIELabValue
 - gdcm::Surface, [768](#)
- SetRecommendedDisplayGrayscaleValue
 - gdcm::Surface, [768](#)
- SetRecommendedPresentationOpacity
 - gdcm::Surface, [768](#)
- SetRecommendedPresentationType
 - gdcm::Surface, [768](#)
- SetRecomputeItemLength
 - gdcm::FileExplicitFilter, [389](#)
- SetRecomputeSequenceLength
 - gdcm::FileExplicitFilter, [389](#)
- SetRedLUT
 - gdcm::LookupTable, [512](#)
- SetRef
 - gdcm::IODEntry, [474](#)
- SetRegion

- gdcm::ImageRegionReader, 461
- SetRenderWindow
 - vtkImageColorViewer, 945
- SetRenderer
 - vtkImageColorViewer, 945
- SetRescaleInterceptSlopeValue
 - gdcm::ImageHelper, 454
- SetRetired
 - gdcm::DictEntry, 327
- SetReversible
 - gdcm::JPEG2000Codec, 491
- SetRoot
 - gdcm::UIDGenerator, 813
- SetRootDirectory
 - gdcm::DICOMDIRGenerator, 320
- SetRows
 - gdcm::Bitmap, 222
 - gdcm::Overlay, 588
- SetSOPInstanceUID
 - gdcm::BaseQuery, 208
- SetSamplesPerPixel
 - gdcm::PixelFormat, 613
- SetScalarType
 - gdcm::PixelFormat, 613
- SetSearchParameter
 - gdcm::BaseQuery, 208
- SetSegmentAlgorithmName
 - gdcm::Segment, 692
- SetSegmentAlgorithmType
 - gdcm::Segment, 692
- SetSegmentDescription
 - gdcm::Segment, 692
- SetSegmentLabel
 - gdcm::Segment, 692
- SetSegmentNumber
 - gdcm::Segment, 692
- SetSegments
 - gdcm::SegmentWriter, 699
- SetSize
 - vtkImageColorViewer, 945
- SetSlice
 - vtkImageColorViewer, 945
- SetSliceOrientation
 - vtkImageColorViewer, 945
- SetSliceOrientationToXY
 - vtkImageColorViewer, 945
- SetSliceOrientationToXZ
 - vtkImageColorViewer, 945
- SetSliceOrientationToYZ
 - vtkImageColorViewer, 945
- SetSlope
 - gdcm::Image, 428
 - gdcm::Rescaler, 677
- SetSortFunction
 - gdcm::Sorter, 734
- SetSource
 - gdcm::network::AAAbortPDU, 147
- SetSourceApplicationEntityTitle
 - gdcm::FileMetaInformation, 394
- SetSpacing
 - gdcm::Image, 428
- SetSpacingValue
 - gdcm::ImageHelper, 454
- SetState
 - gdcm::network::ULConnection, 866
- SetStream
 - gdcm::Reader, 671
 - gdcm::StreamImageReader, 743
 - gdcm::StreamImageWriter, 746
 - gdcm::Trace, 802
 - gdcm::Writer, 971
- SetStreamToFile
 - gdcm::Trace, 802
- SetStyle
 - gdcm::Printer, 642
 - gdcm::XMLPrinter, 975
- SetSurfaceComments
 - gdcm::Surface, 768
- SetSurfaceCount
 - gdcm::Segment, 692
- SetSurfaceNumber
 - gdcm::Surface, 769
- SetSurfaceProcessing
 - gdcm::Surface, 769
- SetSurfaceProcessingDescription
 - gdcm::Surface, 769
- SetSurfaceProcessingRatio
 - gdcm::Surface, 769
- SetSyngoDT
 - gdcm::CSAElement, 274
- SetTag
 - gdcm::AnonymizeEvent, 159
 - gdcm::DataElement, 296
- SetTargetPixelType
 - gdcm::Rescaler, 677
- SetTemplateFileName
 - gdcm::FileStreamer, 406
- SetTileSize
 - gdcm::JPEG2000Codec, 491
- SetTimeout
 - gdcm::ServiceClassUser, 719
 - gdcm::network::ARTIMTimer, 173
- SetToUndefined
 - gdcm::VL, 893
- SetTransferSyntax
 - gdcm::Bitmap, 223
 - gdcm::FileChangeTransferSyntax, 383
 - gdcm::ImageChangeTransferSyntax, 440

- gdcmm::network::PresentationContextAC, [633](#)
- SetTuple
 - gdcmm::network::RoleSelectionSub, [682](#)
 - gdcmm::network::SOPClassExtendedNegociationSub, [730](#)
 - gdcmm::network::ServiceClassApplicationInformation, [714](#)
- SetType
 - gdcmm::ModuleEntry, [543](#)
 - gdcmm::Overlay, [588](#)
- SetTypeOfData
 - gdcmm::Curve, [289](#)
- SetUsage
 - gdcmm::IODEntry, [474](#)
- SetUseSeriesDetails
 - gdcmm::SerieHelper, [713](#)
- SetUseTargetPixelType
 - gdcmm::Rescaler, [677](#)
- SetUseVRUN
 - gdcmm::FileExplicitFilter, [389](#)
- SetUserCodec
 - gdcmm::ImageChangeTransferSyntax, [440](#)
- SetUserData
 - gdcmm::Parser, [592](#)
- SetUserInformation
 - gdcmm::network::AAAssociateRQPDU, [154](#)
- SetVL
 - gdcmm::DataElement, [297](#)
- SetVLToUndefined
 - gdcmm::DataElement, [297](#)
- SetVM
 - gdcmm::CSAElement, [274](#)
 - gdcmm::CSAHeaderDictEntry, [283](#)
 - gdcmm::DictEntry, [327](#)
- SetVR
 - gdcmm::CSAElement, [274](#)
 - gdcmm::CSAHeaderDictEntry, [283](#)
 - gdcmm::DataElement, [297](#)
 - gdcmm::DictEntry, [327](#)
- SetValue
 - gdcmm::Attribute, [179](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [184](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, [190](#)
 - gdcmm::CSAElement, [274](#)
 - gdcmm::DataElement, [296](#)
 - gdcmm::Element, [344](#)
 - gdcmm::Element< TVR, VM::VM1_n >, [348](#)
 - gdcmm::PDBelement, [596](#)
- SetValueFieldLength
 - gdcmm::DataElement, [297](#)
- SetValues
 - gdcmm::Attribute, [180](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, [190](#)
- SetVectorAccuracy
 - gdcmm::Surface, [769](#)
- SetVectorCoordinateData
 - gdcmm::Surface, [769](#)
- SetVectorDimensionality
 - gdcmm::Surface, [769](#)
- SetWarning
 - gdcmm::Trace, [802](#)
- SetWarningStream
 - gdcmm::Trace, [802](#)
- SetWindowId
 - vtkImageColorViewer, [946](#)
- SetWriteDataSetOnly
 - gdcmm::Writer, [971](#)
- SetZSpacingTolerance
 - gdcmm::IPPSorter, [478](#)
- setattribute
 - gdcmm::terminal, [143](#)
- setbgcolor
 - gdcmm::terminal, [143](#)
- setfgcolor
 - gdcmm::terminal, [143](#)
- setmode
 - gdcmm::terminal, [143](#)
- SetupInteractor
 - vtkImageColorViewer, [945](#)
- Shift
 - vtkGDCMImageReader, [913](#)
 - vtkGDCMImageReader2, [919](#)
- ShiftEnd
 - gdcmm::ByteBuffer, [229](#)
- ShowAbort
 - gdcmm::SimpleSubjectWatcher, [726](#)
- ShowAnonymization
 - gdcmm::SimpleSubjectWatcher, [726](#)
- ShowData
 - gdcmm::SimpleSubjectWatcher, [726](#)
- ShowDataSet
 - gdcmm::SimpleSubjectWatcher, [726](#)
- ShowFileName
 - gdcmm::SimpleSubjectWatcher, [726](#)
- ShowIteration
 - gdcmm::SimpleSubjectWatcher, [726](#)
- ShowProgress
 - gdcmm::SimpleSubjectWatcher, [726](#)
- SimpleMemberCommand
 - gdcmm::SimpleMemberCommand, [724](#)
- SimpleSubjectWatcher
 - gdcmm::SimpleSubjectWatcher, [725](#)
- SingleSerieUIDFileSetHT
 - gdcmm::SerieHelper, [713](#)
- SingleSerieUIDFileSetmap

- gdcm::SerieHelper, [712](#)
- Size
 - gdcm::CodeString, [255](#)
 - gdcm::DataSet, [308](#)
 - gdcm::GroupDict, [419](#)
 - gdcm::network::AAAbortPDU, [147](#)
 - gdcm::network::AAAssociateACPDU, [149](#)
 - gdcm::network::AAAssociateRJPDU, [151](#)
 - gdcm::network::AAAssociateRQPDU, [155](#)
 - gdcm::network::AReleaseRPPDU, [170](#)
 - gdcm::network::AReleaseRQPDU, [172](#)
 - gdcm::network::AbstractSyntax, [157](#)
 - gdcm::network::ApplicationContext, [167](#)
 - gdcm::network::AsynchronousOperationsWindow↔
 - Sub, [175](#)
 - gdcm::network::BasePDU, [205](#)
 - gdcm::network::ImplementationClassUIDSub, [466](#)
 - gdcm::network::ImplementationVersionNameSub, [467](#)
 - gdcm::network::MaximumLengthSub, [517](#)
 - gdcm::network::PDataTFPDU, [595](#)
 - gdcm::network::PresentationContextAC, [633](#)
 - gdcm::network::PresentationContextRQ, [637](#)
 - gdcm::network::PresentationDataValue, [639](#)
 - gdcm::network::RoleSelectionSub, [682](#)
 - gdcm::network::SOPClassExtendedNegociationSub, [730](#)
 - gdcm::network::ServiceClassApplicationInformation, [715](#)
 - gdcm::network::TransferSyntaxSub, [807](#)
 - gdcm::network::UserInformation, [885](#)
- size_type
 - gdcm::CodeString, [254](#)
 - gdcm::LO, [507](#)
 - gdcm::String, [756](#)
- SizeType
 - gdcm::DataSet, [304](#)
 - gdcm::FilenameGenerator, [400](#)
 - gdcm::IOD, [471](#)
 - gdcm::NestedModuleEntries, [559](#)
 - gdcm::PresentationContext, [631](#)
 - gdcm::PresentationContextGenerator, [634](#)
 - gdcm::SequenceOfFragments, [702](#)
 - gdcm::SequenceOfItems, [708](#)
 - gdcm::network::AAAssociateACPDU, [149](#)
 - gdcm::network::AAAssociateRQPDU, [153](#)
 - gdcm::network::PDataTFPDU, [594](#)
 - gdcm::network::PresentationContextRQ, [636](#)
- Slice
 - vtkImageColorViewer, [947](#)
- SliceOrientation
 - vtkImageColorViewer, [947](#)
- SmartPointer
 - gdcm::Object, [573](#)
 - gdcm::SmartPointer, [728](#)
- Sort
 - gdcm::IPPSorter, [479](#)
 - gdcm::Sorter, [734](#)
- SortFunc
 - gdcm::Sorter, [735](#)
- SortFunction
 - gdcm::Sorter, [733](#)
- Sorter
 - gdcm::Sorter, [734](#)
- SpacialFiducialsStorage
 - gdcm::MediaStorage, [522](#)
- SpacialRegistrationStorage
 - gdcm::MediaStorage, [522](#)
- Spacing
 - gdcm::Spacing, [736](#)
- SpacingType
 - gdcm::Spacing, [736](#)
- SpatialFiducialsStorage
 - gdcm::UIDs, [822](#)
- SpatialRegistrationStorage
 - gdcm::UIDs, [822](#)
- Spectroscopy
 - gdcm::Spectroscopy, [737](#)
- Split
 - gdcm::ImageFragmentSplitter, [451](#)
 - gdcm::SplitMosaicFilter, [738](#)
- SplitExtent
 - vtkGDCMThreadedImageReader2, [938](#)
- SplitMosaicFilter
 - gdcm::SplitMosaicFilter, [738](#)
- Squeeze
 - gdcm::ApplicationEntity, [168](#)
- StableSort
 - gdcm::Sorter, [734](#)
- StandaloneCurveStorage
 - gdcm::MediaStorage, [522](#)
- StandaloneCurveStorageRetired
 - gdcm::UIDs, [821](#)
- StandaloneModalityLUTStorage
 - gdcm::MediaStorage, [522](#)
- StandaloneModalityLUTStorageRetired
 - gdcm::UIDs, [822](#)
- StandaloneOverlayStorage
 - gdcm::MediaStorage, [522](#)
- StandaloneOverlayStorageRetired
 - gdcm::UIDs, [821](#)
- StandalonePETCurveStorageRetired
 - gdcm::UIDs, [823](#)
- StandaloneVOILUTStorage
 - gdcm::MediaStorage, [522](#)
- StandaloneVOILUTStorageRetired
 - gdcm::UIDs, [822](#)
- Start

- gdcm::network::ARTIMTimer, [173](#)
- StartAssociation
 - gdcm::ServiceClassUser, [720](#)
- StartDataElement
 - gdcm::FileStreamer, [406](#)
- StartElement
 - gdcm::TableReader, [786](#)
 - gdcm::XMLDictReader, [973](#)
 - gdcm::XMLPrivateDictReader, [977](#)
- StartElementHandler
 - gdcm::Parser, [591](#)
- StartEncode
 - gdcm::ImageCodec, [446](#)
 - gdcm::JPEG2000Codec, [491](#)
 - gdcm::JPEGCodec, [498](#)
 - gdcm::JPEGLSCodec, [502](#)
 - gdcm::RLECodec, [681](#)
- StartFilter
 - gdcm::SimpleSubjectWatcher, [726](#)
- StartGroupDataElement
 - gdcm::FileStreamer, [406](#)
- StereometricRelationshipStorage
 - gdcm::UIDs, [822](#)
- Stop
 - gdcm::network::ARTIMTimer, [173](#)
- StopAssociation
 - gdcm::ServiceClassUser, [720](#)
- StopDataElement
 - gdcm::FileStreamer, [406](#)
- StopEncode
 - gdcm::ImageCodec, [446](#)
 - gdcm::JPEG2000Codec, [491](#)
 - gdcm::JPEGCodec, [498](#)
 - gdcm::JPEGLSCodec, [502](#)
 - gdcm::RLECodec, [681](#)
- StopGroupDataElement
 - gdcm::FileStreamer, [406](#)
- StopProtocol
 - gdcm::network::ULConnection, [866](#)
- StorageCommitmentPullModelSOPClassRetired
 - gdcm::UIDs, [820](#)
- StorageCommitmentPullModelSOPInstanceRetired
 - gdcm::UIDs, [820](#)
- StorageCommitmentPushModelSOPClass
 - gdcm::UIDs, [820](#)
- StorageCommitmentPushModelSOPInstance
 - gdcm::UIDs, [820](#)
- StorageServiceClass
 - gdcm::UIDs, [820](#)
- StoredPrintStorageSOPClassRetired
 - gdcm::UIDs, [821](#)
- StrCaseCmp
 - gdcm::System, [782](#)
- StrNCaseCmp
 - gdcm::System, [782](#)
- StrSep
 - gdcm::System, [782](#)
- StrTokR
 - gdcm::System, [782](#)
- Stream
 - gdcm::Writer, [971](#)
- StreamImageReader
 - gdcm::Reader, [671](#)
 - gdcm::StreamImageReader, [741](#)
- StreamImageWriter
 - gdcm::StreamImageWriter, [745](#)
 - gdcm::Writer, [971](#)
- StrictScanner
 - gdcm::StrictScanner, [751](#)
- String
 - gdcm::String, [756](#)
- StringFilter
 - gdcm::StringFilter, [758](#)
- StructureSetDate
 - vtkRTStructSetProperties, [964](#)
- StructureSetLabel
 - vtkRTStructSetProperties, [964](#)
- StructureSetName
 - vtkRTStructSetProperties, [964](#)
- StructureSetTime
 - vtkRTStructSetProperties, [964](#)
- Study
 - gdcm::Study, [760](#)
- StudyComponentManagementSOPClass
 - gdcm::MediaStorage, [522](#)
- StudyComponentManagementSOPClassRetired
 - gdcm::UIDs, [820](#)
- StudyInstanceUID
 - vtkRTStructSetProperties, [964](#)
- StudyRootQueryRetrieveInformationModelFIND
 - gdcm::UIDs, [823](#)
- StudyRootQueryRetrieveInformationModelGET
 - gdcm::UIDs, [823](#)
- StudyRootQueryRetrieveInformationModelMOVE
 - gdcm::UIDs, [823](#)
- Subject
 - gdcm::Subject, [761](#)
- SubstanceAdministrationLoggingSOPClass
 - gdcm::UIDs, [820](#)
- SubstanceAdministrationLoggingSOPInstance
 - gdcm::UIDs, [820](#)
- SubstanceApprovalQuerySOPClass
 - gdcm::UIDs, [824](#)
- Superclass
 - gdcm::AnonymizeEvent, [159](#)
 - gdcm::DataEvent, [300](#)
 - gdcm::DataSetEvent, [310](#)
 - gdcm::FileNameEvent, [399](#)

- gdcm::LO, [507](#)
- gdcm::ProgressEvent, [648](#)
- Surface
 - gdcm::Surface, [766](#)
- SurfaceCount
 - gdcm::Segment, [693](#)
- SurfaceReader
 - gdcm::SurfaceReader, [773](#)
- SurfaceSegmentationStorage
 - gdcm::MediaStorage, [523](#)
 - gdcm::UIDs, [825](#)
- SurfaceVector
 - gdcm::Segment, [691](#)
- SurfaceWriter
 - gdcm::SurfaceWriter, [775](#)
- Surfaces
 - gdcm::Segment, [693](#)
- Swap
 - gdcm::ByteSwap, [230](#)
 - gdcm::SwapperDoOp, [777](#)
 - gdcm::SwapperNoOp, [778](#)
- SwapArray
 - gdcm::SwapperDoOp, [777](#)
 - gdcm::SwapperNoOp, [778](#)
- SwapCode
 - gdcm::SwapCode, [777](#)
- SwapCodeType
 - gdcm::SwapCode, [776](#)
- SwapFromSwapCodeIntoSystem
 - gdcm::ByteSwap, [230](#)
- SwapRange
 - gdcm::ByteSwap, [230](#)
- SwapRangeFromSwapCodeIntoSystem
 - gdcm::ByteSwap, [230](#)
- SyngoDTField
 - gdcm::CSAElement, [275](#)
- SyntaxError
 - gdcm::Parser, [591](#)
- SystemIsBigEndian
 - gdcm::ByteSwap, [230](#)
- SystemIsLittleEndian
 - gdcm::ByteSwap, [230](#)
- T1
 - gdcm::Type, [810](#)
- T1C
 - gdcm::Type, [810](#)
- T2
 - gdcm::Type, [810](#)
- T2C
 - gdcm::Type, [810](#)
- T3
 - gdcm::Type, [810](#)
- TConstMemberFunctionPointer
 - gdcm::MemberCommand, [527](#)
- TM
 - gdcm::VR, [901](#)
- TMComp
 - gdcm, [129](#)
- TMemberFunctionPointer
 - gdcm::MemberCommand, [527](#)
 - gdcm::SimpleMemberCommand, [723](#)
- TRIANGLE
 - gdcm::MeshPrimitive, [531](#)
- TRIANGLE_FAN
 - gdcm::MeshPrimitive, [531](#)
- TRIANGLE_STRIP
 - gdcm::MeshPrimitive, [531](#)
- TS
 - gdcm::Bitmap, [224](#)
- TS_END
 - gdcm::TransferSyntax, [805](#)
- TSName
 - gdcm::UIDs, [818](#)
- TSType
 - gdcm::TransferSyntax, [804](#)
 - gdcm::UIDs, [825](#)
- TYPETOENCODING
 - gdcm, [135](#)
 - gdcmVR.h, [1253](#)
- TYPETOLENGTH
 - gdcmVM.h, [1251](#)
- Table
 - gdcm::Table, [783](#)
- Table16
 - vtkLookupTable16, [959](#)
- TableEntry
 - gdcm::TableEntry, [784](#)
- TableReader
 - gdcm::TableReader, [785](#)
- TableRow
 - gdcm::network::TableRow, [787](#)
- Tag
 - gdcm::Tag, [789](#)
- tag
 - gdcm::Tag, [793](#)
- TagField
 - gdcm::DataElement, [298](#)
- TagMismatchError
 - gdcm::Parser, [591](#)
- TagPath
 - gdcm::TagPath, [794](#)
- TagToValue
 - gdcm::Scanner, [686](#)
 - gdcm::StrictScanner, [751](#)
- TagToValueValueType
 - gdcm::Scanner, [686](#)
 - gdcm::StrictScanner, [751](#)

- tags
 - gdcm::Tag, [793](#)
- TalairachBrainAtlasFrameofReference
 - gdcm::UIDs, [819](#)
- TestAbortOff
 - gdcm::SimpleSubjectWatcher, [726](#)
- TestAbortOn
 - gdcm::SimpleSubjectWatcher, [726](#)
- TestPBKDF2
 - gdcm::ASN1, [174](#)
- Testing
 - gdcm::Testing, [796](#)
- TestsList.txt, [1259](#)
- TextSRStorageTrialRetired
 - gdcm::UIDs, [822](#)
- ThreadedExecute
 - vtkImageRGBToYBR, [955](#)
 - vtkImageYBRToRGB, [957](#)
- ThreadedRequestData
 - vtkGDCMThreadedImageReader2, [939](#)
 - vtkImageMapToColors16, [949](#)
 - vtkImageMapToWindowLevelColors2, [952](#)
- to_string
 - gdcm, [135](#)
- ToPyObject
 - gdcm::PythonFilter, [652](#)
- ToString
 - gdcm::StringFilter, [758](#), [759](#)
- ToStringPair
 - gdcm::StringFilter, [759](#)
- ToUnixSlashes
 - gdcm::Filename, [396](#)
- ToWindowsSlashes
 - gdcm::Filename, [397](#)
- ToshibaPrivateDataStorage
 - gdcm::MediaStorage, [522](#)
- Trace
 - gdcm::Trace, [800](#)
- TransferSyntax
 - gdcm::TransferSyntax, [805](#)
- TransferSyntaxArrayType
 - gdcm::PresentationContext, [631](#)
- TransferSyntaxStringsType
 - gdcm::UIDs, [818](#)
- TransferSyntaxSub
 - gdcm::network::TransferSyntaxSub, [807](#)
- TransferSyntaxes
 - gdcm::PresentationContext, [632](#)
- Transition
 - gdcm::network::Transition, [808](#)
- transitions
 - gdcm::network::TableRow, [787](#)
- Trim
 - gdcm::String, [757](#)
- TrimInternal
 - gdcm::CodeString, [255](#)
- Truncate
 - gdcm::String, [757](#)
- TryJPEG2000Codec
 - gdcm::Bitmap, [223](#)
 - gdcm::ImageChangeTransferSyntax, [440](#)
- TryJPEG2000Codec2
 - gdcm::Bitmap, [223](#)
- TryJPEGCodec
 - gdcm::Bitmap, [223](#)
 - gdcm::ImageChangeTransferSyntax, [440](#)
- TryJPEGCodec2
 - gdcm::Bitmap, [223](#)
- TryJPEGLSCodec
 - gdcm::Bitmap, [223](#)
 - gdcm::ImageChangeTransferSyntax, [440](#)
- TryKAKADUCodec
 - gdcm::Bitmap, [223](#)
- TryPVRGCodec
 - gdcm::Bitmap, [223](#)
- TryRAWCodec
 - gdcm::Bitmap, [223](#)
 - gdcm::ImageChangeTransferSyntax, [440](#)
- TryRLECodec
 - gdcm::Bitmap, [223](#)
 - gdcm::ImageChangeTransferSyntax, [440](#)
- Type
 - gdcm::Element, [344](#)
 - gdcm::Element< TVR, VM::VM1_n >, [347](#)
 - gdcm::Type, [810](#)
 - gdcm::VL, [893](#)
- TypeType
 - gdcm::Type, [810](#)
- UI
 - gdcm::VR, [901](#)
- UIComp
 - gdcm, [129](#)
- UIDGenerator
 - gdcm::UIDGenerator, [812](#)
- UINT12
 - gdcm::PixelFormat, [610](#)
- UINT16
 - gdcm::PixelFormat, [611](#)
- UINT32
 - gdcm::PixelFormat, [611](#)
- UINT64
 - gdcm::PixelFormat, [611](#)
- UINT8
 - gdcm::PixelFormat, [610](#)
- UL
 - gdcm::VR, [901](#)
- ULAction

- gdcm::network::ULAction, [834](#)
- ULActionAE6
 - gdcm::network::ULConnection, [866](#)
- ULBasicCallback
 - gdcm::network::ULBasicCallback, [864](#)
- ULConnection
 - gdcm::network::ULConnection, [865](#)
- ULConnectionCallback
 - gdcm::network::ULConnectionCallback, [868](#)
- ULConnectionInfo
 - gdcm::network::ULConnectionInfo, [869](#)
- ULConnectionManager
 - gdcm::network::ULConnection, [867](#)
 - gdcm::network::ULConnectionManager, [872](#)
- ULEvent
 - gdcm::network::ULEvent, [874](#)
- ULTransitionTable
 - gdcm::network::ULTransitionTable, [875](#)
- ULWritingCallback
 - gdcm::network::ULWritingCallback, [876](#)
- UN
 - gdcm::VR, [901](#)
- UNKNOWN
 - gdcm::PhotometricInterpretation, [607](#)
- UNKNOWN
 - gdcm::CSAHeader, [277](#)
 - gdcm::LookupTable, [510](#)
 - gdcm::Orientation, [581](#)
 - gdcm::PixelFormat, [611](#)
 - gdcm::Spacing, [736](#)
 - gdcm::Surface, [765](#)
 - gdcm::Type, [810](#)
- URI
 - gdcm::MediaStorage, [523](#)
- US
 - gdcm::VR, [901](#)
- US_SS
 - gdcm::VR, [901](#)
- US_SS_OW
 - gdcm::VR, [901](#)
- UT
 - gdcm::VR, [901](#)
- UTComp
 - gdcm, [129](#)
- uid_1_2_840_10008_15_0_3_1
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_15_0_3_10
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_15_0_3_11
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_15_0_3_12
 - gdcm::UIDs, [831](#)
- uid_1_2_840_10008_15_0_3_13
 - gdcm::UIDs, [831](#)
- uid_1_2_840_10008_15_0_3_14
 - gdcm::UIDs, [831](#)
- uid_1_2_840_10008_15_0_3_15
 - gdcm::UIDs, [831](#)
- uid_1_2_840_10008_15_0_3_16
 - gdcm::UIDs, [831](#)
- uid_1_2_840_10008_15_0_3_17
 - gdcm::UIDs, [831](#)
- uid_1_2_840_10008_15_0_3_18
 - gdcm::UIDs, [831](#)
- uid_1_2_840_10008_15_0_3_19
 - gdcm::UIDs, [831](#)
- uid_1_2_840_10008_15_0_3_2
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_15_0_3_20
 - gdcm::UIDs, [831](#)
- uid_1_2_840_10008_15_0_3_21
 - gdcm::UIDs, [831](#)
- uid_1_2_840_10008_15_0_3_22
 - gdcm::UIDs, [831](#)
- uid_1_2_840_10008_15_0_3_23
 - gdcm::UIDs, [831](#)
- uid_1_2_840_10008_15_0_3_24
 - gdcm::UIDs, [831](#)
- uid_1_2_840_10008_15_0_3_25
 - gdcm::UIDs, [831](#)
- uid_1_2_840_10008_15_0_3_26
 - gdcm::UIDs, [831](#)
- uid_1_2_840_10008_15_0_3_27
 - gdcm::UIDs, [831](#)
- uid_1_2_840_10008_15_0_3_28
 - gdcm::UIDs, [831](#)
- uid_1_2_840_10008_15_0_3_29
 - gdcm::UIDs, [831](#)
- uid_1_2_840_10008_15_0_3_3
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_15_0_3_30
 - gdcm::UIDs, [831](#)
- uid_1_2_840_10008_15_0_3_31
 - gdcm::UIDs, [831](#)
- uid_1_2_840_10008_15_0_3_4
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_15_0_3_5
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_15_0_3_6
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_15_0_3_7
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_15_0_3_8
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_15_0_3_9
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_15_0_4_1
 - gdcm::UIDs, [831](#)

uid_1_2_840_10008_15_0_4_2
gdcml::UIDs, 831

uid_1_2_840_10008_15_0_4_3
gdcml::UIDs, 831

uid_1_2_840_10008_15_0_4_4
gdcml::UIDs, 831

uid_1_2_840_10008_15_0_4_5
gdcml::UIDs, 831

uid_1_2_840_10008_15_0_4_6
gdcml::UIDs, 831

uid_1_2_840_10008_15_0_4_7
gdcml::UIDs, 831

uid_1_2_840_10008_15_0_4_8
gdcml::UIDs, 831

uid_1_2_840_10008_1_1
gdcml::UIDs, 825

uid_1_2_840_10008_1_2
gdcml::UIDs, 825

uid_1_2_840_10008_1_20_1
gdcml::UIDs, 826

uid_1_2_840_10008_1_20_1_1
gdcml::UIDs, 826

uid_1_2_840_10008_1_20_2
gdcml::UIDs, 826

uid_1_2_840_10008_1_20_2_1
gdcml::UIDs, 826

uid_1_2_840_10008_1_2_1
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_1_99
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_2
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_100
gdcml::UIDs, 826

uid_1_2_840_10008_1_2_4_101
gdcml::UIDs, 831

uid_1_2_840_10008_1_2_4_102
gdcml::UIDs, 831

uid_1_2_840_10008_1_2_4_103
gdcml::UIDs, 831

uid_1_2_840_10008_1_2_4_50
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_51
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_52
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_53
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_54
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_55
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_56
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_57
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_58
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_59
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_60
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_61
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_62
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_63
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_64
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_65
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_66
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_70
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_80
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_81
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_90
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_91
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_92
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_93
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_94
gdcml::UIDs, 825

uid_1_2_840_10008_1_2_4_95
gdcml::UIDs, 826

uid_1_2_840_10008_1_2_5
gdcml::UIDs, 826

uid_1_2_840_10008_1_2_6_1
gdcml::UIDs, 826

uid_1_2_840_10008_1_2_6_2
gdcml::UIDs, 826

uid_1_2_840_10008_1_3_10
gdcml::UIDs, 826

uid_1_2_840_10008_1_40
gdcml::UIDs, 826

uid_1_2_840_10008_1_40_1
gdcml::UIDs, 826

uid_1_2_840_10008_1_42
gdcml::UIDs, 826

uid_1_2_840_10008_1_42_1
gdcml::UIDs, 826

uid_1_2_840_10008_1_4_1_1
gdcml::UIDs, 826

uid_1_2_840_10008_1_4_1_10
gdcml::UIDs, 826

uid_1_2_840_10008_1_4_1_11
gdcml::UIDs, 826

uid_1_2_840_10008_1_4_1_12
gdcml::UIDs, 826

uid_1_2_840_10008_1_4_1_13
gdcml::UIDs, 826

uid_1_2_840_10008_1_4_1_14
gdcml::UIDs, 826

uid_1_2_840_10008_1_4_1_15
gdcml::UIDs, 826

uid_1_2_840_10008_1_4_1_16
gdcml::UIDs, 826

uid_1_2_840_10008_1_4_1_17
gdcml::UIDs, 826

uid_1_2_840_10008_1_4_1_18
gdcml::UIDs, 826

uid_1_2_840_10008_1_4_1_2
gdcml::UIDs, 826

uid_1_2_840_10008_1_4_1_3
gdcml::UIDs, 826

uid_1_2_840_10008_1_4_1_4
gdcml::UIDs, 826

uid_1_2_840_10008_1_4_1_5
gdcml::UIDs, 826

uid_1_2_840_10008_1_4_1_6
gdcml::UIDs, 826

uid_1_2_840_10008_1_4_1_7
gdcml::UIDs, 826

uid_1_2_840_10008_1_4_1_8
gdcml::UIDs, 826

uid_1_2_840_10008_1_4_1_9
gdcml::UIDs, 826

uid_1_2_840_10008_1_4_2_1
gdcml::UIDs, 826

uid_1_2_840_10008_1_4_2_2
gdcml::UIDs, 826

uid_1_2_840_10008_1_9
gdcml::UIDs, 826

uid_1_2_840_10008_2_16_4
gdcml::UIDs, 826

uid_1_2_840_10008_2_6_1
gdcml::UIDs, 826

uid_1_2_840_10008_3_1_1_1
gdcml::UIDs, 826

uid_1_2_840_10008_3_1_2_1_1
gdcml::UIDs, 826

uid_1_2_840_10008_3_1_2_1_4
gdcml::UIDs, 826

uid_1_2_840_10008_3_1_2_2_1
gdcml::UIDs, 826

uid_1_2_840_10008_3_1_2_3_1
gdcml::UIDs, 826

uid_1_2_840_10008_3_1_2_3_2
gdcml::UIDs, 827

uid_1_2_840_10008_3_1_2_3_3
gdcml::UIDs, 827

uid_1_2_840_10008_3_1_2_3_4
gdcml::UIDs, 827

uid_1_2_840_10008_3_1_2_3_5
gdcml::UIDs, 827

uid_1_2_840_10008_3_1_2_5_1
gdcml::UIDs, 827

uid_1_2_840_10008_3_1_2_5_4
gdcml::UIDs, 827

uid_1_2_840_10008_3_1_2_5_5
gdcml::UIDs, 827

uid_1_2_840_10008_3_1_2_6_1
gdcml::UIDs, 827

uid_1_2_840_10008_4_2
gdcml::UIDs, 827

uid_1_2_840_10008_5_1_1_1
gdcml::UIDs, 827

uid_1_2_840_10008_5_1_1_14
gdcml::UIDs, 827

uid_1_2_840_10008_5_1_1_15
gdcml::UIDs, 827

uid_1_2_840_10008_5_1_1_16
gdcml::UIDs, 827

uid_1_2_840_10008_5_1_1_16_376
gdcml::UIDs, 827

uid_1_2_840_10008_5_1_1_17
gdcml::UIDs, 827

uid_1_2_840_10008_5_1_1_17_376
gdcml::UIDs, 827

uid_1_2_840_10008_5_1_1_18
gdcml::UIDs, 827

uid_1_2_840_10008_5_1_1_18_1
gdcml::UIDs, 827

uid_1_2_840_10008_5_1_1_2
gdcml::UIDs, 827

uid_1_2_840_10008_5_1_1_22
gdcml::UIDs, 827

uid_1_2_840_10008_5_1_1_23
gdcml::UIDs, 827

uid_1_2_840_10008_5_1_1_24
gdcml::UIDs, 827

uid_1_2_840_10008_5_1_1_24_1
gdcml::UIDs, 827

uid_1_2_840_10008_5_1_1_25
gdcml::UIDs, 827

uid_1_2_840_10008_5_1_1_26
gdcml::UIDs, 827

uid_1_2_840_10008_5_1_1_27
gdcml::UIDs, 827

uid_1_2_840_10008_5_1_1_29
gdcm::UIDs, [827](#)

uid_1_2_840_10008_5_1_1_30
gdcm::UIDs, [827](#)

uid_1_2_840_10008_5_1_1_31
gdcm::UIDs, [827](#)

uid_1_2_840_10008_5_1_1_32
gdcm::UIDs, [827](#)

uid_1_2_840_10008_5_1_1_33
gdcm::UIDs, [827](#)

uid_1_2_840_10008_5_1_1_4
gdcm::UIDs, [827](#)

uid_1_2_840_10008_5_1_1_4_1
gdcm::UIDs, [827](#)

uid_1_2_840_10008_5_1_1_4_2
gdcm::UIDs, [827](#)

uid_1_2_840_10008_5_1_1_9
gdcm::UIDs, [827](#)

uid_1_2_840_10008_5_1_1_9_1
gdcm::UIDs, [827](#)

uid_1_2_840_10008_5_1_4_1_1_1
gdcm::UIDs, [827](#)

uid_1_2_840_10008_5_1_4_1_1_10
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_104_1
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_104_2
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_11
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_11_1
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_11_2
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_11_3
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_11_4
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_128
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_129
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_12_1
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_12_1_1
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_12_2
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_12_2_1
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_12_3
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_13_1_1
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_13_1_2
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_13_1_3
gdcm::UIDs, [831](#)

uid_1_2_840_10008_5_1_4_1_1_1_1
gdcm::UIDs, [827](#)

uid_1_2_840_10008_5_1_4_1_1_1_1_1
gdcm::UIDs, [827](#)

uid_1_2_840_10008_5_1_4_1_1_1_2
gdcm::UIDs, [827](#)

uid_1_2_840_10008_5_1_4_1_1_1_2_1
gdcm::UIDs, [827](#)

uid_1_2_840_10008_5_1_4_1_1_1_3
gdcm::UIDs, [827](#)

uid_1_2_840_10008_5_1_4_1_1_1_3_1
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_2
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_20
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_2_1
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_3
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_3_1
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_4
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_481_1
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_481_2
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_481_3
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_481_4
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_481_5
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_481_6
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_481_7
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_481_8
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_481_9
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_4_1
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_4_2
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_5
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_6
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_66
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_66_1
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_66_2
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_66_3
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_66_4
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_66_5
gdcm::UIDs, [831](#)

uid_1_2_840_10008_5_1_4_1_1_67
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_6_1
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_6_2
gdcm::UIDs, [831](#)

uid_1_2_840_10008_5_1_4_1_1_7
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_77_1
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_1
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_2
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_2_1
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_3
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_4
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_4_1
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_5_1
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_5_2
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_5_3
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_5_4
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_6
gdcm::UIDs, [831](#)

uid_1_2_840_10008_5_1_4_1_1_77_2
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_7_1
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_7_2
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_7_3
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_7_4
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_8
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_88_1
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_88_11
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_88_2
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_88_22
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_88_3
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_88_33
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_88_4
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_88_40
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_88_50
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_88_59
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_88_65
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_88_67
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_1_9
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_9_1
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_9_1_1
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_9_1_2
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_9_1_3
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_9_2_1
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_9_3_1
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_1_9_4_1
gdcm::UIDs, [828](#)

uid_1_2_840_10008_5_1_4_1_2_1_1
gdcm::UIDs, [829](#)

uid_1_2_840_10008_5_1_4_1_2_1_2
gdcm::UIDs, [830](#)

uid_1_2_840_10008_5_1_4_1_2_1_3
gdcm::UIDs, [830](#)

uid_1_2_840_10008_5_1_4_1_2_2_1
gdcm::UIDs, [830](#)

uid_1_2_840_10008_5_1_4_1_2_2_2
gdcm::UIDs, [830](#)

- uid_1_2_840_10008_5_1_4_1_2_2_3
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_1_2_3_1
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_1_2_3_2
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_1_2_3_3
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_31
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_32
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_32_1
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_32_2
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_32_3
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_33
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_34_1
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_34_2
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_34_3
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_34_4
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_34_4_1
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_34_4_2
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_34_4_3
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_34_4_4
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_34_5
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_37_1
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_37_2
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_37_3
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_38_1
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_38_2
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_38_3
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_41
 - gdcm::UIDs, [830](#)
- uid_1_2_840_10008_5_1_4_42
 - gdcm::UIDs, [830](#)
- UltrasoundImageStorage
 - gdcm::MediaStorage, [521](#)
 - gdcm::UIDs, [821](#)
- UltrasoundImageStorageRetired
 - gdcm::MediaStorage, [521](#)
 - gdcm::UIDs, [821](#)
- UltrasoundMultiFrameImageStorage
 - gdcm::MediaStorage, [521](#)
- UltrasoundMultiFrameImageStorageRetired
 - gdcm::MediaStorage, [521](#)
- UltrasoundMultiframeImageStorage
 - gdcm::UIDs, [821](#)
- UltrasoundMultiframeImageStorageRetired
 - gdcm::UIDs, [821](#)
- UnInstallPipeline
 - vtkImageColorViewer, [946](#)
- UnRegister
 - gdcm::Object, [573](#)
- UndefinedEntityError
 - gdcm::Parser, [592](#)
- underline
 - gdcm::terminal, [143](#)
- UnexpectedStateError
 - gdcm::Parser, [592](#)
- UnifiedProcedureStepEventSOPClass
 - gdcm::UIDs, [823](#)
- UnifiedProcedureStepPullSOPClass
 - gdcm::UIDs, [823](#)
- UnifiedProcedureStepPushSOPClass
 - gdcm::UIDs, [823](#)
- UnifiedProcedureStepWatchSOPClass
 - gdcm::UIDs, [823](#)
- UnifiedWorklistandProcedureStepSOPInstance
 - gdcm::UIDs, [823](#)
- UnifiedWorklistandProcedureStepServiceClass
 - gdcm::UIDs, [823](#)
- Unknown
 - gdcm::SwapCode, [776](#)
 - gdcm::TransferSyntax, [804](#)
- Unpack
 - gdcm::Unpacker12Bits, [881](#)
- Update
 - gdcm::Curve, [289](#)
 - gdcm::Overlay, [588](#)
- UpdateDisplayExtent
 - vtkImageColorViewer, [946](#)
- UpdateOrientation
 - vtkImageColorViewer, [946](#)
- UpdatePosition
 - gdcm::ByteBuffer, [229](#)
- Usage
 - gdcm::Usage, [883](#)
- UsageType
 - gdcm::Usage, [883](#)

UseDictAlways
 gdcmm::PythonFilter, 652
 gdcmm::StringFilter, 759
 UserInformation
 gdcmm::network::UserInformation, 885
 UserOption
 gdcmm::Usage, 883
 UserOrdering
 gdcmm::SerieHelper, 713

 V
 gdcmm::Validate, 888
 VERBOSE_STYLE
 gdcmm::Printer, 641
 VERTEX
 gdcmm::MeshPrimitive, 531
 VIEWType
 gdcmm::Surface, 765
 VIEWType_END
 gdcmm::Surface, 766
 VL
 gdcmm::VL, 893
 VL16
 gdcmm::VR, 901
 VL32
 gdcmm::VR, 901
 VLEndoscopicImageStorage
 gdcmm::MediaStorage, 523
 gdcmm::UIDs, 822
 VLImageStorageTrialRetired
 gdcmm::UIDs, 822
 VLMicroscopicImageStorage
 gdcmm::MediaStorage, 523
 gdcmm::UIDs, 822
 VLMultiframeImageStorageTrialRetired
 gdcmm::UIDs, 822
 VLPhotographicImageStorage
 gdcmm::MediaStorage, 523
 gdcmm::UIDs, 822
 VLSlideCoordinatesMicroscopicImageStorage
 gdcmm::UIDs, 822
 VLWholeSlideMicroscopyImageStorage
 gdcmm::MediaStorage, 523
 gdcmm::UIDs, 825
 VM
 gdcmm::VM, 897
 VM0
 gdcmm::VM, 896
 VM1
 gdcmm::VM, 896
 VM10
 gdcmm::VM, 896
 VM12
 gdcmm::VM, 896
 VM16
 gdcmm::VM, 896
 VM18
 gdcmm::VM, 896
 VM1_2
 gdcmm::VM, 897
 VM1_3
 gdcmm::VM, 897
 VM1_32
 gdcmm::VM, 897
 VM1_4
 gdcmm::VM, 897
 VM1_5
 gdcmm::VM, 897
 VM1_8
 gdcmm::VM, 897
 VM1_99
 gdcmm::VM, 897
 VM1_n
 gdcmm::VM, 897
 VM2
 gdcmm::VM, 896
 VM24
 gdcmm::VM, 896
 VM256
 gdcmm::VM, 897
 VM28
 gdcmm::VM, 896
 VM2_2n
 gdcmm::VM, 897
 VM2_n
 gdcmm::VM, 897
 VM3
 gdcmm::VM, 896
 VM30_30n
 gdcmm::VM, 897
 VM32
 gdcmm::VM, 896
 VM35
 gdcmm::VM, 896
 VM3_3n
 gdcmm::VM, 897
 VM3_4
 gdcmm::VM, 897
 VM3_n
 gdcmm::VM, 897
 VM4
 gdcmm::VM, 896
 VM47_47n
 gdcmm::VM, 897
 VM4_4n
 gdcmm::VM, 897
 VM5
 gdcmm::VM, 896

- VM6
 - gdcm::VM, [896](#)
- VM6_6n
 - gdcm::VM, [897](#)
- VM7_7n
 - gdcm::VM, [897](#)
- VM8
 - gdcm::VM, [896](#)
- VM9
 - gdcm::VM, [896](#)
- VM99
 - gdcm::VM, [897](#)
- VM_END
 - gdcm::VM, [897](#)
- VMType
 - gdcm::Attribute, [177](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [182](#)
 - gdcm::VM, [896](#)
- VOILUTBoxSOPClass
 - gdcm::UIDs, [821](#)
- VR
 - gdcm::VR, [902](#)
- VR_END
 - gdcm::VR, [901](#)
- VR_VM1
 - gdcm::VR, [901](#)
- VRALL
 - gdcm::VR, [901](#)
- VRASCI
 - gdcm::VR, [901](#)
- VRBINARY
 - gdcm, [135](#)
 - gdcm::VR, [901](#)
- VRDS16ILLEGAL
 - gdcmElement.h, [1051](#)
- VRField
 - gdcm::CSAElement, [275](#)
 - gdcm::DataElement, [298](#)
- VRType
 - gdcm::VR, [900](#)
- VRTypeTemplateCase
 - gdcmVR.h, [1253](#)
- VT100
 - gdcm::terminal, [143](#)
- VTK_CMYK
 - vtkGDCMImageReader.h, [1260](#)
 - vtkGDCMImageReader2.h, [1261](#)
- VTK_INVERSE_LUMINANCE
 - vtkGDCMImageReader.h, [1260](#)
 - vtkGDCMImageReader2.h, [1261](#)
- VTK_LEGACY
 - vtkImageColorViewer, [946](#)
- VTK_LOOKUP_TABLE
 - vtkGDCMImageReader.h, [1260](#)
 - vtkGDCMImageReader2.h, [1261](#)
- VTK_YBR
 - vtkGDCMImageReader.h, [1260](#)
 - vtkGDCMImageReader2.h, [1261](#)
- Valid
 - gdcm::Preamble, [629](#)
- ValidDataSet
 - gdcm::BaseQuery, [208](#)
- Validate
 - gdcm::PixelFormat, [613](#)
 - gdcm::Validate, [887](#)
- ValidateQuery
 - gdcm::BaseQuery, [208](#)
 - gdcm::BaseRootQuery, [211](#)
 - gdcm::FindPatientRootQuery, [410](#)
 - gdcm::FindStudyRootQuery, [412](#)
 - gdcm::ModalityPerformedProcedureStepCreateQuery, [535](#)
 - gdcm::ModalityPerformedProcedureStepSetQuery, [537](#)
 - gdcm::MovePatientRootQuery, [546](#)
 - gdcm::MoveStudyRootQuery, [549](#)
 - gdcm::WLMFindQuery, [966](#)
- Validation
 - gdcm::Validate, [888](#)
- Value
 - gdcm::Value, [889](#)
- value
 - gdcm::STATIC_ASSERTION_FAILURE< true >, [740](#)
 - gdcm::SerieHelper::Rule, [683](#)
- value_type
 - gdcm::CodeString, [254](#)
 - gdcm::LO, [507](#)
 - gdcm::String, [756](#)
- ValueField
 - gdcm::DataElement, [298](#)
 - gdcm::PDBelement, [597](#)
- ValueLengthField
 - gdcm::DataElement, [298](#)
- ValueMultiplicityField
 - gdcm::CSAElement, [275](#)
- ValuePtr
 - gdcm::DataElement, [292](#)
- ValueType
 - gdcm::Scanner, [686](#)
 - gdcm::StrictScanner, [751](#)
- VerificationSOPClass
 - gdcm::UIDs, [818](#)
- Verify
 - gdcm::Defs, [315](#)
 - gdcm::Macro, [514](#)
 - gdcm::Module, [540](#)

- Version
 - gdcmm::Version, 891
- Video
 - gdcmm::MediaStorage, 523
- VideoEndoscopicImageStorage
 - gdcmm::MediaStorage, 522
 - gdcmm::UIDs, 822
- VideoMicroscopicImageStorage
 - gdcmm::UIDs, 822
- VideoPhotographicImageStorage
 - gdcmm::MediaStorage, 523
 - gdcmm::UIDs, 822
- vtkBooleanMacro
 - vtkGDCMImageReader, 911
 - vtkGDCMImageReader2, 917
 - vtkGDCMImageWriter, 922
 - vtkGDCMThreadedImageReader, 936
 - vtkGDCMThreadedImageReader2, 939
 - vtkImageColorViewer, 946
 - vtkImageMapToColors16, 949
- vtkGDCMImageReader, 907
 - ~vtkGDCMImageReader, 909
 - ApplyInverseVideo, 912
 - ApplyLookupTable, 912
 - ApplyPlanarConfiguration, 912
 - ApplyShiftScale, 912
 - ApplyYBRToRGB, 912
 - CanReadFile, 910
 - Curve, 912
 - DirectionCosines, 912
 - ExecuteData, 910
 - ExecuteInformation, 910
 - FileNames, 912
 - FillMedicalImageInformation, 910
 - ForceRescale, 912
 - GetDescriptiveName, 910
 - GetFileExtensions, 910
 - GetIconImage, 910
 - GetOverlay, 910
 - IconDataScalarType, 912
 - IconImageDataExtent, 912
 - IconNumberOfScalarComponents, 912
 - ImageFormat, 912
 - ImageOrientationPatient, 912
 - ImagePositionPatient, 912
 - LoadIconImage, 913
 - LoadOverlays, 913
 - LoadSingleFile, 910
 - LossyFlag, 913
 - MedicalImageProperties, 913
 - New, 910
 - NumberOfIconImages, 913
 - NumberOfOverlays, 913
 - PlanarConfiguration, 913
 - PrintSelf, 910
 - RequestDataCompat, 910
 - RequestInformationCompat, 910
 - Scale, 913
 - SetCurve, 910
 - SetFileNames, 910
 - SetFilePattern, 911
 - SetFilePrefix, 911
 - SetMedicalImageProperties, 911
 - Shift, 913
 - vtkBooleanMacro, 911
 - vtkGDCMImageReader, 909
 - vtkGDCMMedicalImageProperties, 925
 - vtkGetMacro, 911
 - vtkGetObjectMacro, 911
 - vtkGetStringMacro, 912
 - vtkGetVector3Macro, 912
 - vtkGetVector6Macro, 912
 - vtkSetMacro, 912
 - vtkSetVector6Macro, 912
 - vtkTypeRevisionMacro, 912
- vtkGDCMImageReader.h, 1259
 - VTK_CMYK, 1260
 - VTK_INVERSE_LUMINANCE, 1260
 - VTK_LOOKUP_TABLE, 1260
 - VTK_YBR, 1260
- vtkGDCMImageReader2, 913
 - ~vtkGDCMImageReader2, 916
 - ApplyInverseVideo, 918
 - ApplyLookupTable, 918
 - ApplyPlanarConfiguration, 918
 - ApplyShiftScale, 918
 - ApplyYBRToRGB, 918
 - CanReadFile, 916
 - Curve, 918
 - DirectionCosines, 918
 - FillMedicalImageInformation, 916
 - ForceRescale, 918
 - GetDescriptiveName, 916
 - GetFileExtensions, 916
 - GetIconImage, 916
 - GetIconImagePort, 916
 - GetOverlay, 916
 - GetOverlayPort, 916
 - IconDataScalarType, 918
 - IconImageDataExtent, 918
 - IconNumberOfScalarComponents, 918
 - ImageFormat, 918
 - ImageOrientationPatient, 918
 - ImagePositionPatient, 918
 - LoadIconImage, 918
 - LoadOverlays, 918
 - LoadSingleFile, 916
 - LossyFlag, 918

- New, 916
- NumberOfIconImages, 918
- NumberOfOverlays, 918
- PlanarConfiguration, 919
- PrintSelf, 916
- ProcessRequest, 916
- RequestData, 916
- RequestDataCompat, 916
- RequestInformation, 916
- RequestInformationCompat, 916
- Scale, 919
- SetCurve, 917
- SetFilePattern, 917
- SetFilePrefix, 917
- SetMedicalImageProperties, 917
- Shift, 919
- vtkBooleanMacro, 917
- vtkGDCMImageReader2, 916
- vtkGDCMMedicalImageProperties, 925
- vtkGetMacro, 917
- vtkGetObjectMacro, 917
- vtkGetStringMacro, 917
- vtkGetVector3Macro, 917
- vtkGetVector6Macro, 917
- vtkSetMacro, 918
- vtkSetVector6Macro, 918
- vtkTypeRevisionMacro, 918
- vtkGDCMImageReader2.h, 1260
 - VTK_CMYK, 1261
 - VTK_INVERSE_LUMINANCE, 1261
 - VTK_LOOKUP_TABLE, 1261
 - VTK_YBR, 1261
- vtkGDCMImageWriter, 919
 - ~vtkGDCMImageWriter, 921
 - CompressionTypes, 921
 - GetDescriptiveName, 921
 - GetFileExtensions, 921
 - GetFileName, 921
 - JPEG2000_COMPRESSION, 921
 - JPEG_COMPRESSION, 921
 - JPEGLS_COMPRESSION, 921
 - NO_COMPRESSION, 921
 - New, 921
 - PrintSelf, 921
 - RLE_COMPRESSION, 921
 - SetDirectionCosines, 922
 - SetDirectionCosinesFromImageOrientationPatient, 922
 - SetFileNames, 922
 - SetMedicalImageProperties, 922
 - vtkBooleanMacro, 922
 - vtkGDCMImageWriter, 921
 - vtkGDCMMedicalImageProperties, 925
 - vtkGetMacro, 922
 - vtkGetObjectMacro, 922
 - vtkGetStringMacro, 922
 - vtkSetMacro, 923
 - vtkSetStringMacro, 923
 - vtkTypeRevisionMacro, 923
 - Write, 923
 - WriteGDCMData, 923
 - WriteSlice, 923
- vtkGDCMImageWriter.h, 1261
- vtkGDCMMedicalImageProperties, 923
 - ~vtkGDCMMedicalImageProperties, 925
 - Clear, 925
 - GetFile, 925
 - New, 925
 - PrintSelf, 925
 - PushBackFile, 925
 - vtkGDCMImageReader, 925
 - vtkGDCMImageReader2, 925
 - vtkGDCMImageWriter, 925
 - vtkGDCMMedicalImageProperties, 925
 - vtkTypeRevisionMacro, 925
- vtkGDCMMedicalImageProperties.h, 1261
- vtkGDCMPolyDataReader, 925
 - ~vtkGDCMPolyDataReader, 927
 - FileName, 928
 - FillMedicalImageInformation, 927
 - MedicalImageProperties, 928
 - New, 927
 - PrintSelf, 927
 - RTStructSetProperties, 928
 - RequestData, 927
 - RequestData_HemodynamicWaveformStorage, 927
 - RequestData_RTStructureSetStorage, 927
 - RequestInformation, 928
 - RequestInformation_HemodynamicWaveform↔Storage, 928
 - RequestInformation_RTStructureSetStorage, 928
 - vtkGDCMPolyDataReader, 927
 - vtkGetObjectMacro, 928
 - vtkGetStringMacro, 928
 - vtkSetStringMacro, 928
 - vtkTypeRevisionMacro, 928
- vtkGDCMPolyDataReader.h, 1262
- vtkGDCMPolyDataWriter, 928
 - ~vtkGDCMPolyDataWriter, 930
 - InitializeRTStructSet, 930
 - MedicalImageProperties, 931
 - New, 930
 - PrintSelf, 930
 - RTStructSetProperties, 931
 - SetMedicalImageProperties, 930
 - SetNumberOfInputPorts, 930
 - SetRTStructSetProperties, 931
 - vtkGDCMPolyDataWriter, 930

- vtkTypeRevisionMacro, 931
 - WriteData, 931
 - WriteRTSTRUCTData, 931
 - WriteRTSTRUCTInfo, 931
- vtkGDCMPolyDataWriter.h, 1263
- vtkGDCMTesting, 931
 - ~vtkGDCMTesting, 933
 - GetGDCMDataRoot, 933
 - GetMD5MetaImage, 933
 - GetMHDMD5FromFile, 933
 - GetNumberOfMD5MetaImages, 933
 - GetRAWMD5FromFile, 933
 - GetVTKDataRoot, 933
 - MD5MetaImagesType, 933
 - New, 933
 - PrintSelf, 933
 - vtkGDCMTesting, 933
 - vtkTypeRevisionMacro, 934
- vtkGDCMTesting.h, 1263
- vtkGDCMThreadedImageReader, 934
 - ~vtkGDCMThreadedImageReader, 935
 - ExecuteData, 936
 - ExecuteInformation, 936
 - New, 936
 - PrintSelf, 936
 - ReadFiles, 936
 - RequestDataCompat, 936
 - vtkBooleanMacro, 936
 - vtkGDCMThreadedImageReader, 935
 - vtkGetMacro, 936
 - vtkSetMacro, 936
 - vtkTypeRevisionMacro, 936
- vtkGDCMThreadedImageReader.h, 1264
- vtkGDCMThreadedImageReader2, 936
 - ~vtkGDCMThreadedImageReader2, 938
 - GetFileName, 938
 - New, 938
 - PrintSelf, 938
 - RequestInformation, 938
 - SetFileName, 938
 - SetFileNames, 938
 - SplitExtent, 938
 - ThreadedRequestData, 939
 - vtkBooleanMacro, 939
 - vtkGDCMThreadedImageReader2, 938
 - vtkGetMacro, 939
 - vtkGetObjectMacro, 939
 - vtkGetVector3Macro, 939
 - vtkGetVector6Macro, 939
 - vtkSetMacro, 939
 - vtkSetVector3Macro, 939
 - vtkSetVector6Macro, 940
 - vtkTypeRevisionMacro, 940
- vtkGDCMThreadedImageReader2.h, 1265
- vtkGetMacro
 - vtkGDCMImageReader, 911
 - vtkGDCMImageReader2, 917
 - vtkGDCMImageWriter, 922
 - vtkGDCMThreadedImageReader, 936
 - vtkGDCMThreadedImageReader2, 939
 - vtkImageColorViewer, 946
 - vtkImageMapToColors16, 949
 - vtkImageMapToWindowLevelColors2, 952
- vtkGetObjectMacro
 - vtkGDCMImageReader, 911
 - vtkGDCMImageReader2, 917
 - vtkGDCMImageWriter, 922
 - vtkGDCMPolyDataReader, 928
 - vtkGDCMThreadedImageReader2, 939
 - vtkImageColorViewer, 946
 - vtkImageMapToColors16, 949
- vtkGetStringMacro
 - vtkGDCMImageReader, 912
 - vtkGDCMImageReader2, 917
 - vtkGDCMImageWriter, 922
 - vtkGDCMPolyDataReader, 928
 - vtkRTStructSetProperties, 963
- vtkGetVector3Macro
 - vtkGDCMImageReader, 912
 - vtkGDCMImageReader2, 917
 - vtkGDCMThreadedImageReader2, 939
- vtkGetVector6Macro
 - vtkGDCMImageReader, 912
 - vtkGDCMImageReader2, 917
 - vtkGDCMThreadedImageReader2, 939
- vtkImageColorViewer, 940
 - ~vtkImageColorViewer, 943
 - AddInput, 943
 - AddInputConnection, 943
 - FirstRender, 946
 - GetColorLevel, 943
 - GetColorWindow, 943
 - GetInput, 943
 - GetOffScreenRendering, 943
 - GetOverlayVisibility, 944
 - GetPosition, 944
 - GetSize, 944
 - GetSliceMax, 944
 - GetSliceMin, 944
 - GetSliceRange, 944
 - GetWindowName, 944
 - ImageActor, 946
 - InstallPipeline, 944
 - Interactor, 946
 - InteractorStyle, 946
 - New, 944
 - OverlayImageActor, 947
 - PrintSelf, 944

- Render, [944](#)
- RenderWindow, [947](#)
- Renderer, [947](#)
- SLICE_ORIENTATION_XY, [943](#)
- SLICE_ORIENTATION_XZ, [943](#)
- SLICE_ORIENTATION_YZ, [943](#)
- SetColorLevel, [944](#)
- SetColorWindow, [944](#)
- SetDisplayId, [944](#)
- SetInput, [944](#)
- SetInputConnection, [944](#)
- SetOffScreenRendering, [945](#)
- SetOverlayVisibility, [945](#)
- SetParentId, [945](#)
- SetPosition, [945](#)
- SetRenderWindow, [945](#)
- SetRenderer, [945](#)
- SetSize, [945](#)
- SetSlice, [945](#)
- SetSliceOrientation, [945](#)
- SetSliceOrientationToXY, [945](#)
- SetSliceOrientationToXZ, [945](#)
- SetSliceOrientationToYZ, [945](#)
- SetWindowId, [946](#)
- SetupInteractor, [945](#)
- Slice, [947](#)
- SliceOrientation, [947](#)
- UnInstallPipeline, [946](#)
- UpdateDisplayExtent, [946](#)
- UpdateOrientation, [946](#)
- VTK_LEGACY, [946](#)
- vtkBooleanMacro, [946](#)
- vtkGetMacro, [946](#)
- vtkGetObjectMacro, [946](#)
- vtkImageColorViewer, [943](#)
- vtkImageColorViewerCallback, [946](#)
- vtkTypeRevisionMacro, [946](#)
- WindowLevel, [947](#)
- vtkImageColorViewer.h, [1265](#)
- vtkImageColorViewerCallback
 - vtkImageColorViewer, [946](#)
- vtkImageMapToColors16, [947](#)
 - ~vtkImageMapToColors16, [949](#)
 - ActiveComponent, [950](#)
 - DataWasPassed, [950](#)
 - GetMTime, [949](#)
 - LookupTable, [950](#)
 - New, [949](#)
 - OutputFormat, [950](#)
 - PassAlphaToOutput, [950](#)
 - PrintSelf, [949](#)
 - RequestData, [949](#)
 - RequestInformation, [949](#)
 - SetLookupTable, [949](#)
 - SetOutputFormatToLuminance, [949](#)
 - SetOutputFormatToLuminanceAlpha, [949](#)
 - SetOutputFormatToRGB, [949](#)
 - SetOutputFormatToRGBA, [949](#)
 - ThreadedRequestData, [949](#)
 - vtkBooleanMacro, [949](#)
 - vtkGetMacro, [949](#)
 - vtkGetObjectMacro, [949](#)
 - vtkImageMapToColors16, [949](#)
 - vtkSetMacro, [949](#), [950](#)
 - vtkTypeRevisionMacro, [950](#)
- vtkImageMapToColors16.h, [1266](#)
- vtkImageMapToWindowLevelColors2, [950](#)
 - ~vtkImageMapToWindowLevelColors2, [952](#)
 - Level, [952](#)
 - New, [952](#)
 - PrintSelf, [952](#)
 - RequestData, [952](#)
 - RequestInformation, [952](#)
 - ThreadedRequestData, [952](#)
 - vtkGetMacro, [952](#)
 - vtkImageMapToWindowLevelColors2, [952](#)
 - vtkSetMacro, [952](#)
 - vtkTypeRevisionMacro, [952](#)
 - Window, [952](#)
- vtkImageMapToWindowLevelColors2.h, [1266](#)
- vtkImagePlanarComponentsToComponents, [952](#)
 - ~vtkImagePlanarComponentsToComponents, [954](#)
 - New, [954](#)
 - PrintSelf, [954](#)
 - RequestData, [954](#)
 - vtkImagePlanarComponentsToComponents, [954](#)
 - vtkTypeRevisionMacro, [954](#)
- vtkImagePlanarComponentsToComponents.h, [1267](#)
- vtkImageRGBToYBR, [954](#)
 - ~vtkImageRGBToYBR, [955](#)
 - New, [955](#)
 - PrintSelf, [955](#)
 - ThreadedExecute, [955](#)
 - vtkImageRGBToYBR, [955](#)
 - vtkTypeRevisionMacro, [955](#)
- vtkImageRGBToYBR.h, [1267](#)
- vtkImageYBRToRGB, [956](#)
 - ~vtkImageYBRToRGB, [957](#)
 - New, [957](#)
 - PrintSelf, [957](#)
 - ThreadedExecute, [957](#)
 - vtkImageYBRToRGB, [957](#)
 - vtkTypeRevisionMacro, [957](#)
- vtkImageYBRToRGB.h, [1268](#)
- vtkLookupTable16, [957](#)
 - ~vtkLookupTable16, [958](#)
 - Build, [958](#)
 - GetPointer, [959](#)

- MapScalarsThroughTable2, 959
- New, 959
- PrintSelf, 959
- SetNumberOfTableValues, 959
- Table16, 959
- vtkLookupTable16, 958
- vtkTypeRevisionMacro, 959
- WritePointer, 959
- vtkLookupTable16.h, 1268
- vtkRTStructSetProperties, 959
 - ~vtkRTStructSetProperties, 961
 - AddContourReferencedFrameOfReference, 961
 - AddReferencedFrameOfReference, 962
 - AddStructureSetROI, 962
 - AddStructureSetROIObservation, 962
 - Clear, 962
 - DeepCopy, 962
 - GetContourReferencedFrameOfReferenceClassUID, 962
 - GetContourReferencedFrameOfReferenceInstanceClassUID, 962
 - GetNumberOfContourReferencedFrameOfReferences, 962
 - GetNumberOfReferencedFrameOfReferences, 962
 - GetNumberOfStructureSetROIs, 962
 - GetReferencedFrameOfReferenceClassUID, 962
 - GetReferencedFrameOfReferenceInstanceUID, 962
 - GetStructureSetObservationNumber, 962
 - GetStructureSetROIDescription, 962
 - GetStructureSetROIGenerationAlgorithm, 962
 - GetStructureSetROIName, 962
 - GetStructureSetROINumber, 962
 - GetStructureSetROIObservationLabel, 962
 - GetStructureSetROIRefFrameRefUID, 962
 - GetStructureSetRTROIInterpretedType, 962
 - Internals, 963
 - New, 962
 - PrintSelf, 963
 - ReferenceFrameOfReferenceUID, 963
 - ReferenceSeriesInstanceUID, 963
 - SOPInstanceUID, 964
 - SeriesInstanceUID, 964
 - StructureSetDate, 964
 - StructureSetLabel, 964
 - StructureSetName, 964
 - StructureSetTime, 964
 - StudyInstanceUID, 964
 - vtkGetStringMacro, 963
 - vtkRTStructSetProperties, 961
 - vtkSetStringMacro, 963
 - vtkTypeRevisionMacro, 963
- vtkRTStructSetProperties.h, 1269
- vtkSetMacro
 - vtkGDCMImageReader, 912
 - vtkGDCMImageReader2, 918
 - vtkGDCMImageWriter, 923
 - vtkGDCMThreadedImageReader, 936
 - vtkGDCMThreadedImageReader2, 939
 - vtkImageMapToColors16, 949, 950
 - vtkImageMapToWindowLevelColors2, 952
- vtkSetStringMacro
 - vtkGDCMImageWriter, 923
 - vtkGDCMPolyDataReader, 928
 - vtkRTStructSetProperties, 963
- vtkSetVector3Macro
 - vtkGDCMThreadedImageReader2, 939
- vtkSetVector6Macro
 - vtkGDCMImageReader, 912
 - vtkGDCMImageReader2, 918
 - vtkGDCMThreadedImageReader2, 940
- vtkTypeRevisionMacro
 - vtkGDCMImageReader, 912
 - vtkGDCMImageReader2, 918
 - vtkGDCMImageWriter, 923
 - vtkGDCMMedicalImageProperties, 925
 - vtkGDCMPolyDataReader, 928
 - vtkGDCMPolyDataWriter, 931
 - vtkGDCMTesting, 934
 - vtkGDCMThreadedImageReader, 936
 - vtkGDCMThreadedImageReader2, 940
 - vtkImageColorViewer, 946
 - vtkImageMapToColors16, 950
 - vtkImageMapToWindowLevelColors2, 952
 - vtkImagePlanarComponentsToComponents, 954
 - vtkImageRGBToYBR, 955
 - vtkImageYBRToRGB, 957
 - vtkLookupTable16, 959
 - vtkRTStructSetProperties, 963
- WIREFRAME
 - gdcm::Surface, 766
- WLMFindQuery
 - gdcm::WLMFindQuery, 966
- WarningOff
 - gdcm::Trace, 802
- WarningOn
 - gdcm::Trace, 802
- Waveform
 - gdcm::MediaStorage, 523
 - gdcm::Waveform, 964
- WaveformStorageTrialRetired
 - gdcm::UIDs, 821
- what
 - gdcm::Exception, 368
- white
 - gdcm::terminal, 143
- Window
 - vtkImageMapToWindowLevelColors2, 952

WindowLevel

vtkImageColorViewer, 947

Write

gdcm::ByteValue, 236

gdcm::CSAHeader, 279

gdcm::CommandDataSet, 259

gdcm::DataElement, 297

gdcm::DataSet, 308

gdcm::Element, 344

gdcm::Element< TVR, VM::VM1_n >, 348

gdcm::EncodingImplementation< VR::VRASCII >, 361

gdcm::EncodingImplementation< VR::VRBINARY >, 362

gdcm::ExplicitDataElement, 371

gdcm::File, 377

gdcm::FileAnonymizer, 380

gdcm::FileMetaInformation, 394

gdcm::Fragment, 415

gdcm::ImageWriter, 465

gdcm::ImplicitDataElement, 469

gdcm::Item, 482

gdcm::PGXCodec, 606

gdcm::PNMCodec, 628

gdcm::PixmapWriter, 625

gdcm::Preamble, 629

gdcm::SegmentWriter, 699

gdcm::SequenceOfFragments, 704

gdcm::SequenceOfItems, 710

gdcm::StreamImageWriter, 746

gdcm::SurfaceWriter, 775

gdcm::Tag, 793

gdcm::VL, 894

gdcm::VR, 903

gdcm::VRVLSize< 0 >, 906

gdcm::VRVLSize< 1 >, 906

gdcm::ValueIO, 890

gdcm::Writer, 971

gdcm::network::AAAbortPDU, 147

gdcm::network::AAssociateACPD, 149

gdcm::network::AAssociateRJPD, 151

gdcm::network::AAssociateRQPD, 155

gdcm::network::AReleaseRPPD, 170

gdcm::network::AReleaseRQPD, 172

gdcm::network::AbstractSyntax, 157

gdcm::network::ApplicationContext, 167

gdcm::network::AsynchronousOperationsWindow← Sub, 175

gdcm::network::BasePDU, 205

gdcm::network::ImplementationClassUIDSub, 466

gdcm::network::ImplementationUIDSub, 467

gdcm::network::ImplementationVersionNameSub, 467

gdcm::network::MaximumLengthSub, 517

gdcm::network::PDataTFPDU, 595

gdcm::network::PresentationContextAC, 633

gdcm::network::PresentationContextRQ, 637

gdcm::network::PresentationDataValue, 639

gdcm::network::RoleSelectionSub, 682

gdcm::network::SOPClassExtendedNegociationSub, 730

gdcm::network::ServiceClassApplicationInformation, 715

gdcm::network::TransferSyntaxSub, 807

gdcm::network::UserInformation, 886

vtkGDCMImageWriter, 923

Write16

gdcm::VL, 894

WriteASCII

gdcm::Element< TVR, VM::VM1_n >, 348

WriteBuffer

gdcm::ByteValue, 236

gdcm::SequenceOfFragments, 704

WriteBufferAsRGBA

gdcm::LookupTable, 512

WriteData

vtkGDCMPolyDataWriter, 931

WriteFooter

gdcm::DictConverter, 324

WriteGDCMData

vtkGDCMImageWriter, 923

WriteHeader

gdcm::DictConverter, 325

WriteHelpFile

gdcm::BaseQuery, 208

WriteImageInformation

gdcm::StreamImageWriter, 747

WriteImageSubregionRAW

gdcm::StreamImageWriter, 747

WritePointer

vtkLookupTable16, 959

WriteQuery

gdcm::BaseQuery, 208

WriteRTSTRUCTData

vtkGDCMPolyDataWriter, 931

WriteRTSTRUCTInfo

vtkGDCMPolyDataWriter, 931

WriteRawHeader

gdcm::StreamImageWriter, 747

WriteSlice

vtkGDCMImageWriter, 923

Writer

gdcm::Writer, 970

XML

gdcm::Printer, 641

XMLDictReader

gdcm::XMLDictReader, 973

- XMLEncoding
 - gdcm::UIDs, [819](#)
- XMLPrinter
 - gdcm::XMLPrinter, [975](#)
- XMLPrivateDictReader
 - gdcm::XMLPrivateDictReader, [977](#)
- XRay3DAngiographicImageStorage
 - gdcm::MediaStorage, [523](#)
 - gdcm::UIDs, [822](#)
- XRay3DCraniofacialImageStorage
 - gdcm::UIDs, [822](#)
- XRayAngiographicBiPlaneImageStorageRetired
 - gdcm::MediaStorage, [522](#)
 - gdcm::UIDs, [822](#)
- XRayAngiographicImageStorage
 - gdcm::MediaStorage, [522](#)
 - gdcm::UIDs, [822](#)
- XRayRadiationDoseSR
 - gdcm::MediaStorage, [523](#)
- XRayRadiationDoseSRStorage
 - gdcm::UIDs, [823](#)
- XRayRadiofluoroscopicImageStorage
 - gdcm::UIDs, [822](#)
- XRayRadiofluoroscopicImageStorage
 - gdcm::MediaStorage, [522](#)
- YBR2RGB
 - gdcm::ImageChangePhotometricInterpretation, [433](#)
- YBR_FULL
 - gdcm::PhotometricInterpretation, [607](#)
- YBR_FULL_422
 - gdcm::PhotometricInterpretation, [607](#)
- YBR_ICT
 - gdcm::PhotometricInterpretation, [607](#)
- YBR_PARTIAL_420
 - gdcm::PhotometricInterpretation, [607](#)
- YBR_PARTIAL_422
 - gdcm::PhotometricInterpretation, [607](#)
- YBR_RCT
 - gdcm::PhotometricInterpretation, [607](#)
- YES
 - gdcm::Surface, [765](#)
- yellow
 - gdcm::terminal, [143](#)
- ZEROED_OUT
 - gdcm::CSAHeader, [277](#)
- ZSpacing
 - gdcm::IPPSorter, [479](#)
- ZTolerance
 - gdcm::IPPSorter, [479](#)